論　文

任意の公平性を提供できるスケーラブル IP-VPN フロー制御機構

本田　治[†a)]　大崎　博之[††b)]　今瀬　真[††c)]　村山　純一[†††d)]
松田　和浩[†††e)]

On Scalable IP-VPN Flow Control Mechanism Supporting Arbitrary Fairness Criteria

Osamu HONDA[†a)], Hiroyuki OHSAKI[††b)], Makoto IMASE[††c)], Junichi MURAYAMA[†††d)], and Kazuhiro MATSUDA[†††e)]

あらまし　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) が注目を浴びている。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。本稿では、公平な IP-VPN サービスを実現するための、IP-VPN 公平性制御機構 I2VFC (Inter- and Intra-VPN Fairness Control) を提案する。I2VFC は、IP-VPN サービスプロバイダのプロバイダエッジルータ (PE ルータ) 間で動作する、AIMD (Additive Increase and Multiplicative Decrease) 型のウィンドウフロー制御である。提案する I2VFC では、AIMD 型ウィンドウフロー制御の解析結果を利用することにより、VPN 間公平性の基準を、IP-VPN のサービスプロバイダが自由に規定できるという点が特徴である。また、PE ルータのみを変更するだけでよく、既存の IP ネットワークへ容易に導入が可能である。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 がどの程度設計目標を満たしているかを定量的に評価する。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 公平性を含んだ任意の公平性を実現できること、(3) 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、(4) 転送速度および収容する VPN 数に関して高いスケーラビリティを持つこと、などを明らかにした。

キーワード　IP-VPN (IP-based Virtual Private Network)，公平性，ウインドウフロー制御，AIMD (Additive Increase and Multiplicative Decrease)

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) [1-3] 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。しかし現実には、IP-VPN のサービスプロバイダは、公平な IP-VPN サービスを提供することが強く求められている [4]。近年、IP ネットワークのトラヒックエンジニアリング技術に関して、さまざまな研究が行なわれている。しかし、既存のトラヒックエンジニアリング技術は、公平な IP-VPN サービスの実現には不十分である [2]。

本研究では、文献 [5] で提案されている、L3-PPVPN (Provider Provisioned VPN)、すなわち、サービスプロバイダが、顧客に対して第 3 層の VPN サービスを提供するというフレームワークに注目する。そして L3-

† 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
Gradauate School of Engineering Science, Osaka University
†† 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
Graduate School of Infomation Science and Technology, Osaka University, Yamadaoka 1-5, Suita, Osaka 565-0871, Japan
††† 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
NTT Information Sharing Platform Laboratories, NTT Corporation, 3-9-11 Midori-cho, Musashino, Tokyo 180-8585, Japan
a) E-mail: o-honda@ics.es.osaka-u.ac.jp
b) E-mail: oosaki@ist.osaka-u.ac.jp
c) E-mail: imase@ist.osaka-u.ac.jp
d) E-mail: murayama.junichi@lab.ntt.co.jp
e) E-mail: matsuda.kazuhiro@lab.ntt.co.jp

PPVPN のフレームワーク上で、公平な IP-VPN サービスを、サービスプロバイダの IP ネットワークへ容易に導入可能な方法で実現することを目標とする。

本稿では、まず、公平な IP-VPN サービスを実現するための制御機構に対する設計目標を議論する。具体的には、以下の 4 つの設計目標を掲げ、それぞれの設計目標の必要性および要求条件を議論する。

- (1) VPN 間公平性 (inter-VPN fairness) を実現
- (2) VPN 内公平性 (intra-VPN fairness) を実現
- (3) サービスプロバイダの IP ネットワークへの導入が容易
- (4) 転送速度 / VPN 数に関して高いスケーラビリティを実現

本稿では、公平な IP-VPN サービスを実現するための、IP-VPN 公平性制御機構 I2VFC (Inter- and Intra-VPN Fairness Control) を提案する。I2VFC は、IP-VPN サービスプロバイダのプロバイダエッジルータ (PE ルータ) 間で動作する、AIMD (Additive Increase and Multiplicative Decrease) [6] 型のウィンドウフロー制御である。具体的には、入口側のプロバイダエッジルータ (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 (プロトコル種別や送信/受信アドレスなどが等しいパケットの列) を、単一のフローとして集約する。さらに、入側 PE ルータと、出側の PE ルータの間で、AIMD 型のウインドウフロー制御を行う。

さらに、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 がどの程度設計目標を満たしているか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。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 が転送速度および収容する 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 公平性制御に関する関連研究を紹介する。3 章では、IP-VPN 公平性制御機構の設計目標を議論する。4 章では、我々が提案する IP-VPN 公平性制御機構 I2VFC の概要、アーキテクチャと動作アルゴリズムを説明する。5 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。6 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、7 章において、本稿のまとめと今後の課題を述べる。

2. 関 連 研 究

文献 [7, 8] では、ネットワーク中のすべてのルータが DiffServ に対応した IP ネットワークにおいて、DiffServ ルータのキュー管理機構を変更することにより、公平な IP-VPN サービスを実現する手法を提案している。しかし、文献 [7, 8] で提案されている手法は、ネットワーク上のすべてのルータを、独自のキュー管理機構を有する DiffServ ルータに置き換える必要があるため、IP-VPN サービスプロバイダの IP ネットワークに容易に導入することができない。

