

# 動的ネットワークにおける流量制御システムの保守性と適用 範囲の改良及び性能評価

渡邊 将一郎<sup>†</sup> 北形 元<sup>†</sup> 菅沼 拓夫<sup>†</sup> 木下 哲男<sup>\*</sup> 白鳥 則郎<sup>†</sup>

<sup>†</sup> 東北大学電気通信研究所

E-mail: {mash,minatsu,suganuma,norio}@shiratori.riec.tohoku.ac.jp

<sup>\*</sup> 東北大学情報シナジーセンター

E-mail: kino@riec.tohoku.ac.jp

我々は、アプリケーションの利用状況に応じて優先度を決定し、流量を調整するアプリケーション指向フロー制御システムを提案してきた。本稿では提案システムの保守性と適用範囲の向上を目的とし、(S1)QoS 解決知識の分離、(S2)FN 層・LN 層間の効果的な接続、という2つの改良手法の提案を行う。更に評価実験を通じ、提案手法の有効性を示す。

## Maintainability and Availability Improvement and Evaluation of Flow Control System in Dynamic Networking

Shoichiro Watanabe<sup>†</sup>, Gen Kitagata<sup>†</sup>, Takuo Suganuma<sup>†</sup>, Tetsuo Kinoshita<sup>\*</sup>  
and Norio Shiratori<sup>†</sup>

<sup>†</sup>Research Institute of Electrical Communication, Tohoku University

E-mail: {mash,minatsu,suganuma,norio}@shiratori.riec.tohoku.ac.jp

<sup>\*</sup>Information Synergy Center, Tohoku University

E-mail: kino@riec.tohoku.ac.jp

We have proposed application-oriented flow control system which decides priority of flow according to application utilization situation and controls application flow. In this paper, for the purpose of improving the maintainability/availability of our system, we propose two improving methods, which are (S1)separation QoS resolution knowledge,(S2)effective connection between FN layer and LN layer. Furthermore, through the evaluation experiment, we show validity of our proposal.

### 1 はじめに

デジタル回線等の比較的高品質な接続環境の普及に伴い、家庭やオフィス等からのインターネット利用が増加している。一般的に家庭等から用いられている ISDN や ADSL 等のアクセスネットワークは、上流ネットワークの帯域に比べ、狭域な環境になっている。このようなネットワーク環境にて複数のアプリケーションを同時に利用する場合、利用者が最も優先して使いたいアプリケーションのトラフィックが他のアプリケーションのトラフィックにより圧迫され、サービス品質が低下する可能性がある。具体例として、web ブラウザとビデオストリーミングアプリケーションを同時に使用している場合に、web ブラウザによりファイルをダウンロード中に、ビデオストリーミングの品質が低下するなどが挙げられる。このため、アプリケーションのフロー単位での流量

調整など、きめの細かいネットワークサービス品質、すなわちネットワーク QoS(Quality of Service) の保証の実現が重要な課題となっている [1]。そこで我々は、次世代ネットワークアーキテクチャとして提唱する動的ネットワークアーキテクチャにおける流量制御モデルの提案を行っている [2, 3]。

動的ネットワークアーキテクチャとは、アプリケーションと論理ネットワーク間のギャップによる生じる種々の問題を解決するために、我々が提案している新たなネットワークアーキテクチャである。動的ネットワークアーキテクチャの特徴は、利用者/アプリケーションレベルでの変動や、基盤となる現行の IP ネットワークなどの論理ネットワークレベルでの変動を、自律的に監視・調整・制御する「やわらかいネットワーク層 (Flexible Network layer : FN 層)」と呼ぶ機能層を新たにアプリケーション層 (Application layer : AP 層) と論理ネットワーク層 (Logical Network layer : LN 層) 間に

導入する点にある。FN層は、LN層の上位で稼働するため、すべてのIPルータに実装する必要は無く、広域ネットワーク上に点在するように配置されていても、その性能を発揮できる。このため、本アーキテクチャは高域ネットワークヘシームレスに適用可能である。本稿が取り上げる流量制御システムは、やわらかいネットワーク層を持つ機能群のうち、QoS制御機能とネットワーク運用機能に焦点を当てている。

