Vol. 44 No. 3 情報処理学会論文誌 Mar. 2003

エージェント指向ミドルウェアにおける利用者指向流量調整方式の設計と実装

北形 元[†] 今野 将[†] 加藤貴司[†]
菅沼拓夫[†] 木下哲男[††]

ISDNやADSL等のデジタル回線の普及にともない，自宅等からのインターネット利用者が急増している．このような，上流のネットワークに比べ加入者回線の方が低速な環境においては，同時に利用しているアプリケーションのトラフィックにより，他のアプリケーションのトラフィックが圧迫され，サービス品質が低下する場合がある．そこで我々は，アプリケーション層と論理ネットワーク層を効果的に運用することを目的とした動的ネットワーキングアーキテクチャに関する研究を行っている．本論文では，同アーキテクチャを効果的に活用した利用者指向流量調整方式を提案する．本方式は，利用者のアプリケーションの利用状況を監視し，これに応じてアプリケーションのフローごとの優先度を決定し流量を調整する方式であり，次のような特徴を持つ．(1) 論理ネットワーク層の上位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．(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が不要である．本論文では，提案方式を実現する試作システムの設計および実装について述べ，実験を通じ提案方式の有効性を示す．

Design and Implementation of User-centered Flow Control Method on Agent Oriented Middleware

Gen Kitagata,[†] Susumu Konno,[†] Takashi Katoh,[†]
Takuo Suganuma[†] and Tetsuo Kinoshita[††]

As the widely spread of digital subscriber lines such as ISDN and ADSL, people who access to the Net from their home or office room are rapidly increasing. When we use a number of applications at the same time in such a narrow bandwidth network environment, the data flows of some applications such would be pushed out due to the heavy traffic applications. As a result, QoS of the former applications become low. To deal with this problem, we have been promoting the research of Dynamic Networking Architecture to bridge the application layer and the logical network layer effectively. In this paper, we propose the User-centered Flow Control Method applying the architecture effectively. The method monitors applications' states, decides priority of each application's network flow and controls the flow. The system has two remarkable characteristics. (1) Because the system controls network flow in the Flexible Network Layer which lays on the logical network layer, it requires no special implementation of network environments. (2) Because the system decides priority of each application's flow by monitoring the applications, it requires no special implementation of applications. In this paper, we show design and implementation of the prototype system, and verify the effectiveness of this method by experimental results.

1. まえがき

近年，ISDNやADSL等のデジタル回線の普及にともない，自宅等からのインターネット利用者が急増している．このような，上流のネットワークに比べ加入者回線の方が低速な環境においては，同時に利用しているアプリケーションのトラフィックにより，他のアプリケーションのトラフィックが圧迫され，サービス品質が低下する場合がある．具体的には，Webブラウザによるファイル転送とTelnetによる遠隔ログインを同時に行った際に生じるTelnetのキャラクタエ

† 東北大学電気通信研究所
Research Institute of Electrical Communication, Tohoku University
†† 東北大学情報シナジーセンター
Information Synergy Center, Tohoku University

コー遅延等のレスポンス低下や，ファイル転送とビデオストリーミング受信を同時に行った際のビデオストリーミングのフレーム落ち等があげられる．このため，アプリケーションのフロー単位での流量調整等，きめの細かいネットワークサービス品質（以後 QoS と表記）制御の実現が重要な課題となっている[1),2)]．

エンドツーエンドでの QoS 制御[3)]を行う代表的な方式として，IETF により IntServ[4)]や DiffServ[5)]が提案されている．IntServ は RSVP[6)]等のシグナリングプロトコルを用いて，エンドノード間での QoS 保証を行う仕組みを提供し，DiffServ はサービスクラスごとの相対的な優先制御を行うことにより，IntServ 利用時のスケーラビリティの問題を解決している．また，利用者要求やネットワーク負荷が変動する環境下において，ビデオ会議システムのような利用者間マルチメディア通信アプリケーションの適応的な QoS 制御を目的としたフレームワークの研究もさかんである[7),8)]．

しかしながら，IntServ や DiffServ を用いるには経路上の IP ルータに QoS 保証のための特別な実装を必要とするため，利用できるネットワーク環境が限定され，Ethernet や無線 LAN 等の一般的なネットワークでの使用は難しい．また，すでに提案されているアプリケーション層における QoS 制御フレームワークにおいても，アプリケーションにフレームワークを利用するためのなんらかのライブラリを組み込む必要がある．そのため，利用者にとって有用な既存アプリケーションの多くが，これらのフレームワークをそのまま利用できないという問題がある．