一方、ルータ間で AIMD 型のウインドウフロー制御を行なうことにより、IP-VPN の公平性を実現する手法も提案されている [9, 10]。文献 [9, 10] では、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フローとして集約し、集約したフローに対して独自の AIMD 型ウィンドウフロー制御を行うことにより、IP-VPN 間の公平性を実現する。しかし、文献 [9, 10] で提案されている手法では、ネットワーク中のすべてのコアルータが独自のアクティブキュー管理機構を実装する必要があり、現実のネットワークへの導入が用意ではないという問題がある。

3. 設 計 目 標

(1) VPN 間公平性 (inter-VPN fairness) を実現

第一の設計目標は、VPN サービスを契約している顧客間の公平性を実現すること、すなわち、 VPN 間公平性 (inter-VPN fairness) を実現することである。既存の IP-VPN では、基盤となる 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に、VPN 間の公平性が十分に提供されていないのが現状である。しかし、 IP-VPN サービスとしては、ある VPN が大量のトラヒックを発生させた場合でも、ボトルネックリンクの帯域が各 VPN 間で公平に使用され、他の VPN のスループットが不当に低く抑えられないことが望ましい。つまり、サイト間を接続する VPN フロー (同じ VPN に収容されており、通過する入側および出側 PE ルータが共通であるフロー) のスループット (実効転送速度) の比が、IP-VPN サービスプロバイダが規定する比となっていることが求められる。

特に、IP-VPN サービスを提供するのはプロバイダ

であることを考えると、VPN 間の公平性の基準は、IP-VPN のサービスプロバイダが自由に規定できることが望ましい。例えば、あるボトルネックリンクを共有している複数の VPN フローに対して、均等に帯域を配分するのではなく、VPN の地理的条件 (サイト間の距離やホップ数など) や契約回線速度等に応じて配分できることが望ましい。どのような公平性の基準が適切かは、IP-VPN のサービスプロバイダごとに異なると考えられるため、IP-VPN 公平性制御機構としては、さまざまな公平性の基準に対応できることが求められる。

具体的には、VPN フロー i $(1 \leqq i \leqq N)$ のスループットを T_i、IP-VPN サービスプロバイダが規定する VPN フロー i のスループットの重みを r_i とすれば、すべての i、j $(i \neq j)$ に対して、

$$\frac{T_i}{r_i} = \frac{T_j}{r_j} \tag{1}$$

を実現することが求められる。

また、どのようなタイムスケールで VPN 間公平性を実現するかも重要である。数 10 ミリ秒 ~ 数 100 ミリ秒といった細かな粒度での公平性が必要となるのか、それとも数十秒 ~ 数分といった荒い粒度での公平性で十分なのかによって、IP-VPN 公平性制御機構に要求される機能が大きく異なる。

現実には、以下の 2 つの理由により、ラウンドトリップ時間の 100 倍程度のタイムスケールで、VPN 間公平性を実現することが望ましいと考えられる。(1) 現在、IP-VPN 上を転送されるトラヒックの大部分がデータ系トラヒックであり、ラウンドトリップ時間の 100 倍程度のタイムスケールの公平性が実現されれば十分であること。(2) ラウンドトリップ時間オーダのタイムスケールで公平性を実現するためには、エッジルータの変更だけでは不可能であり、コアルータ自体を変更する必要があること。

(2) VPN 内公平性 (intra-VPN fairness) を実現

第二の設計目標は、同じ VPN に収容されている利用者間の公平性を実現すること、すなわち、VPN 内公平性 (intra-VPN fairness) を実現することである。IP-VPN サービスとして、VPN 間公平性が実現され、VPN を契約している顧客間の公平性が実現されたとしても、実際に 同じ VPN に収容されている利用者間で不公平性が発生することは望ましくない。本稿では、同じ VPN に収容されているフローのスループットの比がすべて等しい時、VPN 内公平性が実現されていると定義する。

ただし、 IP-VPN サービスの特性を考えると、VPN 内公平性に対する要求は、VPN 間公平性に対する要求に比べて、比較的緩いものであると考えられる。一般に、ある同じ VPN に収容されている利用者に対して、どのように帯域を配分するかは、VPN を契約している顧客が決定すべき内容であり、IP-VPN のサービスプロバイダは関知しない。そこで、 IP-VPN 公平性制御機構としては、ある特定のフローのスループットが不当に抑えられなければ十分であると考えられる。

(3) サービスプロバイダの IP ネットワークへの導入が容易

第三の設計目標は、IP-VPN 公平性制御機構が、既存の IP ネットワークへ容易に導入できることである。IP-VPN がこれだけ普及した要因として、ネットワークのインフラとして、既存の IP ネットワークがそのまま利用できるという点が挙げられる。従って、既存の IP-VPN の枠組をできるだけ変更せずに、公平な IP-VPN サービスを実現できることが望ましい。

