マルチホームネットワークにおけるUDP通信の透過的トラヒック分散

山井成良[1]　久保武志[2]　岡山聖彦[3]　宮下卓也[1]

[1]岡山大学 総合情報処理センター
[2]岡山大学 大学院自然科学研究科
[3]岡山大学 工学部

概　要

ネットワークサービスの応答時間の悪化に対処する1つの方法として，複数のバックボーンを通信先に応じて使い分けるマルチホームネットワークが注目されている．我々はマルチホームネットワークを容易に構築でき，かつ効率よく利用することを目的として，TCP通信を対象とした透過的トラフィック分散方法を提案した．しかし，この方法ではTCP通信のコネクション確立時間に基づいてバックボーンを選択しているため，そのままではUDP通信のトラフィック分散に適用できない．そこで，本稿ではUDP通信を対象とした透過的トラフィック負荷分散手法を提案する．これにより，マルチメディア通信などUDPを用いたアプリケーションに対してもマルチホームネットワークを用いて応答時間の短縮を容易に図ることが可能となる．

A Transparent Load Sharing of UDP Traffic on Multihomed Networks

Nariyoshi Yamai[1], Takeshi Kubo[2], Kiyohiko Okayama[3] and Takuya Miyashita[1]

[1]Computer Center, Okayama University
[2]Graduate School of Natural Science and Technology, Okayama University
[3]Faculty of Engineering, Okayama University

Abstract

Multihomed network, that is a kind of network connected to the Internet via more than one backbones, is one of the most interesting networks to improve response time of network services. In order to operate multihomed networks easily and efficiently, we proposed a transparent dynamic load sharing method for TCP traffic. However, this method cannot be applied to load sharing for UDP traffic since criterion of backbone selection was TCP connection setup time. In this paper, we propose a transparent dynamic load sharing method for UDP traffic. With the proposed method, we can easily improve response time of many UDP applications such as multimedia communication and so on.

1　はじめに

インターネットの利用は年々増加の一途を辿っており，これに伴いWWW等のネットワークアプリケーションにおける応答時間の劣化が顕著になってきている．これに対処する一つの方法として，自組織のネットワークを複数のバックボーンネットワーク(以下，単にバックボーンと呼ぶ)と接続し，通信先に応じて利用するバックボーンを使い分けることにより応答時間の改善を図るマルチホームネットワークが最近注目されるようになってきた．

しかし，マルチホームネットワークでトラヒックを分散する場合，従来の経路制御方法ではバックボーンから入手した経路情報と通信先アドレスのみで利用するバックボーンが一意に定まるため，通信先に偏りが生じると効率的なトラヒック分散が行われず，特定のバックボーンにトラヒックが集中する危険性がある．また，一般にマルチホームネットワークで

は接続先バックボーンから経路情報を入手できるようにバックボーン管理者と協調して設定作業を行う必要があり，導入や管理にかなりの技術レベルと管理コストが要求される点も問題である．このような理由から，マルチホームネットワークの導入は特に中小規模の組織では困難であった．

この問題に対処するため，我々の研究グループでは，複数のバックボーンと自組織のネットワークとの接続を受け持つルータにおいて，各バックボーンの状態を自らが判断して，コネクション単位で適切なバックボーンを選択する方法を提案した [1]．しかし，この方法は TCP 通信のコネクション確立時間に基づいてバックボーンを選択しているため，そのままでは UDP 通信のトラフィック分散に適用できない．そこで，本稿では UDP 通信を対象とした透過的トラフィック負荷分散手法を提案する．これにより，マルチメディア通信など UDP を用いたアプリケーションに対してもマルチホームネットワークを用いて応答時間の短縮を容易に図ることが可能となる．

