社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

IP-VPN 公平性制御機構 I2VFC の性能評価

本田　泰之† 　本田　治†† 　大崎　博之† 　今瀬　真† 　石塚　美加††† 　村山　純一†††

† 大阪大学 大学院情報科学研究科
〒 565-0871 大阪府吹田市山田丘 1-5
†† 大阪大学 大学院基礎工学研究科
〒 565-0871 大阪府吹田市山田丘 1-5
††† 日本電信電話株式会社 NTT 情報流通プラットフォーム研究所
〒 180-8585 東京都武蔵野市緑町 3-9-11
E-mail: †{yasu-hnd,oosaki,imase}@ist.osaka-u.ac.jp, ††o-honda@ics.es.osaka-u.ac.jp,
†††{ishizuka.mika,murayama.junichi}@lab.ntt.co.jp

あらまし　近年、既存の IP ネットワークを利用して仮想的な専用網を実現する、IP-VPN (IP-based Virtual Private Network) が注目を浴びている。しかし、従来の IP-VPN (IP-based Virtual Private Network) では、 IP-VPN の顧客間の公平性が実現されないという問題がある。そこで我々は、文献 [1] において、IP-VPN の顧客間の公平性を実現する、IP-VPN 公平性制御機構 I2VFC (Inter- and Intra-VPN Fairness Control) を提案した。本稿では、シミュレーションおよびプロトタイプシステムを用いた実験により、I2VFC の有効性を定量的に示す。我々は、I2VFC の VPN 間公平性、VPN 内公平性、帯域および VPN 数に関するスケーラビリティに着目した評価を行った。その結果、さまざまな制御パラメータ設定のもとで、VPN 間公平性および VPN 内公平性が実現できることが分かった。また、我々が提案する I2VFC は、帯域および VPN 数に関して高いスケーラビリティを持つことが分かった。例えば、汎用の計算機を用いた場合、帯域が 1.6 Gpps 程度、VPN 数が 1300 程度まで実現できることが分かった。

キーワード　IP-VPN (IP-based Virtual Private Network)、I2VFC (Inter- and Intra-VPN Fairness Control)、公平性、スケーラビリティ

Performance Evaluation of I2VFC: Inter- and Intra-VPN Fairness Control Mechanism

Yasuyuki HONDA†, Osamu HONDA††, Hiroyuki OHSAKI†, Makoto IMASE†, Mika ISHIZUKA†††, and Junichi MURAYAMA†††

† Graduate School of Infomation Science and Technology, Osaka University, Yamadaoka 1-5, Suita, Osaka 565-0871, Japan
†† Gradauate School of Engineering Science, Osaka University
††† NTT Information Sharing Platform Laboratories, NTT Corporation, 3-9-11 Midori-cho, Musashino-shi, Tokyo, 180-8585 Japan
E-mail: †{yasu-hnd,oosaki,imase}@ist.osaka-u.ac.jp, ††o-honda@ics.es.osaka-u.ac.jp,
†††{ishizuka.mika,murayama.junichi}@lab.ntt.co.jp

Abstract　In recent years, IP-VPN (IP-based Virtual Private Network) that realizes a virtual private network using the existing IP network has been capturing the spotlight. However, in the conventional IP-VPN, there is a problem that fairness among IP-VPN customers is not realized. In [1], we have therefore proposed an IP-VPN fairness control mechanism called I2VFC (Inter- and Intra-VPN Fairness Control) that realizes fairness among IP-VPN customers. In this paper, we quantitatively show effectiveness of our I2VFC using simulation experiments and prototype systems measurements. Focusing on inter-VPN fairness, the intra-VPN fairness, and scalability, we have analyzed the performance of I2VFC. Consequently, we have found that I2VFC could realize both inter-VPN fairness and intra-VPN fairness under diverse control parameter configurations. Moreover, we have found that I2VFC had a high scalability in terms of the link bandwidth and the number of VPNs accomodated. For instance, we found that, when a modern desktop computer was used, I2VFC could support about 1.6 Gbps bandwidth and about 1,300 numbers of VPNs.

Key words　IP-VPN (IP-based Virtual Private Network), I2VFC (Inter- and Intra-VPN Fairness Control), Fairness, Scalability

1 はじめに

近年、IP ネットワークを利用して仮想的な私設網を実現する、IP-VPN (IP-based Virtual Private Network) [2–4] が注目を浴びている。IP-VPN を用いることにより、従来の専用線に比べてはるかに安価に、仮想的な私設網を IP ネットワーク上に構築することができる。

既存の IP-VPN は、IP-VPN の顧客間の公平性が保証されないという問題がある。これは、IP ネットワークがベストエフォート型のネットワークであるため、その上に構築される IP-VPN もベストエフォート型のネットワークとなるからである。我々はこれまで、文献 [1] において、L3-PPVPN のフレームワーク [5] 上で、公平な IP-VPN サービスを実現する、IP-VPN 公平性制御機構 I2VFC (Inter- and Intra-VPN Fairness Control) を提案した。

