マルチメディア通信と分散処理 107－14
（２００２．３．２８）

Diffserv と連動した経路選択機能の実装

頴原 桂二郎[†] 小川 晃通[†]
杉浦 一徳[††] 村井 純[†††]

本研究では，BSD の経路表を多段化し，Diffserv におけるサービス区分を表わす DSCP の値ごとに経路表を選択できる機構およびそれらを操作するインターフェイスを実装した．DSCP を経路制御に結びつく要素とし，有線・衛星など明らかに異なる特性を持つリンクを利用した経路を効率良く切り替えることは有用である．このように DSCP に特別の意味を持たせ Diffserv のサービス区分をもとに経由すべき経路を選択できるようにすることで，よりサービスに適したパケット転送が可能となる．本実装は 2001 年春の WIDE 合宿のバックボーンで 地上・衛星回線の選択に使用され，その有用性を確認するとともに評価をおこなった．また，追加される経路が小規模であると予想されたため，経路表の切り替えおよび経路計算は線的なものとしているが，規模性を考慮した場合，メモリ使用量・計算時間など改善の余地は多く，この点についても考察する．

Implementation of diffserv linked routing engine

Keijiro Ehara ,[†] Akimichi Ogawa ,[†] Kazunori Sugiura [††]
and Jun Murai [†††]

We designed and implemented a system which looks up a next-hop with DSCP vale, source address and destination address in the IP packet. IP routing referring a DSCP value is useful in case there are two or more links which clearly has different feature like between a satellite link and a ground link. This way, routers can forward packets more suitably for provisioning service of customers. We evaluated this system by operating a live Diffserv ready network with dynamic service level agreement. In this paper, we present design, evaluation and conclusion of the operational experiment.

1. はじめに

インターネットにおけるパケットの到達性は，経路制御技術の発展と運用経験の蓄積により著しく向上した．また，VoIP といったリアルタイムアプリケーションの充実により，ユーザのインターネット通信技術への関心は単なるパケットの到達性から，通信品質へと遷移しつつある．

ユーザの要求である通信品質を保証する技術として Diffserv, MPLS や QoS 経路制御が挙げられる．我々はこれらの技術を利用して，ユーザから動的な資源予約を可能にする機構を設計・実装した．この機構を 250 人規模のネットワークで実際に運用実験をした．

この機構は，ユーザが資源を予約する Web インターフェイス，予約や帯域資源のアドミッションを行なうサーバ，各ルータを予約にしたがって設定するモジュール，設定された内容にしたがってパケットを転送するルータといった各モジュールにわけられる．本論文ではこの機構のうち，ルータに実装した，パケットヘッダ中の宛先アドレスと DSCP の値による経路選択機能について述べる．

† 慶應義塾大学 政策・メディア研究科
Keio University Graduate School of Media and Governance
†† 独立行政法人通信総合研究所
Communications Research Laboratory
††† 慶應義塾大学 環境情報学部
Keio University Faculty of Environmental Information

2. WIDE 合宿での運用実験

2.1 実験の概要

WIDE プロジェクト[1)] はインターネットとその関連技術に関する研究と開発を行なう非営利団体である．

WIDE プロジェクトでは春と夏の年 2 回，メンバによる 4 日間の合宿を行なっている．WIDE 合宿では “Camp-net” と呼ばれる一時的な運用ネットワークを構築し，インターネットとの接続性を確保するとともにさまざまな実験を行なう．例えば，IPv6 や MPLS のような最新のネットワーク技術を合宿参加者に提供している．

我々は WIDE プロジェクト 2001 年春の合宿においてユーザが動的に使用する対外線の種類と帯域を選択できる Diffserv / MPLS ネットワークを運用・実験した．この合宿は愛知県西浦温泉で行なわれ，259 名が参加した．

2.2 ネットワークトポロジ

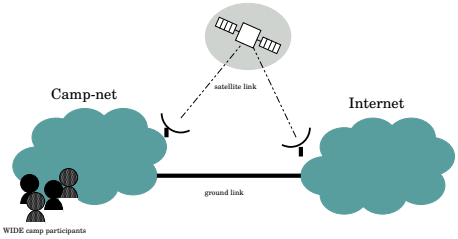

図 1 Camp-net の対外線

Camp-net の対外線は，**図 1** に示すように，地上回線経由と衛星回線経由の 2 種類の経路が用意された．地上回線は，合宿地と慶應義塾大学湘南藤沢キャンパス間に HSD 128 Kbps の回線を 5 本用意し，PPP/MP で 1 回線に束ねて使用した．衛星回線は，双方向 768 Kbps で同キャンパスとの間を結んだ．