我々は，上述の問題を含む，アプリケーションと論理ネットワーク間の機能のギャップにより生じる種々の問題を解決するため，動的ネットワーキングアーキテクチャに関する研究を行っている[9),10)]．動的ネットワーキング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の変動（e.g. 利用者要求の変化や，CPU 負荷の変動等）や，通信基盤となる IP ネットワーク等の論理ネットワークレベルでの変動（e.g. 使用可能な空き帯域の変化等）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」と呼ぶネットワークミドルウェアを新たに導入する点にある．

やわらかいネットワーク層は大きく分けて 4 つのユニット（ミドルウェアサービスユニット，高レベル通信ユニット，QoS 制御ユニット，ネットワーク運用ユニット）から構成されるが（**図 2**），本論文では，このうち QoS 制御ユニットとネットワーク運用ユニットに焦点をあて，同ユニットを効果的に活用した利用者指向流量調整方式を提案し，上述のネットワーク QoS 保証方式の問題を解決する．提案方式の特徴は，(1) 論理ネットワーク層の上位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という点である．本論文では，提案方式を実現するエージェント指向ミドルウェアの設計および，試作システムの実装を行い，実験を通じ提案方式の有効性を検証する．

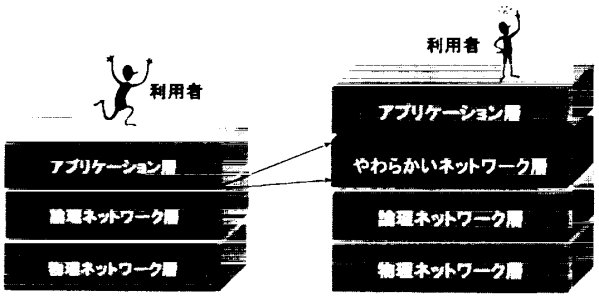
以下，まず 2 章で，動的ネットワーキングの概要を述べる．次に 3 章で利用者指向流量調整方式の提案を行い，4 章で試作システムの設計と実装について述べる．さらに，5 章で試作システムを用いた実験とその評価を行い，提案方式の有効性を示す．

2. 動的ネットワーキング

従来のネットワークアプリケーションの多くは，TCP/UDP 等が提供するトランスポート通信サービスを直接用いて構成されてきた．しかしながら，これらトランスポート通信サービスを直接利用することによる，次のような問題が生じつつある．まず，通信相手との通信路確保のために，通信相手に対応する IP アドレス，ポート番号を推定しなければならないといった接続時の不自由さがあげられる．次に，トランスポート通信サービスが提供する QoS 制御等の高度な機能を利用しようとする場合，個々のアプリケーションが独自にそれらの機能への対応を行わなければならず，アプリケーションの複雑化をまねくという問題があげられる．これらの問題は，様々な通信基盤上で頻繁に移動しながら通信を行う将来のネットワーク利用形態を考えると，今後さらに大きな問題となるであろう．これらの問題を解決するためには，ソフトウェア工学的，あるいは利用者指向的な観点を考慮した新たなネットワークアーキテクチャを構成する必要がある．

そこで我々は，これらの問題点を解決するために，これまで行ってきたやわらかいネットワークに関する研究成果[11)~14)]に基づき，次世代広域ネットワーク環境のモデルとして，動的ネットワーキングアーキテクチャに関する研究を行っている[9),10)]．

図 1 に動的ネットワーキングアーキテクチャの層構成を示す．本アーキテクチャでは，IP ネットワーク等のサービス機能層である論理ネットワーク層（以下 LN 層と表記）とアプリケーション層（以下 AP 層と表記）の中間に位置するネットワークミドルウェアと



a) 従来型ネットワークアーキテクチャ　b) 動的ネットワーキングアーキテクチャ

図1 動的ネットワーキングアーキテクチャ
Fig. 1 Dynamic Networking Architecture.