本稿では、シミュレーション実験およびプロトタイプシステムを用いた実験により、提案する I2VFC の有効性を定量的に評価する。シミュレーション実験では、VPN 間公平性および VPN 内公平性に着目した評価を行う。その結果、I2VFC の制御パラメータの設定によらず、非常に高い精度で VPN 間公平性が実現できることを示す。また、ボトルネックリンクが複数存在する複雑なネットワークにおいても、Max-Min 公平性 [6] を含んだ任意の公平性を実現できることを示す。提案する I2VFC は VPN 内公平性を実現するための積極的な制御を行わないが、ネットワーク全体の輻輳を分散させることにより、結果としてエンドホスト上で動作する TCP の公平性 (VPN 内公平性) を向上させることを示す。プロトタイプシステムを用いた実験では、シミュレーション実験の妥当性を示すとともに、I2VFC の制御のために必要な CPU 時間およびメモリ量を計測する。その結果、提案する I2VFC がリンク帯域および収容する VPN 数に関して高いスケーラビリティを持つことを示す。

本稿の構成は以下の通りである。まず、2 章では、IP-VPN 公平性制御機構 I2VFC の概要および基本となるアイデアを簡単に紹介する。3 章では、シミュレーション実験により、提案する I2VFC がどの程度 VPN 間公平性および VPN 内公平性を実現できるかを定量的に評価する。4 章では、I2VFC のプロトタイプシステムを用いた実験により、I2VFC の帯域および VPN 数に関するスケーラビリティを評価する。最後に、5 章において、本稿のまとめを述べる。

2 IP-VPN 公平性制御機構 I2VFC

本章では、我々が提案する IP-VPN 公平性制御機構 I2VFC [1] の概要を説明する。アルゴリズムの詳細については、文献 [1] を参照されたい。

I2VFC の核となるのは、IP-VPN のサービスプロバイダの PE ルータ上で動作する、AIMD 型のウィンドウフロー制御である。

具体的には、入側 PE ルータにおいて、VPN に収容されている複数のフローを、単一の VPN フローとして集約し、VPN ごとの論理キューに格納する。さらに、入側 PE ルータと出側 PE ルータ間で、各 VPN ごとに管理パケットを定期的に交換することにより、ネットワークのラウンドトリップ時間およびパケット棄却率を測定する。

入側 PE ルータは、これらの情報をもとに、各 VPN フローごとに AIMD 型のウインドウフロー制御 [7] を行い、VPN フローからネットワークに送出されるパケット数を調整する。PE ルータ間では、ウィンドウフロー制御のみを行い、再送制御や誤り制御等は行わない。なお、通常 VPN のトラヒックは双方向に転送されるため、上り/下り両方の VPN フローに対してそれぞれのウインドウフロー制御を行なう必要がある。

PE ルータにおいて、各 VPN ごとに AIMD 型のウィンドウフロー制御を行い、VPN 間公平性を実現する。つまり、測定したラウンドトリップ時間およびパケット棄却率と IP-VPN サービスプロバイダが規定した公平性の基準 (各 VPN フローの重み) をもとに、AIMD 型ウィンドウフロー制御のパラメータ (ウインドウサイズの線形増加量 a および乗算減少量 b [7]) を適切に設定する。これにより、 VPN フローのスループットの比を、サービスプロバイダが設定する任意の比率に制御することが可能となる。

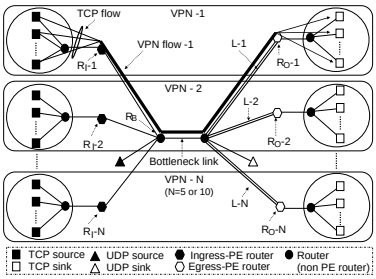

図 1: 単一ボトルネックリンクのネットワークトポロジ
Fig. 1 Network topology with single bottleneck link

VPN 内公平性は、エンド–エンド間で動作する、TCP の輻輳制御機構を利用することによって実現する。つまり、 IP-VPN 公平性制御自体は、VPN 内公平性を実現するための積極的な制御は行わない。同じ VPN 内に収容されているフローは、すべてラウンドトリップ時間およびパケット棄却率が等しくなるため、TCP の輻輳制御機構によって十分な VPN 内公平性が実現できると考えられる。

3 シミュレーション

本章では、シミュレーション実験により、提案する I2VFC の有効性を検証する。特に、VPN 間公平性および VPN 内公平性がどの程度実現できるかに着目して評価を行う。

VPN 間公平性および VPN 内公平性の評価指標として、次式で定義される重みつき公平性指標 (Weighted Fairness Index) F を用いる [8,9]。

$$F = \frac{(\sum_i^N \frac{x_i}{r_i})^2}{N\sum_i^N (\frac{x_i}{r_i})^2} \tag{1}$$