Camp-net のトポロジは**図 2** のようになっており，合宿参加者のために 6 つのセグメントを用意した．

4 つのセグメントは NAT ルータの下にあり，プライベートアドレスがこの NAT セグメントのユーザに配られた．そのうち 2 セグメントは IEEE 802.11b による無線 LAN であり，2 セグメントは FastEthernet である．

残りの 2 つのセグメントは FastEthernet であり，グローバルアドレスが配られた．

2.3 帯域資源要求

本実験では，合宿参加者からの動的な資源要求に応じてアドミッション制御をし，品質予約が設定される．合宿参加者には資源要求を発行できる Web ページを提供した．このページにアクセスするため，合宿初日

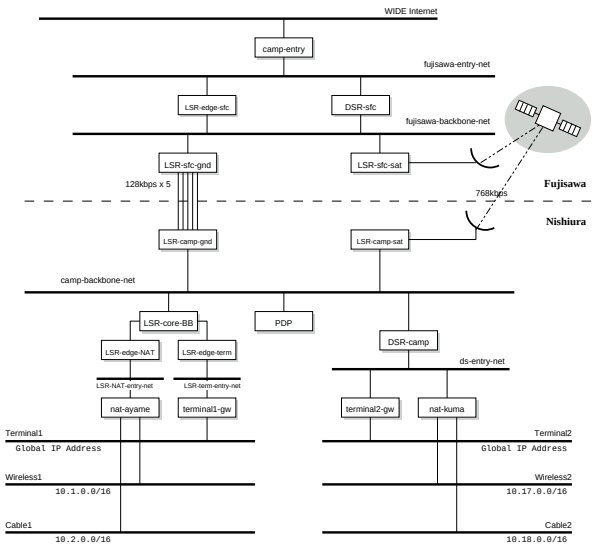

図 2 Camp-net トポロジ

に合宿参加者の各個人に対して固有の ID とパスワードを配布した．

Web ページでは，衛星回線と地上回線のどちらを利用するか，必要な帯域，フロー情報等を入力することで予約ができる．現在の予約状況，残り帯域といった情報等も Web ページで提供しており，合宿参加者はこのページを通じて資源予約に必要な情報の取得および資源予約自体のすべての操作をできる．

我々実験チームは，回線選択機構に，MPLS によるパス設定と IP の経路表によるものの 2 種類を用意した．これらは参加者のいるセグメントによりどちらが実際に使われるかが決定されるが，参加者はそれを意識する必要はない．

また，予約・回線の指定のない場合には，遅延の大きな衛星回線を通るようにした．

2.4 各ルータの役割

各ユーザセグメントを収容するエッジルータは，参加者からの帯域資源要求にしたがってトラフィックを識別し，パケットに DSCP を設定する．各ルータには AltQ[2] が実装されており，パケットは DSCP にしたがった転送時のふるまいを受ける．

回線選択は，Camp-net トポロジ図 2 の右下の 3 つのユーザセグメントでは本論文の IP の経路表によるもので，左下の 3 つのユーザセグメントでは MPLS によるものとなる．

3. 設計と実装

3.1 設 計 方 針

通常，ルータはパケットヘッダ中の宛先アドレスのみを次ホップの選択に利用する．この単純さのため，BSD スタックでは経路の計算方法や経路情報を格納する構造体は確立されており，ラディックスツリーを使用するものが主流である．

本実装では，次ホップの選択にパケットヘッダ中の宛先アドレスのほかに，DSCP の値をキーとして利用する．このため，複数のキーを利用したなんらかの検索方法を考えなければならない．

実装するにあたり，250 人程度の中規模ネットワークで利用するものであり追加される経路数に限りがあることと，新しい試みの有用性を確認する目的の実装であることを考慮した結果，経路表を DSCP の値ごとに多段化し，それらを切り替えて従来のルーチンを再利用する方法をとることとした．このため，カーネルへの変更は比較的少量となり，容易なものとなる．

また，新たな経路を DSCP の値とともにカーネルに加えることができるようにルーティングソケットを変更し，従来の route コマンドにこの機能を付加させる必要がある．