文献[2]で提案した流量制御モデルの特徴としては、(1)論理ネットワーク層の上に位置するやわらかいネットワーク層内で流量調整を行うため、IPルータ等への特別な実装を必要とせず、利用するネットワーク環境が限定されない点、(2)やわらかいネットワーク層からアプリケーションの状態を監視し、フローの優先度を決定するため、既存アプリケーションに特別な実装をせずに利用可能である点である。しかしながら、プロトタイプの実装を通じて明らかになった新たな問題点として、(P1)新規アプリケーション対応の際、エージェント内の知識の変更が必要であり、保守性が低い、という問題点と、(2)LN層とFN層の統一的な連結方法が確立されていない事により、FN層内の流量制御機能とアプリケーションとの接続性が不十分であり、適用範囲が狭い、という問題点がある。そこで本稿では、2つの問題それぞれに対し、(S1)「QoS解決知識の分離」、(S2)「FN層・LN層間の効果的な接続」という解決法を提案し、プロトタイプの実装・評価実験を通じ、提案手法の有効性を示す。

## 2 流量制御システム

### 2.1 FN層における流量制御システム

IPネットワークにおけるQoS保証の困難性の解決と、ユーザ指向性の実現を目標に、我々はFN層における流量制御システムを提案している[2]。本システムのエージェント構成を図1に示す。

本システムではまず、IntServ等により論理ネットワーク層レベルで行われていたフロー制御を、FN層、つまり従来のAP層レベルで行うことで、QoS保証機能を持たない一般的なIPネットワークでのQoS制御を実現している。一般に、上流ネットワークに比べアクセスネットワークの方が狭帯域である家庭や小規模オフィスからのインターネット接続環境や、モバイル環境においては、上流ネットワークとアクセスネットワークを繋ぐルータへ流入トラフィックが集中する。このため、アクセスネットワークへ流入する時点で、フロー毎の流量を調整することが効果的である。そこで本システムでは、図1に示すように、FNゲートウェイからFNクライアントに向かってトラフィックが発生する際、FNゲートウェイ上で一旦FN層までトラフィックを引き上げ、流

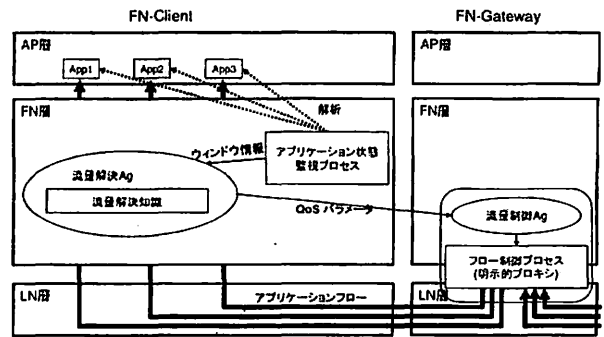


図1: 動的ネットワークにおける流量制御システム

量制御エージェントがフロー毎にキューイングする。そしてキューからデータを取り出す頻度をアプリケーション毎の優先度に応じて制御することで、FN層における流量制御を実現している。

また、FNクライアント側では、アプリケーション状況監視が各アプリケーションのウィンドウの状態を監視し、得られたウィンドウ情報をもとに、流量解決知識を持った流量解決エージェントがアプリケーションのフロー間の優先度からなるQoSパラメータを生成する。ここで、ウィンドウ情報とは、アプリケーションウィンドウの重なり順序(z軸)や、最大化/最小化等を含むウィンドウの大きさなどである。ウィンドウの状態から間接的に利用者要求を読み取ることで、利用者への負担が少ないQoS制御システムが実現できる。

### 2.2 FN層における流量制御システムの問題分析

上述した流量制御システムは、QoS保証機能を持たないEthernet等の一般的なネットワークにおいても流量制御が可能、という点と、アプリケーションのウィンドウ状態から利用者のアプリケーションに対する優先度要求を間接的に取得することによりQoS指定にかかる負担が小さい、という点の、2つの利点を持っている。しかしプロトタイプの実装/実験を通して、次の二つの問題点(P1)、(P2)が明らかとなっている。

