

パーフロー QoS を実現するエージェント型リソース制御システム

笠原 英樹[†] 増田 暁生[†] 吉田 彰 顕^{††}

IP ネットワークにおいて映像ストリーム等のリアルタイムアプリケーションのフローを高品質に転送するために、フロー単位の QoS を実現するエージェント型リソース制御システム RA (Resource Agent) の実装および評価を行った。本論文では、本システムを経済的かつ高い処理性能で実現するために、従来のネットワークリソースのモデル化手法に対して高速性、スケール性を確保することが可能となるバーチャルリンクモデル化法、またネットワーク故障等によるネットワーク状態変化に対して即時に同期化が可能でかつメモリ上の DB で高速検索、判定を可能とするリソース管理テーブル構成法を用いた新たな実装方式について示した。また本論文では、提案方式に基づく実装とその評価を行い、提案システムが要求条件に応じスケール可能であることを明らかにした。

Agent Based Network Resource Control System for Per-flow QoS Services

HIDEKI KASAHARA,[†] AKEO MASUDA[†] and TERUAKI YOSHIDA^{††}

We have designed a network resource control system for large-scale networks, based on a bandwidth broker model, called the RA (Resource Agent). RA provides parameter-based admission control for cost-effective per-flow QoS services in Diffserv enabled IP networks. In this paper, we propose a novel network resource mapping model, called the “virtual link multi-mapping model,” which makes the RA scalable. Also, we show that the “n-th route retention method” enables the RA to keep the route management table synchronized with the real network dynamically. System architecture of the implementation and evaluation of the call processing performance is shown. The evaluation results show that the virtual link multi-mapping model eliminates the performance bottleneck at the database access, and we confirmed that the RA is scalable enough to increase its processing performance, up to several million BHCC.

1. ま え が き

ADSL や光アクセス技術の進展により、ブロードバンドネットワークサービスの加入者数は急激に増加している。また、その拡大したブロードバンドネットワーク上に展開されるアプリケーションについても、従来の電子メールや Web アクセスに加えてネットワークの高速、広帯域性を活かした映像配信や双方向映像コミュニケーション等のサービスが開始され始めている。IP ネットワークによるこれらの映像サービスは既存の TV 放送メディアにない双方向性を有する等の特長により新たなメディアとして期待されている。

しかし、高品質な映像ストリーミングサービスを現在の QoS を提供していない IP ネットワーク上で実現するには課題がある。電子メールや Web アクセス等のアプリケーションでは転送の即時性が要求されないことに対して、映像ストリーミングでは転送の実時間性が要求されるため、従来のエンドホスト間で実現していたパケットロスに対する再送制御等が有効に活用できない。また、アプリケーションレベル品質のパケットロス耐性を向上させる方法として FEC (Forward Error Correction) を用いることも考えられるが、FEC はネットワーク品質が大きく変動する場合には効果的に機能しない。これらの課題を解決するためには、IP ネットワークの転送品質を制御、向上させることが必要になる¹⁾。

本論文では、IP ネットワークの品質制御を実現するためのエージェント型リソース制御システム RA (Resource Agent) の実装およびその評価について示す。実装したシステムは、経済的にネットワークの品質制

[†] 日本電信電話株式会社ネットワークサービスシステム研究所
Network Service Systems Laboratories, NTT Corporation

^{††} 広島市立大学大学院情報科学研究科
Faculty of Information Sciences, Hiroshima City University

御を実現するために高速性，スケール性を有している点の特徴としている。

まず，2章においてIPネットワーク品質制御に関する関連研究について示し，3章においてシステムの実装の前提となるネットワークアーキテクチャを示し，RAの機能条件を述べる．4章において，受付制御処理に関する提案方式について述べ，5章においてシステムの実装およびその評価を示す．最後に6章でまとめを行う．

2. 関連研究

IPネットワークの転送品質制御については，過去多くの研究および実装がなされている．以下，代表的なものについて述べ，本研究との関係について述べる．

2.1 Intserv と Diffserv

アプリケーションが要求するネットワーク品質を提供する方式として Intserv モデルおよびその信号プロトコルである RSVP が IETF を中心に提案されている²⁾．しかし，RSVP を大規模ネットワークに適用することについては，転送ノード装置にフローごとのメモリの消費および処理負荷が大きくなるため，その適用が困難であるといわれている³⁾．