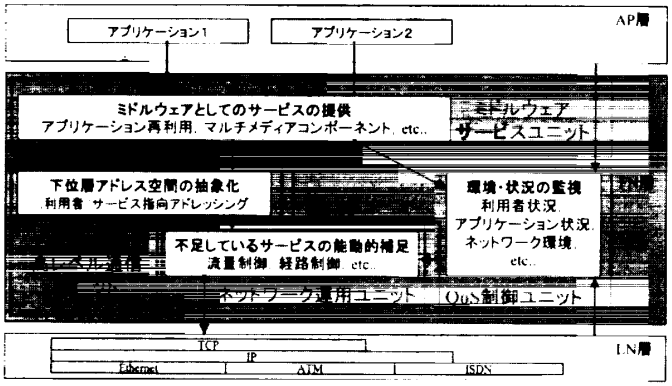


図2 やわらかいネットワーク層のユニット構成
Fig. 2 Units of Flexible Network Layer.

して，新たに「やわらかいネットワーク層（以下FN層と表記）」を導入する．FN層は，AP層からの要求とLN層のサービスを効果的に連結し，AP層とLN層の双方における大・小，一時的・長期的変動を自律的に調整する機能を持つ．以下に，図2に示すFN層の各ユニットの役割について概要を述べる．

（1）ミドルウェアサービスユニット

アプリケーションから共通的に利用される，品質調整をともなう動画，静止画，音声等のメディア送受信機能をコンポーネントとしてアプリケーションに提供する．

（2）高レベル通信ユニット

LN層で用いられるIPアドレスのようなホスト指向のアドレス形式ではなく，より利用者に近い形式（人物名やサービスの種類等）で，利用者が接続相手を指定する機能を提供する．

（3）QoS制御ユニット

QoS制御ユニットは，以下の2つの機能から構成される．

- QoS要求獲得機能：上位層からのQoS要求を獲得する．
- ネットワーク情報収集機能：LN層あるいは上位層から，ネットワークにかかわる様々な情報を収集，管理する．

（4）ネットワーク運用ユニット

ネットワーク運用ユニットは，以下の4つの機能から構成される．

- 流量解決機能：個々のアプリケーションが送受信するデータのストリーム（以下，フローと表記）ごとの流量を決定する．
- 経路解決機能：LN層の提供する通信路の利用法を決定する．
- 流量調整機能：流量解決機能と連携して，フローごとの流量を調整する．
- 経路多重化機能：経路解決機能と連携して，LN層が提供する通信路の切替え，多重化等を行う．

FN層がAP層とLN層の双方における変動を自律的に調整するためには，性能的調整すなわちパラメータ的な調整のみならず，必要であればFN層自体の機能構成を変更し，より大きな変動にも適応できることが求められる．そこで，これらのユニットの設計に際しては，それぞれのユニットが持つべき機能を分割し，自律・協調動作可能な小規模な機能部品，すなわちエージェトとして設計する．これにより，アプリケーションやネットワーク環境の違いに応じたユニットの機能調整や，ユニット間の連携・協調が容易となる．

3. 利用者指向流量調整方式の提案

3.1 既存のQoS保証方式の性質

論理ネットワーク層における代表的なQoS保証方式であるIntServやDiffServでは，経路上のIPルータにRSVP[6]等のシグナリングプロトコルの実装や，DSフィールドを扱うための実装が要求される．このため，アプリケーションが要求する帯域や最大遅延を保証できる反面，利用できるネットワークが限定されてしまう．また，IntServやDiffServを利用するアプリケーションには，シグナリングプロトコルを扱ったり，DSフィールドにDSCPをセットしたりするための実装が不可欠である．このため，既存アプリケーションをこれらの両方式に対応させるためには既存アプリケーションの変更が必要となり，アプリケーションの再利用性の確保が困難となる．

3.2 提案方式

本論文では，FN層の2つのユニット，すなわち，QoS制御ユニットとネットワーク運用の具体的設計を与え，これらのユニットを効果的に活用した利用者指向流量調整方式を提案する．

提案方式ではまず，前述した利用できるネットワークが限定されるという問題（P1）を解決するために，

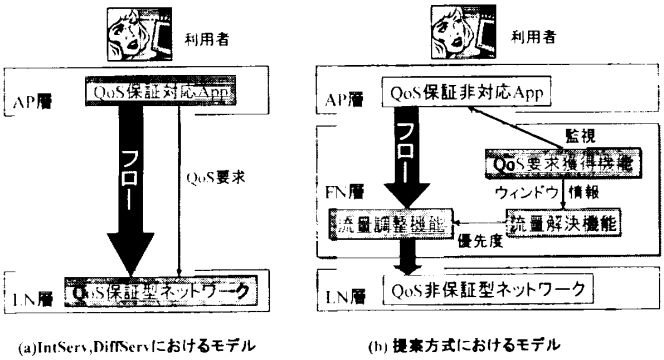


図3 QoS制御機能モデルの比較
Fig. 3 Comparison of QoS control models.

