Diffserv をもちいたフロー間帯域比指定方式の提案

Specifying Bandwidth Ratio between Network Flows for Diffserv Routers

二階堂 友秀　　佐藤 直人　　弓場 敏嗣
NIKAIDO, Tomohide　　SATO, Naohito　　YUBA, Toshitsugu

電気通信大学
University of Electro-Communications

概要

インターネットに代表される広域環境では，ルータをもちいた帯域制御による QoS の実現が重要になっている．既存の RSVP や Diffserv などのシステムでは，アプリケーションにおけるフロー間の関係を考慮せずに，フローごとに，あるいは複数のフローを集約したサービスクラスごとに帯域確保（帯域制御）をおこなっている．しかし実際には，いくつかのクライアント・サーバ型のアプリケーションで典型的にみられるように，フロー間の帯域比を一定にすることで性能が改善されるアプリケーションは多い．そこで本論文では，個々のアプリケーションがもちいる複数のフロー間の使用帯域比を直接アプリケーションが指定する方式を提案し，Diffserv ルータに組みこむキューイング・システムとして実現する．これにより，ルータがフローごとに帯域を確保しなくてもフロー間の帯域比を一定に保つことができ，多くのアプリケーションで性能改善が図られる．Network Simulator をもちいて評価実験をおこない，フロー間の帯域比を一定に保てることを確認した．

１．はじめに

インターネットに代表される広域環境では，ルータをもちいた QoS の実現，特に帯域制御による QoS の実現が重要になっている．

従来，帯域制御は主に ATM [1] や RSVP(Resource ReSerVation Protocol) [2] などの Intserv(Integrated services) 型のシステムを用いて，フローごとに帯域を確保する方法で実現されてきたが，これを広域環境へ適用するには困難がある．また Diffserv (Differentiated services) [3] ルータをもちいた，フローをいくつかのサービスクラスに分類して帯域制御をおこなう方法も最近注目されているが，各フローごとに限られた数のサービスクラスを適切に選択するのは一般に困難である．これらの問題は，アプリケーションにおけるフローの使われ方，特にフロー間の関係を考慮せずにフローごとに（またはサービスクラスごとに）独立に帯域制御（帯域確保）をおこなうことから生じている．

しかし，広域環境で動作する実際のアプリケーションを調べると，個々に固定幅の帯域を確保する必要のあるリアルタイム性のフローや，個々にサービスクラスを定められるフローばかりではなく，むしろアプリケーションが用いる複数のフロー間で相対的な帯域制御をおこなうことで性能が改善される．

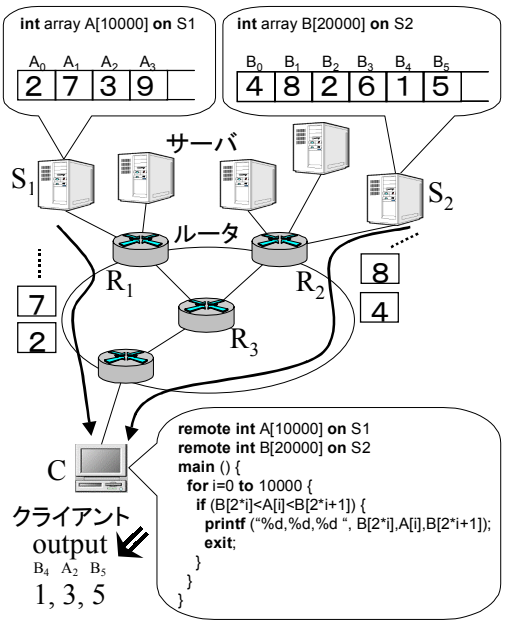


図 1. 配列の転送・走査

例えば，図 1 に示すように 2 つのサーバ S_1, S_2 から配列 A, B をクライアント C に転送して $B_{2i}<A_i<B_{2i+1}$ となる要素（i=2）をもとめる場合, A, B を転送するフローに各々等しい帯域を割り当てると，計算を完了するまでに A_0, A_1, ..., A_5 と B_0, B_1, ..., B_5 の計 12 個の要素が転送される．しかし，フロー間の帯域比を 1:2 にすることで A_0, A_1, A_2 と

B_0, B_1, ..., B_5 の計 9 個にでき，不用なデータ転送をなくし，さらに（R_3-C 間での使用帯域幅が同じ場合）計算時間も短縮できる．また，R_2-R_3 間の輻輳によって配列 B の転送スループットが低下した場合，配列 A の転送スループットも下げてフロー間の帯域比 1:2 を保ち，B にくらべて A の要素が不必要に多量に転送されないようにして S_1-C 間の帯域を有効利用できる．この様な仕組みはネットワークを介してのビデオデータの転送・再生においても有用である． 図 1 において S_1 を MPEG4 画像サーバ，S_2 を音声サーバとし，クライアント C で画像と音声を合成し再生する場合，R_1- R_3 間の帯域が不足するとフレームの欠落が生じ，その結果転送された音声データの一部が不要になる．そこで，画像と音声の 2 つの帯域比を一定の値にすることで両者の品質を同等に保ち，不要な転送をなくせば帯域を有効に利用できる．