一方，転送ノードに対して次ノードに対する転送優先度のみを規定した Diffserv モデルが同じく IETF で提案されている³⁾．Diffserv は各ノードにおいてフローごとの管理は行わず，フローを集約した単位で扱うことによって，ノードのスケラビリティの問題はない．しかし，Diffserv で規定される EF (Expedited forwarding) クラスで最優先転送する場合でも，集約したフローの総トラフィック量がネットワークの物理リソースを超過すれば転送品質を維持することはできない．そのため，フローごとの受付制御を行う機能が別途必要であり，当該機能を一般的に BB (Bandwidth Broker) と呼ばれるサブシステムで実現する研究がなされている．

2.2 Diffserv + BB

Internet2/Qbone において，次に示すフレームワークが提案されている⁴⁾．図1に示すように，BBは担当する Diffserv ドメインのネットワーク内のリソースを管理し，フロー単位の利用要求を受け付けフローが要求する転送品質の提供可否の判断を行う役割を持つ．なお，Diffserv ドメインとは，単一の QoS ポリシ，つまり該当ドメインの境界ルータにてパケットに付与される DSCP (Diffserv Code Point) 値とそれへのリソースのマッピング基準が統一されているネットワーク単位のことであり，一般にはある事業者 (ISP) の

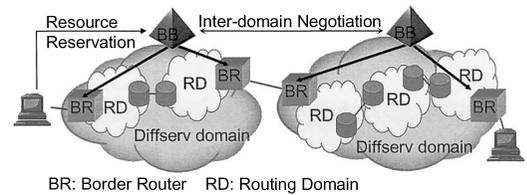


図1 Diffserv ドメインと BB
Fig. 1 Diffserv domains and BBs.

管理範囲であるネットワークドメインに相当すると考えられる．したがって，1つの Diffserv ドメインは大規模である可能性もあり，論理的には1つの Diffserv ドメインに1つの BB が配備されるが，実際には複数のサーバにより実装されることもある．

2.3 測定ベースと申告パラメータベース判定

BB における受付判定の基準となるリソースの利用状況を把握する方式は，トラフィック監視やモニタパケットの導通等によって通信開始時点での品質状態の情報を基にして受付制御を行う測定ベース受付判定方式⁵⁾と，利用者の申告した要求量に基づいて確保するリソースを記録し，その総量とリソースの最大容量と比較することによって受付判定を行う申告パラメータベース受付判定方式^{6),7)}に大別することができる．測定ベース判定方式はフローに対して確定的なリソースを確保しないため，品質保証の正確性が低くなる可能性がある．一方，申告パラメータベース判定方式は利用者からの申告パラメータおよびネットワークのリソース情報を用いて受付制御を行うため精度の高い品質制御が可能であるが，フローごとの状態や各リンクのリソース使用状況を管理する必要があり，受付制御のために保持する情報量や処理量が多く，BB の処理能力がスケールネックとなる可能性がある．

このように，BB のスケール性が解決できれば，Diffserv モデルにおける申告パラメータベースの受付制御方式は IP ネットワークの転送品質を制御し向上させるための有効な手段である．しかしながら，この方式に基づくシステムはいくつか提案されているものの，その実装に関する具体的な方法や，BB を実際に大規模なネットワークに適用する場合に課題となる高速処理性やスケール性，ネットワーク状態の変化への対応方法に関する報告はほとんどない．本論文では，申告パラメータベース判定方式に基づく品質制御システム，RA の実装を行い，その評価を行う．また，高速処理およびスケール性を確保するためのネットワークリソースのモデル化方法と，ネットワーク状態変化に瞬時に対応することを可能とする経路管理方法について示し，あわせてその有効性について示す．

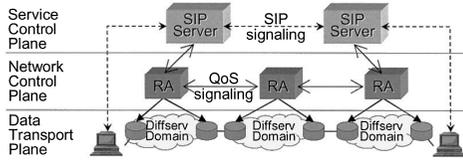


図 2 品質制御 IP ネットワークのアーキテクチャモデル
Fig. 2 Architecture model for flow based QoS services.

3. 前提とするアーキテクチャモデル

本論文における RA の機能条件を定義する前提として、図 2 に示すような品質制御 IP ネットワークのアーキテクチャモデル⁸⁾を採用する。本モデルは、(1) パケット転送機能を有する IP データ転送プレーンと (2) SIP セッション制御等のサービス制御を行うサービス制御プレーンに加えて、(3) サービス制御機能からの QoS 要求を受け付け、QoS 機能を提供するネットワーク制御プレーンの 3 つのプレーンで構成される。