(P1) 新規アプリケーション対応の際、エージェント内の知識の変更が必要であり、保守性が低い

(P2) LN層とFN層の統一的な連結方法が確立されていない事により、FN層内の流量制御機能とアプリケーションとの接続性が不十分であり、適用範囲が狭い

まず問題点(P1)について問題分析を行う。図1に示すように、QoS解決エージェントは、ウィンドウ監視プロセスからのウィンドウ情報と、自身が持つQoS解決

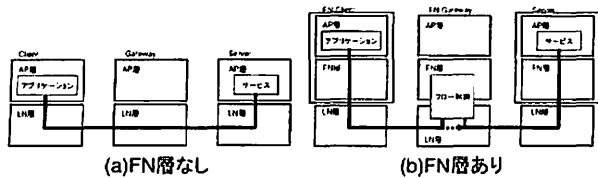


図 2: FN 層あり/なしによるフローの違い

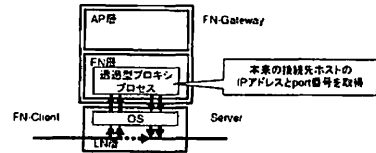


図 4: 透過型プロキシを用いたフローの引き込み

### 3 流量制御システムの改良

#### 3.1 改良手法の提案

2.2 で述べた問題点 (P1), (P2) を以下の解決手法 (S1), (S2) によりそれぞれ解決する。(S1) QoS 解決知識の分離まず、問題点 (P1) を解決するために、QoS 解決エージェントが一元的に管理していた QoS 解決知識から、アプリケーション毎に個有の知識を分離する。そしてアプリケーション毎に個有の性質を保持するアプリケーション QoS エージェントを新たに導入し、QoS 解決エージェントとの協調により QoS パラメータを決定する。具体的には、アプリケーションそれぞれの QoS エージェントに、アプリケーション識別子、ウィンドウ識別子、ネットワークリソース要求 (帯域など) から成る QoS 要求知識を持たせ、それら QoS エージェントが、QoS 解決エージェントと関係・協調し、QoS パラメータを生成する。このように、アプリケーション個有の性質をそれぞれの QoS エージェントに持たせることで、新規アプリケーションに対応する場合に QoS エージェントを追加するだけで済み、QoS 解決エージェントが持つ知識に変更を行う必要がなくなり、システムの保守性の向上が期待できる。(S2) FN 層・LN 層間の効果的な接続問題点 (P2) を解決するためには、ゲートウェイ上の LN 層を流れるフローを、アプリケーションに対して透過的に FN 層内へ引き込むことが有効である。そこで FN 層/LN 層間に、透過型プロキシプロセスを用いた新たな接続方式を導入する (図 4)。透過型プロキシとは、LN 層が持つ IP フィルタ機能を利用し、FN 層を通過しようとするフローをプロキシプロセスへリダイレクトさせる既存技術である。プロキシプロセスはリダイレクトされたフローから、本来の接続先サーバの IP アドレス、ポート番号を取得し、本来の接続先へ新たな接続 (セッション) を確立する。その後、クライアントからのフローと新たに確立したセッションのフロー間のフォワーディングを行う。さらに、QoS 機能から伝達された QoS パラメータ、すなわちアプリケーション毎のフロー優先度に応じ、各フロー毎のフォワーディング速度を調整する。この時、FN クライアント上のアプリケーションは、本来の接続先に接続するだけでよい。このことにより、プロキシに対応していないアプリケーション使用時にも、FN 層における流量制御機能を適用可能となるため、アプリケー