このように，特にクライアントが同時に複数のサーバとの間でデータ転送をおこなうクライアント・サーバ型のアプリケーションにおいて，フロー間帯域比指定により実行性能の改善やネットワーク資源の有効利用が期待できる．本論文では，フロー間の帯域比を一定に保つキューイング・システムを新たに提案する．個々のアプリケーションは転送パケットのヘッダ内に直接指定帯域比を設定し，Diffserv ルータに組みこんだキューイング・システムが帯域比に従ったフロー制御をおこなう．シミュレーション・ソフトウェア Network Simulator をもちいた予備実験の結果からこのキューイング・システムが十分な実現可能性と有効性を持つことが示された．

２．Diffserv ルータによる帯域制御

2.1. ルータを用いた帯域制御

従来，帯域制御は主に ATM や RSVP に代表される Intserv 型のシステムを用いて，フローごとに帯域を確保する方法で実現されてきた．ATM における帯域制御は，VC (Virtual Channel)と呼ばれるルーティング処理をする最低単位である論理コネクションごとにおこなわれる．そのため，ATM は同一 VC 上を流れるトラフィックの一部を選択的に帯域制御することができない．一方，RSVP は end-to-end で個別のフローごとに帯域を割り当てる．しかし，ネットワークの端から端まで帯域を予約するためにバックボーン回線上では膨大な数の予約を処理する必要があり，ルータに大きな負担が掛かる．そのため，RSVP を広域ネットワーク環境へ適用することは困難である．

そこで，広域環境ではルータごとのパケット・キューイングによる帯域制御が重要になる．特に Diffserv ではフローを，EF, AF, BE の 3 つのサービスクラスに分類し，サービスクラス単位に異なる優先度でパケット・キューイングをおこなうことで，ルータの負担を低く抑えながら QoS のための帯域制御を実現している．個別のキューイング・アルゴリズムとしてこれまで Weighted Round Robin[5] や Weighted Fair Queueing [6] が提案され，実際に Diffserv ルータなどでもちいられているが，フロー間の帯域比を指定して相対的な帯域制御をおこなうには新たなキューイング・アルゴリズムが必要である．そこで，本研究では Weighted Fair Queueing アルゴリズムを応用して，指定帯域比に従ってフロー制御をおこなうキューイング・アルゴリズムを新たに作成し，Diffserv ルータ内のキューイング・システムとして実現する．

2.2. Diffserv ルータの構成

Diffserv ルータ [7] は，クラシファイア，アクション・エレメント，キュー，シェーパからなり，各々以下の機能をもつ（図 2).

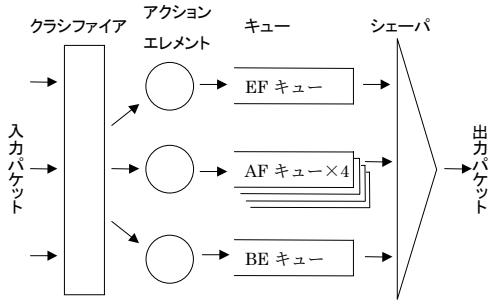


図 2. Diffserv **ルータ**

- クラシファイア（classifier）は，到着したパケットをあらかじめ指定された条件に従ってアクション・エレメントに振り分ける．クラシファイアは入力パケットの source/destination IP アドレス，source/destination ポート番号，サービスクラス（DSCP）[8] を識別できる．DSCP は ToS フィールド（IPv6 では class フィールド）を再定義した領域である（図 3, 図 4).