IP データ転送プレーンの具体的な構成や機能はネットワーク制御プレーンが隠蔽するため、各種アプリケーションを実現するサービス制御プレーンのサーバは物理ネットワークの構成を意識することなく、ネットワーク制御プレーンが提供する簡易で抽象化されたプリミティブにより QoS 機能を利用することが可能である。また、クライアントが送信するサービス要求はサービス制御プレーンがいったん終端し、要求サービスの QoS の要否判定やネットワーク制御プレーンへのインタフェース機能を代行 (Proxy) するため、QoS 機能の導入にあたってはクライアントへの機能追加が不要である。このように、ネットワーク制御プレーン以下は複数の異なるアプリケーションに対して統合的に QoS を提供することを可能とする。

RA は、上記のモデルにおいてネットワーク制御プレーンに位置し、IP データ転送プレーンのトポロジ・リソース状態を把握し、サービス制御プレーンが QoS 要求を行えるインタフェースを提供する。QoS 要求はネットワークリソースを確保すべきフローの発着アドレスを含むフロー識別子と、要求するリソース量のパラメータを含む信号であり、RA はこの要求に対して 4.3 節に後述する方法で受付判定を行い、応答する。また、QoS 機能を要求したフローが RA が管理する Diff-serv ドメインを複数経由する場合には、RA 間で QoS 要求信号を伝播させ End-End の QoS を提供する。

4. 受付制御処理方式

4.1 トポロジ・リソースのモデル化

申告パラメータベースの受付制御を行うためには、

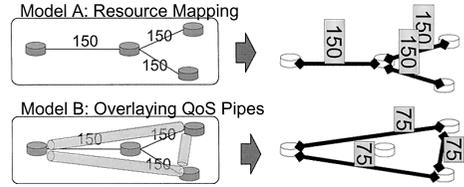


図 3 提案されている抽象化モデル A, B
Fig. 3 Mapping models A, B.

ネットワークの各リンクのリソース使用状況を把握するために、物理ネットワークのトポロジとリソースをサーバ上で管理する情報 (論理ネットワーク) としてモデル化する必要がある。このトポロジとリソースのモデル化としてこれまでに図 3 に示す 2 つのモデル化手法が提案されている。

4.1.1 提案されているモデル化手法

モデル A: リソースマップへの直接写像

このモデルでは、すべてのリンクのリソースとトポロジをそのまま写像した論理ネットワークを BB 上に作成し、本論理ネットワークに基づきリソースの管理・確保を行う⁷⁾。この方法ではリソースをリンク単位でそれぞれ管理するため、フローが通過するすべてのリンクについて、ホップ数分のリソースのチェックと確保を行う必要があり、処理性能への影響が懸念される。

処理性能の向上を目的として管理するネットワークのエリアを分割し、分割された小規模なサブエリアを管理する複数の BB 間の協調動作によりエンドエンドフローに対するリソース受付を行う方法が考えられるが、分割したサブエリアが小規模であるとエンドエンドフローが通過するサブエリアの数が多くなり BB 間の協調動作に必要となる処理が相対的に大きくなり処理性能を向上させることが難しい。また、処理性能向上の一般的手法として複数 CPU マシンによるマルチプロセッサ処理が考えられるが、このモデル化では処理系を複数にしても参照・更新するリソースはすべての処理系で共有する必要があるため、データベース (DB) アクセス競合の発生により、複数プロセッサによる分散処理を効果的に実現することが困難である。

モデル B: QoS パイプによる管理単位の集約

このモデルは、リソースの管理単位をリンクごとではなく、論理的な 2 点間のパスを単位とする。まず、ある発着の 2 点間を結び論理パス、“QoS パイプ”を (仮想的に、または MPLS 等を用いて実際に) 定義し、パイプの経路上のすべてのリンクにおいて、そのパスに対する排他的なリソースをあらかじめ割り当てておく。BB ではこの QoS パイプのリソース量とトポロジを論理ネットワークとして用いる。

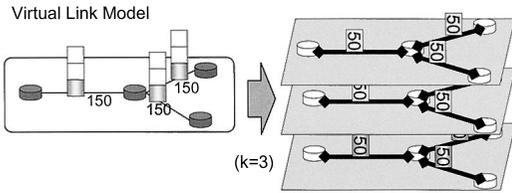


図 4 提案モデル：バーチャルリンクモデル
Fig. 4 Virtual Link Multi-mapping Model.