IntServ等では論理ネットワーク層で行っているフロー制御を，FN層で実行する．具体的には，アプリケーション層と論理ネットワーク層間に流れるトラフィックを，流量調整機能によりフローごとにキューイングし，それぞれのキューからデータを取り出す頻度を制御する．これより，論理ネットワーク層のIPルータによるフロー制御機能に依存することなく，様々なネットワーク環境においてフロー制御が可能となる．

加えて，前述の既存アプリケーションの変更が必要であるという問題（P2）を解決するために，QoS要求獲得機能と流量解決機能により，利用者が操作しているアプリケーションのウィンドウ状態に応じ各アプリケーションのフローの優先度を決定する．具体的には，QoS要求機能がアプリケーションのウィンドウの重なり順序や大きさ等のウィンドウ状態を監視し，これに基づき流量解決機能がアプリケーション間の優先順位を決定する．これにより，既存のアプリケーションに特別な変更を施すことなく，利用者がアプリケーションごとのフロー優先度を調整することが可能となる．

図3に，従来のIntServ，DiffServにおけるQoS制御機能モデルと提案方式におけるQoS制御機能モデルの比較を示す．従来方式に対する提案システムの特徴は，QoS制御に必要となる機能をFN層に配置することにより，QoS保証に非対応なアプリケーション，およびQoS非保証型のネットワークにおいてもフロー制御機能が利用可能である点である．

4. 設計と実装

本章では，前述の提案方式を実現する以下の3つの機能の具体的設計，および試作システムの実装について述べる．

- QoS要求獲得機能
- 流量解決機能
- 流量調整機能

4.1 QoS要求獲得機能の設計

QoS要求獲得機能では，利用者がどのアプリケーションのフローを優先したいのかを把握するために，アプリケーションのウィンドウ情報を取得する．ウィンドウ情報としては具体的に

- ウィンドウ識別子（handle）
- アプリケーション識別子（classname）
- アクティブ・非アクティブの状態（activity）
- ウィンドウの重なり（z-order）
- ウィンドウサイズ（width，hight）
- 最大化・最小化の状態（man，mix）

等の情報を取得する．なおアクティブとは，ウインドウがマウスやキーボードの入力フォーカスを取得しており，利用者の入力を受け付けられる状態であることを示す．これら取得したウィンドウ情報は，流量解決機能に伝達される．

4.2 流量解決機能の設計

流量解決機能では，QoS要求獲得機能から得られたウィンドウ情報からQoSパラメータを生成し，流量調整機能に伝達する．QoSパラメータは，以下の3つから構成される．

- ウィンドウ識別子（handle）
- アプリケーション識別子（classname）
- ネットワークリソース配分率（ratio）

流量解決機能には，ウィンドウ情報を条件部，QoSパラメータを動作部として持つIF-THENルールを，流量解決知識として組み込む．この流量解決知識とQoS要求獲得機能から得られたウィンドウ情報に基づいて推論を行い，実際のQoSパラメータを生成する．このQoSパラメータは，アプリケーション間の優先度を比率で表したものである．

ネットワーク情報とウインドウ情報は，ともにOAV形式で表現されるファクトとして，流量解決機能に保持される．OAV形式は，オブジェクト名と，複数の属性名・属性値の組からなり，下記のように表される．

(オジェクト名 :属性名1 属性値1 ...)

また，QoSパラメータを導出するルールは，下記のような，条件とアクションを「-->」でつないだ形式で表現する．

```
(条件1)
(条件2)
...
-->
(アクション1)
(アクション2)
...
```

```
//ネットワーク情報//
(network_state :width low :delay normal :reliability high)

//ウィンドウ情報//
(window_info :classname IEFrame :handle 8D5 :activity no
             :z-order 2 :max no :min no :width 223 :height 156)
(window_info :classname VTwin32 :handle D77 :activity yes
             :z-order 1 :max no :min no :width 303 :height 236)

//ルール//
(window_info :classname ?c :handle ?h :activity yes
             :z-order 1 :max ?max :min ?min :width > 100 :height > 100)
(network_state :width low :delay normal :reliability high)
-->
//QoS パラメータ//
(make (qos_param :classname ?c :handle ?h :ratio 0.8))
```

図4 ネットワーク情報とウィンドウ情報，およびQoSパラメータ生成ルールの例
Fig. 4 Example of network information, window information and QoS parameter reasoning rules.