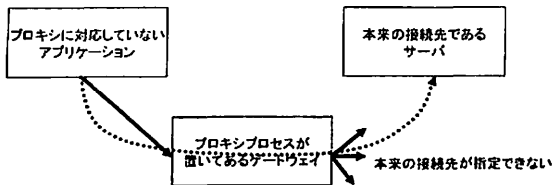


図 3: 従来型プロキシを用いる場合の、接続先指定に関する問題

知識を元に QoS パラメータを生成する。この QoS 解決知識は、if-then 形式のルールとして格納されている。アプリケーション個有の知識や解決知識が一元的に QoS 解決知識内に埋めこまれているために、新規アプリケーションに対応する場合、QoS 解決知識を大幅に書き直す必要があり、結果として新規アプリケーション対応のための保守コストが多くかかってしまう。

次に問題点 (P2) について問題分析を行う。通常、クライアントからのフローがゲートウェイを通過する際、そのフローはゲートウェイ上の LN 層のみを通過する (図 2(a))。しかし、図 2 から分かる通り文献 [2] で提案した流量制御システムでは FN 層により流量を制御するため、FN ゲートウェイにおいて、LN 層から FN 層へフローを引きあげる必要がある (図 2(b))。そのため、従来の AP 層に当たる層に設置したプロキシプロセスを用いて、クライアントアプリケーションからのフローを一旦 FN 層へ引き込んでいる。つまりアプリケーション側から見た場合、本来接続したいサーバに接続するのではなく、FN ゲートウェイ上のプロキシプロセスに接続するよう設定する必要がある。ここで例えば、ウェブブラウザのようにプロキシを用いることを前提に設計されているアプリケーションの場合、プロキシの存在するホストと、本来接続したいサーバの、2つのホストの IP アドレスとポート番号を指定できるので問題はないが、telnet 等、プロキシを用いることを前提としていないアプリケーションの場合、本来の接続先を指定することは困難であり、また指定するための仕組みを追加したとしても、利用者が何らかの追加的な手続きを行う必要がある (図 3)。

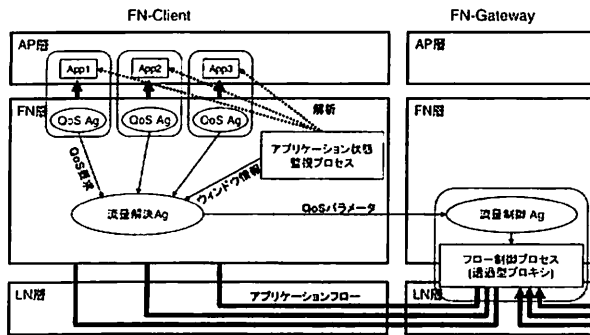


図 5: 改良後の FN 層における流量制御機能構成

```

(agent WMPlayerApp
(property
(create :author "mash")
)
)
(instrat_facts
(worker :name mash)
(chara :name WMPlayerApp port 80 priority 1024)
)
(knowledge
(rule inspector
(Msg :performative _INIT_1) = ?fact
-->
(inspect nonstop)
(bind ?agent (yp ClientAgent2))
(send
:performative mit
:to ?agent
:content (chara :name WMPlayerApp :port 80 :priority 1024))
)))

```

図 6: 新規アプリケーション対応時のテンプレート

ションに対する適用範囲が広がる。

## 4 設計と実装

### 4.1 設計

3章で提案した問題解決手法を FN 層における流量解決システムに適用し、設計を行った。図 5 に、解決手法を適用した機能構成を示す。また、アプリケーション用 QoS エージェントの記述を容易にするため、図 6 に示すようなテンプレートを用意した。新規に QoS エージェントを作成するために必要な作業は、図 6 中のイタリックの部分、すなわち、エージェント名(アプリケーション識別子)、アプリケーションが使用するポート番号、要求優先度である。

### 4.2 実装

前章の設計に基づき、提案方式を実現する試作システムの実装を行った。OS は Microsoft Windows2000, および Vine Linux 2.1 を使い、実装言語には Visual C++, Java 言語, およびエージェントフレームワークである DASH[4] を用いた。アプリケーション状態監視プロセスにおけるウィンドウ情報の取得には、Microsoft Windows のメッセージフック API を利用した。