QoS 要求受信時の受付判定では、要求元のフローの発着点に該当する QoS パイプを検索し、その QoS パイプに割り当てられているリソース量を上限として受付可否判定、リソース確保を行う⁹⁾。このモデルでは、リソースのチェックと確保はリンクごとには行わず、フローのホップ数にかかわらず該当 QoS パイプに対して 1 度行うだけなので、前述のモデル A に比べて処理能力向上の面で有利である。しかし、該当 QoS パイプでのリソース限度を超えるとあるリンクに残余帯域があっても使用できないという分割損が発生し、リソースの有効利用の面で問題がある。

4.1.2 提案モデル：バーチャルリンクモデル

これらに対し我々は、モデル A, B が持つ課題を解決する「バーチャルリンクモデル」を提案し、これを用いてシステムの実装を行った。図 4 に、提案モデルを示す。このモデルは、サーバ負荷分散や処理系増強（マルチプロセッサ）による処理能力向上を可能とすることを目的としている。処理サーバ数を複数にしたり、1 サーバで DB アクセス競合を起こさずに処理系増強の効果を得るためには論理ネットワークを分割し分散配備する必要があるが、前述のようにネットワークのエリアで分割しても処理能力向上の効果は得られない。本提案モデルでは、各リンクのリソース量を均等に分割し、それぞれを同一のトポロジを持つ複数の論理ネットワークに写像することで分割を実現する。つまり任意の数 k の論理ネットワークは、ネットワーク全体のトポロジを持つが、リソース量に関しては各リンクの容量の一部 ($1/k$) を互いに排他的に管理する。

受付制御処理時には、リソース確保要求をラウンドロビン等の任意の方法によってそれぞれの論理ネットワークが格納されている DB に振り分ける。どの DB も全リンクのリソースを保持しているため、リソースのチェックと確保の処理は 1 つの DB に対する検索・更新処理のみで処理を完了することができ、DB 間（またはサーバ間）の協調動作によるオーバーヘッドが発生しない。

また、提案モデルではモデル B と同様に物理リン

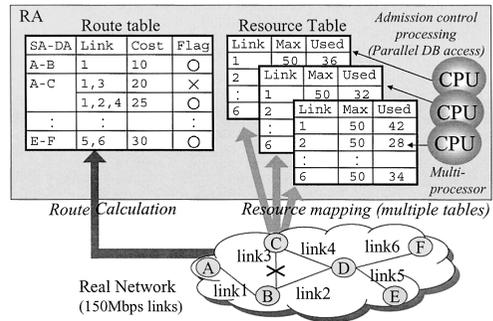


図 5 RA のテーブル構成表
Fig. 5 Table configuration of Resource Agent.

クリソースの分割を行うため、あるリンクについてある DB ではリソースが不足し、別の DB では空きがある状況が発生する可能性がある。このときモデル B では複数の物理リンクにわたる QoS パイプ単位でリソース量を保持するので、トラヒックの偏りに応じた QoS パイプのリソース容量の再割当ては困難であるが、提案モデルではリンク単位でリソース量を保持するので、たとえば受付制御時に要求呼ごとにリソース不足リンク単位で他 DB に不足分を要求する方法や、定期的なバックグラウンド処理によって DB 間の残りリソース量の偏りを監視し再割当てを行う方法によって DB 間での残りリソース量の調整が簡易に実現でき、処理能力を低下させることなくリソース量の分割損を回避できる¹⁰⁾。

4.1.3 提案モデルに基づく論理ネットワーク管理

提案方式によってモデル化されたトポロジとリソース情報はそれぞれ、図 5 に示す経路管理テーブルとリンクリソース管理テーブルとして RA の DB に格納される。経路管理テーブルはインタフェースの接続先情報やリンクコスト情報を収集しルータ間で動作するルーティングプロトコル (OSPF 等) と等価な経路計算を行うことにより生成される。リンクリソース管理テーブルはリンク速度情報を収集し、各論理ネットワークに割り当てるリソース量を計算することによって生成される。これらのネットワーク情報は事前および定期的に各ルータ・スイッチからネットワーク管理プロトコルを用いて収集する。

受付制御処理では、まず発着アドレス (SA-DA) をキーとして経路管理テーブルを検索してどのリンクを通過するかを調べ、検索されたそれぞれのリンクについてリンクリソース管理テーブルを検索し、受付可否の判断と、利用中リソース量の更新を行う。このように、経路管理テーブルを事前に作成しておくことによってリソース確保要求の度に経路計算を行う必要が

なく、高速な受付判定を行うことができる。ここで、リソース管理テーブルは1つの要求処理の中でアクセス回数が多く、かつデータの検索だけでなく更新処理が必要であるので、マルチプロセッサまたはマルチスレッドによる並列処理時には排他制御（データベースロック等）が不可欠となり、並列処理性が低下してしまう恐れがあるが、RA では前述のバーチャルリンクモデルによってリソース管理テーブルを複数化することが可能なので、並列処理性を維持し、高い処理性能を実現することが可能となっている。