具体的には、 IP-VPN のサービスプロバイダが所有する PE ルータだけに変更を加え、既存のコアルータには変更を加えずに、IP-VPN 公平性制御機構を実現することが求められている [5, 11, 2]。IP-VPN 公平性制御機構は、コアルータが単純 Drop-tail ルータであっても、公平な VPN サービスを提供できなければならない。さらに、カスタマエッジ (CE) ルータは VPN を契約している顧客の管理下にあるため、IP-VPN のサービスプロバイダの都合で変更することは事実上不可能である。このため、IP-VPN 公平性制御は、IP-VPN サービスプロバイダが所有する PE ルータのみによって実現できることが必要である。

(4) 転送速度 / VPN 数に関して高いスケーラビリティを実現

第四の設計目標は、IP-VPN 公平性制御機構が、VPN フローの転送速度および収容する VPN 数に関して高いスケーラビリティを持つことである。近年、ネットワークの高速化が急速に進んでいるため、各 VPN ごとに数 Gbps から数十 Gbps 程度のスループットを実現できることが望ましい。一方、現在は、企業や団体といった組織単位で IP-VPN サービスに加入しているため、IP-VPN のサービスプロバイダが管理する VPN 数はそれほど多くない。しかし今後は、個人の利用者単位で IP-VPN サービスに加入することも想定される。この場合、IP-VPN のサービスプロバイダが管理する

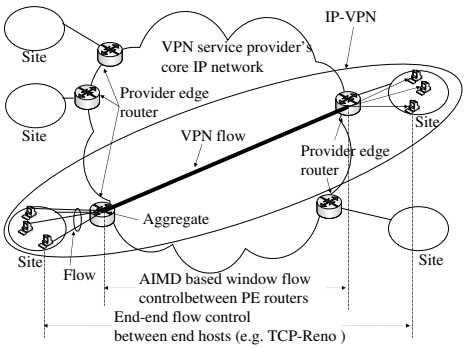


図 1 Overview of I2VFC (Inter- and Intra-VPN Fairness Control)

VPN 数は膨大となることが予想されるため、IP-VPN 公平性制御機構が、VPN 数に関して高いスケーラビリティを持つことも重要であると考えられる。

4. 提案アーキテクチャとアルゴリズム

4.1 I2VFC の概要

図 1 に、提案する I2VFC の概要を示す。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、同じ 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をもとに、AIMD 型ウィンドウフロー制御のパラメータを適切に設定する。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。

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。

PE ルータ間で転送されるパケットに対しては、カプセル化等の処理は行わない。つまり、 VPN に収容されている複数のフローを構成するパケットは、IP-VPN サービスプロバイダのネットワークをそのまま転送される。これにより、PE ルータの処理を単純にし、転送速度および VPN 数に関して高いスケーラビリティを実現する。

4.2 アルゴリズム

I2VFC は、VPN フローごとに以下のような処理を行う。

- 管理パケットの交換によるフィードバック情報の取得

I2VFC のウィンドウフロー制御に必要な情報 (各 VPN のパケット棄却率、ラウンドトリップ時間、送信が完了したパケットのシーケンス番号) を得るために、管理パケットを入側と出側の PE ルータ間で交換する。

入側 PE ルータは、パケットを一定個数 Δ 送信するごとに、管理パケットを出側 PE ルータに送信する。管理パケットには、管理パケットの送信時刻、送信した最後のパケットのシーケンス番号を記録する。

出側 PE ルータは、入側 PE ルータから送信された管理パケットから、各 VPN フローのパケット棄却率を計算する。計算したパケット棄却率とパケットの受信時刻を管理パケットに記録し、管理パケットを入側 PE ルータに返送する。

入側 PE ルータは、出側 PE ルータから返送された管理パケットから、各 VPN フローのパケット棄却率および出側 PE ルータに到着したパケットのシーケンス番号を得る。また、管理パケットに記録されている、パケット送信時刻およびパケット受信時刻より、VPN フローのラウンドトリップ時間を計算する。

- AIMD 型のウインドウフロー制御

入側 PE ルータは、AIMD 型のウインドウフロー制御によってウィンドウサイズを更新し、各 VPN から

のパケットの送信量を制御する。

入側 PE ルータは、管理パケットを受信すると、VPN フローのパケット棄却率 p をもとにウィンドウサイズ w を以下のように更新する。

$$w \leftarrow \begin{cases} \min(\overline{W}, w + \frac{a\Delta}{w}) & \text{if } p = 0 \\ \max(\underline{W}, w - b\,w) & \text{otherwise} \end{cases} \quad (2)$$

ここで、$\overline{W}$ はウインドウサイズの最大値、$\underline{W}$ はウインドウサイズの最小値である。通常、入側 PE ルータは、1 ラウンドトリップ時間に、w/Δ 個の管理パケットを受信する。このため、式 (2) は、パケット棄却率が 0 の時、1 ラウンドトリップ時間あたりに、a だけウィンドウサイズを増加させることを意味している。