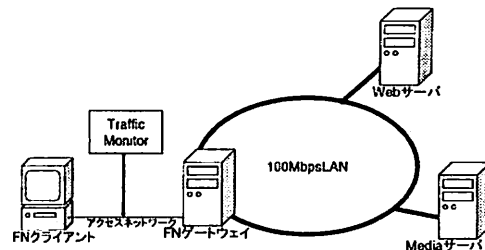


図 7: 実験環境

## 5 評価実験

前章の設計に基づき実装した試作システムを用いて、改良方式を適用した流量制御システムの有効性を検証するための 2 つの実験、すなわち、実験 1: アプリケーション利用状況に応じたスループット制御性能の評価と、実験 2: オーバーヘッドの評価を行った。

### 5.1 実験環境

図 7 に実験環境を示す。クライアントとゲートウェイには PC を用いた。クライアント用 PC の CPU は Pentium3 1.0GHz, OS は Microsoft Windows 2000 であった。ゲートウェイ用の PC の CPU は Pentium3 1.0GHz, OS は Vine Linux 2.1 であった。Web サーバ, Media サーバには Sun Microsystems の UltraSPARCStation (OS は Solaris8) を用いた。アクセスネットワーク、すなわちクライアントとゲートウェイ間のリンクには QoS 保証機能を持たない Ethernet や無線 LAN 等の一般的なリンク方式を使用し、ここにトラフィックを観測する Traffic Monitor を接続した。実験に使用するアプリケーションとしては、Web ブラウザと動画プレーヤをクライアントにて使用した。Web ブラウザ、動画プレーヤには、QoS 保証機能を持たない一般的なアプリケーションである、Microsoft Internet Explorer, Microsoft Media Player をそれぞれ用いた。

### 5.2 実験 1: アプリケーション利用状況に応じた流量調整の性能

提案方式を用いることで、利用者のアプリケーション利用状況(どのアプリケーションを優先的に使用しているかなど)に応じて適切な流量調整が行えることを検証するために、利用者が 2 つのアプリケーション、すなわち Web ブラウザと動画プレーヤを同時に使用し、それぞれのアプリケーションの優先度を変更した場合のスループットの変化と、優先度を変更してからスループットに変化が生じるまでの応答速度を測定した。実験条件を以下に示す。

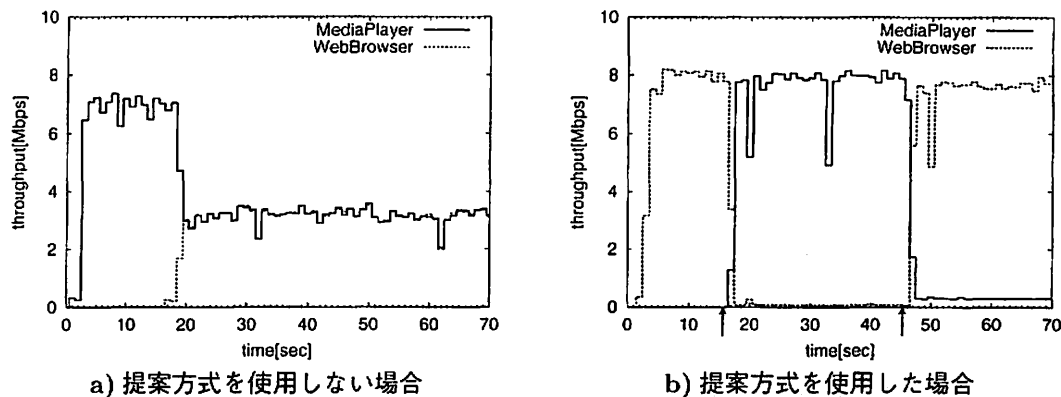


図 8: 実験結果 1: スループットの推移 (10Mbps)

### 実験条件

- アクセスネットワークのリンク帯域は、ゲートウェイより上流のネットワークの帯域に比べ十分狭い帯域として、10Mbps とする
- それぞれのアプリケーションは、プロキシを使わない設定で使用する
- クライアントにて、Web ブラウザによるファイルのダウンロードと、動画プレーヤによる動画のストリーミング再生を行う
- ストリーミングを行う動画のデータはビットレート 6Mbps の MPEG1 ファイルとする
- ゲートウェイと Web サーバ・動画サーバは 100Mbps の LAN で接続する