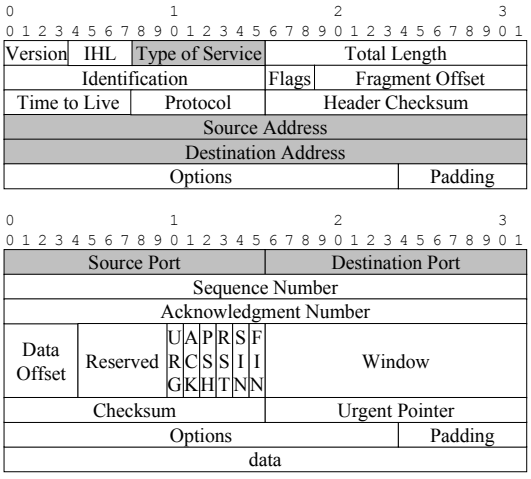


図 3. IP **ヘッダ**(IPv4)**と** TCP **ヘッダ**

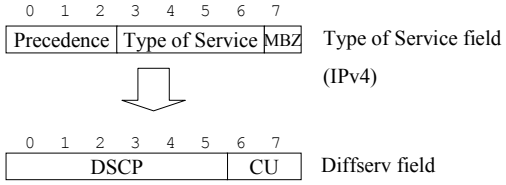


図 4. ToS **フィールドの再定義**

- アクション・エレメントは，クラシファイアから振り分けられたパケットを処理してキューに挿入する．これらの処理にはパケットの廃棄や DSCP 値の変更が含まれ，その他管理者が自由に設定できる．本研究では独自のキューイング・システムをアクション・エレメントの一つとして実現する．
- キューはサービスクラスごとに用意され，各々送出待ちのパケットを蓄える．Diffserv ルータは，1 本の EF キュー，4 本の AF キュー，1 本の BE (best effort) キューを持つ．EF (Expedited Forwarding) [9] は完全優先転送用であり，帯域確保にもちいられる． AF (Assured Forwarding) [10] は相対優先転送用で，最低帯域の保証にもちいられる．AF は 4 つの転送クラスと 3 つの廃棄レベルを持つ．
- シェーパ (shaper) はキューからパケットを取り出して送出する．各キューには重みを設定でき，WRR (Weighted Round Robin)，WFQ (Weighted Fair Queueing) [11] などのキューイング・アルゴリズムにしたがってパケットが取り出される．

2.3. Diffserv **をもちいた帯域比指定**

本研究では Diffserv ルータの DSCP 識別機能と，アクション・エレメントの拡張性を利用し，フロー間帯域比を制御するためのキューイング・システムを Diffserv ルータのアクション・エレメントとして実現する．アプリケーションは，指定帯域比をパケットの DSCP に設定し，Diffserv ルータがその帯域比に従った重み付けキューイングをおこない，指定された比によるフロー間の帯域比分割を実現する．ここで，次の点が検討課題になる．

- DSCP（6 ビット） をもちいた帯域比指定の方法
- 指定されたフロー間の帯域比を実現するキューイング・アルゴリズム

次節では，これらの問題を解決する帯域比指定の方式を提案する．

３．フロー間帯域比指定の仕組み

本節では，最初に帯域比指定のための DSCP の値の定め方を説明し，次に Diffserv ルータに組みこむ新たなキューイング・システムについて説明する．

3.1. アプリケーションによる帯域比指定

パケット・ヘッダにフロー間の帯域比指定値を付加してルータにフロー間の比を指示するために，DSCP を以下のように再定義する (図 5).

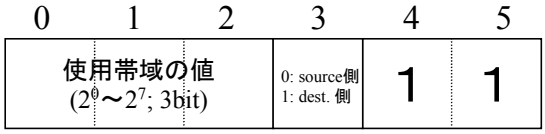
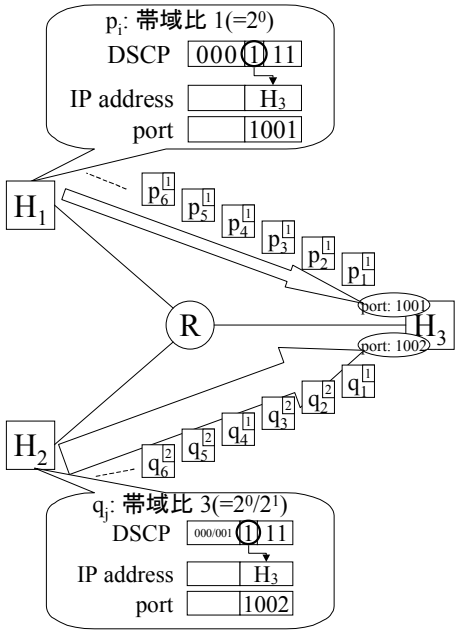


図 5. 提案する DSCP の構成（6 ビット）

- 本システムが処理するパケットをその他のパケットと区別するために，上位 2 ビットを "11" とする．
- クライアント・プログラムの IP アドレスと TCP/UDP ポート番号によりフローを識別する．またフローのどちら側でクライアント・プログラムが動作しているかを指示するために 1 ビットを用いる．
- DSCP の下位3ビットで使用帯域の比を指定する． 3 ビットでは直接には 0～7 の値しか取り得ないため，各々2^0～2^7 を表すこととし，また異なる値のパケットを混在させることで 1～255 の帯域比を指定できるようにする．