入側 PE ルータは、1 ラウンドトリップ時間中に、w 個のパケットを送信することができる。管理パケットがネットワーク中で連続して廃棄された場合のデッドロックを防ぐため、タイムアウトを利用した管理パケットの再送制御を行う。具体的には、入側 PE ルータが管理パケットを送信してから、T 秒間経過しても 出側 PE ルータから管理パケットが返送されない場合、管理パケットを再送する。さらに、タイムアウトが発生した場合、入側 PE ルータは、式 (2) に従って、ウインドウサイズを乗算的に減少させる。なお、タイムアウトによる管理パケットの再送は、送信すべきパケットが存在しない場合でも行なわれる。また、管理パケットのサイズは、IP ヘッダを加えて 32 バイト程度である。例えば、$\Delta = 4$ とすると、管理パケットのオーバーヘッドは、パケット長が 1500 バイトの時で帯域の約 0.5%、パケット長が 500 バイトの時で帯域の約 5%、パケット長が 100 バイトの時で帯域の約 24%となる。

4.3 設計目標の実現

4.3.1 VPN 間公平性の実現

VPN 間公平性を実現するための基本的なアイディアは、入側 PE ルータおよび出側 PE ルータ間で動作する、VPN ごとの AIMD 型のウィンドウフロー制御のパラメータを調整することにより、(ラウンドトリップ時間よりも十分大きい、ある程度大きな時間スケールで考えた場合に) 任意の公平性を実現するというものである。

AIMD 型のウィンドウフロー制御 [6] は、ネットワーク中で輻輳が発生していない時には、ウィンドウサイズ W を a だけ加算的に増加させる。一方、ネットワーク中で輻輳が発生している時には、ウィンドウサイズを $b \times W$ だけ乗算的に減少させる。AIMD 型のウィンドウフロー制御は、TCP の輻輳回避フェーズでも採用されており、これまでに数多くの研究が行なわれている

例えば、文献 [12] では、決定的 AIMD モデル (Deterministic AIMD model) を用いて、定常状態における AIMD 型ウインドウフロー制御のスループット T が近似的に次式で与えられることが示されている。

$$T = \frac{pa(b-2) + \sqrt{p(b-2)a(pab - 8b - 2pa)}}{4pbR} \quad (3)$$

$$\simeq \frac{\sqrt{2-b}\sqrt{a}}{\sqrt{2b}R\sqrt{p}} \quad (4)$$

ここで、a および b は、AIMD 型のウィンドウフロー制御のパラメータである。つまり、1 ラウンドトリップ時間あたりの、ウインドウサイズの線形増加量および乗算減少量である。R はネットワークのラウンドトリップ時間、p はパケット棄却率である。

ここで、式 (3) を違った視点から眺めてみる。すると、AIMD 型のウィンドウフロー制御を用いることにより、すべてのフローに対して帯域を公平に配分するだけでなく、「それぞれのフローに対して帯域を任意の比率で配分できる」ことが分かる。つまり、式 (3) は「ネットワークのラウンドトリップ時間 R およびパケット棄却率 p に応じて、パラメータ a および b を適切に設定すれば、フローのスループットを任意の値に制御することができる」ことを意味している。

ここで複数の VPN フローを考える。i 番目の VPN フローに対応する AIMD 型のウィンドウフロー制御のパラメータ a および b を、それぞれ a_i および b_i とする。同様に、i 番目の VPN スループットを T_i 、i 番目の VPN のラウンドトリップ時間、パケット棄却率を、R_i 、p_i などと表記する。

まず、VPN フロー i および VPN フロー j 間の公平性を考える。ここで、$R_i/R_j = \gamma$ 、$p_i/p_j = \delta$ であれば、式 (3) より、T_i および T_j の比 η は次式で与えられる。

$$\eta = \frac{T_i}{T_j} \quad (5)$$

$$\simeq \sqrt{\frac{a_i b_j (2-b_i)}{a_j b_i (2-b_j)\gamma^2 \delta}} \quad (6)$$

従って、上式で与えられる η が望む値となるように、a および b を設定することにより、任意の公平性を実現できる。ただし、式 (1) を満たす a および b の組み

合せは無数に存在するため、実際には、その中からAIMD 型のウィンドウフロー制御の過渡特性を考慮して決定する必要がある。

なお、任意の公平性を実現するようにパラメータ a および b を設定するためには、全ての VPN フローのラウンドトリップ時間 R_i およびパケット棄却率 p_i が既知でなければならない。提案する IP-VPN 公平性制御では、PE ルータ間でフィードバック情報を交換し、各 VPN フローのラウンドトリップ時間 R_i およびパケット棄却率 p_i を測定する。なお、 PE ルータ間のラウンドトリップ時間やパケット棄却率が分かれば十分であり、エンドホスト間のラウンドトリップ時間やパケット棄却率は必要ないことに注意されたい。

4.3.2 VPN 内公平性の実現

VPN 内公平性は、エンド–エンド間で動作する、TCP の輻輳制御機構をそのまま利用することで実現する。つまり、 IP-VPN 公平性制御機構は、PE ルータにおいて、VPN 内に収容されている各フローを識別しない。VPN 内に収容されているエンドホスト上で動作する、TCP の輻輳制御機構を利用することによって VPN 内公平性を実現する。

