卒業論文 1999年度 (平成11年度)

柔軟性のあるパケット転送機構の設計と実装

指導教員

村井 純
徳田 英幸
楠本 博之
中村 修

慶應義塾大学 環境情報学部
安田 歩

平成12年2月28日

卒業論文要旨

柔軟性のあるパケット転送機構の設計と実装

インターネットの発展に伴って利用者や規模が増大し、ネットワークを構成する通信路も多様化している。また、インターネット上で利用されるアプリケーションも、広帯域や実時間性など、ネットワークに対して様々な伝送品質を要求するようになった。このように、現在のインターネット上では、利用者の要求やネットワークの状態に見合ったパケット転送機構の実現に対する要望は非常に高い。しかし、インターネットで用いられている既存の経路制御プロトコルに基づいた経路制御機構では、こうしたアプリケーションからの要求を満たしたパケット転送は行なえない。

本研究では、これらの問題点を解決するための新しいパケット転送機構を提案した。パケットを転送するポリシーと、ポリシー間の優先順位を定義するモデルを提案した。提案したモデルに従って、拡張性を持ち、自由度の高いパケット転送機能を実現するソフトウェアをルータ上に設計し、実装した。

本研究で提案したモデルに基づいて、設計、実装した手法を、他の研究と比較した。この結果、本研究で提案する手法が、利用者の要求とネットワークの状態を考慮した、柔軟性のあるパケット転送機構を実現するのに有効であることを示した。

Abstract of Bachelor's Thesis

The Design and Implementation of Flexible Packet Forwarding System

As the Internet grows, various kinds of datalinks came up. Applications became more demanding in terms of bandwidth and real-time communications. It is necessary to realize a packet forwarding system that can apply users' demand and dynamically changing network status. Ongoing Internet system has necessity of packet forwarding structure which utilizes network. However, the existing routing system including routing protocols can not satisfy these demand.

This paper proposes a design and implementation of a packet forwarding system that can solve the problem above.

And this paper proposes the model that have policy of packet forwarding and definition of priority between the policies.

The router model proposed in this paper was designed and implemented, and high flexibility in forwarding IP packet was made possible by this router. Throughout the evaluation based on the experiments, the system can forward IP packet with high flexibility which considers user requirement and network status.

目 次

図目次

表 目 次

第1章　序論

1.1　研究背景

現在、インターネットを利用する利用者は急増し、広帯域のネットワークを必要とするアプリケーションも急速に普及している。このため、実際にネットワークを流れるトラフィックは、爆発的に増加している。それに伴い、インターネットを構成するネットワークは拡張されてきている。インターネット上で利用されている通信路は、種類が多様化してきており、実際にネットワークを構成している回線には、銅線、光ファイバー、無線、衛星など、異なる媒体が混在している。

一方、アプリケーションの視点からネットワークを見ると、新しいサービスの普及が目覚しい。今までは、WWW などの、一般的に実時間性を強く要求しないアプリケーションが、インターネットのトラフィックの多くを占めていた。しかし現在では、real video や IP 電話など、実時間性が要求されるインターネット上でのサービスが急速に普及してきている。また、これらは動画や音声のデータの組み合わせで構成されていることが多く、円滑な通信には広い帯域や、遅延の少ないネットワークを必要とする。

しかし現在、通信や回線の種類に応じてパケットの経路を制御する機構は普及していない。低遅延性、広帯域性、高信頼性など、アプリケーションが要求する特性を考慮した経路制御の仕組みが必要である。

現在のインターネットの経路制御機構では、ネットワーク層で転送の処理を行なう。インターネットにおけるネットワーク層プロトコルである IP[1] の場合、経路制御をする時、全てのデータが IP パケットとして等しく扱われる。

インターネットでは、通信路の種類ごとに各々特徴がある。既存の経路制御機構では、ルータがパケットを転送する際に、回線を指定することができない。このため、ネットワークを構成する様々な通信路が持つ特徴は生かされていない。

このように、現在の経路制御機構では、回線の特徴とアプリケーショ

ンの要求が考慮されない。

この問題を解決するにためには、ネットワークの資源や特性を有効に活用し、様々なアプリケーションの要求を満たして通信路を利用するパケット転送機構の構築が必要である。

1.2　本研究の目的

本研究では、回線の特徴、ユーザの要求などを考慮し、その動的な変化に対応した汎用性の高いパケット転送機構を実現する。そのためのモデルとして、OROCHI を提案する。