4.2 経路管理テーブルの更新

システムの動作中にネットワークに故障が発生すると、転送ネットワークのリルーティングと同期化させるための経路管理テーブルの更新を行う必要がある。RA は故障検知から同期化処理の完了までの間は誤判定を防ぐため新規の利用受付を一時停止する必要があるため、同期化処理は少なくとも転送ネットワークのリルーティング処理よりも早く完了する必要がある。

経路管理テーブルを再構成するには、各ルータの経路制御と同様にダイクストラ法（OSPF の場合）による最短経路計算を行う必要がある。ここでルータが計算する経路数は自身を始点とした各ルータへ経路のみであり、RA ではすべてのルータを始点終点とする経路の計算が必要であることから、1 経路の計算時間が同等とした場合、ダイクストラ法の特徴からルータ、RA での経路計算時間はルータ数 N に対してそれぞれ $T_{NW}(N) = O(N^2)$ 、 $T_{RA}(N) = O(N^3)$ と考えることができる¹²⁾。つまり RA での経路計算時間はルータでの計算時間に比べネットワーク規模が大きくなるにつれ不利となる。文献 7) でも上記の方法によって経路情報の同期化を図っているが、ネットワーク規模が大きくなったときの問題については言及されていない。

この課題に対し我々は、経路管理テーブルの構成法として n 次ルート保持法を適用した。 n 次ルート保持法は、あらかじめ故障が発生した場合に想定される迂回経路も計算しておき保持することによって、故障発生時には再計算の必要なく瞬時に経路を切り替える方法である。ある 2 点間の最短経路において、経路上のリンクのそれぞれが不通となった場合の迂回先の最短経路を 1 次迂回経路とし、さらに 1 次迂回経路の経路上のリンクのそれぞれが不通になった場合の迂回経路を 2 次迂回経路とする。これらの経路の抽出を繰り返し、各始点終点の組合せに対する n 次経路を RA の経路管理テーブルとして保持する（図 5）。受付判定時には経路管理テーブルにおいて合計コストの低い順に検索を行い、合計コストが最小で故障フラグが立っ

ているリンクを含まない経路が現在の経路となる。これにより、故障検出時には故障リンクを含む経路に対して故障フラグを立てる処理を行うだけで、経路管理テーブルをネットワークの新経路と同期化させることが可能となる。なお、仮に経路テーブル上のすべての候補が故障フラグが立つような場合、すなわち n 次以上の故障が発生した場合にはその時点で新たな経路計算を行い、経路テーブルに候補を追加する。

ここで RA が保持する迂回経路の次数 n は、対象とするネットワークの規模に応じて決定すればよい。大規模なネットワークでは n を増やしていくと保持すべき経路数は増大し現実的ではない計算時間がかかってしまう。ところが n 次迂回経路が使用されるような多重故障が発生する確率は n の値が大きくなるにつれて飛躍的に小さくなるので、対象とするネットワークの規模や冗長度と、ネットワーク装置の不稼働率を考慮して現実的な値を設定すれば、十分に信頼性の高いサービスを提供することができる¹²⁾。

4.3 リソース判定アルゴリズム

ネットワークが提供する QoS のパラメータとしてはパケット廃棄率、遅延時間、遅延揺らぎがある。本システムでは、新規要求フローに対して排他的に当該フローが経路する各リンクの帯域を与えることによって各 QoS パラメータの制御を行う。受付判定は新規のトラヒック (R_{new} [bit/s]) と現使用帯域の総和 ($\sum R(i)$ [bit/s]) の加算とリンク帯域 (W [bit/s]) の比較により行う。

$$W \geq \sum R(i) + R_{new} \quad (1)$$

各ルータの出力パケットバッファが多重フロー数以上具備されている条件で、上記判定式によりルータのバッファ溢れによるパケット廃棄率をゼロにする安全側判定が実現できる。QoS を要求するフローのトラヒックがパケットレベルでのバースト性を有している場合においても、クライアントを収容するエッジルータにおいて適切なレートへのパケットレベルのシェーピングを実施する場合には上記式が適用可能である。なお、フロー単位のレートシェーピング機能をエッジルータが具備していない場合もあるため、本システムでは周期的バースト性を有するフローの場合の統計的な目標廃棄率に基づく安全側判定論理もあわせて実装している¹³⁾。

また、各ノードに実装するパケットバッファ量が既知であれば、本システムはフローが通過するリンク数を DB 内に保持しているためフローの End-End の遅延、遅延揺らぎの最悪値は簡易に算出可能であり、遅

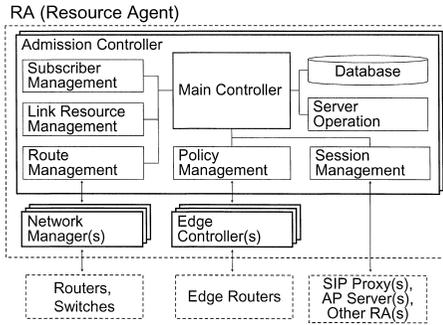


図 6 RA の機能ブロック図

Fig. 6 Resource Agent diagram.