3.2 経路決定までの流れ

データリンク層で処理されたパケットは，**図 3** に示されるように ip_input() を通じてネットワーク層での処理を移行する．ip_input() では，チェックサムの検証やフラグメントされたパケットの再構築が行なわれる．その後，自分宛てのパケットかどうかを調べ，そうでなければ ip_forward() を呼んで次ホップにパケットを転送する．ip_forward() 内での次ホップの決定は rtalloc() を呼ぶことで行なわれる．

rtalloc() の引き数である route 構造体は**図 7** に示されるとおりであり，宛先アドレスしか格納できない．宛先アドレスに加えて，DSCP の値をキーとして経路を決定するためには rtalloc() を使用できないことがわかる．

また，DSCP が設定されていないパケットを転送する際には rtalloc() を使用する．このため rtalloc() 自体は変更せずに残しておき，DSCP に値が設定されている場合のみ異なる経路表を検索するようにした．具体的には，**図 5** に示されるように ip_forward() では関数 qosrtalloc() を呼び，その先で DSCP が設定されていなければそのまま rtalloc() を呼ぶことで対応した．

```
...
struct  route ipforward_rt;
...
void
ip_input(struct mbuf *m)
{
        ...
                ip_forward(m, 0);
        ...
}
...
void
ip_forward(m, srcrt)
        struct mbuf *m;
        int srcrt;
{
        ...
                rtalloc(&ipforward_rt);
        ...
}
...
```

図 3 ip_input() から rtalloc() が呼ばれるまでの流れ (sys/netinet/ip_input.c)

```
struct route {
        struct  rtentry *ro_rt;
        struct  sockaddr ro_dst;
};
```

図 4 route 構造体 (sys/net/route.h)

関数 qosrtalloc() では引き数で与えられた DSCP の値により検索するラディックスツリー自体を切り換える．これら複数のラディックスツリーへのポインタはリスト構造で管理した．切り換えられたラディックスツリーを検索することで DSCP ごとに異なる次ホップを決定できる．

3.3 ルーティングソケットの変更

ユーザからの動的な帯域資源要求はアドミッション

```
void
ip_forward(m, srcrt)
        struct mbuf *m;
        int srcrt;
{
        ...
        qosrtalloc(&ipforward_rt, dscp);
        ...
}
```

図 5 qosrtalloc() の呼び出し (sys/netinet/ip_input.c の変更)

制御を経て，サーバから各ルータで実行されているデーモンに通達される．このデーモンがルーティングソケットに RTM_ADD メッセージを送ることでカーネルに経路が加えられる．このとき，DSCP 情報も含めなければならない．本実装では RTM_ADD メッセージを発行する際，宛先アドレスを格納する**図 6** で示される sockaddr_in 構造体中の sin_port に着目した．変数 sin_port はポート番号を通常格納されるがルーティングソケットでは使用されないものである．この sin_port 変数に DSCP の値を入れ，RTM_ADD メッセージを発行することでルーティングメッセージに DSCP の値を含めることができる．

```
struct sockaddr_in {
        u_int8_t        sin_len;
        sa_family_t     sin_family;
        u_int16_t       sin_port;
        struct in_addr  sin_addr;
        int8_t          sin_zero[8];
};
```

図 6 sockaddr_in 構造体 (sys/netinet/in.h)

ルーティングソケットは RTM_ADD メッセージを受けとると，与えられた DSCP に対応する経路表が存在するかを調べる．存在する場合は経路表を切り替えたのち，rtrequest() 関数により経路をラディックスツリーに加える．存在しない場合は新たにラディックスツリーを作成し，DSCP の値に関連付けたのちに rtrequest() 関数を呼ぶ．

経路の削除を意味する RTM_DELETE メッセージも RTM_ADD メッセージ同様の処理を行ない，すべての経路が削除された場合は経路表自体も削除される．

3.4 route コマンドの変更

ルーティングソケットへのインターフェイスとして route コマンドがある．図 7 に示されるように，route コマンドに DSCP の値をオプションとして設定できるように変更した．

```
% route add -inet 203.178.143.31 \
  -dscp 0x10 203.178.143.30
% route delete -inet 203.178.143.31 \
  -dscp 0x10 203.178.143.30
```