ここで、x_i は i 番目のフローのスループット、r_i は i 番目のフローに対する重み (VPN 内公平性を評価する時はすべて 1)、N はネットワーク中のすべてのフローの数である。重みつき公平性指標 F は 0 から 1 の値をとり、公平性が完全に満たされたとき $F = 1$ となり、公平性が低下するにつれ F は 0 に近い値を取る。

シミュレーションには、OPNET Modeler 9.1A [10] を変更して使用した。シミュレーション時間は 50 秒である。10 回のシミュレーションを実行し、重みつき公平性指標 F の平均値を計算した。すべてのシミュレーションにおいて、重みつき公平性指標 F の 95% 信頼区間は、すべて計測値の 2% 以内に収まっていたため、図中には信頼区間を示していない。

3.1 単一ボトルネックリンクの場合

まず、ボトルネックリンクが単一のネットワーク (図 1) において、VPN 間公平性および VPN 内公平性がどの程度実現できるかを明らかにする。送信側ホストから受信側ホストに向けて、時刻 $t = 5$ [s] から、複数の TCP フローを用いて連続的にデータ転送を行った。シミュレーションにおいて設定した、各 VPN の重みおよびリンクの伝搬遅延を、それぞれ表 1 および表 2 に示す。

バックグラウンドトラヒックとして、ボトルネックリンク上に UDP トラヒックを転送した。バックグランドトラヒックの平均到着レートをボトルネックリンク帯域の 30%、パケット長を 1,500 バイトとし、パケット到着間隔を指数分布とした。特に断りのない限り、シミュレーションでは以下のパラメータを

表 1　各 VPN フローの重み (単一ボトルネックリンクの場合)
Table 1　Weight of each VPN flow (case of a single bottleneck link)

VPN フロー	VPN フローの重み (r_i)
VPN 1	1.0
VPN 2	2.0
VPN 3	2.0
VPN 4	3.0
VPN 5	4.0

表 2　各リンクの伝搬遅延 (単一ボトルネックリンクの場合)
Table 2　Propagation delay of each link (case of a single bottleneck link)

リンク	伝搬遅延 [s]
L1	0.050
L2	0.025
L3	0.075
L4	0.050
L5	0.025

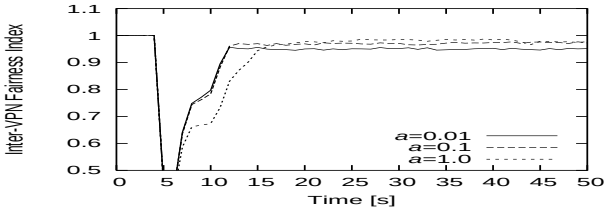

図 2: VPN 間公平性の重みつき指標指標 F の時間的変動 (VPN フロー 1 の乗算減少量 $b = 0.1$)
Fig. 2　Evolution of inter-VPN fairness index (multiplicative decrease factor $b = 0.1$ for VPN flow 1)

用いている。VPN フロー数 5、ボトルネックリンクの帯域 50 [Mbit/s]、ルータのバッファサイズ 50 [packet]、各 VPN フローを構成する TCP フロー数 30、管理パケット送信間隔 $\Delta = 4$。

まず、ウインドウサイズの線形増加量 a および乗算減少量 b をさまざまな値に設定した時に、VPN 間公平性がどの程度実現されるかを明らかにする。エンドホスト上で動作する TCP の輻輳回避フェーズは、$a = 1.0$ および $b = 0.5$ の AIMD 型ウィンドウフロー制御に相当する。このため、I2VFC のウィンドウフロー制御は、$a < 1.0$ および $b < 0.5$ のパラメータ設定の下で、良好に動作すると考えられる。

ウインドウサイズの線形増加量を $a = 0.01, 0.1, 1$ と変化させ、ウィンドウサイズの乗算減少量 b を、表 1 の公平性が実現できるように設定した場合のシミュレーション結果を示す。図 2 および図 3 は、VPN フロー 1 のウィンドウサイズの乗算減少量をそれぞれ $b = 0.1$ および $b = 0.25$ と設定した時の、VPN 間公平性の公平性指標の時間的変動を示している。他の VPN フローのウィンドウサイズの乗算減少量 b の値は、計測したパケット棄却率およびラウンドトリップ時間をもとに設定している。これらの図では、1 [s] ごとの VPN フローのスループットを計算し、式 (1) で定義される重みつき公平性指標 F の値をプロットしている。

これらの図より、ウインドウサイズの線形増加量 a および乗算減少量 b の設定にかかわらず、非常に高い精度で VPN 間公平性を実現できている (F が 0.9 以上) ことがわかる。また、VPN 間公平性の時間的変動 (過渡特性) に着目すると、ウインドウサイズの線形増加量 a が 1.0 の時は過渡特性が若干劣化しているが、それより小さい時、a の設定は過渡特性にほとんど影響を与えないことが分かる。これは、$a = 1.0$ の時、I2VFC のウィンドウフロー制御が、エンドホスト上で動作する TCP のウインドウフロー制御と干渉するためだと考えられる。また図 2 および図 3 を比較すると、ウィンドウサイズの乗算減少量 b の設定は、VPN 間公平性に大きな影響を与えないことがわかる。た