2 TCP 通信の透過的トラヒック分散方法

マルチホームネットワークは，1つのネットワークを複数のバックボーンによりインターネットに接続する形態で，トラヒック分散による応答性の改善や耐故障性の向上などを図る方法として注目されている．これまでに知られているマルチホームネットワークの構成方法としては，AS(Autonomous System) 番号を取得する方法 (方法 1)[2]，NAT(Network Address Translation) [3, 4] を用いる方法 (方法 2)[5]，アプリケーションゲートウェイ (Application Level Gateway, ALG) を用いる方法 (方法 3)[6] などが挙げられる．

このうち，方法 1，2 では BGP4[7] を用いて経路情報の交換を行う必要があり，導入・管理にかなりの技術レベルと管理コストが要求される点が問題となる．また，これらの方式では，経路制御は送信先アドレスのみに依存して行われ，現在のトラヒック量などバックボーンの利用状況が反映されないため，通信先に偏りがあった場合に特定バックボーンにトラヒックが集中する可能性も残されている．

一方，方法 3 では適用できるアプリケーションが WWW など ALG に対応した一部のものに限られ，また ALG に対応したアプリケーションであってもユーザが ALG の存在を意識する必要がある点が問題となる．

これらの問題を解決するため，我々の研究グループでは独自の経路制御方法に基づいてこれらの問題を解決する新しい動的トラヒック分散方法を提案した．以下ではその概要を説明する．

2.1 ネットワーク構成

以下の説明では比較的小規模のネットワークを対象とし，図 1 に示すように，自組織のネットワーク (LAN) を 1 つのルータ R により 2 つのバックボーン (B1, B2) に接続した構成を取るものとする．また，自組織のネットワークには B1 から与えられたアドレスが割り当てられており，B2 とは NAT を経由してアクセスするように設定されているものとする．なお，バックボーン B1 を以下ではデフォルトバックボーンと呼ぶ．

このような構成のネットワークにおいて，本方法では内部から外部への TCP コネクションを対象とし，コネクション確立時にルータが適切なバックボーンを選択する．

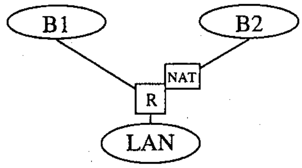

図 1: マルチホームネットワーク

2.2 往路・復路でのトラヒック分散

次に，往路と復路それぞれにおけるトラヒック分散について述べる．

往路において，ルータはそれぞれのバックボーンの状態を監視し，新たなコネクション確立を要求するパケットが来ると，その時点での各バックボーンの状態から適切なバックボーンを選択する．一度コネクションが確立されると，そのコネクションに属する以降のパケットは同一のバックボーンを利用する．

復路のトラヒック分散は，NAT を用いることにより行う．この様子を，図 2 において H1 が H2 と通信をする場合を例にとり説明する．

まず，ルータが往路で左図の実線矢印で示すように B1 を選択した場合を考える．この場合，ルータ R ではアドレス変換されずに H2 にそのまま届き，復路では H2 は H1 宛にパケットを送り返す．ここで，H1 は B1 から割り当てられたアドレスを用い

ているため，このパケットは右図の実線矢印で示すように往路と同じB1を経由してH1に届く．

一方，ルータが往路でB2を選択した場合を考えると，ルータはNATを用いて通信元アドレスをH1からR2(B2から割り当てられたアドレス)に変換するため，復路ではH2はR2宛にパケットを送り返す．ここで，R2はB2から割り当てられたアドレスであるため，このパケットは右図の破線矢印で示すように往路と同じB2を経由してルータRに届き，発信先アドレスがR2からH1に変換されて最終的にH1に届く．

以上のように，NATを用いることにより往路と復路は同一のバックボーンを経由することになるため，往路でトラヒック分散を行うと復路でも自動的にトラヒック分散が行われることになる．