既存の経路制御プロトコルでは、パケットの終点アドレスとルータが持つ経路表に基づき、パケットの転送先を決定している。また、現在実装されている、経路制御プロトコルに依存しないパケット転送機構においても、転送先経路の決定として扱う要素は、パケットの始点アドレス、終点アドレス、始点ポート番号、終点ポート番号のみであり、また、これらを自由に取り扱う枠組は提供されていない。従って、既存の機構では、多様な要求を経路制御に反映させることはできない。

本研究では、ネットワーク資源を効率的に活用し、アプリケーションの要求を満たす機構として、首振りルータに着目した。首振りルータとは、既存の経路制御プロトコルに影響されることなく、特定の要素によりパケットの転送を行なうルータのことである。

本研究では、既存の首振りルータで実現されていない機能を実装し、柔軟にパケット転送を制御できる機構を提供する。これにより、パケット転送に様々な条件を付与できるようになる。また、管理者は、各々のポリシーにより、パケットが通る経路を、自由度を持って、決定できる。

1.3　本論文の構成

本論文は、本章を含めて8章から成る。本章では、本研究の背景および目的を述べた。第2章では、既存の経路制御機構の概要と問題点を明白にする。第3章で、本研究で着目した首振りを概説し、現状の問題点について述べる。第4章で問題点の解決手法を示し、解決を実現するモデルである OROCHI を提案する。第5章で OROCHI の設計を行ない、第6章で実装に関する説明をする。第7章では本研究で実装した OROCHI の性能評価を行なう。最後に、第8章で本研究の結論と、今後の課題を

挙げる。

第2章　既存の経路制御機構

本章では、既存の経路制御機構の概要を述べ、その問題点について論じる。

2.1　経路制御機構の現状

2.1.1　ホップバイホップ経路制御

インターネットは分散型ネットワークである。接続されている各ネットワークは、自律的に管理され、ホップバイホップによる経路制御が行なわれている。この方法では、パケットを中継するノードが、次ホップへの経路を決定する。この際、中継ノードであるルータは、パケットの保持している終点アドレスを検索キーとして経路表を検索し、次に送信するべきルータへのインターフェイスを決定する。一般的に経路表の作成には、経路制御プロトコルが用いられる。代表的な経路制御プロトコルとして、RIP[2] や OSPF[3] が挙げられる。

このホップバイホップ経路制御によるパケット転送の仕組みを図2.1に示す。各ルータの経路表において、終点ノードGへの経路を表すエントリをまとめたものを、表2.1に示す。

各ルータは経路表に基づいて、バケツリレー方式でパケットを中継する。ルータAから ルータGに向かうパケットは、矢印のような経路を辿る。

2.1.2　始点経路制御

ホップバイホップの経路制御とは異なり、送信されるパケットに、あらかじめ通過するルータのアドレス情報を与え、経路を指定する方法がある。これは、始点経路制御と呼ばれる。これには、通過するルータを送信者が直接指定できるという利点がある。しかし、通過するルータの数の増加に伴い、パケットヘッダが増大する。さらに、中継ルータも、始