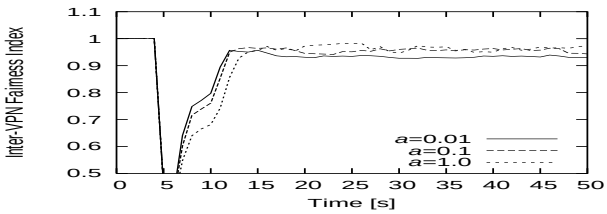

図 3: VPN 間公平性の重みつき指標指標 F の時間的変動 (VPN フロー 1 の乗算減少量 $b = 0.25$)
Fig. 3　Evolution of inter-VPN fairness index (multiplicative decrease factor $b = 0.25$ for VPN flow 1)

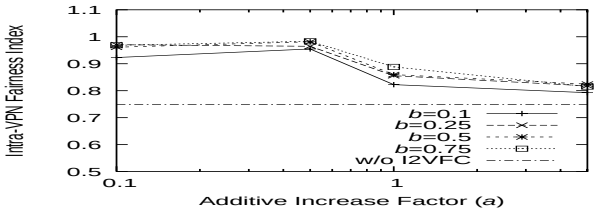

図 4: VPN 内公平性の重みつき公平性指標 F (各 VPN フローを構成する TCP フロー数が 2 の時)
Fig. 4　Weighted fairness index for intra-VPN fairness (2 TCP connections in each VPN flow)

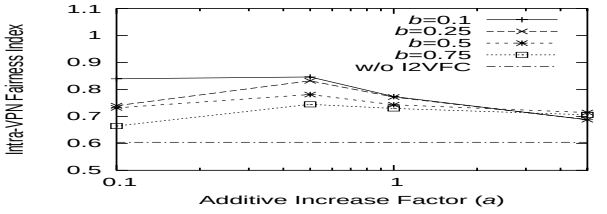

図 5: VPN 内公平性の重みつき公平性指標 F (各 VPN フローを構成する TCP フロー数が 10 の時)
Fig. 5　Weighted fairness index for intra-VPN fairness (10 TCP connections in each VPN flow)

だし、VPN 間公平性を最大化する a の値は、b の値に応じて変化することがわかる。

以上の考察より、I2VFC のウィンドウフロー制御は、$a < 1.0$ および $b < 0.5$ のパラメータ設定の下で、非常に高い VPN 間公平性を実現することが分かる。

次に、ウインドウサイズの線形増加量 a および乗算減少量 b をさまざまな値に設定した時に、VPN 内公平性がどの程度実現されるかを明らかにする。VPN フロー数を 4 とし、ウインドウサイズの線形増加量を $a = 0.1, 0.25, 0.5, 0.75$ と変化させ、ウインドウサイズの乗算減少量を $b = 0.1, 0.5, 1.0, 5.0$ と変化させた。この時の、VPN 内公平性の公平性指標 F の値を、図 4 および図 5 に示す。図 4 は、各 VPN フローを構成する TCP フローの数を 2 とした時の結果である。図 5 は、各 VPN フローを構成する TCP フローの数を 10 とした時の結果である。

なお、I2VFC の制御によって、どの程度 VPN 内公平性が向上するかを調べるために、I2VFC の制御を行わないシミュレーションもあわせて行った。その結果、各 VPN フローを構成する TCP フロー数が 2 の時の公平性指標は 0.748、各 VPN フローを構成する TCP フロー数が 10 の時の公平性指標は 0.604 であった。

図 4 および図 5 より、ウインドウサイズの線形増加量 a および乗算減少量 b の値によって、VPN 内公平性が大きく変化していることが分かる。特に、a および b の値が小さい時に、より VPN 内公平性が向上していることが分かる。これは、a および

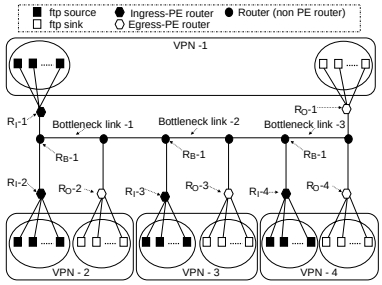

図 6: 複数ボトルネックリンクのネットワークトポロジ
Fig. 6　Network topology with multiple bottleneck links

表 3　各 VPN フローの重み (複数ボトルネックリンクの場合)
Table 3　Weight of each VPN flow (case of multiple bottleneck links)

	B_3 [Mbit/s]		
VPN フロー	10	20	30
VPN 1	1.0	1.0	1.0
VPN 2	3.0	3.0	3.0
VPN 3	1.0	1.0	1.0
VPN 4	1.0	3.0	5.0

表 4　各リンクの伝搬遅延 (複数ボトルネックリンクの場合)
Table 4　Propagation delay of each link (case of multiple bottleneck links)