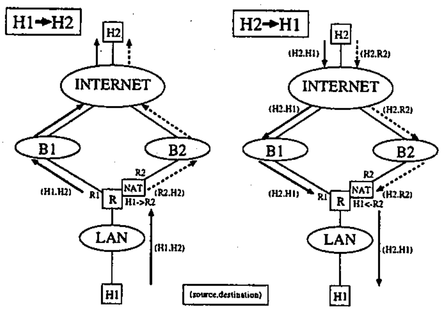

図 2: 往路および復路でのパケットの流れ

2.3 TCP通信におけるバックボーン選択

バックボーンの選択基準としては種々のものが考えられるが，我々はこのうちコネクション確立時間を採用した．すなわち，ルータは内部ネットワークからSYNフラグ付きパケット(SYNパケット)を受け取るとこれを全てのバックボーンに対して同時に送出して通信先とのコネクションの確立を試み，このうち最も早くACKフラグ付きパケット(ACKパケット)を受け取ったバックボーンを選択してSYNフラグ及びACKフラグ付きパケット(SYN+ACKパケット)を送出し，それ以外のバックボーンにはACKパケットを受け取ると直ちにRSTフラグ付きパケット(RSTパケット)を送出してコネクションを破棄する．これにより，負荷が小さいと思われるバックボーンを比較的小さなオーバヘッドで求めることができる．また，ICMP echoパケットによるRTT(Round Trip Time)の測定と比較すると，ICMP echoパケットはフィルタリングにより応答が返されない場合があるのに対して，コネクション確立時間は通信が可能であれば必ず測定できるため，通信先に依存せず確実に測定できるという特徴を有する．

3 UDP通信の透過的トラヒック分散方法

前節で述べた透過的トラヒック分散方法は原理的にはUDP通信にも適用可能である．すなわち，コネクション単位ではなく，フロー単位でバックボーンを選択すればよい．しかし，UDP通信にはコネクションの概念が存在しないため，バックボーン選択基準は前節のものをそのまま利用することはできない．そこで，以下ではUDP通信におけるバックボーン選択基準について検討する．

3.1 UDP通信におけるバックボーン選択

一般にバックボーン選択基準は以下の条件を満足することが望ましい．

条件1: 過去の状態ではなく通信時における通信先までのネットワークの状態を反映するものであること．

条件2: バックボーンの状態の評価を短時間かつ低コストで行えるものであること．

条件3: 通信先に関わらず適用できるものであること．

通信路のネットワーク特性を測定する方法として，pathchar[8]，NEPRI[9]などが知られている．しかし，これらは多くのICPMパケットやUDPパケットを送出してネットワーク特性を測定するため，これらの測定値をそのままバックボーン選択基準としても条件2を満たさない．また，ラウンドロビンや各バックボーンを流れるフロー数など，ルータの内部情報に基づく選択基準は条件1を満たさない．更に，ICMP echoパケットによるRTTの測定は前節で述べたようにフィルタリングにより応答が返されずタイムアウトするまでバックボーンを選択できない危険性があるため，条件2あるいは条件3を満たさない．

そこで，本稿では，(a)殆どの通信ではフロー確立前に通信先のIPアドレスを得るためにDNSサーバに問合せを行う，(b)一般に通信先IPアドレスを管理するDNSサーバは通信先と同一ネットワークあるいは近い位置にあるネットワークに属している場合が多い，という性質に着目し，UDP通信に対

するバックボーン選択基準として，DNS サーバに対する応答速度を用いる手法を提案する．すなわち，新しいフローに属する UDP パケットを内部ネットワークから受け取ると，ルータは全てのバックボーンに対して送信先 IP アドレスを管理する DNS サーバに問合せパケットを送出し，このうち最も早く応答が返されたバックボーンを選択する．

この手法では，(b) の性質により，通信先の応答速度の代わりに DNS サーバの応答速度を利用することができ，また (a) の性質によりルータは事前に通信先に対応する問合せ先 DNS サーバを見つけることができるため，殆どの UDP 通信に対して適用できる．この手法ではバックボーン選択基準が満たすべき条件のうち条件 3 を除く全ての条件を満たし，また DNS サーバへの問合せを行わない一部の UDP 通信については，直ちにデフォルトバックボーンを選択できるため，ICMP echo パケットによる RTT の測定のような問題は生じない．