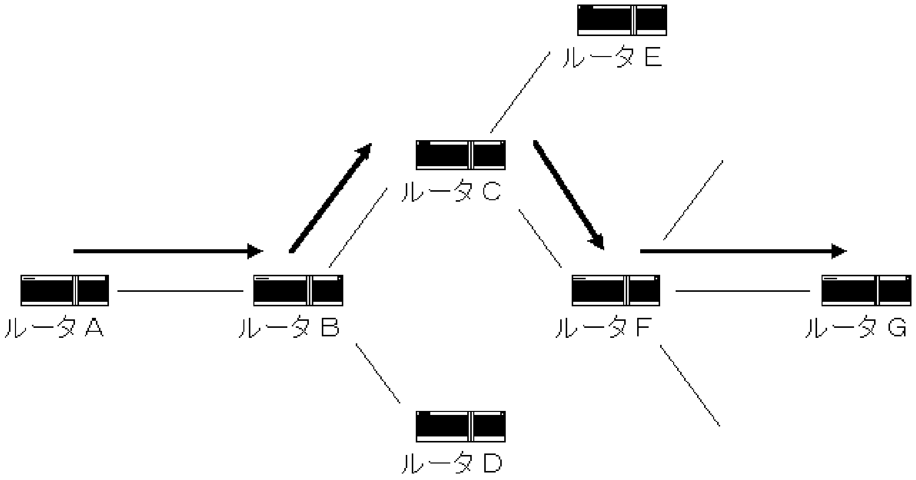

図 2.1: ホップバイホップによる経路制御

点経路制御に対応していなければならない。このため、大規模なネットワークであるインターネットで用いるのは、効率的にも、規模性の面からも、非現実的である。

2.2 既存の経路制御機構の問題点

経路制御プロトコルによって動作する経路制御機構は、ネットワーク全体を考慮したものである。パケットを転送する際、パケットの終点アドレスを経路表で検索する方法が用いられている。パケットが持つ情報

表 2.1: 各ルータの簡略化した経路表

router	destination	next hop
A	G	B
B	G	C
C	G	F
D	G	B
E	G	C
F	G	G

の中で、転送に利用されているものは、終点アドレスのみである。このことにより生じる不具合として、2つのルータ間に複数の経路が存在する時に起きる問題がある。本論文では、このことに着目する。本節の議論は、図2.2に示すような部分的トポロジーを持つネットワークを前提とする。この上で、問題となる状況と、その問題点を以下に挙げる。

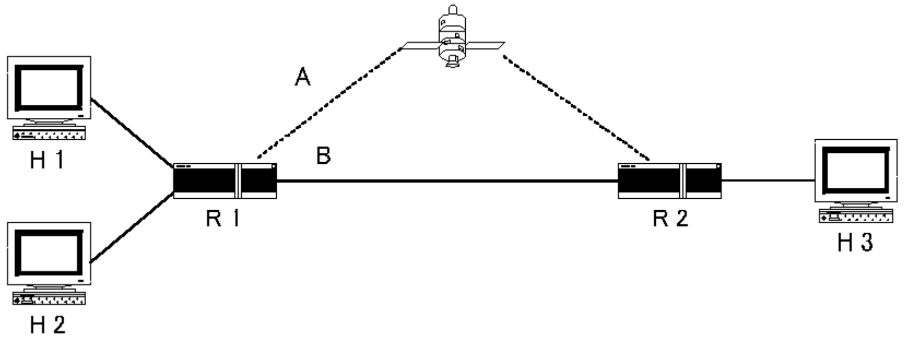

図 2.2: 前提となるネットワーク

2.2.1 始点アドレスによる経路決定

特定のホストや発信者からのパケットのみを、優先的に転送したり、特定の経路に向けたりすることができない。

この問題が生じる例を挙げる。

前提 ホストH1と、ホストH2の2つの送信ノードが存在する。目的ノードは、ホストH3である。

状況 ネットワークを構成するlink A、Bには、回線容量限界のトラフィックが流れている。

問題点 ネットワークを流れるデータに優先順位をつけて、経路制御に反映させることはできない。このため、重要度の低いデータのトラフィックに帯域が占有され、重要な通信が阻害されてしまう。

2.2.2 アプリケーションによる経路決定

ある程度の遅延が許容されるアプリケーションのトラフィックが、低遅延の回線を通ることにより、帯域を圧迫したり、逆に遅延が許されないパケットが高遅延の回線を通ることは、非効率的である。

この問題が生じる例を挙げる。

前提 link A は大きな遅延が発生し、link B は遅延がほとんど発生しない。

状況 この時、ある程度の遅延が許されるサービスと、遅延はなるべく小さくするべきであるサービスの、通信路に対する要求の異なる通信が、図 2.2 のネットワークを利用している。2 つのサービスのトラフィック量の合計は、link A,B を合わせた回線容量の範囲内である。

問題点 既存の経路制御機構では、遅延のようなアプリケーションの特性を認識して経路制御に反映することはできない。

2.2.3 上位層プロトコルによる経路決定

上位層のプロトコルでパケットを識別し、転送先の経路を制御することはできない。

この問題が生じる例を挙げる。

前提 link A は信頼性が高く、link B は相対的に信頼性が低い。

状況 TCP や UDP、RSVP[4] などのトランスポート層のプロトコルのパケットが、一つの回線に混在する。

問題点 TCP のパケットのみ信頼性の高い回線を通したい、ICMP のパケットだけ低遅延の回線を通したいといった要求が発生する。しかし、ネットワーク層でパケットの転送を制御する現在の機構では、トランスポート層の情報は経路制御に利用されない。