リンク	伝搬遅延 [s]
L1	0
L2	0
L3	0
L4	0
L5	0

b の値が小さい場合は、I2VFC のウィンドウフロー制御が TCP のウィンドウフロー制御に与える影響が小さくなるためと考えられる。

図 4 および図 5 において、ウインドウサイズの線形増加量 a および乗算減少量 b の設定にかかわらず、I2VFC の制御を行うことにより、I2VFC の制御を行わない場合と比較して、VPN 内公平性が向上している点は注目すべきである。例えば、図 4 において、$a = 5.0$ および $b = 0.75$ のように、TCP のウィンドウフロー制御に比べて、より急激にウィンドウフロー制御を行った場合でも、VPN 内公平性が 0.748 から 0.815 へ向上している。これは、入側 PE ルータおよび出側 PE ルータ間で AIMD 型のフロー制御を行うことにより、ボトルネックルータにおける輻輳が緩和されたことが原因と考えられる。

以上の考察より、I2VFC の制御を導入することによって、ウインドウサイズの線形増加量 a および乗算減少量 b の設定にかかわらず、VPN 内公平性が向上することが分かった。I2VFC は、VPN 内公平性を向上させるための積極的な制御を行っていないが、ネットワーク全体の輻輳を分散させることにより、結果としてエンドホスト上で動作する TCP の公平性を向上させることが分かった。

3.2　複数ボトルネックリンクの場合

次に、ボトルネックリンクが複数存在するネットワーク (図 6) において、VPN 間公平性が実現できることを示す。ここでは特に、提案する I2VFC によって、複数ボトルネックが存在する複雑なネットワークにおいて、Max-Min 公平性が実現できることを示す。シミュレーションにおいて設定した、各 VPN の重み、リンクの伝搬遅延を表 3 および表 4 に示す。

以下のシミュレーションでは、リンク 1 の帯域を 20 [Mbit/s]、リンク 2 の帯域を 10 [Mbit/s] と固定し、リンク 3 の帯域 (B_3) を 10, 20, 30 [Mbit/s] と 3 種類に変化させた。表 3 に示した、各 VPN の重みは、Max-Min 公平性に基づき計算した値である。

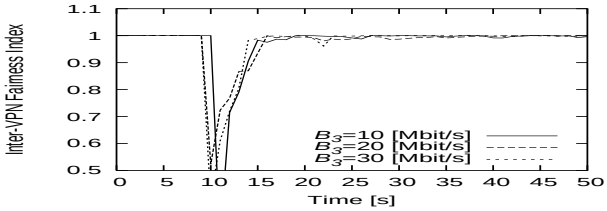

図 7: VPN 間公平性の重みつき指標指標 F の時間的変動 (複数ボトルネックリンクの場合)
Fig. 7　Evolution of weighted fairness index for inter-VPN fairness (case of multiple bottleneck links)

ルータのバッファサイズを 200 [packet]、各 VPN フローを構成する TCP フロー数 を 30 とした。その他のパラメータについては、単一ボトルネックリンクの場合と同じ値を用いた。

すべての VPN フローに対して、ウインドウサイズの線形増加量を $a = 0.5$ と設定した。VPN フロー 1 のウィンドウサイズの乗算減少量を $b = 0.01$ と設定した。他の VPN フローのウィンドウサイズの乗算減少量 b の値は、計測したパケット棄却率およびラウンドトリップ時間をもとに設定した。この時の、VPN 間公平性の公平性指標の時間的変動を図 7 に示す。

図 7 より、リンク 3 の帯域によらず、すべての場合において非常に高い精度で VPN 間公平性が実現できている ($F > 0.95$) ことがわかる。提案する I2VFC では、各 VPN フローが経由するボトルネックリンクがそれぞれ異なる場合であっても、各 VPN フローのパケット棄却率およびラウンドトリップ時間をもとにウィンドウサイズの線形増加量/ 乗算減少量を設定することにより、任意の公平性を実現できている。

VPN 間公平性がどれだけ速く収束するか (過渡特性) に着目する。図 7 より、リンク 3 の帯域によらず、すべての VPN フローが転送を開始してから 10 秒以内に VPN 間公平性の公平性指標が収束していることがわかる。例えば、リンク 3 の帯域が 10 [Mbit/s] の時、最もホップ数の多い VPN フロー 1 のラウンドトリップ時間が 0.312 [s] であることを考えると、非常に良好な過渡特性を示していると言える。

以上の考察より、提案する I2VFC を用いることで、ボトルネックリンクが複数存在するネットワークにおいて、Max-Min 公平性のような任意の公平性を実現できることが分かった。また、VPN 間公平性は良好な過渡特性を持つ (ラウンドトリップ時間の 16 倍程度のタイムスケールで収束する) ことが分かった。

