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

島本裕志[1]　山井成良[2]　宮下卓也[2]　岡本卓爾[3]

[1]岡山大学 大学院自然科学研究科
[2]岡山大学 総合情報処理センター
[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．しかし，従来のマルチホームネットワークの構成法では，導入や管理にかなりの技術とコストが要求される，特定のバックボーンにトラヒックが集中する可能性があるなどの問題点がある．本稿では，この問題点を解決するため，ルータが自らがバックボーンネットワークの状態を監視し，現在の状態に基づいて利用するバックボーンを選択する方法を提案する．本手法ではアプリケーションゲートウェイとは異なり透過的にトラヒック分散を行えるため，クライアントプログラムに特別の機能や設定を必要としない．

A Transparent Dynamic Traffic Balancing on Multihomed Networks

Hiroshi Shimamoto[1], Nariyoshi Yamai[2], Takuya Miyashita[2], and Takuji Okamoto[3]

[1]Graduate School of Natural Science and Technology, Okayama University
[2]Computer Center, 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. However, multihomed network is hard to introduce or manage because the existing construction methods have several problems such that it requires much technical skill and administrative cost for the administrator, traffic congestion may occur on a backbone while others have little traffic, and so on. In this paper, we propose a dynamic traffic balancing technique to solve these problems. Using our technique, the router connecting the inside network and backbones monitors the condition of each backbone and selects the appropriate backbone according to the current condition. Moreover, our technique balances traffic transparently and does not require additional functions or configuration to client programs.

1　はじめに

近年，インターネット利用の急激な増加により，WWW，FTP などの広域ネットワークサービスにおける応答時間の悪化が深刻な問題となってきている．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．

そこで，本稿では，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する．これにより，通信先に偏りが生じた場合でもコネクション毎に個別のバックボーンを用いて効率的にトラヒックを分散することが可能になる．また，本方法では必ずしもバックボーンとの間で経路情報を交換する必要がないため，導入や管理が容易である．

同様の方法として，WWW など一部のアプリケーションではプロクシなどのアプリケーションゲートウェイ (Application Level Gateway, ALG) によりトラヒック分散を行う方法が提案されている [1]．しかし，この方法では，ユーザが ALG に対応したクライアントを用い，かつ ALG を経由してサーバにアクセスするようにクライアントを設定する必要があり，ALG の存在をユーザが意識する必要がある点が問題となる．本稿で提案する方法では，ルータの機能を変更するだけで，クライアントの変更を必要とせず，透過的に動的トラヒック分散を行うことができる点で ALG を用いる方法より優れている．

2 従来のマルチホーム化技法

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 番号を取得する方法 (方法 1)，NAT(ネットワークアドレス変換 [2, 3]) を用いる方法 (方法 2)，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) などが挙げられる．

以下では，それぞれの方式について説明し，その問題点を挙げる．

2.1 AS 番号取得によるマルチホーム化

現在，インターネットではネットワーク全体を AS (Autonomous System)[4] と呼ばれる部分ネットワークの集合として扱い，AS 間で BGP4[5] を用いて経路情報の交換を行う方法が一般的である．方法 1 は自組織のネットワークに対する AS 番号を取得し，各バックボーンとの間で経路情報を交換して経路制御を行う方法である．これによりネットワークトポロジ，障害の有無，経路制御のポリシなどに応じて適切なバックボーンを選択することが可能になる．

ところが，この方法では，BGP4 の運用に関して次のような問題がある．

1. 中小規模のネットワークについては，AS 番号の取得が困難である．
2. BGP4 の運用を行うには，経路制御技術についての詳しい知識が必要である．
3. 経路情報を交換するため，それぞれの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る．

このような問題から，特に中小規模のネットワークではこの方法でマルチホーム化することは困難である．

また，別の問題として，特定バックボーンへトラヒックが集中する可能性がある点が挙げられる．

通常，ルータは，バックボーンなどから入手した経路情報に IP パケットの通信先を照らし合わせることにより経路制御を行う．この経路情報には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，通信先アドレスが同じパケットに対して必ず同じバックボーンが選択される．従って，通信先に偏りがあると，特定バックボーンにトラヒックが集中する可能性がある．

この問題については，いわゆる首振りルータ [6] を用いることによりある程度解決することができる．しかし，インターネットでは一般に復路と往路とで経路制御は独立して行われるため，たとえ首振りルータを用いて往路でトラヒックを分散したとしても，復路では 1 つのバックボーンにトラヒックが集中する可能性がある．

2.2 NAT によるマルチホーム化

方法 2 は，各バックボーンから個別のアドレスの割当を受け，外部との通信の際に NAT を用いて内部アドレスをバックボーンから割り当てられたアドレスに変換することによりマルチホームネットワークを実現する方法である [7]．この方法では，バックボーンから割り当てられたアドレスをそのまま使うため，自組織ネットワークの経路情報のアナウンスが不要で，また往路と復路で同一のバックボーンを利用するため復路でもトラヒック分散を行えるという特徴を持つ．