2.2.4　ロードバランス

２つの回線の回線使用率が、経路制御に影響を及ぼすことがない。
この問題が生じる例を挙げる。

前提 link B の方が、ネットワークのコストが低い。

状況 link B ではパケットの処理能力を超えるトラフィックが流入し、一方で link A には回線容量に充分余裕がある。

問題点 link A、link B を合わせた帯域では充分にトラフィックを処理できるにも拘らず、実際には処理できない。

第3章　「首振り」機構の現状と問題点

本章では、「首振り」の必要性について論じ、「首振り」を行なう機構についてのべる。また、関連研究とその問題点を明らかにする。

3.1　首振りの必要性

本節では、本研究の前提となる、首振り機構の必要性について論じる。

3.1.1　ネットワークトポロジー

想定するネットワークトポロジーが、図3.1に示すように単純である場合、目的ノードに対する経路は１つしか存在しないため、首振りは不要である。しかし、現実のネットワークには、２つのノード間に複数の経路があるトポロジーが存在する。

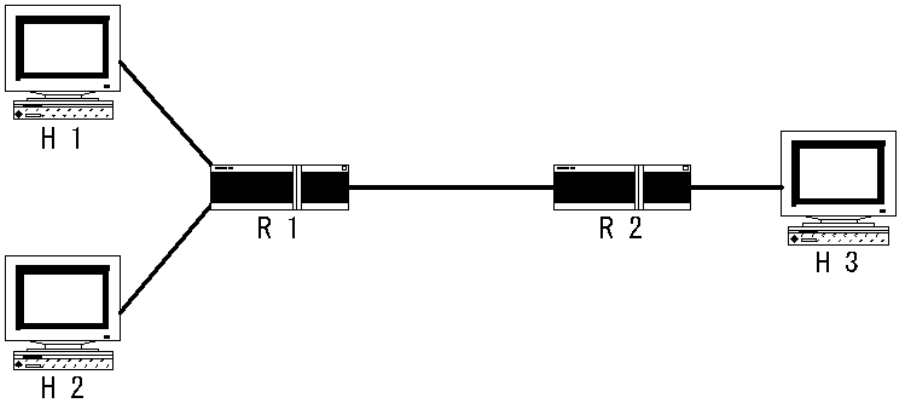

図3.1: ルータ間の回線が１つである単純なネットワークトポロジー

3.1.2 データリンク

ネットワークを構成しているデータリンクが全て同じ条件である場合、目的ノードに対する経路が複数存在していても、首振りは不要である。なぜなら、ネットワークを構成する全ての回線が同一の品質を持っている場合、あらゆる種類のパケットは同じ品質で伝送されるので、パケット転送時の通信路品質の差異を考慮する必要がない。しかし現実では、2つのノード間に、条件が全く同じである経路が存在することは考えにくく、ほとんどのネットワークは、多様な経路をもつトポロジーから成る。

3.1.3 必要性

上述のように、現実には、ネットワークは複雑なトポロジーと多様なデータリンクで、構成されている。ネットワーク上で、経路制御プロトコルに代表されるような単一の枠組でパケット転送を行なった時、アプリケーションが要求する通信品質を満たすことができない。

そこで、様々な要求を許容できるパケット転送の機構として、首振り機構が必要となる。

3.2 首振りの概要

本研究で提案するモデルでは、「首振り機構」の存在が前提となっている。本節では、「首振り」について説明する。「首振り」とは、ある目的ノードに対し、複数の経路へのインターフェイスを持つルータが、特定のポリシーに基づきパケットを送出するインターフェイスを選択し、送信することである。首振りは、一台のルータと、そのルータとの間に2つ以上の経路を持つノードが存在すれば、可能である。首振りを行なうルータを、「首振りルータ」と呼ぶ。

図3.2において、ホストH1から ホストH2に向かってパケットを送信する時を考える。

経路途中のルータR1とR2の間には、link Aとlink Bの2つの経路が存在する。R1では、経路上の次ノードとなるR2へパケットを転送する。この時R1は、route Aとroute Bのいづれか一方の経路を選択する。