提案方式を使用しない場合と使用した場合の実験の結果を、図 8a),b) にそれぞれ示す。横軸は実験開始からの経過時間 (秒) を表す。縦軸は、ゲートウェイからクライアントへ流入する Web ブラウザと動画プレーヤのフローのスループットを 1 秒単位で示している。図 8b) の横軸下の矢印は、クライアントにおいて利用者がアプリケーションの優先度を変化させたタイミングを示している。

まず、提案方式を使用しない場合 (図 8a)), 利用者が動画プレーヤでストリーミング再生を開始すると (0-20 秒), 約 7Mbps のスループットが得られ、スムーズに動画が再生された。次に利用者が Web ブラウザを起動しファイルのダウンロードを開始すると (20 秒以降), 動画プレーヤと Web ブラウザが受信するデータのスループットはどちらも約 3.5Mbps となった。その結果、動画の再生に必要な 6Mbps の帯域が得られなくなり、音声の断絶や画像のフレーム落ち等が発生し、十分な再生品質が得られなくなった。

これに対し提案方式を使用した場合 (図 8b)), まず Web ブラウザでファイルのダウンロードを開始 (0-20 秒) し、その後動画プレーヤをアクティブにしてストリーミング再生を行ったところ (20-45 秒), 動画プレーヤ

が受信するデータのスループットは約 8Mbps に保たれ、クライアント上の 2 つのアプリケーションが同時にデータを受信している状況でも、スムーズな再生が得られることが確認された。これは、クライアント上のアプリケーション状態監視プロセスによる利用者が優先的に利用しているアプリケーションの検知、流量解決 Ag による QoS パラメータの導出、及びゲートウェイ上の流量制御 Ag によるフロー制御が有効に働いたことによる。

また、クライアントにおいて利用者がアプリケーションの優先度を変更してから、ゲートウェイからのフローの流入量が増えるまでの反応時間、すなわち図 8b) の横軸下に矢印で示した時間から優先されたアプリケーションのスループットが上昇するまでの遅延時間は、概ね 1.4 秒以下であった。これには、クライアント上のアプリケーション状態監視プロセスがアプリケーションの状況変化を検知し、流量解決 Ag が QoS パラメータ導出のための推論を行い、ゲートウェイ上の流量制御 Ag へ伝達されるまでの処理が、実用上十分短い時間内で行われることを示している。

### 5.3 実験 2: 流量調整に伴うオーバーヘッドの影響

提案方式のオーバーヘッドを評価するために、提案方式を使用した場合と使用しない場合、すなわち、ゲートウェイからクライアントへのフローのフォワーディングを FN 層で行った場合と、IP 層のみで行った場合の平均スループットを測定する。

#### 実験条件

- アクセスネットワークのリンク帯域は、115Kbps, 2Mbps, 10Mbps, 100Mbps の 4 種類を用意する
- クライアントとゲートウェイの MTU (Max Transfer Unit) は 1500bytes に設定する
- クライアントにて、Web ブラウザによるファイルのダウンロードを行う