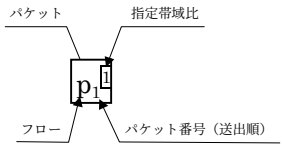


図 6. DSCP の設定

例えば，図 6 において，H_n (n=1,2,3) をホスト，R をルータとする．このとき，H_3 上のアプリケーションが H_1 と H_2 からそれぞれフロー p, q によってデータを転送するとする．そしてフロー間の帯域比を BW(p):BW(q) = 1:3 としたいとする（BW(p) はフロー p の帯域を表すとする）．H_1 から H_3 へのパケットを $p_1, p_2, p_3, \ldots$ ，H_2 から H_3 へのパケットを $q_1, q_2, q_3, \ldots$ とすると，p_i には帯域比 1 を指定する 0 ($1=2^0$) を，q_j には帯域比 3 を指定する 0 と 1 ($3=2^0+2^1$) をそれぞれ DSCP の下位 3 ビットで指示する．また，クライアント・プログラムがフローの destination 側で動作していることを指示するために 3 ビット目を 1 とする．

3.2. キューイングによる指定帯域比の実現

本節では，指定された帯域比の値に基づい

てルータ内でどのようにキューイングするかを，単一のアプリケーションが複数のフローを発生させた場合を例にして説明する．

1. クラシファイアは入力パケットのうち DSCP の上位 2 ビットが "11" のものを識別し，専用のアクション・エレメントに振り分ける．
2. アクション・エレメントにおいて，ポート番号および DSCP 中の帯域比の値をラベルとして持つキューを生成し，パケットを格納する．このキューはメモリ上に生成されるもので，以下仮想キューと呼ぶ．
3. 仮想キューごとの帯域比の値により，WFQ アルゴリズムを用いてパケットを取り出す．

例えば図 6 において，ルータは p_1, p_2, p_3, ... のフローと q_1, q_2, q_3, ... のフローをクラシファイアで選別し，本システムで作成したアクション・エレメントへ送る．アクション・エレメントではパケットの IP アドレス，ポート番号，帯域比の値ごとに仮想キューが生成される．そして WFQ アルゴリズムによって仮想キューからパケットが取り出される (図 7).

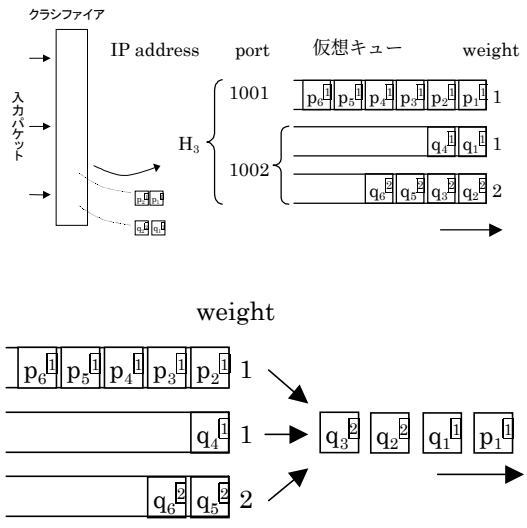


図 7. 仮想キューの生成と取り出し

3.3. 一般の場合のキューイング

3.2 節では単一のアプリケーションが複数の同方向なフローを用いる場合の帯域制御を取り上げた．しかし，実際のネットワークでは，複数のアプリケーションが各々向きの異なる複数のフローを用いる可能性がある．

そこで，まず各クライアント・プログラムが動作しているホストが source/destination のどちらの IP アドレスかを DSCP の 1 ビットで示す．そして，ルータは IP アドレスが同じフローの集まりを一つの組としてまとめて，その組の間で帯域比が指定値になるようにキューイングをおこなう．

例えば，図 8 の 2 つのアプリケーション（App.1, App.2）が，以下のフローを発生させているとする．

	flow	port	帯域比
App. 1	$p: H_1 \rightarrow H_3$,	1001	1
	$q: H_2 \rightarrow H_3$,	1002	3
App. 2	$r: H_1 \rightarrow H_4$,	2001	1
	$s: H_2 \leftarrow H_4$,	2002	2

このとき，パケット r_i, s_j のヘッダは図 8 のように設定され，これによってルータはフロー r, s を H_4 に属する 1 つの組として識別できる．

また, App.1, App.2 がともに H_3 で動作している場合（図 9）には，DSCP 値の定める際に BW(p) :BW(q)=1:3,BW(r):BW(s)=1:2 より p, q, r, s の重みを 3, 9, 4, 8 と変更し，これによりアプリケーションごとの割り当て帯域を等しくする．