条件はそれぞれOAV形式で表現されたマッチングパターンである．

推論の手順としてはまず，ファクトがワーキングメモリと呼ぶ記憶域に読み込まれる．次いで，ルールの条件部すべてにマッチするファクトがワーキングメモリ内に存在すれば，そのルールが発火し，アクション部に記述されたアクションが実行される．この動作を，発火するルールがなくなるまで繰り返す．

図4にネットワーク情報とウィンドウ情報の例，およびQoSパラメータ生成ルールの例を示す．図4の例では，利用者が使用しているネットワークに関する情報を，オブジェクト名network-stateで表したファクトにより表現している．このファクトでは，ネットワークに関する情報として，帯域は狭く，遅延は標準的，信頼性は高めであることを示している．同様にウインドウ情報は，オブジェクト名window_infoで表した2つのファクトにより表現している．この2つのファクトはそれぞれ，アプリケーション識別子がIEFrameであるアプリケーションとVTwin32であるアプリケーションのウインドウがあり，後者のアプリケーションのウインドウが一番手前（z-order=1）にあり，かつアクティブであること等を示している．さらに，QoSパラメータ生成ルールとしては，一番手前（z-orderが1）で，かつウインドウサイズが縦横それぞれ100ピクセル以上のウインドウ情報があり，さらにネットワーク情報として帯域が狭く，遅延が標準的，信頼性が高めである場合，そのアプリケーションに，利用可能なネットワークリソースの80%を配分することを示すファクト（オブジェクト名：qos_param）をワーキングメモリに追加することを示している．以上のような仕組みで，ネットワーク情報とウインドウ情報から，アプリケーションごとのネットワークリソース配分率を導出する．

4.3 流量調整機能の設計

流量調整機能では，流量解決機能で生成されたQoSパラメータに基づき実際に通信データの送受信制御，すなわちフロー制御を行う．本機能では，アプリケーションのフローごとに個別にキューを設け，本機能が受信したフローをいったん各キューに格納する．そして，前述の流量解決機能により導出されたアプリケーションごとのネットワークリソース配分率に基づき，単位時間にそれぞれのキューからデータを取り出し転送する量を制御することで，フロー単位の流量制御を行う．ここで，本機能はフローをパケット単位ではなく連続したストリームとして扱うため，キューから一度に取り出すデータ量（スライス）は容易に変更可能である．そこで，即応性を優先したい場合はスライスを小さくし，反対にスループットを優先したい場合はスライスを大きくする等，柔軟なキューイングが行えるよう工夫している．

4.4 エージェント指向設計

本節では，前節で設計した各機能のエージェント指向設計について述べる．**図5**に，提案方式を実現するエージェント構成を示す．図5中のクライアントは利用者端末を表し，ゲートウェイは上流ネットワークとの境界に位置するルータを表している．次いで，各エージェントの設計を以下に述べる．

- **QoS要求獲得Ag** QoS要求獲得機能を実現するエージェントであり，アプリケーションのウインドウ情報の取得を行う．
- **QoS Ag** 各アプリケーションのQoS要求を知識として保持するエージェントである．具体的には，アプリケーションが要求する優先度等の情報を保持する．

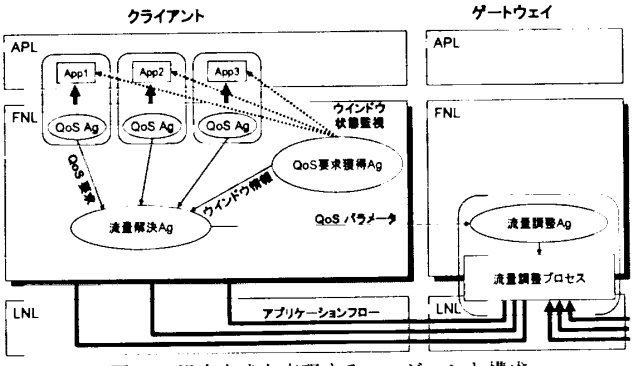


図5 提案方式を実現するエージェント構成
Fig. 5 Agent structure for realizing the proposing method.