現在、インターネット上のトラヒックの 90 % 以上が TCP によって転送されているため、エンドホスト上で動作する、TCP の輻輳制御機構を利用するという方法は非常に有効であると考えられる。TCP の輻輳回避フェーズは AIMD 型のウィンドウフロー制御を採用しているため、ラウンドトリップ時間およびパケット棄却率が等しい環境下では、ボトルネックリンクの帯域を公平に共有することができる。特に、同じ VPN 内に収容されているすべてのフローは、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ことが期待できるため、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。

なお、本方式の欠点としては、同じ VPN 内に収容されている、プロトコル種別の異なるトラヒック間 (例えば、TCP トラヒックと UDP トラヒック) の公平性は実現できないという点が挙げられる。しかし、 IP-VPN サービスの目的を考えると、VPN 内公平性の厳密な制御は、VPN を契約している顧客が管理している、カスタマエッジルータで行うべきであろう。例えば、カスタマエッジルータにおいて、プロトコルごとの優先制御などを行うことにより、より厳密な VPN 内公平性が実現できると考えられる。

提案する I2VFC では、PE ルータ間で AIMD 型のウィンドウフロー制御が動作し、エンドホスト間で TCP のウィンドウフロー制御が動作する。つまり、 PE ルータ間、およびエンドホスト間で、二重にフィードバック型のウィンドウフロー制御が動作することになる。TCP over ABR でさまざまな問題点が指摘されたように [13] 、複数のフィードバック制御が相互干渉することにより、全体の性能が低下してしまう危険性も存在する。この制御の干渉は、制御のタイムスケールを変えることで回避できると考えられる。

TCP の輻輳回避フェーズは、1 ラウンドトリップ時間あたりのウインドウサイズの増加量と減少量が $a = 1$ および $b = 1/2$ である AIMD 型のウィンドウフロー制御と考えられる。I2VFC の ウィンドウフロー制御は、1 ラウンドトリップ時間あたりのウインドウサイズの増加量 a と減少量 b を TCP よりも小さな値に設定する。これにより、I2VFC のウインドウサイズの変化を TCP より緩やかにし、2 種類のウィンドウフロー制御のタイムスケールを変化させ制御の相互干渉を回避する。制御の相互干渉の程度とウインドウサイズの増加量 a と減少量 b の関係については、6.1 で検討している。

4.3.3 導入容易性およびスケーラビリティの実現

提案する I2VFC は、サービスプロバイダの IP ネットワークへの導入を容易にするため、PE ルータのみを変更することによって IP-VPN 公平性制御を実現する。AIMD 型のウィンドウフロー制御は、エンド–エンド間で動作する輻輳制御機構である。提案する I2VFC では、入側 PE ルータおよび出側 PE ルータ間で AIMD 型のウィンドウフロー制御を行うため、既存のコアルータに変更を加える必要がない。また、エンドホスト上で動作する、TCP の輻輳制御機構を利用することによって VPN 内公平性を実現するため、エンドホストにも変更を加える必要がない。

また、提案する I2VFC は、PE ルータにおいて、VPN 内に収容されている各フローを識別しないことにより、高いスケーラビリティを実現する。逆に、VPN に収容されている各フローを識別し、各フローに対し何らかのフロー制御を適用することで、UDP と TCP も公平となる VPN 内公平性を実現することも可能と考えられる。しかしながら、個々のフローを識別する機構は、パケットのトランスポート層の情報が必要であり、PE ルータの処理負荷が高くなる。また、IPsec などで、IP パケットが暗号化されている場合は、必要な情報が得られないため、VPN 内に収容されている各フ

ローの識別が困難である。I2VFC は、PE ルータにおいて、VPN 内に収容されている各フローを識別する必要がないため、PE ルータの高速化が可能である。また、 IPsec などを用いて、IP パケットのペイロード部分が暗号化されている場合でも、I2VFC は問題なく動作する。

5. 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 [9, 14]。

$$F = \frac{(\sum_i^N \frac{x_i}{r_i})^2}{N\sum_i^N (\frac{x_i}{r_i})^2} \tag{7}$$

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。

シミュレーションは、OPNET Modeler 9.1A [15] 上に I2VFC を実装して実行した。シミュレーション時間は 45 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。

5.1 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 2) 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 $t = 0$ [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。バックグラウ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 $\Delta = 4$ 、リンクの伝搬遅延 $5.06 \times 10^{-6}[s]$。

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

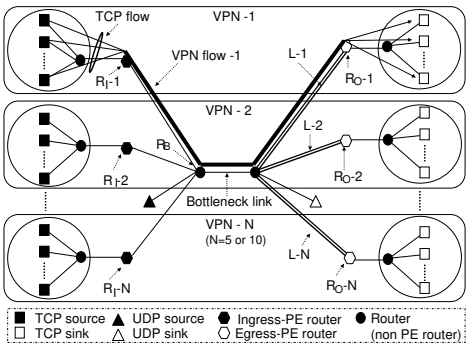


図 2 Network topology with single bottleneck link

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]
L-1	0.050
L-2	0.025
L-3	0.075
L-4	0.050
L-5	0.025