一方ルータでは，各ホストの使用帯域が等しくなるように重みを正規化する（H_3, H_4 の重みの合計はともに 1，図 10 参照）．

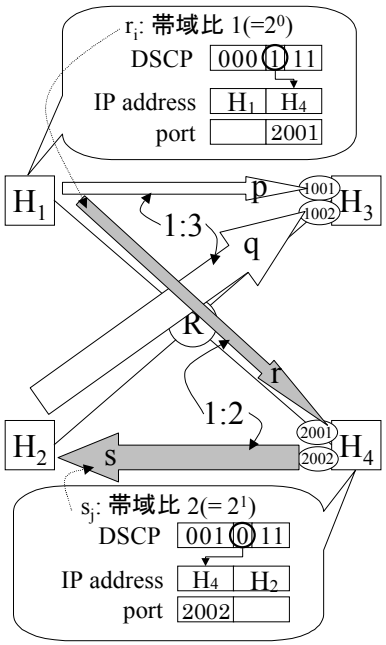


図 8. 複数のアプリケーション実行時

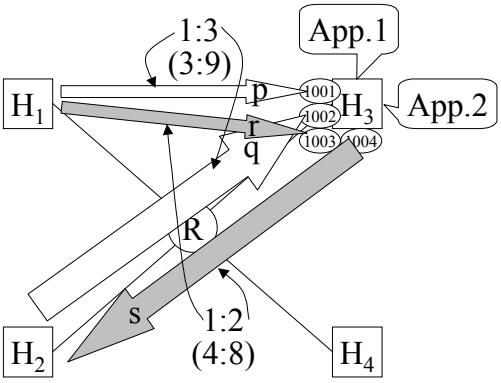


図 9. App1 と App2 を H3 で実行した場合

ここで，パケットのサイズが個々に異なることを考慮していない，という問題が生じる．実際，パケットのサイズが大きいほどより多くの帯域幅が割り当たってしまう．この問題は，パケットごとの Round Robin の代わりに，バイトごとの Round Robin によって解決できる [11].

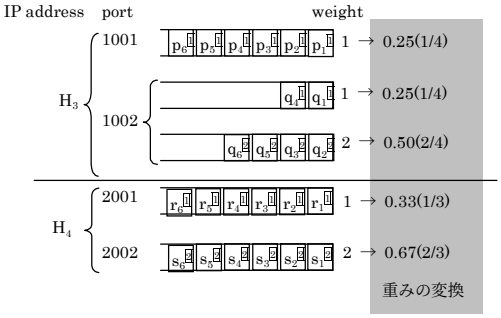


図 10. 重みの変換

3.4. 重みの動的な変更

各ホストのキューの重みを固定すると，実際のトラフィックによっては，フロー間の指定帯域比を実現できない場合がある．例えば図 8 で，フロー p の H_1-R 間のスループットが低下した場合 BW(p):BW(q) = 1:3 とするには p, q への割り当てる重みを小さくする必要がある．そこで以下の手順で重みを動的に変更する．

1. 各フローを監視してスループットが低下したものがあれば，そのフローが属するホスト全体の重みを一定の割合で小さくする（未使用帯域・資源の回収）.
2. その分を他のホストに等分して割り当て，各ホストでは，キューの重みを割り当てられた分だけ大きくする（再配分）.

例えば図 10 においてフロー p のキューからのパケットの取り出しが 2 回に 1 回失敗するとする（キューが空のため）．この時の帯域利用率を 50% として，H_3 の 3 つのキューの重みを各々 0.25→0.125, 0.25→0.125, 0.5→0.25 と変更する．そして H_4 の 2 つのキューの重みを 0.33→0.5, 0.67→1.0 と変更する．

一方，帯域利用率が 100% のホスト間では，使用する帯域が等しくなるように，必要に応じて重みを変更する．

3.5. 物理キューへのパケットの格納

仮想キューの数は，実際にルータの持つキュー（以下物理キュー）よりも多くなるため，仮想キューに収められたパケットをそのまま物理キューに移してルータから送出することはできない．そこでアクション・エレメントにおいて，以下のように仮想キューのパケットを物理キューに移しかえる（図 11).

(1) 仮想キューを WFQ アルゴリズムに従って１つにマージする.
(2) マージしたキューから物理キューに，重みに従ってパケットを分配する.

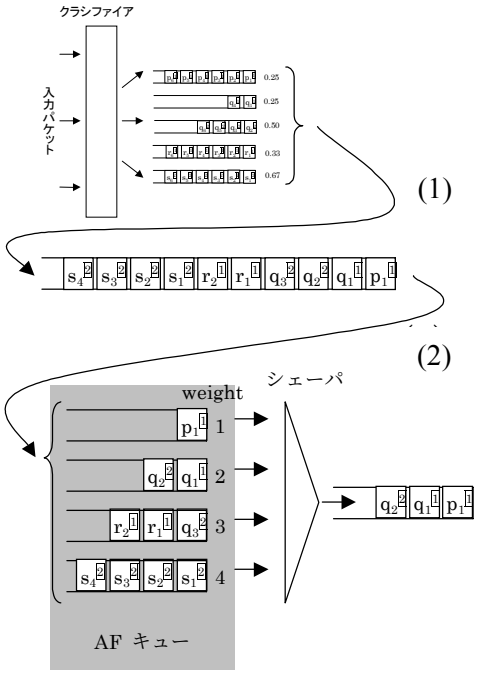


図 11. AF キューへのパケットの移しかえ

本システムで用いるルータの構成は図 12 のようになる.