しかし，この方法でもバックボーンの選択のためには private BGP を用いて，バックボーンから経路情報を取得する必要があり，そのためにかなりの技術レベルと管理コストが必要となる．また，方法 1 と同様に，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．

2.3 ALG によるマルチホーム化

方法 3 は，WWW などの一部のアプリケーションにおいて各バックボーンに属するアドレスを持つ ALG をそれぞれ導入し，これらの ALG を経由して外部にアクセスすることでマルチホームネットワークを実現する方法である．この場合，ALG に経路制御機能を持たせることによりトラヒック分散を行うことができる [1]．

この方法では，方法 2 と同様に自組織ネットワークの経路情報のアナウンスが不要で，また往路と復路で同一のバックボーンを利用するため復路でもト

ラヒック分散を行えるという特徴を持つ．しかし，この方法を利用できるアプリケーションは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．

3 動的トラヒック分散方法

2章で述べたように，従来のマルチホーム化技法によるトラヒック分散は，いずれの方法も導入・管理に要する技術レベルや管理コストが高い，特定のバックボーンにトラヒックが集中する可能性がある，ユーザが透過的に通信を行えないなどの問題がある．これらの問題点は，いずれもバックボーンから入手した経路情報に基づいてバックボーンを選択する従来の経路制御方法に根本的な原因がある．

そこで，本章ではNATによるマルチホーム化技法をもとに，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する．

3.1 提案方法の概要

本稿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[1]．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，B 2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．

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．なお，外部から内部へのTCPコネクションの分散については，DNSラウンドロビンなど従来の方法で対処することができる．以下では，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．

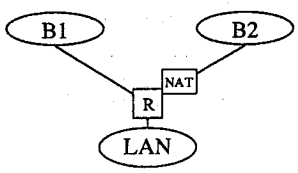

図 1: マルチホームネットワーク

3.1.1 往路のトラヒック分散

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．

ルータにおいて観測できるバックボーンの選択基準として，例えば以下のようなものが考えられる．

- 現在のコネクション数
- コネクション確立に要する時間
- 最近の通信のエラー率
- 現在の各バックボーンの利用率

これらの選択基準はネットワークの利用状況に応じて変化するため，たとえ通信先が偏ってもトラヒック分散が可能となる．また，これらの情報の収集は，BGP4での経路情報の取得と異なりバックボーン管理者との協調作業を必要としないため，容易に行える．

なお，バックボーンの選択基準は例えばアプリケーションや通信先／通信元アドレスに応じて変更することができ，これによりポリシを考慮した経路制御も可能である．

3.1.2 復路のトラヒック分散

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．

3.2 バックボーンの選択

前節で述べたように，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，試作においてはこのうちコネクション確立時間を採用した．これはコネクション確立時に全てのバックボーンを用いて通信先とのコネクションの確立を試み，このうち最も早く確立できたバックボーンを選択するもので，次のような特徴を持つ．

[1] 3つ以上のバックボーンに接続することも可能であるが，説明の都合上省略する．

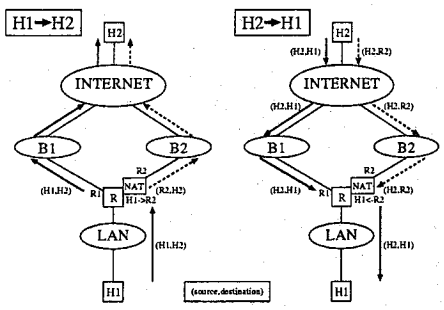

図 2: 往路および復路でのパケットの流れ

- コネクション確立時間は通信先との距離やネットワークの状態の影響を受けやすく，トラヒック分散を動的に行うための指標として適している．
- コネクション確立時間はコネクション確立時に測定すれば良いため，最近の通信のエラー率やバックボーンの利用率など継続的に情報を収集する必要のある他の基準より測定が容易である．
- ping による RTT(Round Trip Time) の測定と比較すると，ping 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，確実である．
- いくつかのバックボーンに障害が発生している状況でも，コネクション確立時点で利用可能なバックボーンの中から 1 つを選択するため，耐故障性がある．

以下では，実際にルータで行われるバックボーン選択処理について述べる．

よく知られているように，TCP コネクションの確立には 3-ウェイハンドシェイクが用いられる．この 3-ウェイハンドシェイクでは，H1 から H2 に対してコネクションを確立しようとする場合，実際に確立するまでに以下の 3 つのパケットの送受を必要とする．