- それぞれのアプリケーションは、プロキシを使わない設定で使用する
- ゲートウェイと Web サーバは 100Mbps の LAN で接続する

実験 2 の結果を表 1 に示す。なお、実験結果は 10 回の試行の平均である。表 1 より、全てのリンク帯域において、流量制御 Ag が FN 層で稼働することによるオーバーヘッドによって、クライアントへの到着パケット数は僅かに減少していることがわかる。しかし、上流ネットワークの帯域よりアクセスネットワークの帯域のほうが狭い 3 つのケース (115Kbps, 2Mbps, 10Mbps) では、提案方式が IP 層のみを使用した場合に比べてスループットが向上することが分かる。これは、ゲートウェイの IP 層のみでフォワーディングした場合、ゲートウェイに到着する IP パケットがそのままクライアントへフォワードされるのに対して、提案方式では流量制御 Ag がパケット単位ではなく連続したストリームとしてフローを扱うため、クライアントへフォワードされる時点で IP パケットが新たに生成されることによる。また、ゲートウェイより上流のリンク帯域に比べアクセスネットワーク (クライアントとゲートウェイ間) のリンク帯域のほうが狭い場合、一つの IP パケットに可能な限りのペイロードが格納され、結果としてペイロード長が大きくなり、狭帯域であるアクセスネットワークを効果的に利用可能となる。上記の結果より、提案手法は、上流ネットワークよりアクセスネットワークのほうが狭帯域である場合に特に効果的であることが確認された。すなわち、今後ますます増加するであろう、自宅やオフィスから ADSL 等の加入者回線を経由して接続する環境、また無線 LAN や携帯電話などを経由してネットワークへ接続するモバイル環境において、特に有効であるといえる。

#### 5.4 考察

5.2, 5.3 にて、提案手法の性能評価を行った。次に本稿で提案した改良手法に対し、定性的な議論を行う。まず改良手法 (S1) 「QoS 解決知識の分離」について、アプリケーション個々の情報を QoS エージェントに分離することにより、QoS エージェントの追加のみで新規アプリケーションへの対応が可能となった。さらに 4.1 でテンプレートを用意することで、最小限の書き換えのみで、新たなアプリケーション用の QoS エージェントを追加可能となり、システムの保守性が大幅に向上したことが分かる。次に提案手法 (S2) 「FN 層・LN 層間の効果的な接続」について、5.2, 5.3 の実験結果より、アプリケーションにプロキシに関する特別な設定を行わなくても、FN 層での流量制御を適用可能であることが実証された。このことにより、本来プロキシプロセスの使用を前提に設計されていないアプリケーションにも FN 層で

表 1: 平均スループット

a) IP 層のみ使用した場合			
リンク帯域 [bps]	ペイロード長 [bytes]	到達パケット数 [/s]	スループット [bps]
115K	1362	8.16	86.8K
2M	1365	119.45	1.24M
10M	1365	681.64	7.10M
100M	1365	5365.81	55.9M

b) 提案方式を使用した場合			
リンク帯域 [bps]	ペイロード長 [bytes]	到達パケット数 [/s]	スループット [bps]
115K	1445	7.83	88.4K
2M	1459	114.68	1.28M
10M	1458	675.35	7.51M
100M	1449	3641.28	40.3M

の流量制御が適用可能となり、適用範囲が広がったと言える。

## 6 むすび

我々は、アプリケーションの利用状況に応じて優先度を決定し、流量を調整するアプリケーション指向フロー制御システムを提案してきた。本稿では提案システムの保守性と適用範囲の向上を目的とし、(S1)QoS 解決知識の分離、(S2)FN 層・LN 層間の効果的な接続、という 2 つの改良手法の提案を行った。更に評価実験を通じ、提案手法の有効性を確認した。今後の課題として、ゲートウェイ下に複数のクライアントが存在する場合におけるクライアント間でのネットワークリソース配分方式の開発などがある。

## 参考文献

- [1] N. Nahrstedt and J.M. Smith: Design, Implementation, and Experiences of the OMEGA End-Point Architecture, IEEE J.Select. Areas Commun., Vol.14, No.7, pp.1263-1279 (1996).
- [2] 北形 元, 菅沼 拓夫, 木下 哲男, 白鳥 則郎: 動的ネットワークアーキテクチャにおける応用の一例: アプリケーション指向フロー制御システムの提案, 情報処理学会 第 9 回 DPS ワークショップ論文集, pp.103-108,(2001).
- [3] G. Kitagata, T. Suganuma and T. Kinoshita: Application-Oriented Flow Control in Agent-Based Network Middleware. Proc. of 5th Pacific Rim International Workshop on Multi-Agents (PRIMA2002), LNAI 2413 (Springer), pp.178-189, 2002.
- [4] Distributed Agent System based on Hybrid architecture, <http://www.agent-town.com/>.