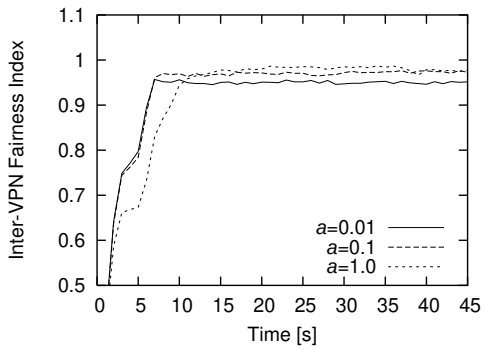


図 3 Evolution of inter-VPN fairness index (multiplicative decrease factor $b = 0.1$ for VPN flow 1)

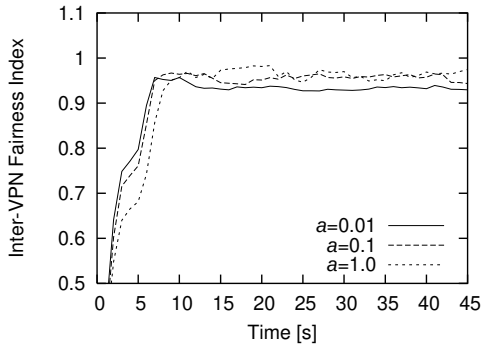


図 4 Evolution of inter-VPN fairness index (multiplicative decrease factor $b = 0.25$ for VPN flow 1)

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 3 および図 4 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、式 (6) 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 (7) 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。なお、すべてのシミュレーションにおいて、ボトルネックリンクの利用率はほぼ 100%であった。また、同様のシミュレーションを I2VFC を使用しないで行った場合も、ボトルネックリンクの利用率はほぼ 100%であった。

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 3 および図 4 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。

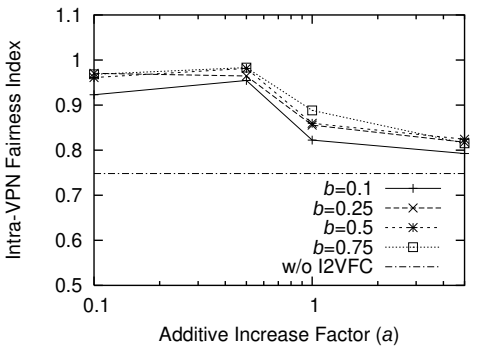


図 5 Weighted fairness index for intra-VPN fairness (2 TCP connections in each VPN flow)

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 $a = 0.1, 0.5, 1.0, 5.0$ と変化させ、ウインドウサイズの乗算減少量を $b = 0.1, 0.25, 0.5, 0.75$ 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 5 および図 6 に示す。図 5 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 6 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

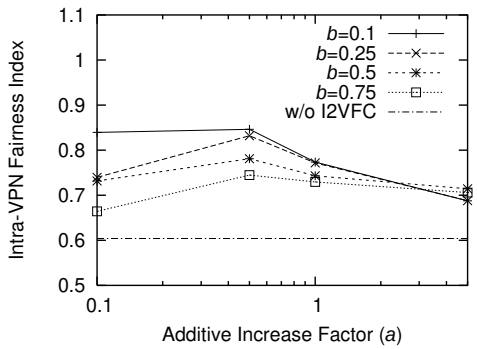


図 6 Weighted fairness index for intra-VPN fairness (10 TCP connections in each VPN flow)

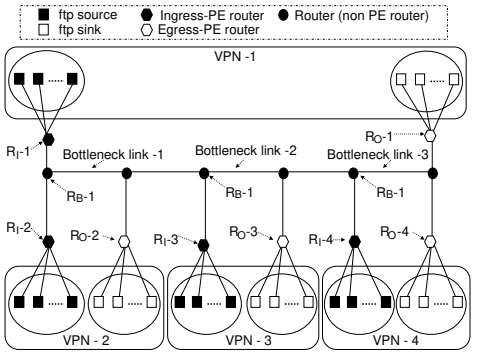


図 7 Network topology with multiple bottleneck links

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

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 5 および図 6 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 b の値が小さい場合は、I2VFC のウィンドウフロー制御と TCP のウィンドウフロー制御の干渉が発生していないためと考えられる。逆に、a および b の値が大きくなり、特に $a \geq 1$ および $b \geq 0.5$ になると、VPN 内公平性が低下していくことが分かる。これは、制御の干渉が発生しているためと考えられる。

図 5 および図 6 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 5 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。

5.2 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 7) 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 の重みを表 3 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。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 、リンクの伝搬遅延 $5.06 \times 10^{-6}[s]$。とした。

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

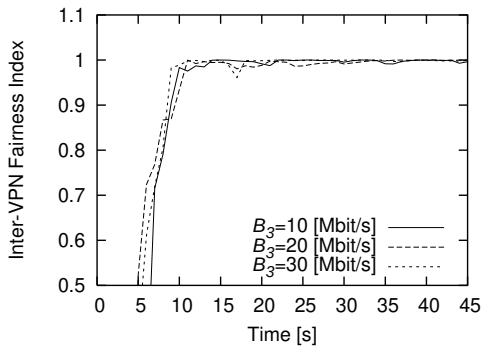


図 8　Evolution of weighted fairness index for inter-VPN fairness (case of multiple bottleneck links)