延, 遅延揺らぎに関する受付判定も簡易に可能である.

5. 実装と評価

5.1 機能構成

図 6 に, 実装した RA の機能構成を示す. RA は, 以下の (1) ~ (3) の機能ブロックからなる. それぞれの機能ブロックは導入するネットワーク規模やユーザ数によって同一または異なるサーバに搭載することができ, またそれぞれにおいて処理性能に応じた負荷分散や信頼性向上のための冗長構成をとることができる.

(1) 受付制御部 (Admission Controller)

前述の経路/リンクリソース管理テーブルや各フローのセッション状態等運用に必要なデータをメモリ上の DB で管理し, SIP サーバ等のサービスサーバから送信されるリソース確保要求に対して受付判定を行う. また, 他のドメインを管理する RA との間でインタードメイン間交渉を行う.

(2) ネットワーク管理部 (Network Manager)

ネットワークのルータからトポロジやリンク速度情報を収集し, 経路計算を行う. また, ルータやインタフェースの故障を検知し, 受付制御部に通知する.

(3) エッジ制御部 (Edge Controller)

受付許可したフローに対し, 優先クラスの利用許可, 申告帯域を越える転送レートのポリシーリングを行うため, エッジルータへの優先制御・帯域制御設定を行う.

5.2 処理手順

図 7 に, SIP サーバからのリソース確保要求に対する RA の受付制御処理手順を示す. 本シーケンスは RFC3312¹⁴⁾ に規定された手順をベースとしている. RFC3312 では QoS に関するセッション確立の前提条件 (Precondition) について SDP を用いて End-End でネゴシエーションする方式が定義されており, リソース確保方法として RSVP を用いる場合が例示されているが, 一部モディファイして用いている. な

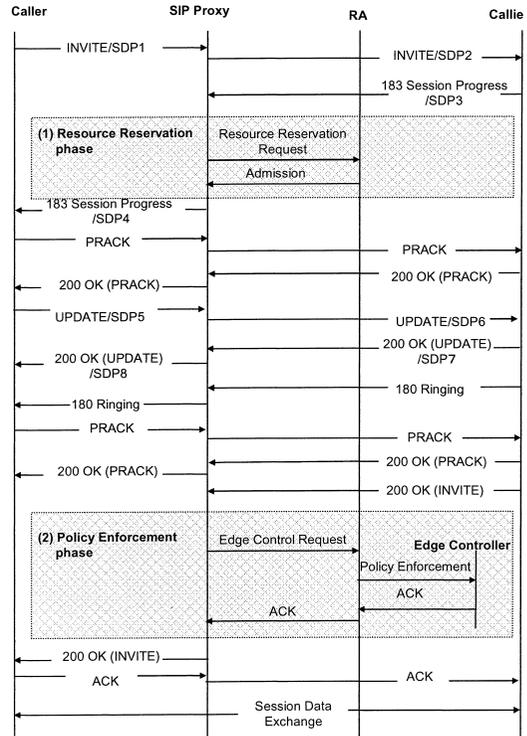


図 7 受付制御処理手順例: ユニキャスト

Fig. 7 Admission control sequence diagram (Unicast).

お図中の (2) Policy Enforcement Phase は双方向映像コミュニケーションサービス等でユーザ発トラヒックに対してエッジルータにおいて優先制御・帯域制御が必要な場合のみ処理を行う.

5.3 評価

5.3.1 受付制御の応答時間とスケール性

実装した RA の受付制御性能の評価を行った. RA の受付制御部は C++ でコーディングし, メモリを共有する複数の CPU (UltraSPARC III Cu 1.2 GHz) が搭載された 1 台のサーバ (Sun SolarisTM プラットフォーム) 上に動作させた. 評価における管理対象ネットワークは複数の迂回経路を確保できる二重帰属型のツリートポロジ (1000 ノード, 2000 リンク) のものを用い, 経路・リソース情報を 4 章で述べたパッチャルリンクモデルに基づいて複数の DB に格納した. 受付制御プログラムとインメモリ DB が使用した最大のメモリ量は, DB 数 $k = 3$, 保持迂回経路次数 $n = 4$ の場合で約 2 GB であった. 評価は, RA に対して擬似呼発生サーバからリソース確保・解放要求の高負荷をかけ, その要求に対する処理時間 [msec] と, 処理能力 [BHCC] (1 時間あたりに確保・解放の処理を完了できるセッション数) を測定することにより行った.