1. H1 は H2 に対して SYN フラグ付きのパケット (SYN パケット) を送る．
2. H2 はこのパケットを受け取ると H1 に対して SYN フラグと ACK フラグの両方が立ったパケット (SYN+ACK パケット) を送る．
3. H1 はこのパケットを受け取ると H2 に対して ACK フラグ付きのパケット (ACK パケット) を送る．

このとき最後の ACK パケットの代わりに H1 が RST フラグ付きのパケット (RST パケット) を H2 に送ると，このコネクションは確立されず直ちに破棄される．この性質を利用してルータは最も早く SYN+ACK パケットを送り返したコネクションだけを選択する．すなわち，ルータは以下のように動作する．

1. ルータは内部から外部への SYN パケットを受け取ると，このパケットを複製して全てのバックボーンに送出する．このとき，アドレス変換が必要なバックボーンについては，送信元のアドレスをバックボーンのアドレスに変換する．
2. ルータはあるバックボーン B から上記の SYN パケットに対する最初の SYN+ACK パケットを受け取ると，そのパケットを (必要であればアドレス変換を行った上で) 本来の送信先に中継する．また，今後このコネクションに対してバックボーン B を利用するように記録する．
3. ルータは他のバックボーンから 2 番目以降の SYN+ACK パケットを受け取ると，そのパケットを中継せずに破棄し，代わりにそのバックボーンを用いて RST パケットを送出する．

例として，図 2と同様のネットワーク構成において，B1 が選択される場合のパケットの流れを図 3 に示す．

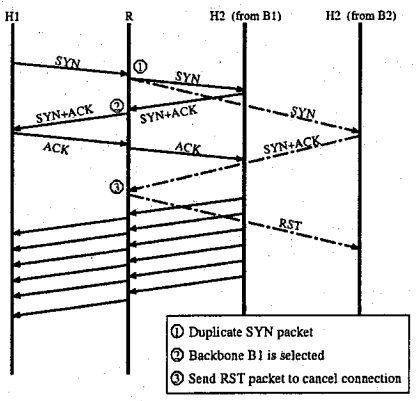

図 3: バックボーン選択時のパケットの流れ

3.3 コネクションの管理

通常，NAT を経由して内部ネットワークから外部ネットワークへアクセスする場合，NAT の内部ではどのようにアドレス変換を行えばよいかをコネクション単位で管理している．本方式においても NAT の利用を前提としているためコネクション管理が必要であるが，NAT を経由しないコネクションについてもどのバックボーンを選択したかを記録するため同様のコネクション管理を行っている．

ルータには現在確立されているコネクションを管理するための表(コネクション表)が設けられており，通信元の内部アドレス及びポート番号，アドレス変換後のアドレス及びポート番号，通信相手のアドレス及びポート番号，選択したバックボーンへのインタフェースなどが記録されている．ルータはこのコネクション表を利用して以下のように動作する．

1. 内部から外部へのSYNパケットを受け取ると，前節で述べた方法でバックボーンの選択を行い，このコネクションに関する新しいエントリを選択されたバックボーンとともにコネクション表に追加する．

2. RSTパケットやFINパケットを受け取るなどしてコネクションが解放される時には，当該コネクションに関するエントリをコネクションから削除する．

3. それ以外のパケットを受け取った時には，このパケットがどのコネクションに属するかをコネクション表より求め，アドレス変換を行った上で，内部から外部へのパケットの場合にはコネクション表に登録されているバックボーンへ，外部から内部の場合には内部のネットワークへ中継する．

なお，上記の動作では，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．

4 動的トラヒック分散機能の実装と性能評価

前章で述べた方法の有効性を検証するため，動的トラヒック分散機能を持つルータを試作し，その性能評価を行った．本章では試作ルータの実装方法と性能評価について述べる．

4.1 実装方法

試作ルータは，OSとしてFreeBSD 2.2.7Rを搭載したAT互換機(Gateway 2000社GP6-450)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た．

4.2 性能評価

4.2.1 実験環境

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2種類のネットワークで接続され，一方は10Mbpsあるいは100Mbpsの2種類の速度を切り替えて使うことができる専用LAN(B1)で，一方は他の計算機と共用される10MbpsのLAN(B2)となっている．この環境において，サーバ上でHTTPサーバプログラムを動作させ，クライアントからサーバに同時に複数のデータ転送要求を発生させ，B1のみを用いて10Mbpsで通信を行う場合(ケース1)，B1を10Mbpsに設定してB1とB2でトラヒック分散を行う場合(ケース2)，B1を100Mbpsに設定してB1とB2でトラヒック分散を行う場合(ケース3)の3種類の場合についてコネクション単位の平均スループットを測定した．また，ケース2及び3においては，B1, B2それぞれを経由するコネクション数の時間変移を求めた．