トリップ時間をもとに、式 (6) を満たす値に設定した。この時の、VPN 間公平性の公平性指標の時間的変動を図 8 に示す。

図 8 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 8 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。

6. プロトタイプシステムによる評価

6.1 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 9)。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。

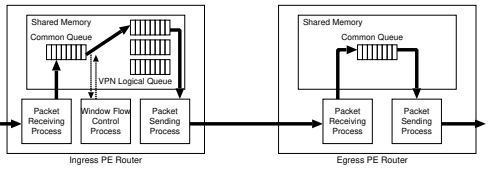


図 9　I2VFC prototype system overview

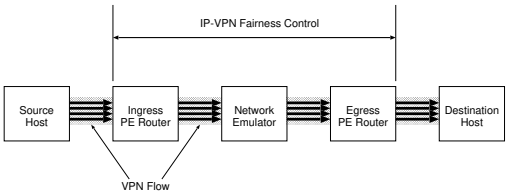


図 10　Network topology used in prototype system experiments

6.2 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 10 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 [16] 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 [17] 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 4 に示す。なお、特に断りのない限り、実験

表 4　実験に使用した機器の仕様
Table 4　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	384 Mbyte	Linux 2.4.22	e100-2.3.18
受信側ホスト	Celeron 1.06 GHz	384 Mbyte	Linux 2.4.22	e100-2.3.18
入側 PE ルータ	Pentium4 1.70 GHz	256 Mbyte	Linux 2.4.20	e1000-4.4.19
出側 PE ルータ	Celeron 2.00 GHz	512 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	256 Mbyte	FreeBSD 5.2.1	fxp, tx

表 5　実験におけるパラメータ設定
Table 5　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]

では表 5 に示すパラメータ設定を用いた。

6.3　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 11 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 $t = 30[s]$ 以降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 3) 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。

プロトタイプシステムを用いた実験の結果 (図 11) 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 5 では $F = 0.92$) ともおよそ一致している。

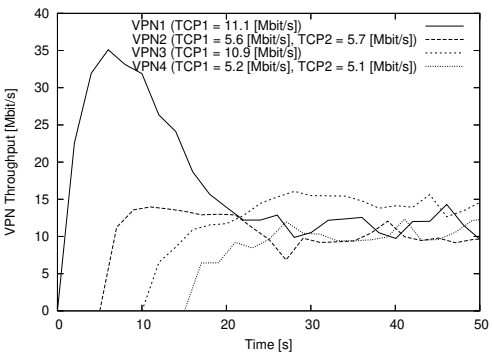


図 11　Evolution of each instantenous VPN throghput

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。

6.4　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

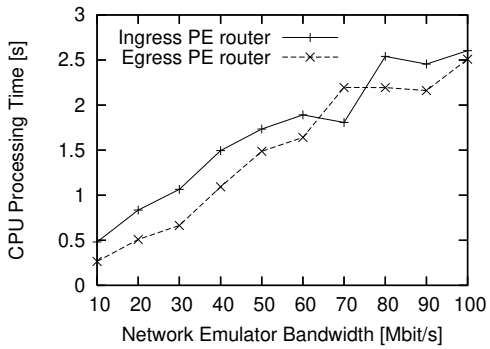


図 12 Relation between network emulator bandwidth and total CPU processing time consumed by ingress/egress PE router

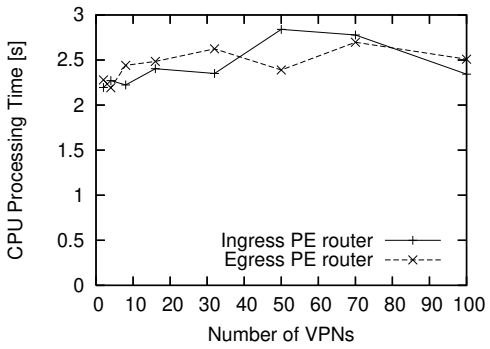


図 13 Relation between the number of VPN flows and total CPU processing time consumed by ingress/egress PE router

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 5 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 12 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 13 に示す。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、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 14 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。

7. ま と め

本稿では、まず、公平な IP-VPN サービスを実現するための制御機構に対する設計目標を議論した。具体的には、以下の 4 つの設計目標を掲げ、それぞれの設計目標の必要性および要求条件を議論した。

- (1) VPN 間公平性 (inter-VPN fairness) を実現
- (2) VPN 内公平性 (intra-VPN fairness) を実現
- (3) サービスプロバイダの IP ネットワークへの導入が容易
- (4) 転送速度 / VPN 数に関して高いスケーラビ

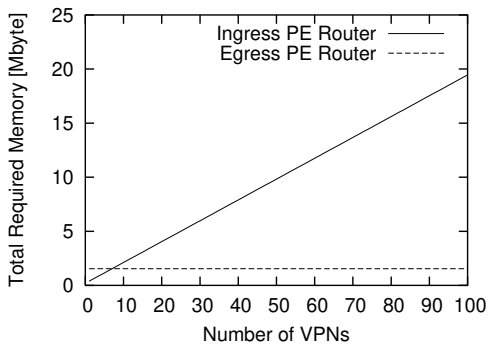


図 14 Relation between the number of VPN flows and memory usage in ingress/egress PE router