- **流量解決 Ag** 流量解決機能を実現するエージェントであり，QoS Ag から通知される QoS 要求と，QoS 要求獲得 Ag から通知されるウインドウ情報から，QoS パラメータ，すなわちアプリケーションに配分するネットワークリソースの配分率を決定する．
- **流量調整 Ag** 流量調整機能を実現するエージェントであり，流量解決 Ag から通知され QoS パラメータに基づき，クライアントへ流れるフローを調整するために，流量調整プロセスを制御する．

ここで，クライアントへ流入するフローは，ゲートウェイからクライアントへフォワードされる時点で制御する必要がある．そのため，流量調整 Ag は，ゲートウェイ上に配置する．

4.5 実　　装

前節までの設計に基づき，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 を用い，実装言語には Visual C++，Java 言語を用いた．さらにエージェントフレームワークとして，DASH[15]を用いた．QoS 要求獲得機能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．また，流量調整機能については，透過型プロキシとして実装した．実装したエージェントの総コード量は 1,774 行であった．

5. 実験と評価

前章の設計に基づき実装した試作システムを用いて，提案方式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流量調整の性能，および，実験 2：流量調整にともなうオーバヘッドの影響に関する評価実験を行った．

5.1 実 験 環 境

図 6 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 1.0 GHz，OS は Microsoft Windows 2000 である．ゲートウェイ用の PC の CPU は Pentium3 1.0 GHz，OS は Vine Linux 2.1 である．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で使用した．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．

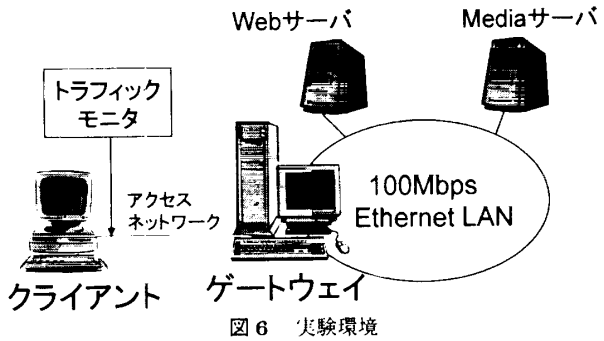


図6 実験環境
Fig. 6 Experimental environment.

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等）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．

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，10 Mbps とする．
- クライアントで，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 6 Mbps の MPEG1 ファイルとする．