4　プロトタイプシステムによる評価

4.1　プロトタイプシステムの概要

I2VFC のプロトタイプシステムを、C 言語を用いてユーザ空間で動作するアプリケーションとして実装した (図 8)。入側 PE ルータは、パケット送信処理、パケット受信処理、ウィンドウフロー制御をそれぞれ処理する 3 種類のプロセスによって、出側 PE ルータは、パケット送信処理、パケット受信処理をそれぞれ処理する 2 種類のプロセスによって構成されている。パケットの受信には libpcap バージョン 0.6.2 を用い、パケットの送信には RAW ソケットを用いて実装した。プロセス間通信には共有メモリを使用して実装した。

4.2　実 験 環 境

プロトタイプシステムを用いた実験に使用した、ネットワークのトポロジを図 9 に示す。実験では、以下のような機器を使用した。

- 送信側ホスト、受信側ホスト

Linux オペレーティングシステムが稼働する計算機を用い、TCP ベンチマークソフトウェア [11] によって複数の TCP フローを生成した。今回の実験では、TCP の受信側ポート番号に

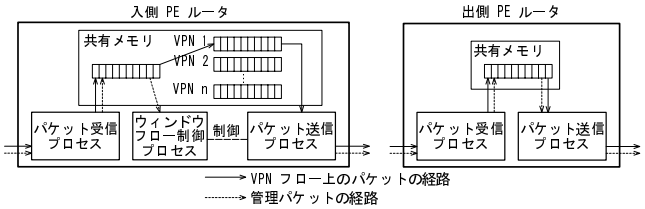

図 8: I2VFC プロトタイプシステムの概要
Fig. 8　I2VFC prototype system overview

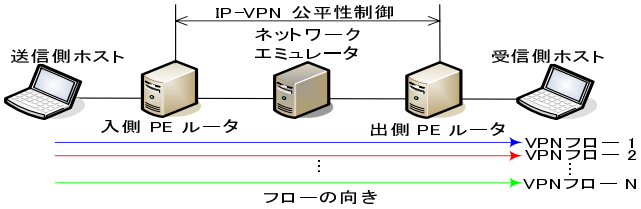

図 9: 実験に使用したネットワークトポロジ
Fig. 9　Network topology used in prototype system experiments

表 6　実験におけるパラメータ設定
Table 6　Parameter configuration in prototype system experiment

VPN フロー数	4	
VPN フローを構成する TCP フロー数	1, 2	
VPN フローの重み	1.0	
VPN フロー単位のバッファサイズ	128	[packet]
ウィンドウサイズの線形増加量 a	0.1	
ウィンドウサイズの乗算減少量 b	0.1	
管理パケット送信間隔 Δ	4	
ネットワークエミュレータの帯域	50	[Mbit/s]
ネットワークエミュレータの遅延	2	[ms]
ネットワークエミュレータのバッファサイズ	50	[packet]

よって VPN を識別した。

- 入側 PE ルータ、出側 PE ルータ

Linux オペレーティングシステムが稼働する計算機を用い、実装した I2VFC のプロトタイプを動作させた。

- ネットワークエミュレータ

さまざまなネットワーク環境を模擬するためにネットワークエミュレータを使用し、ボトルネックとなるリンクの帯域および遅延を変化させた。FreeBSD オペレーティングシステムが稼働する計算機を用い、ネットワークエミュレータとして dummynet [12] を使用した。dummynet では、帯域、遅延、バッファサイズを設定することができるため、この機能を利用した。

それぞれの機器の仕様 (CPU、メモリ量、OS 種別など) を表 5 に示す。なお、特に断りのない限り、実験では表 6 に示すパラメータ設定を用いた。

4.3　VPN 間公平性および VPN 内公平性の評価

プロトタイプシステムを用いた実験では、VPN フロー数を 4 とし、VPN フロー 1 および VPN フロー 3 を構成する TCP フロー数を 1 に、VPN フロー 2 および VPN フロー 4 を構成する TCP フロー数を 2 とした。VPN フロー 1 から VPN フロー 4 まで、5 [s] ごとに順番に転送を開始させ、その時の VPN スループットおよび各 TCP フローのスループットを計測した。

プロトタイプシステムを用いた実験では、各 VPN フローの挙動を詳細に調べるため、重みつき公平性指標ではなく、VPN スループットの時間的変動を計測した。実験によって得られた、VPN スループットの時間的変動を図 10 に示す。この図では、2 [s] ごとの平均 VPN スループットの時間的変動をプロットしている。図中には各 VPN フローを構成する TCP フローのスループットもあわせて示している。この図より、VPN 間公平性が実現できていることが分かる。つまり、VPN フローを構成する TCP フロー数が VPN ごとに異なっているにもかかわらず、25 [s] 前後で VPN スループットがほぼ等しくなっていることが分かる。シミュレーション結果 (図 2) と比較すると、VPN スループットの収束時間がほぼ等しいことが分かる。