既存の経路制御の枠組での経路選択は、以下の手順による。

静的経路制御では、あらかじめ手動で設定されている経路を使用する。

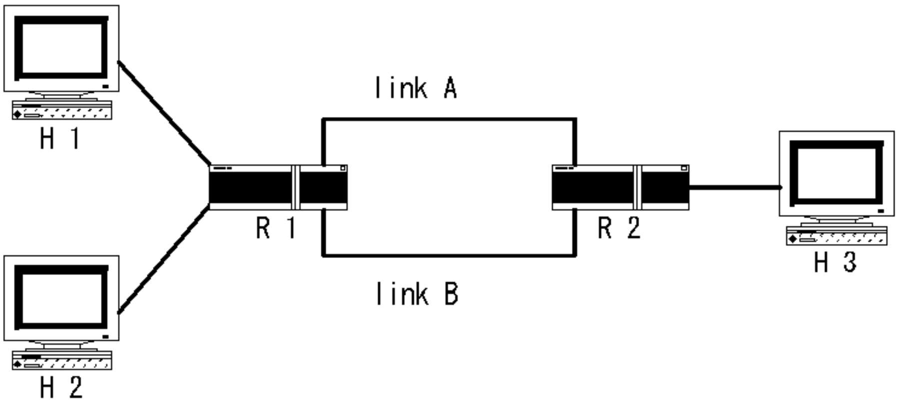

図 3.2: 本研究の前提となるネットワークトポロジー

動的経路制御機構の場合、経路制御プロトコルによって計算される経路のコストに従って、経路の選択を行なう。

一方、R1 において首振りを利用してパケット転送を行なう場合、管理者から与えられたポリシーに基づき、R1 で経路の選択を行なう。

「特定のポリシー」としてポート番号を用いた場合の、首振りの例を以下に挙げる。ルータ R1 において、http ポートの通信には route A を使用し、telnet ポートの通信には route B を使用するというポリシーを設定する。

この時のネットワーク上でのデータの流れを、図 3.3、図 3.4 に示す。

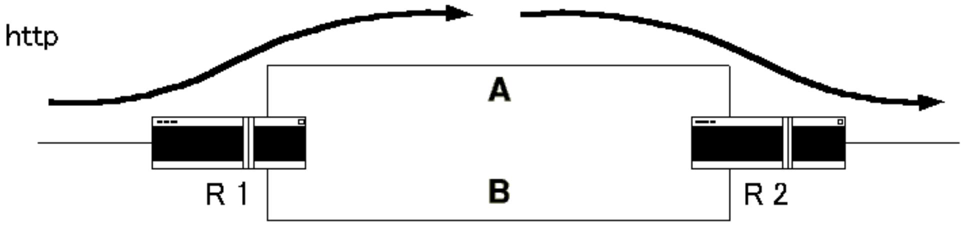

図 3.3: ポート番号による首振りの例 :http

この図で、ルータ R 1 から ルータ R2 に向かって流れているデータについて考える。R1 と R2 の間には、2 つのリンク A、B が存在する。この場合、ルータ R1 は、次ノードである R2 に対して 2 つの経路へのインターフェイスを持っており、このルータ R1 で首振りが行なわれる。

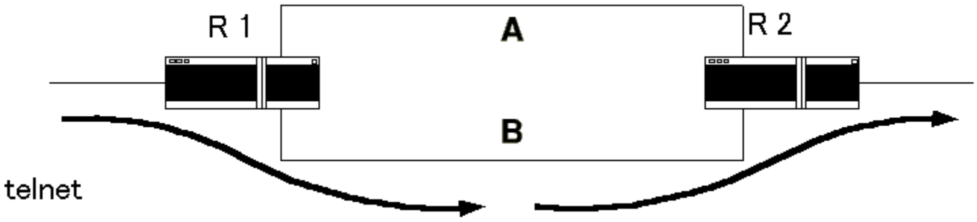

図 3.4: ポート番号による首振りの例 :telnet

首振りの判断基準となる要素は、宛先ポート番号である。ポート番号がhttpであるものはAを、telnetであるものはBを経由してR2に到達する。

首振りシステムを使用すれば、ネットワーク層から見た上位層であるトランスポート層の情報に応じて、使用する回線を選択できる。一般的には、アプリケーションごとに独自のポート番号を使用するので、アプリケーションごとに異なる経路を設定できる。