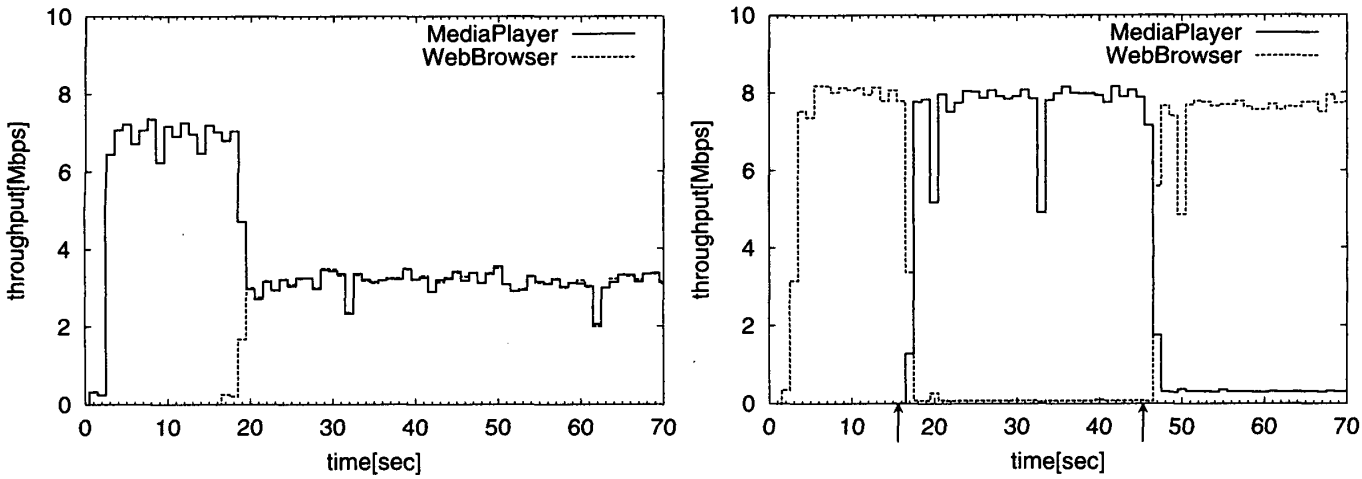


図7 実験結果1：スループットの推移（10 Mbps）
Fig. 7 The results of experiment1: Throughput of application flow.

- ゲートウェイと Web サーバ・動画サーバは 100 Mbps 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7 a)，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7 b) 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7 a)），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7 Mbps 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3.5 Mbps となった．その結果，動画の再生に必要な6 Mbps 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7 b)），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8 Mbps 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 QoS 要求獲得機能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，流量解決機能による QoS パラメータの導出，およびゲートウェイ上の流量調整機能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変化するまでの反応時間，すなわち図7 b) 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，おおむね1.4秒以下であった．これには，クライアント上の QoS 要求解決機能がアプリケーションの状況変化を検知し，流量解決機能が 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調整機能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：流量調整にともなうオーバヘッドの影響

提案方式のオーバ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，115 Kbps，2 Mbps，10 Mbps，100 Mbps 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 1,500 bytes に設定する．
- クライアントで，Web ブラウザによるファイルのダウンロードを行う．
- ゲートウェイと Web サーバは 100 Mbps 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

表1 平均スループット
Table 1 Average of throughput.

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]
115 K	1,362	8.16	86.8 K
2 M	1,365	119.45	1.24 M
10 M	1,365	681.64	7.10 M
100 M	1,365	5,365.81	55.9 M

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]
115 K	1,445	7.83	88.4 K
2 M	1,459	114.68	1.28 M
10 M	1,458	675.35	7.51 M
100 M	1,449	3,641.28	40.3 M

回の試行の平均である．表1より，すべてのリンク帯域において，流量調整機能がFN層で稼働することによるオーバヘッドによって，クライアントへの到着パケット数はわずかに減少していることが分かる．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（115 Kbps, 2 Mbps, 10 Mbps）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，提案方式では流量調整機能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，1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や携帯電話等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節，5.3節では，提案方式の性能評価を行った．その結果，提案方式は，十分実用に耐える性能で実現できることが検証された．また，実験結果より，本方式の以下の2つの特徴，すなわち，(1) 論理ネットワーク層の上位に位置するLN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こと，および，(2) FN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という点が実現できること，を確認することができた．以上2点の特徴により，提案方式は高機能なネットワークアプリケーションの開発効率の向上に貢献するものと考えられる．

さらに，提案方式の実装において，FN層を構成する2つのユニット，すなわちQoS制御ユニットとネットワーク運用ユニットが持つべき機能であるQoS要求獲得機能，流量解決機能，流量調整機能の具体的な設計と実装が与えられたことにより，提案方式で利用されたアプリケーションに関する種々の情報（ウインドウ情報等）を，FN層内の他のユニットで利用することも可能になると考えられる．また，本提案方式をFN層内のミドルウェアサービスユニット等と連携させることで，ビデオ会議システム等のマルチメディアアプリケーションの効果的な運用等を行えることが期待できる．

6. むすび

本論文では，やわらかいネットワーク層が持つユニットのうち，QoS制御ユニットとネットワーク運用ユニットに焦点をあて，これらのユニットの具体的な設計を与えるとともに，同ユニットを効果的に活用した利用者指向流量調整方式を提案した．提案方式の特徴は，(1) 論理ネットワーク層の上位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という点にある．さらに，提案方式を実現する試作システムの実装により，アプリケーション利用状況を反映して動作する提案方式の有効性が検証された．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等がある．