４．ネットワーク上での実現

Diffserv では, ネットワークを DS ドメインと呼ばれる部分ネットワークの集まりと

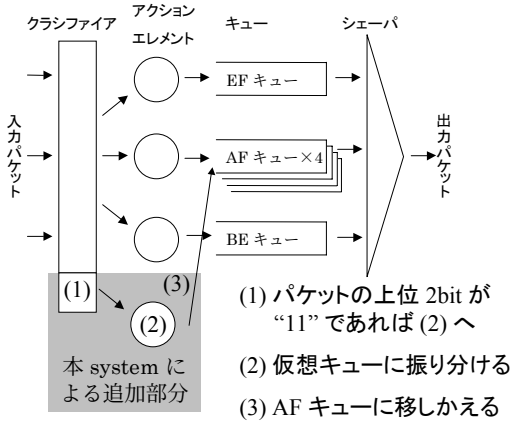


図 12. 本システム におけるルータの構成

して構成する．DS ドメインの境界上のルータはエッジルータと呼ばれ，個別のフロー制御や DS ドメイン間の接続を担当する．一方, DS ドメイン内のルータはコアルータと呼ばれ，エッジルータが集約したフローをサービスクラス（EF, AF, BE）ごとに処理する．個別のフロー制御からサービスクラスごとの制御への変換の仕組みはポリシーと呼ばれ，DS ドメイン内で一貫したポリシー管理をおこなうために, Bandwidth Broker (BB) [12] によるポリシー管理機構が提案されている.

コアルータは主にバックボーン回線で大量のトラフィックの処理をおこなうため負荷が高く，前節で述べたフロー単位のキューイングはコアルータには負担が大きすぎる．そこで，指定帯域比に従ったキューイングはエッジルータでおこない，コアルータでは通常の AF サービスクラスのパケットとして扱い，そのまま転送する.

例えば図 13 に示すように p (S_1-R_1-R_3-R_4-C) と q (S_2-R_2-R_3-R_4-C) の２つのフローの間で帯域比を設定した場合，指定帯域比による帯域制御は p と q が合流する R_3 で最初におこなわれる．R_3 は p, q のうち，スループットの小さい方を基準に，指定帯域比にあわせてフロー毎の重みを設定する．例え

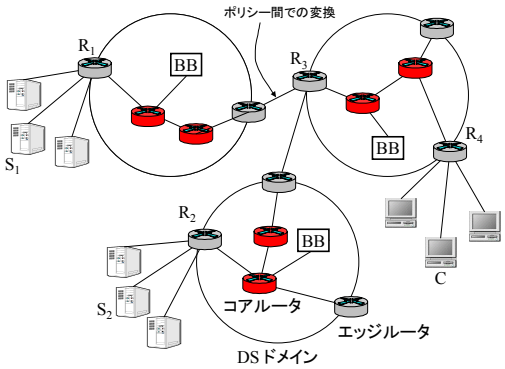


図 13. エッジルータとコアルータ

ば, p と q のうち, R_1-R_3 間の輻輳により p のスループットが小さくなっているとする. このとき R_3 は, 帯域比を保つために p のスループットに合わせて q の帯域を狭めるように重みを変更する. その結果, R_2 においても RED [13] などの輻輳回避機構の働きにより S_2-R_3 間の q のスループットが低下し, S_2-R_3 間の帯域を他のアプリケーションが有効利用できるようになる.

5. 予備評価

性能評価として, Linux PC (Pentium III, 600MHz) 上で Network Simulator 2.1 (NS-2) をもちいて実験をおこなった. NS-2 はパケット単位でトラフィック・シミュレーションをおこなうソフトウェアである. 本論文で提案するキューイング・システムは, NS-2 用の Diffserv コンポーネントである Diffserv addition [14] を改変して作成した.

実験には図 14 に示す 3 つのネットワークをもちいた. 各実験において物理キュー EF, AF, BE の重みは 1:1:1 の比で設定した. また, リンクの帯域は各ルータ間が 5 Mb/s, ルータと各ホスト (クライアント, サーバ) 間が 10 Mb/s とし, 各実験とも全体の実験時間は 60 秒とした.