4 トラヒック分散機能の設計

前節で述べたバックボーン選択基準に基づき，我々はトラヒック分散機能の設計を行った．以下では，まずトラヒック分散プログラムの構成と動作の概略について述べる．

4.1 プログラムの構成と動作の概略

トラヒック分散機能を実現するため，ルーティングを担当するプログラム (以下ルーティングプログラム) は，DNS サーバの IP アドレスを管理する DNS サーバ管理表及び現在処理中のフローを管理するフロー管理表を持たせた．このうち，DNS サーバ管理表は (ホスト名 N，ホスト名に対する IP アドレス A，DNS サーバの IP アドレス D) の 3 つ組を記録しておくものであり，またフロー管理表は (通信元 IP アドレス A_s，同ポート番号 P_s，通信先 IP アドレス A_d，同ポート番号 P_d，アドレス変換後の通信元 IP アドレス A_a，同ポート番号 P_a，プロトコル P，選択したバックボーン B) の 8 つ組を記録しておくものである．

ルーティングプログラムは内部ネットワーク用及び各バックボーン用のネットワークインタフェースで受信するパケットを監視し，以下のように動作する．

- 内部ネットワーク・外部ネットワーク間の DNS 問合せ・応答パケットについては無条件に全てデフォルトバックボーンを経由するように中継する．このうち，外部ネットワークから内部ネットワークへ返される DNS 応答パケットはその中に含まれるホスト名とこれに対応する IP アドレス並びに応答パケットの送信元である DNS サーバの IP アドレスを取り出して DNS サーバ管理表に登録する．
- DNS 問合せ・応答パケット以外の内部ネットワークからの UDP パケットについては，まずこのパケットに対応するフローがフロー管理表に登録されているかどうかを調べ，登録されている場合には以前に選択したバックボーンに (必要であればアドレス変換した上で) このパケットを中継する．登録されていない場合にはこれを新しいフローに属するものとして扱い，DNS サーバ管理表に送信先 IP アドレス A_d を含むレコード (N, A_d, D) が存在するかどうかを調べる．もしこのようなレコードが存在すれば，DNS サーバ D にホスト名 N を問い合わせるパケットを生成して全てのバックボーンに送出し，このうち最も早く応答パケットを返したバックボーンを選択してフロー管理表に登録する．もし DNS サーバ管理表に該当するレコードが存在しなければ，デフォルトバックボーンを選択してフロー管理表に登録する．
- DNS 問合せ・応答パケット以外の外部ネットワークからの UDP パケットについては，まずこのパケットに対応するフローがフロー管理表に登録されているかどうかを調べ，登録されている場合には必要であればアドレス変換した上で内部ネットワークに中継する．もし該当するフローがフロー管理表に登録されていなければ，デフォルトバックボーンからのパケットである場合に限りこれを新しいフローに属するものとして扱い，フロー管理表に登録する．フロー管理表に登録されていないデフォルトバックボーン以外からの UDP パケットは全て破棄する．

4.2 典型的な動作手順

次に，図 3 のような構成のネットワークを例にとり，ホスト H_1 がホスト host.domain.jp(H_2) と UDP 通信を行う場合の典型的な動作手順を示す．なお，以下の説明では例えばホスト H_2 の IP アドレスも単に H_2 と記すことにする．