参考文献

1) Campbell, A., Coulson, G. and Hutchison, D.: A Quality of Service Architecture, *ACM Computer Communications Review* (Apr. 1994).
2) Campbell, A., Coulson, G. and Hutchison, D.: Supporting Adaptive Flows in a Quality of Service Architecture, *Multimedia Systems Journal* (Nov. 1995).
3) Nahrstedt, N. and Smith, J.M.: Design, Implementation and Experiences of the OMEGA End-Point Architecture, *IEEE J. Select, Areas Commun.*, Vol.14, No.7, pp.1263–1279 (1996).
4) Wroclawski, J.: The Use of RSVP with IETF Integrated Services, IETF Request for Comments 2210 (1997).
5) Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z. and Weiss, W.: An Architecture for Differentiated Service, IETF Request for Comments 2475 (1998).
6) Braden, R. (Ed.), Zhang, L., Berson, S., Herzog, S. and Jamin, S.: Resource ReSerVation Protocol (RSVP) — Version 1 Functional, IETF Request for Comments 2205 (1997).
7) 中沢 実，須田飛志，阿部倫之，服部進実：ユーザアダプティブエージェントによる自律的ネットワーク，電子情報通信学会論文誌，Vol.J81-B-I, No.2, pp.58–69 (1998).
8) 小菅昌克，山崎達也，荻野長生，松田 潤：エージェントによる適応的 QoS 制御方式，電子情報通信学会論文誌，Vol.J82-B, No.5, pp.702–7100 (1999).
9) Kinoshita, T., Sugiura, S., Suganuma, T., Sugawara, K. and Shiratori, N.: Evolutional Flexible Network, IPSJ Technical Report, Vol.99, No.56, pp.143–148 (1999).
10) Suganuma, T., Kinoshita, T. and Shiratori, N.: Flexible network layer in dynamic networking architecture, *Proc. 1st International Workshop on Flexible Networking and Cooperative Distributed Agents* (*FNCDA2000*), pp.473–478 (2000).
11) Shiratori, N. and Sugawara, K.: Developing of a Next Generation Network Based on Knowledge-based Design Methodology, IEICE Technical Report, AI93-28 (July 1993).
12) Shiratori, N. and Sugawara, K.: Towards Developing Flexible Network—Knowledge-based Design Methodolody, IEICE Technical Report, AI93-46 (Sep. 1993).
13) Shiratori, N., Sugawara, K., Kinoshita, T. and Chakraborty, G.: Flexible Networks: Basic concepts and Architecture, *IEICE Trans. Comm.*, Vol.E77-B, No.11, pp.1287–1294 (1994).
14) Shiratori, N., Suganuma, T., Sugiura, S., Chakraborty, G., Sugawara, K., Kinoshita, T. and Lee., E.S.: Framework of a flexible computer communication network, *Computer Communications*, Vol.19, pp.1268–1275 (1996).
15) Distributed Agent System based on Hybrid architecture.
http://www.agent-town.com/

(平成 14 年 7 月 9 日受付)
(平成 14 年 11 月 5 日採録)

北形 元

1972 年生．2002 年東北大学大学院同研究科博士後期課程修了．現在，東北大学電気通信研究所助手．博士（情報科学）．エージェント指向コンピューティングに興味を持つ．電子情報通信学会会員．

今野 将（正会員）

1973 年生．1996 年千葉工業大学工学部情報工学科卒業．2001 年同大学大学院博士後期課程情報工学専攻期間満了退学．現在，東北大学電気通信研究所助手．博士（工学）．エージェント指向コンピューティングに興味を持つ．IEEE，電子情報通信学会各会員

加藤 貴司

1971 年生．2001 年東北大学大学院情報科学研究科博士後期課程修了．現在，東北大学電気通信研究所助手．博士（情報科学）．マルチエージェントシステムにおけるエージェントの協調に興味を持つ．人工知能学会，電子情報通信学会各会員．

菅沼 拓夫（正会員）

1966年生．1997年千葉工業大学大学院博士後期課程情報工学専攻修了．現在，東北大学電気通信研究所助手．博士（工学）．やわらかいネットワーク，エージェント指向コンピューティングに興味を持つ．IEEE，電子情報通信学会各会員．

木下 哲男（正会員）

1953年生．1979年東北大学大学院修士課程修了．同年，沖電気工業（株）入社．1996年東北大学電気通信研究所助教授．2001年同大学情報シナジーセンター教授．知識工学，エージェント技術，知識情報システム等の研究開発に従事．工学博士．1989年度情報処理学会研究賞，1996年度同論文賞，2000年度電子情報通信学会業績賞各受賞．電子情報通信学会，人工知能学会，日本認知科学会，AAAI，IEEE，ACM各会員．