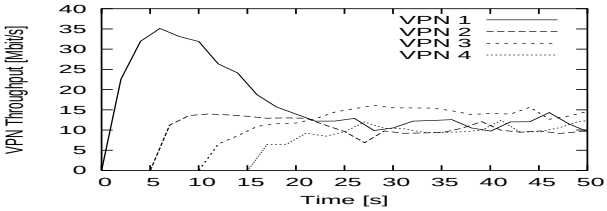

図 10: 各 VPN スループットの時間的変動
Fig. 10　Evolution of each instantenous VPN throghput

プロトタイプシステムを用いた実験の結果 (図 10) をもとに、VPN 内公平性の重みつき公平性指標 F を計算した。その結果、VPN フロー 2 の重みつき公平性指標が $F = 0.99$ であり、VPN フロー 4 の重みつき公平性指標が $F = 0.99$ であった。これから、VPN 内公平性が実現できていることが分かる。これらの値は、シミュレーション結果 (図 4) ともおおよそ一致している。

以上の考察から、実装したプロトタイプシステムにおいても、VPN 間公平性および VPN 内公平性が実現できていることが確認できた。さらに、シミュレーション結果とプロトタイプシステムによる実験結果がほぼ一致していることも確認できた。これにより、シミュレーション実験およびプロトタイプシステムを用いた実験の妥当性が確認できたと考えられる。

4.4　スケーラビリティの評価

I2VFC のスケーラビリティを評価するために、プロトタイプシステムを用いて (1) 入側 PE ルータおよび出側 PE ルータが使用する CPU 時間、および (2) 入側 PE ルータおよび出側 PE ルータが使用するメモリ量を計測した。使用する CPU 時間は、C コンパイラのプロファイラを用いてモジュール単位の実行時間を計測した。具体的には、すべての VPN フローが転送を開始してから 180 [s] 間の制御を行い、使用した CPU 時間の総和を計測した。また、I2VFC では動的なメモリ確保は行わず、静的なメモリ確保のみ行う。このため、I2VFC が必要とするメモリ量は、I2VFC のプロトタイプが確保するメモリ量の合計として計算した。

帯域および VPN 数に関するスケーラビリティを評価するため、ネットワークエミュレータの帯域を 10–100 [Mbit/s] と変化させ、VPN フロー数を 2–100 と変化させて実験を行なった。各 VPN フローを構成する TCP フローの数は 1 または 2 とした。その他のパラメータについては、表 6 の値を用いた。

まず、VPN フロー数を 100 と固定し、ネットワークエミュレータの帯域を変化させた時の、入側 PE ルータおよび出側 PE ルータが使用する CPU 時間を図 11 に示す。この図では、入側 PE ルータおよび出側 PE ルータが、180 [s] 間の制御に要した CPU 時間の総和を示している。この図より、入側 PE ルータおよび出側 PE ルータが使用する CPU 時間は、帯域に応じてほぼ線形に増加していることが分かる。例えば、ネットワークエミュレータの帯域が 100 [Mbit/s] の時、入側 PE ルータが使用した CPU 時間は 10.9 [s] であるが、これは CPU の利用率に換算すると約 6 % である。このことから、実験に使用した機器を用いて、VPN フロー数が 100 の時に、約 1600 [Mbit/s] 程度までの帯域をサポートできると考えられる。

次に、ネットワークエミュレータの帯域を 100 [Mbit/s] に固定し、VPN フロー数を変化させた時の、入側 PE ルータおよび出側 PE ルータが使用する CPU 時間を図 12 に示す。この図よ

表 5　実験に使用した機器の仕様
Table 5　Device specifications used in prototype system experiments

	CPU	メモリ	OS	NIC ドライバ
送信側ホスト	Celeron 1.06 GHz	376 Mbyte	Linux 2.4.22	e100-2.3.18
受信側ホスト	Celeron 1.06 GHz	376 Mbyte	Linux 2.4.22	e100-2.3.18
入側 PE ルータ	Pentium4 1.70 GHz	254 Mbyte	Linux 2.4.20	e1000-4.4.19
出側 PE ルータ	Celeron 2.00 GHz	505 Mbyte	Linux 2.4.20	epic100-1.11, 8139to-0.9.24
ネットワークエミュレータ	Pentium4 2.26 GHz	228 Mbyte	FreeBSD 5.2.1	fxp, tx

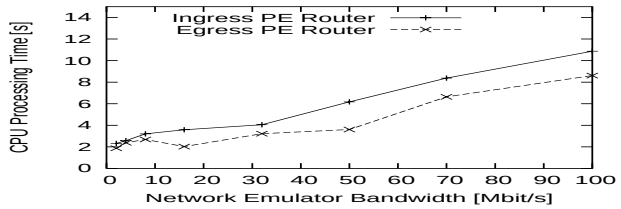

図 11: ネットワークエミュレータの帯域と入側/出側 PE ルータが使用する CPU 時間の関係
Fig. 11　Relation between network emulator bandwidth and total CPU processing time consumed by ingress/egress PE router