実験 1：フロー間の帯域比の維持

ネットワークは, 図 14(1a) に示す通り, 2 台のルータ R_1, R_2 と, R_1 につながれたサーバ群 S1~S4 および R_2 につながれた 2 台のクライアント C_1, C_2 からなる. 4 つの TCP フロー p(S_1-C_1), q(S_2-C_1), r(S_3-C_2), s(S_4-C_2) をつくり, BW(p):BW(q)=1:2, BW(r):BW(s)=1:4 と指定した. また, 他に EF, AF, BE クラスの TCP フローを R_1-R_2 間に流した. 各フローの送出時間は以下のようにした.

- フローp, q ：実験開始から 40 秒まで
- フロー r, s ：20 秒から実験終了まで
- EF, AF, BE クラスのフロー各 1 つ：実験開始から終了まで

実験の結果, 図 14(1b) に示す通り, p, q および r, s 間の帯域比は指示通り各々 1:2, 1:4 になった. また, 図 14(1c) のスループットの変化に示す通り, (1) p, q の組と r, s の組の間の帯域比が 1:1 になり (BW(p)+BW(q):BW(r)+BW(s)=1:1), (2) EF, BE と p, q, r, s, AF を合わせた帯域の比は 1:1:1 であることがわかる.

実験 2：動的な重みの変更

ルータ R_1, R_2, R_3 とサーバ S_1~S_4, クライアント C_1, C_2 を図 14(2a) に示すように接続した. そして, フロー p, q, r, s を実験 1 と同じ帯域比を指定して 60 秒間流し, さらに S_1-R_1 間に UDP のフローを $20 \leqq T \leqq 40$ の間 4.8 Mb/s で流した.

結果は図 14(2b) に示す通り, 実験を通して, BW(p):BW(q)=1:2, BW(r):BW(s)=1:4 が保たれている. また 図 14(2c) より, $0 \leqq T \leqq 20$ および $40 \leqq T \leqq 60$ では, UDP フローのために S_1-R_1 間の r のスループットが 0.2 Mb/s に低下している. その結果, R_3 において p, q, r, s の重みが変更され (3.4 節), 各々

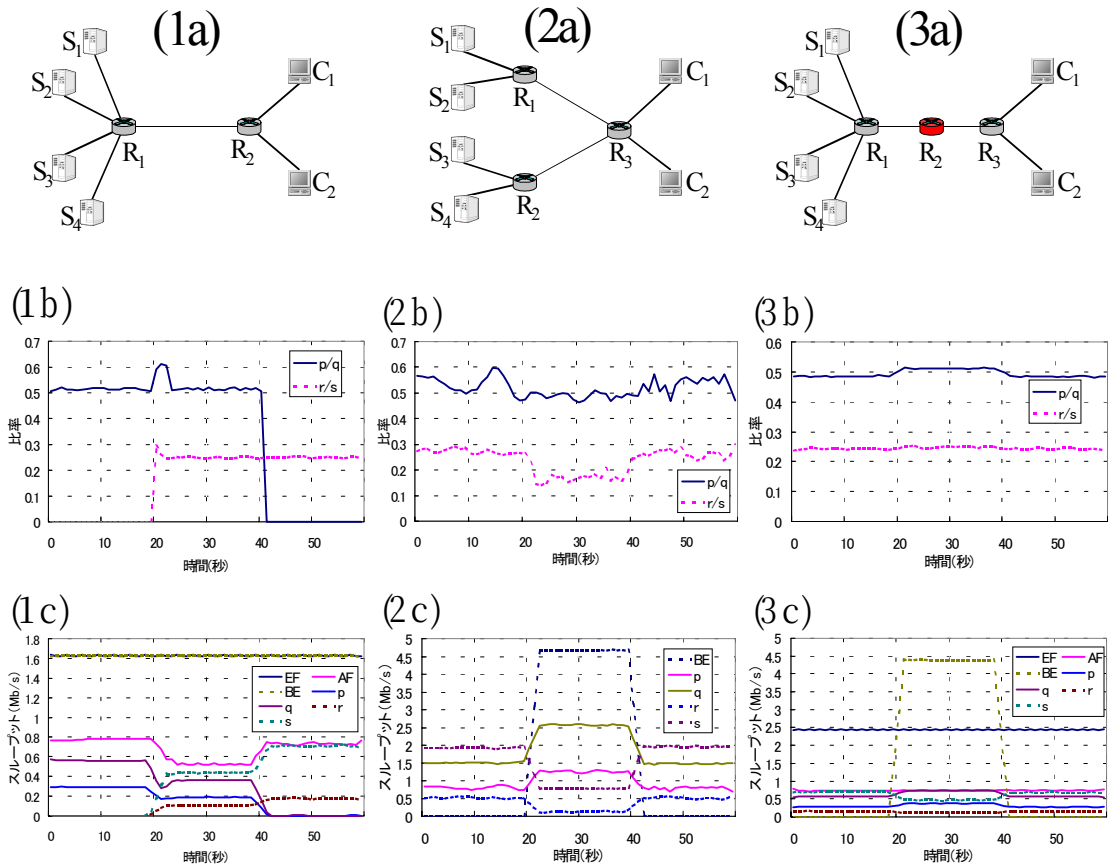


図 14. 評価実験（帯域比およびスループット）

のスループットが r: 0.2 Mb/s, s: 0.8 Mb/s, p: 1.3 Mb/s, q: 2.6 Mb/s となるように重みが変更された.