(1) ホスト H_x は内部ネットワーク内の DNS サーバ D_x にホスト名 host.domain.jp に対する IP

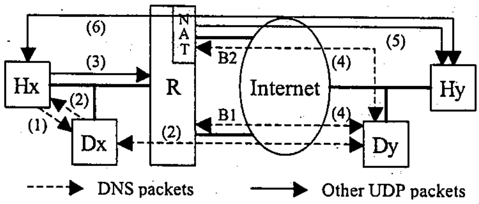

図 3: 典型的な動作手順

アドレス H_y を問い合わせる．DNS サーバ D_x は問合せパケットを最終的に DNS サーバ D_y に送る．

(2) DNS サーバ D_y はホスト名 host.domain.jp を解決し，その IP アドレス H_y を含む応答パケットをホスト H_x に送出する．ルータはこのパケットを解析し，3 つ組 (host.domain.jp, H_y, D_y) を DNS サーバ管理表に登録した後，これを DNS サーバ D_x に中継する．最終的にホスト H_x は host.domain.jp に対する IP アドレス H_y を得る．

(3) ホスト H_x はホスト H_y 宛に UDP パケット (送信元 IP アドレス H_x, 同ポート番号 P_x, 送信先 IP アドレス H_y, 同ポート番号 P_y, プロトコル UDP) を送出する．ルータはこのパケットを受け取るとフロー管理表にこのパケットに対応するフローが登録されているかどうかを検索し，該当するフローが登録されていないことを知る．引き続いてルータは IP アドレス H_y を 2 番目の要素として含むレコードが DNS サーバ管理表に存在するかどうかを検索し，該当するレコード (host.domain.jp, H_y, D_y) を得る．

(4) ルータは DNS サーバ D_y に host.domain.jp の IP アドレスを問い合わせるパケットを生成し，全てのバックボーンに送出する．

(5) DNS サーバ D_y はこれらの問合せパケットを受け取り，ルータに送り返す．ルータはこのうち早く応答パケットを返したバックボーン (ここでは $B2$ とする) を選択し，アドレス変換後の IP アドレス A_a, P_a を決定してフロー管理表に 8 つ組 (H_x, P_x, H_y, P_y, A_a, P_a, UDP, $B2$) を登録した後，最初のパケット (H_x, P_x, H_y, P_y, UDP) をアドレス変換したもの (A_a, P_a, H_y, P_y, UDP) をバックボーン $B2$ に送出する．

(6) これ以降，ルータがこのフローに属するパケットを受け取ると，アドレス変換を行いながら内部ネットワーク・バックボーン $B2$ 間で中継する．

なお，上記の動作手順では，従来の経路情報に基づく経路制御が全く行われていないことに注意する．

5 トラヒック分散機能の実装と性能評価

5.1 トラヒック分散機能の実装

前節で述べた方法の有効性を検証するため，UDP 通信を対象として動的トラヒック分散機能を持つルータを試作し，その評価を行った．本章では試作ルータの実装方法と評価結果について述べる．

5.2 実装方法

試作ルータは，OS として FreeBSD 4.1R を搭載した AT 互換機 (Gateway 2000 社 ESSENTIAL) に 3 枚のネットワークインタフェースを装着したものを用いた．本来 FreeBSD では経路制御はカーネルで行われるが，試作ルータでは実装を容易にするため，カーネルを一切変更せずユーザプロセスで全ての経路制御処理を行った．このため，各バックボーンから到着するパケットは全て divert 機能を用いてカーネルが中継する前にユーザプロセスに渡されるようにした．また，逆にバックボーンへのパケットの送出は，ソケットインタフェースを用いた通常の方法で送出するとカーネルで本来の経路制御が行われるため，bpf(Berkeley Packet Filter) を用いてフレームを直接ネットワークインタフェースに書き出すようにしている．

5.3 性能評価

性能評価は，図 4 に示すようにクライアントとサーバの間に試作ルータを配置した環境で行った．この環境では，サーバと試作ルータの間はバックボーンに見立てた 2 つのネットワークで接続され，通信速度は両者とも 10Mbps である．また，クライアント・ルータ間の通信速度は 100Mbps である．なお，クライアント，サーバはともに DNS サーバとしても動作するように設定されている．