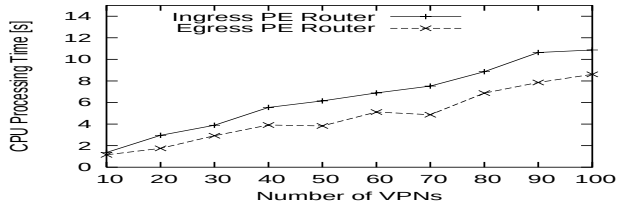

図 12: VPN フロー数と入側/出側 PE ルータが使用する CPU 時間の関係
Fig. 12　Relation between the number of VPN flows and total CPU processing time consumed by ingress/egress PE router

り、入側 PE ルータおよび出側 PE ルータが使用する CPU 時間は、VPN フロー数に応じてほぼ線形に増加していることが分かる。例えば、VPN フロー数が 100 の時に、入側 PE ルータが使用した CPU 時間は 10.9 [s] であるが、これは CPU の利用率に換算すると約 6 % である。このことから、実験に使用した機器を用いて、帯域が 100 [Mbit/s] の時に、約 1600 VPN フロー程度までサポートできると考えられる。

最後に、ネットワークエミュレータの VPN フロー数を変化させた時の、入側 PE ルータおよび出側 PE ルータが使用するメモリ量を図 13 に示す。なお、入側 PE ルータおよび出側 PE ルータが使用するメモリ量は、帯域によらず一定である。この図より、入側 PE ルータが使用するメモリ量は、VPN フロー数にほぼ比例することが分かる。一方、出側 PE ルータが使用するメモリ量は、VPN フロー数によらずほぼ一定であることが分かる。これは、使用されるメモリの大半が、入側 PE ルータのウィンドウフロー制御に必要なバッファとして確保されるためである。例えば、VPN フロー数が 100 の時に、入側 PE ルータが使用するメモリ量は 19.5 [Mbyte]、出側 PE ルータが使用するメモリ量は 1.55 [Mbyte] である。このことから、実験に使用した機器を用いて、約 1300 VPN フロー程度までサポートできると考えられる。

5 ま と め

本稿では、シミュレーション実験およびプロトタイプシステムを用いた実験により、I2VFC の有効性を定量的に評価した。その結果、(1) I2VFC の制御パラメータの設定によらず、非常に高い精度で VPN 間公平性が実現できること、(2) ボトルネックリンクが複数存在する複雑なネットワークにおいても、Max-Min

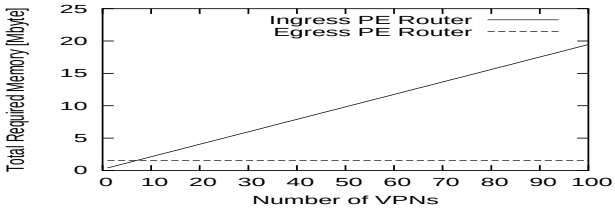

図 13: VPN フロー数とメモリ量の関係
Fig. 13　Relation between the number of VPN flows and memory usage in ingress/egress PE router

公平性を含んだ任意の公平性を実現できること、(3) I2VFC を用いることにより、エンドホスト上で動作する TCP の公平性 (VPN 内公平性) が向上すること、(4) I2VFC がリンク帯域および収容する VPN 数に関して高いスケーラビリティを持つこと、などが明らかになった。

謝　　辞

本研究の一部は、平成 16 年度科学技術振興調整費「サイバーソサエティを実現する仮想網技術」の援助による。

文　　献

[1] 本田 治, 大崎 博之, 今瀬 真, 村山 純一, 松田 和浩, “AIMD 型のウィンドウフロー制御を利用した IP-VPN 公平 性制御機構,” 電子情報通信学会技術研究報告 (IN2004-8), pp. 43–48, May 2004.

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

[3] M. Carugi and J. D. Clercq, “Virtual private network services: Scenarios, requirements and architectural constructs from a standardization perspective,” *IEEE Communication Magazine*, June 2004.

[4] A. Nagarajan, “Generic requirements for provider provisioned VPN,” *Internet Draft* `<draft-ietf-ppvpn-generic-reqts-02.txt>`, Jan. 2003.

[5] R. Callon, M. Suzuki, J. D. Clercq, B. Gleeson, A. G. Malis, K. Muthukrishnan, E. C. Rosen, C. Sargor, and J. J. Yu, “A framework for layer 3 provider provisioned virtual private networks,” *Internet Draft* `<draft-ietf-ppvpn-framework-08.txt>`, Mar. 2003.

[6] D. Bertsekas and R. Gallager, *Data Networks*. Englewood Cliffs, New Jersey: Prentice-Hall, 1987.

[7] 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, 1989.

[8] 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.

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

[10] Opnet Technologies, Inc., “OPNET.” `http://www.opnet.com/`.

[11] “The TCP/UDP bandwidth measurement tool.” `http://dast.nlanr.net/Projects/Iperf/`.

[12] L. Rizzo, “Dummynet: a simple approach to the evaluation of network protocols,” *ACM Computer Communication Review*, vol. 27, pp. 31–41, Jan. 1997.