なお，HTTPのデータ転送要求の発生には，webjamma[8]を用いた．このプログラムは親プロセスが多数の子プロセス(この実験では30)を生成した後にURLをファイルから次々に読み込んで子プロセスに渡し，子プロセスは渡されたURLへのアクセスを完了した後に再び親プロセスから次のURLを取得するような動作を行う．従って，データ転送速度が十分早い場合には，親プロセスから子プロセスへのURLの供給が間に合わず，必ずしも30プロセスが同時にサーバにアクセスするとは限らない点に注意する．

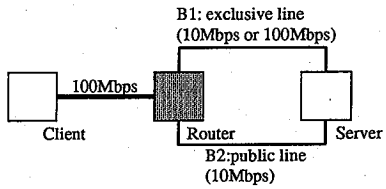

図 4: 実験環境

4.2.2 実験結果と考察

まず，各ケースにおけるコネクション単位の平均スループットを表1に示す．この表からわかるように，トラヒック分散を行った場合(ケース2，3)は行わなかった場合に比べて大幅に平均スループットが改善されており，本方法によるトラヒック分散が有効に機能していることがわかる．ケース1の平均スループットがケース2に比べて著しく悪化しているのは，ケース1では帯域が狭いためコネクションが長時間解放されず，結果として30プロセスが同

表 1: コネクション単位の平均スループット

Case	Throughputs
Case 1	32 (kbps)
Case 2	132
Case 3	668

時にサーバにアクセスするのに対して，ケース 2 では B1，B2 の合計の帯域が広く，10〜20 程度のプロセスしか同時にサーバにアクセスしないためであると思われる．

次にケース 2，3 におけるコネクション数の時間変化をそれぞれ図 5，6に示す．

図 5を見ると，B1 と B2 がほぼ均衡して使われていることがわかる．B1 を利用するコネクション数の割合が少し高いのは，B1 が専用 LAN であるのに対して B2 は他の計算機と共用されるためであると思われる．また，150 秒前後では B2 の方がよく利用されているが，これはルータが SYN パケットを複製して中継する時に実際には同時ではなく先に B2 に送出するための影響であると思われる．

一方，図 6を見ると，殆んどのコネクションが高速な B1 を利用しており，B2 を利用するコネクションがほとんどないことがわかる．

以上の結果から，提案方法は図 4の実験環境ではネットワークの利用状況や帯域に応じて適切にバックボーンを選択し，動的にトラヒックを分散できると言える．

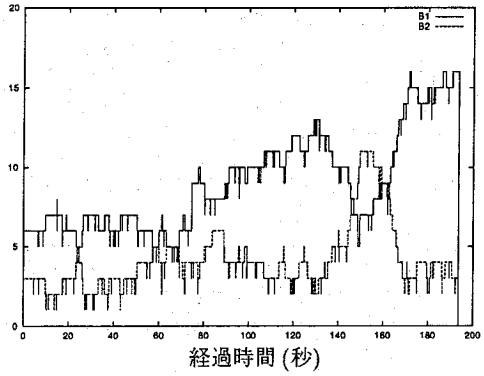

図 5: ケース 2 におけるコネクション数の時間変移

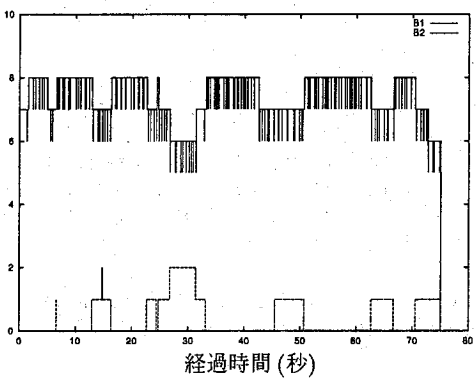

図 6: ケース 3 におけるコネクション数の時間変移

5 おわりに

本稿では，マルチホームネットワークにおいて，ルータが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．また，バックボーンの選択基準としてコネクション確立時間を用いたルータを試作し，実験によりその有効性を確認した．この方法により，マルチホームネットワークにおける効率的なトラヒック分散が高い技術レベルや運用コストを必要とせずに可能となり，マルチホームネットワークの普及に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や，バックボーンのより効果的な選択方法の確立などが挙げられる．

謝辞 本研究の一部は，文部省科学研究費補助金奨励研究 (A) による補助を受けている．ここに記して感謝の意を表する．

参考文献

[1] 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.

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

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

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

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

[6] 小巻賢二郎，所真理雄: “フローを考慮した経路制御機構”，電子情報通信学会技術研究報告，IN98-27, pp.25–32, 1998.

[7] 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.

[8] R. P.Wooster and M. Abrams, “Proxy Caching that Estimates Page Load Delays”, Proc. 6th World Wide Web Conference, pp.325–334, 1997.