この環境において，クライアント及びサーバで評価用プログラムを動作させ，全体のスループット及びバックボーン選択率を 10 回測定してその平均を

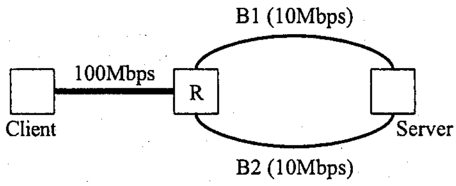

図 4: 実験環境

求めた．クライアント側のプログラムは 10 本のスレッドを同時に起動するもので，各スレッドはサーバにデータ転送を要求し，サーバからのデータ転送を受信してスループットを算出する動作を行う．また，サーバ側のプログラムはクライアントからデータ転送を要求されるとその発信者アドレスを記録し，一定帯域 (本実験では約 1.1Mbps) でデータ送出を行うものである．なお，比較の対象として，トラヒック分散を行わない場合についても同様の実験を行い，平均スループットを求めた．

実験の結果得られたスループットを表 1 に，またバックボーン選択率を表 2 に示す．これらの図からわかるように，トラヒック分散が行われていない場合には 1 つのバックボーンしか利用できないため，スループットがバックボーン B1 の帯域である 10Mbps を超えることができないが，トラヒック分散を行うと 2 つのバックボーンをほぼ均等に利用し，その結果 10Mbps を超えるスループットが得られることがわかる．

表 1: 全体のスループット

(単位: Mbps)	平均	最大	最小
トラヒック分散なし	9.1	9.6	8.7
トラヒック分散あり	10.6	10.9	9.8

表 2: バックボーン選択率

(単位: %)	B1	B2
トラヒック分散なし	100	0
トラヒック分散あり	56	44

6 まとめ

本稿では，マルチホームネットワークにおいて UDP 通信を対象として透過的トラフィック分散を行う場合にバックボーン選択基準が問題となることを示し，実際に効果的にトラヒック分散を行うためのバックボーン選択基準として，DNS サーバの応答時間を用いる方法を提案した．また，提案方法を取り入れたルータを試作し，実験によりその有効性を確認した．この方法により，従来の TCP 通信だけでなく UDP 通信についても効率的なトラヒック分散が容易に実現可能となり，マルチホームネットワークの普及に一層貢献できると思われる．今後の課題としては，実際のインターネットにおける本方法の性能評価が挙げられる．

参考文献

[1] 岡山聖彦，山井成良，島本裕志，宮下卓也，岡本卓爾：“マルチホームネットワークにおける透過的な動的トラヒック分散”，情報処理学会論文誌，Vol.41, No.12, pp.3255–3264, 2001.

[2] Hawkinson, J. and Bates, T.: “Guidelines for creation, selection, and registration of an Autonomous System (AS)”, RFC1930, 1996.

[3] Egevang, K. and Francis, P.: “The IP Network Address Translator(NAT)”, RFC1631, 1994.

[4] Srisuresh, P. and Holdrege, M.: “The IP Network Address Translator(NAT) Terminology and Considerations”, RFC2663, 1999.

[5] 梶田将司，結縁祥治: “NAT によるプライベートネットワークの準マルチホーム化技法”，情報処理学会分散システム／インターネット運用技術研究報告，No.16, pp.73–78, 1999.

[6] 中川郁夫，上谷一，鍋島公章，樋地正浩，今野幸典: “マルチホーム環境におけるアプリケーションルーティング技術の提案”，情報処理学会分散システム運用技術研究報告，No.12, pp.37–42, 1998.

[7] Rekhter, Y. and Li, T.: “A Border Gateway Protocol 4”, RFC1771, 1995.

[8] Jacobson, V.: “pathchar — A Tool to Infer Characteristics of Internet Paths”, MSRI, `ftp://ftp.ee.lbl.gov/pathchar/msri-talk.pdf`, 1997.

[9] 青木武司，菊池慎司，高橋英一，岡野哲也，安達基光，勝山恒男: “IP ネットワークの性能測定技術”，電子情報通信学会技術研究報告，IN98-90, pp.9–16，1998.