図 7 変更された route コマンド

3.5 実 装 環 境

本機構は NetBSD-current 上に実装した．ルータ PC の主要スペックを**表 1** に示す．

表 1 ルータ PC の主要スペック

CPU	Intel PentiumIII 750MHz
メモリ	128MB
NIC	Intel Pro/100+ x2

4. 評価と考察

4.1 Camp-net における実験の評価

3 日間にわたり，本実装によるルータを動作させたが直接的な問題は発生せず，良好な結果が得られた．資源予約の Web インターフェイスから回線設定を行なうと，衛星回線から地上回線へと経路が切り替わるため合宿地外への RTT が**図 8** に示されるように大幅に改善される．また，6000 回の ICMP ECHO_REQUEST による各回線の RTT 測定結果を**表 2** に示す．

表 2 RTT 測定結果

	地上回線	衛星回線
最小値 (msec)	57.315	516.138
平均値 (msec)	116.505	625.877

```
% ping 203.178.143.31
PING puel.sfc.wide.ad.jp (203.178.143.31): 56 data bytes
64 bytes from 203.178.143.31: icmp_seq=0 ttl=255 time=630.451 ms
64 bytes from 203.178.143.31: icmp_seq=1 ttl=255 time=625.218 ms
64 bytes from 203.178.143.31: icmp_seq=2 ttl=255 time=593.209 ms
64 bytes from 203.178.143.31: icmp_seq=3 ttl=255 time=642.224 ms
64 bytes from 203.178.143.31: icmp_seq=4 ttl=255 time=651.268 ms
64 bytes from 203.178.143.31: icmp_seq=5 ttl=255 time=93.231 ms
64 bytes from 203.178.143.31: icmp_seq=6 ttl=255 time=83.214 ms
64 bytes from 203.178.143.31: icmp_seq=7 ttl=255 time=102.228 ms
64 bytes from 203.178.143.31: icmp_seq=8 ttl=255 time=96.224 ms
64 bytes from 203.178.143.31: icmp_seq=9 ttl=255 time=103.211 ms
64 bytes from 203.178.143.31: icmp_seq=10 ttl=255 time=98.239 ms
```

図 8 回線選択による RTT の改善

また，MPLS とは違い IP 層での経路制御であるため traceroute コマンドや RECORD_ROUTE オプションを含めた ICMP ECHO_REQUEST などにより経路が動的に変わるのを確認できるため，ユーザへのフィードバックという点においては効果があると言える．

4.2 考　　察

本実装では，各ルータでパケットの宛先アドレスと DSCP の値をキーとして経路を検索し，次ホップを決定した．Camp-net に特化した設計であるため，検索のキーがさらに増えたりした場合への対応は難しい．特に検索キーが増えていくと，CPU 使用率，メモリ消費量および検索時間に多大な影響が出る．検索キーを増やすためにはラディックスツリーを使用せずに，さまざまな考案がなされている 2 次元フィルタなどを実装すべきであろう．2 次元フィルタを使用しても，CPU 使用率とメモリ消費量は相反する要素である．ルータの使用される環境や経路数などに応じて動的に最適な検索方法に切り替わるの機構も面白いだろう．また，本機構の機能をネットワークプロセッサ上に構築するのもパフォーマンスの向上に繋がると思われる．

5. ま と め

本研究では，Camp-net からインターネットへの対外線に対して品質予約を行なうことにより，衛星回線および地上回線を選択的に利用できる機構を構築・運用した．この機構は Diffserv ネットワークにおけるユーザからの動的資源予約を可能とした包括的な機構のひとつのモジュールである．

Diffserv は，VoIP などの新しいリアルタイムインターネットアプリケーションの登場とともに高まる通信品質向上への要求を満たすために標準化された．この Diffserv においてサービスクラスを意味するパケットヘッダ内の DSCP の値に特別な意味を持たせ，この値を参照して各ルータでの次ホップを決定することはサービスを提供していく上で有用である．例えば，明らかに遅延やジッタなどの異なる特性を持つリンクが存在する場合，よりサービスに適した経路を通すことが可能となる．

今後は，動的経路制御との協調などの検討を進めていくとともに，大規模ネットワークでも使用に耐えうるカーネル内での経路情報の持ち方や検索方法について研究を行なっていく予定である．

参 考 文 献

1) WIDE プロジェクト, http://www.wide.ad.jp
2) ALTQ: Alternate Queueing for BSD UNIX, http://www.csl.sony.co.jp/person/kjc/software.html