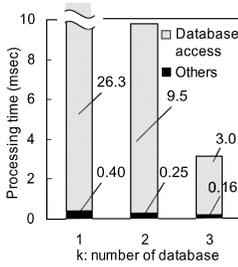


図 8 受付制御の処理時間
Fig. 8 Processing time.

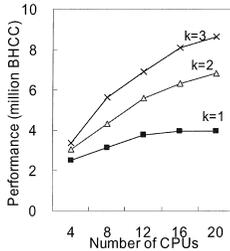


図 9 処理能力の推移
Fig. 9 Processing performance.

また、この測定は DB 分散数 k が 1 (分散なし), 2, 3 のそれぞれについて、サーバ上で稼働させる CPU 数を変化させて行った。

測定の結果として、図 8 に 20CPU 稼働時に 360 万 BHCA の高負荷を与えたときの 1 つの受付制御 (リソース確保) 処理の処理時間 [msec] とその内訳を、図 9 に処理能力の推移を示す。これらの結果から、以下のことが分かった。

(1) 高速処理性：図 8 から、DB の負荷分散がない状態でも、高負荷状態でも 30[msec] 以内に受付制御処理を完了できることが分かり、さらに DB の負荷分散によって約 3[msec] ($k = 3$) までに高速化されていることが分かった。この測定結果は 20CPU 稼働時のものだが、4CPU 稼働時でも 250 万 BHCC 処理時の処理時間を計測したところ 24.6[msec] と十分に高速であることが分かった。

(2) スケール性：図 9 から、DB の負荷分散がない状態 ($k = 1$) でかつ CPU 数が 4 の場合でも 200 万 BHCC 以上の処理性能が得られた。また、 $k = 1$ では CPU 数を増加させても性能がほとんど向上しないが、DB を分散すること ($k = 2, 3$) によって、分散数 k が大きいほど性能限界点が高くなり、また CPU 数増加にともなって性能限界が向上していくことが分かる。すなわち、要求される処理能力条件に対してハードウェア性能の増強によって処理能力を向上させることが可能なスケール性を有していることが分かった。

表 1 1 セッション処理あたりのロック待ち発生回数の推移

Table 1 Number of lock-waits per 1 session processing.

稼働 CPU 数	2	3	4
ロック待ち回数	0.28	0.48	0.72

(3) パーチャルリンクモデルによる DB 負荷分散の効果：受付制御処理時間の内訳 (図 8) を見ると、DB の分散数 k を多くするにつれて 1 処理の処理時間、特にその大部分を占める DB アクセス処理部分が減少していることが分かり、DB 負荷分散が高速化に効果的に作用していることが分かる。ここで、DB 分散がマルチプロセッサの並列処理性を高めていることを確認するために、利用者からのリソース確保要求と解放要求の組合せ 1 セッションを、DB の負荷分散を行わない状態 ($k = 1$) の RA が処理するとき発生する平均の DB のロック獲得待ち回数を計測した結果を表 1 に示す。本システムは 1 セッションを処理するには約 10 回の DB アクセスを行い、データ更新処理を行う際には DB の特定の部分のロックを獲得する。測定は 144 万 BHCA の高負荷を与えて行った。測定結果によると、稼働 CPU 数を増やしていくと、同時に複数の処理が DB アクセスを試みるためにロック待ちが発生し、稼働 CPU 数が多ければ多いほど、その発生回数は高くなっている。ロックの獲得を待たされた場合、スレッドの切替えが行われるまでは CPU がアイドルになってしまう。このことから、DB の分散がない場合は CPU 数を増強しても DB アクセス部分がボトルネックとなり、処理能力の向上に結び付かない。これに対し、DB 分散 ($k = 2$) を適用した場合のロック待ちの発生回数を計測したところ、稼働 CPU 数が 4 のときに 0.21 であり、 $k = 1$ の場合の 0.72 と比べ減少していることが分かった。このことから、DB の分散がアクセス競合を削減し並列処理の効果を保つことで、CPU 増強に対する処理能力のスケール性を実現していることが分かる。