3.3 関連研究

ここで、既に実装が公開されている首振り究について述べる。以下で使用する「フロー」とは、ある始点ノードから終点ノードまでの一連のパケットの流れである。

3.3.1 SuMiRe

SuMiRe[5]は、終点アドレス、始点アドレス、終点ポート番号、始点ポート番号の組合せにより経路制御を行なう機能を持つ。

経路表は変更せず、フローを識別する経路のリストを、別に保持している。パケットの転送時には、経路リストをまず検索し、適合するエントリがあればその経路にパケットを転送する。適合するエントリがなければ通常の経路表を検索してパケット転送する。

また、SuMiReは、NAT[6]、パケットフィルタリングの機能を持っている。

3.3.2 froute

froute[7] は、SuMiRe と同じく、終点アドレス、始点アドレス、終点ポート番号、始点ポート番号でフローを識別する。経路表に拡張を加え、既存の経路制御機構を扱うことができる。

ルータ間にある複数の経路へのインターフェイスを、一つのデバイスにまとめることにより複数の経路を活用できるようにしている。

3.3.3 STAR

STAR[8] は、終点アドレスの他に、始点アドレスを考慮した経路制御機構である。地域 ISP(Internet Service Provider) と大手 ISP を接続する際、大手 ISP が契約している ISP のパケットのみを転送するために考案された。始点アドレスごとに経路表を保持し、パケットの始点アドレスによって検索する経路表を選択する。ポート番号を考慮することはできない。

3.4 関連研究の問題点

前節で述べた既存の首振り機構の問題点を述べる。

3.4.1 条件の自由度

前節で挙げた首振り機構は、いづれの実装においても、限られたいくつかの条件を指定できるだけであり、設定の自由度に欠ける。このため、通信に対する要求を満足させる性能を持っていない。

3.4.2 回線状態の動的な反映

既存の首振りルータを含めた静的経路制御では、ルータに接続されている回線の状態を検知し、経路制御に反映させる機構は存在しない。

図 3.2 において、ホスト H3 に向かう経路が静的にリンク A に向けられていたとする。

この状態の時、リンク A のどこかで障害が発生するなどしてリンク A がネットワークの経路として機能しなくなったことを考える。この場合、

リンク B による迂回経路が存在するにも関わらず、H1 から H3 に向かうパケットは R1 より先に転送されることがなく、H3 に到達できない。

第4章　モデルの提案

本章では、これまで述べてきた問題点を解決するための、統合的な首振りルータの枠組であるモデルとして、OROCHI を提案する。

4.1　問題解決へのアプローチ

経路の制御は、ネットワーク全体のトポロジーから経路を決定する機構と、ルータ内で行なわれるパケット転送に大別できる。局所的なものに、大別できる。これまでに述べてきた問題点は、ネットワークの一部分で限定的に解決することが現実的である。ネットワーク全体を考慮した経路制御プロトコルによって問題点を克服することは、規模性の問題からも困難である。

よって、本研究では、解決のための手段として、局所的な経路制御、つまりパケット転送のための処理である、首振りに着目した。

4.2　モデルの概要

OROCHI は、ルータの一部の機能として、既存の経路制御機構で不足しているパケット転送処理を実現し、必要に応じたパケット転送を可能にすることを目的とする。

OROCHI では、既存の経路制御機構で、パケット転送の判断材料として扱われている、終点アドレス以外の要素を取り入れる。本論文では、パケット転送の判断材料として、5 つの項目を取り上げる。しかし、提案するモデルでは、将来的に新たな要素で首振りを行なえるよう、要素が追加できるように拡張性を持つ。また、経路制御機構の枠組と親和性を保ち、パケット転送のポリシーと既存の経路制御の両方を扱える多様な選択肢を持ったルータが実現する。