リティを実現

その後、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。I2VFC は、IP-VPN 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。I2VFC では、AIMD 型ウィンドウフロー制御の解析結果を利用することにより、VPN 間公平性の基準を、IP-VPN のサービスプロバイダが自由に規定できるという点が特徴である。また、PE ルータのみを変更するだけでよく、既存の IP ネットワークへ容易に導入が可能である。I2VFC が、4 つの設計目標をどのように実現しているかを議論し、さらに、I2VFC の動作アルゴリズムを説明した。

また、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 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 が転送速度および収容する 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。

今後の課題は、VoIP などのリアルタイム系トラヒックをどのように扱うかである。将来、これらのトラヒックが増加すると考えられる。そこで、例えば、I2VFC では、これらのトラヒックを別の VPN フローとして制御するなどの対策が必要と考えられる。

文　献

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

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

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

[4] T. Braun, M. Guenter, and I. Khalil, “Management of quality of service enabled VPNs,” *IEEE Communications Magazine*, May 2001.

[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.-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.

[7] I. Khalil and T. Braun, “Edge provisioning and fairness in VPN-DiffServ networks,” *JNSM*, vol. 10, pp. 11–38, Mar. 2002.

[8] 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.

[9] 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.

[10] 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.

[11] N. Kavak, “Ericsson’s network-based IP-VPN solutions,” *Ericsson Review No.3*, pp. 178–191, 2000.

[12] S. Floyd, M. Handley, and J. Padhye, “A comparison of equation-based and AIMD congestion control,” *available at http://www.icir.org/tfrc/*, 2000.

[13] 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.

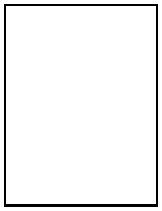
[14] R. Jain, *The Art of Computer Systems Performance Analysis*. New York: Wiley-Interscience, Apr. 1991.

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

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

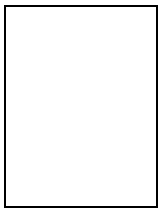
[17] L. Rizzo, “Dummynet: a simple approach to the evaluation of network protocols,” *ACM Computer Communication Review*, vol. 27, pp. 31–41, Jan. 1997.

（平成 xx 年 xx 月 xx 日受付）



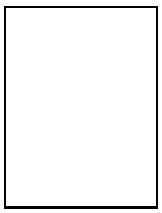
本田　　治　（正員）

平成 10 年大阪大学基礎工学部情報工学科退学．平成 12 年同大学大学院博士前期課程修了．平成 17 年同大学大学院博士後期課程退学．現在，同大学大学院情報科学研究科助手．分散システム，大規模ネットワークに関する研究に従事



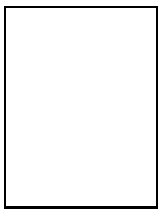
大崎　博之　（正員）

平成 7 年大阪大学大学院基礎工学研究科物理系専攻博士前期課程 修了．同年，日本学術振興会特別研究員．平成 9 年大阪大学大学院基礎工学研究科物理系専攻博士後期課程 修了．同年，大阪大学大学院基礎工学研究科情報数理系助手．平成 11 年大阪大学情報処理教育センター助手．平成 12 年大阪大学サイバーメディアセンター助手．平成 14 年大阪大学大学院情報科学研究科情報ネットワーク学専攻助教授．工博．この間，高速ネットワークにおけるトラヒック制御などの研究に従事．IEEE，IEICE 各会員．



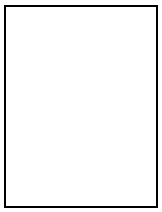
今瀬　　真　（正員）

昭５０阪大・基礎工・情報卒．昭５２年同大大学院修士課程了．同年日本電信電話公社武蔵野研究所入所．ＮＴＴマルチメディアネットワーク研究部長などを歴任．平成 14 年４月大阪大学大学院情報ネットワーク学専攻教授．現在、同職．ネットワーク理論、分散アルゴリズム、情報ネットワークなどの研究、開発に従事．工博．情報処理学会、応用数理学会各会員．



村山　純一　（正員）

平成元年・早大・理工卒．平成３年・同修士課程了．同年・NTT 入社．以来、IP-VPN サービスプラットフォーム、テラビット級スーパーネットワークの研究開発等に従事．現在、NTT 情報流通プラットフォーム研究所・セキュアコミュニケーション基盤プロジェクト・主任研究員．



松田　和浩　（正員）

昭和 58 北大・工・電子卒．昭和 60 同大大学院電子工学専攻修士課程了．同年 NTT に入社．以来，LSI CAD システム，プロトコル処理用 LSI，コンテンツ・デリバリ・ネットワークの研究開発に従事．現在，レイヤ２／３ VPN の研究開発に従事．NTT 情報流通プラットフォーム研究所主幹研究員．IEEE 会員．