5.3.2 経路管理テーブルの更新時間

ネットワーク故障発生時の経路管理テーブルの更新のためには、従来方式では最短経路計算処理を行う必要があるが、提案方式では経路テーブルに保持経路数分の故障フラグ設定処理を行うのみである。ここで、文献 11) では Pentium4 2.5 GHz の PC を用いた約 4 百万経路 (2,000 エッジ) の最短経路計算時間が約 60[sec] であることが報告されているが、同様に 4 百万経路を検索し、提案方式に基づいて故障リンクを含む経路に故障フラグを設定する処理時間を測定したところ、Pentium4 1.8 GHz の PC を用いて 0.18[sec] で

あった。この結果から、提案方式では対象ネットワークが大規模であっても瞬時に経路管理テーブルを実ネットワーク状態と同期化させることが可能であることが分かる。

6. む す び

本論文では、IP ネットワークにおいて映像ストリームを高品質に転送するためのフロー単位の QoS を実現するエージェント型リソース制御システム RA の実装およびその評価結果を示した。特に高速性・スケール性を確保するためのバーチャルリンクモデル、ネットワーク状態変化に対して即時に同期化を可能とする n 次ルート保持法を用いた新たな実装方式について示した。また、実装システムの性能評価からその有効性を明らかにした。今後の課題としては、本システムが提供する品質制御の主観品質まで含めた実際のアプリケーション品質に対する効果の評価や、論理ネットワークの分割数の最適値の検証等を行っていく。

参 考 文 献

- 1) 戸田 巖：ネットワーク QoS 技術，オーム社 (2001).
- 2) Braden, R., et al.: Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification, RFC 2205 (1997).
- 3) Blake, S., et al.: An architecture for differentiated services, RFC2475 (1998).
- 4) Teitelbaum, B. and Chimento, P.: QBone Bandwidth Broker Architecture Work in Progress.
<http://qbone.internet2.edu/bb/bboutline2.html>
- 5) 間瀬憲一：インターネットにおけるスケーラブルなアドミッションコントロール方式，信学会誌，Vol.85, No.9, pp.655–661 (2002).
- 6) Masuda, A., et al.: Prototype Implementation of a Bandwidth Agent for Content Delivery Service, *Technical Proceedings of ISS/WTC2002*, Paris, France (2002).
- 7) Schelen, O. and Pink, S.: Resource Sharing in Advance Reservation Agents, *Journal of High Speed Networks, Special issue on Multimedia Networking*, Vol.7, No.3-4 (1998).
- 8) ITU-T Recommendation Y.1291: An architectural framework for support of Quality of Service (QoS) in packet networks (2004).
- 9) Clercq, J., et al.: An architecture for a gradual deployment of end-to-end QoS on an Internet-widescale (Virtual Service-aware Network), Internet Draft, draft-declercq-vsn-arch-01.txt (2003).
- 10) 増田暁生, 佐藤豊治, 笠原英樹：分散型受付制御のスケーラビリティ評価，信学技報，Vol.103, No.689, NS2003-271, pp.89–92 (2004).
- 11) 三好 優, 木村辰幸：“迂回経路を考慮した IP 網リソース管理方式” 信学技報，NS2003-196, pp.137–142 (2003).
- 12) 増田暁生, 廣中智浩, 笠原英樹：帯域管理のための経路情報の保持方法に関する一考察，2003 信学ソ大，B-6-90 (2003).
- 13) 北村健彦, 笠原英樹：ストリームトラフィックに対する帯域設計方法の検討，信学技報，NS2002-294, pp.177–180 (2003).
- 14) Camarillo, G., et al.: Integration of Resource Management and Session Initiation Protocol (SIP), RFC3312 (2002).

(平成 16 年 5 月 13 日受付)

(平成 16 年 11 月 1 日採録)



笠原 英樹 (正会員)

昭和 59 年埼玉大学工学部電子工学科卒業。昭和 61 年同大学大学院修士課程修了。同年日本電信電話 (株) 入社。以来、ATM、IP ネットワークの QoS に関する研究開発に従事。現在 NTT ネットワークサービスシステム研究所主幹研究員グループリーダー。平成 5 年情報処理学会研究賞。電子情報通信学会会員。



増田 暁生

平成 9 年東京大学理学部卒業。同年日本電信電話 (株) 入社。以来、NTT ネットワークサービスシステム研究所にてネットワークシステムの研究開発に従事。電子情報通信学会会員。



吉田 彰顕 (正会員)

昭和 48 年大阪大学基礎工学部電気工学科卒業。昭和 50 年同大学大学院修士課程修了。同年日本電信電話公社 (現 NTT) 横須賀電気通信研究所入所。主にデジタル無線通信システム、高速ネットワークコンピュータ技術の研究開発に従事。工学博士。平成 11 年広島市立大学情報科学部教授。現在、メディア融合型情報ネットワークの研究に従事。電子情報通信学会、IEEE 各会員。