実験 3：コアルータの追加

実験 1 の 2 つのルータの間にコアルータを 1 台追加して R_1-R_2-R_3 とした. そして, フローp, q, r, s を実験 2 と同様に流し, さらに S_1-R_1 間に UDP フローを 20≦T≦40 の間 4.5 Mb/s で流した.

図 14(3b) に示す通り, 実験を通して BW(p):BW(q)=1:2, BW(r):BW(s)=1:4 が保たれている. また, 実験 2 と同様に 20≦T≦40 では s のスループットが低下したために r, s の重みが低下し, その分 p, q の重みが増大している. 40≦T で s のスループットが回復すると重みも T≦20 と同じ値に戻っている. コアルータ R_2 を挿入したことによる重みの変更の遅れ等は, 今回の実験では観察できなかった.

以上 3 つの実験では, 2 つのアプリケーションが各々2 つのフローを要求し, その間の使用帯域比を指定した場合を想定した. 各フローは, 要求された比を維持しており, 本システムが機能していることを示している. また, 複数のフローの組がある場合, 1 つのあるフローのスループットが低下した場合に, 帯域比を指定した別のフローのスループットも下がり, 同時に他の組のフローのスループットが上がることを確認した.

6. おわりに

本論文では, アプリケーションが使用する複数のフローの間の帯域比を指定した値に保

つためのキューイング・システムを新たに提案した．そして，Diffserv ルータ内のアクション・エレメントの 1 つとして実現し，Network Simulator をもちいて予備評価をおこない，実際に機能することを確認した．

今後は，より大規模なネットワークをもちいて実験をし，本システムが広域環境でも有効に機能することを検証する．これについては，ソフトウェア・ルータ（gated [15] や ALTQ [16]）上での評価もあわせて検討している．また，提案したキューイング・システムを実際の Diffserv ルータのアクション・エレメントとして実装できるかどうかも検証が必要である．仮想キューを実現する場合，メモリが不足していたり，CPU の性能が低いと十分な性能が得られない．実際の Diffserv 上での検証もあわせておこなっていく予定である．

参考文献

[1] ATM Traffic Management Specification Version 4.0 <af-tm-0056.000>, ATM Forum (Apr. 1996).

[2] Braden, B., Zhang, L., Berson S., Herzog, S. and S. Jamin. Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification. RFC 2205 (Sep. 1997).

[3] Blake, S. et al. An Architecture for Differentiated Services. IETF RFC2475 (Dec. 1998)

[4] McCanne, S. and Floyd, S. Network Simulator -ns (version2). http://www-mash.cs.berkeley.edu/ns/

[5] J. Nagle, On Packert Switches with Infinite Storage. IEEE Transactions on Communications. COM-35, No. 4, pp. 435-438 (Apr. 1987).

[6] Demers, A., Keshav, S., and S. Shenker. Analysis and Simulation of a Fair Queuing Algorithm. Proceedings of SIGCOMM '89.

[7] Bernet, Y. et al. A Conceptual Model for Diffserv Routers. IETF INTERNET-DRAFT (May 2000).

[8] Nichols, K. et al. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. IETF RFC2474 (Dec. 1998).

[9] Jacobson, V. et al. An Expedited Forwarding PHB. IETF RFC2598 (Jun. 1999).

[10] Heinanen, J. et al. Assured Forwarding PHB Group. IETF RFC2597 (Jun. 1999).

[11] Shreedhar, M. and Varghese, G. Efficient Fair Queuing using Deficit Round Robin. (Oct. 1995).

[12] Neilson, R. et al. A Discussion of Bandwidth Broker Requirements for Internet2 Qbone Deployment. http://www.merit.edu/working.groups/i2-qbone-bb/

[13] Floyd, S., and Jacobson, V. Random Early Detection Gateways for Congestion Avoidance, IEEE/ACM Transactions on Networking, Vol.1 No.4 (August 1993).

[14] Murphy, S. Diffserv additions to ns-2. http://www.teltec.dcu.ie/~murphys/ns-work/diffserv/

[15] Merit GateD Consortium. http://www.gated.org/

[16] Cho, K. A Framework for Alternate Queueing:Towards Traffic Management by PC-UNIX BasedRouters. http://www.csl.sony.co.jp/person/kjc/papers/usenix98/altq.html