OROCHI の概念的な構造を、図 4.1 に示す。

OROCHI は、全体を統括し、首振りの際の優先度づけを決定する制御

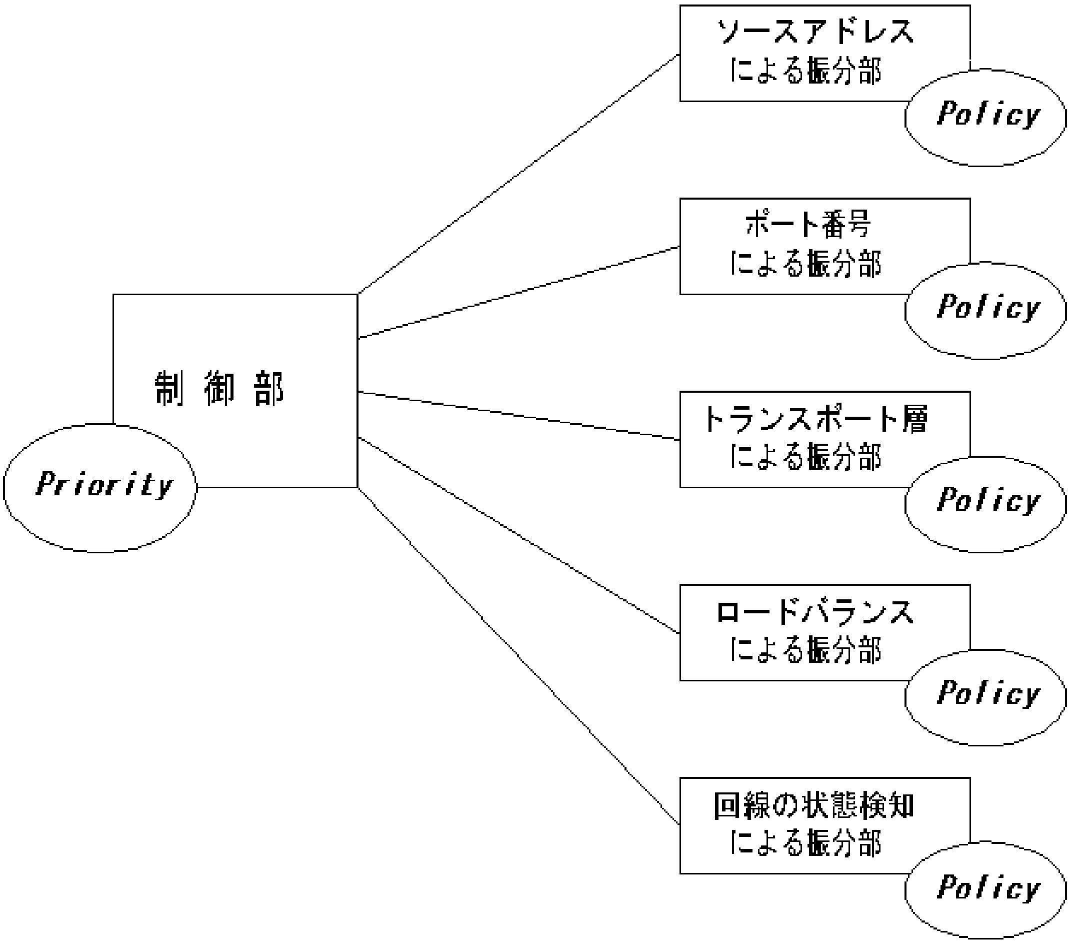

図 4.1: OROCHI の概念的な枠組

部と、実際に首振りのための処理を行なう５つの振分部から成る。これらはモジュールとして独立しており、振分部については、モジュールを追加することで、新たな要素による首振りが可能となる。

4.3　ポリシーの定義

各振分部では、ユーザから与えられた要素によって、パケットの転送先へのインターフェイスを決定する。このユーザから与えられた要素を、本論文では「ポリシー」と定義する。本論文で扱うポリシーには、次の５つがあり、それぞれが一つの振分部を構成する。

- 始点アドレス
- 終点ポート番号
- トランスポート層プロトコル
- ロードバランス
- 回線の状態

4.4 優先度の順位づけの定義

アプリケーションからの要求や、ネットワークからの状態報告について、いつでもこれらの要求を全て満たしたパケット転送の処理ができるとは限らない。よって、制御部では、各首振りの要素であるポリシーに対して、首振りの条件となる優先度の順位づけを行なう。

4.5 本モデルが果たす役割

本モデルは、自由なポリシーの記述、新たなポリシーの追加、ポリシー間相互の優先度づけの記述を可能にする。各振分部の機能に応じたパケットの転送と、これらの統合的な運用を実現する。4.1 に示す本モデルでは、パケットの転送処理を自由に組み合わせて以下に挙げることなどが実現できる。

- 特定の発信者からのパケットのみを、特定の経路に向けられる。
- アプリケーションの相違を判別し、特性に見合った適切な経路を選択できる。これにより、実時間性の要求される通信は低遅延の回線を通すことができる。
- より早いパケットの到達が求められる ICMP のみを、低遅延の回線に向けられる。
- 図 3.2 において、リンク A に帯域容量限界のトラフィックが流れ、リンク B にはトラフィックが流れていない、という状況を回避できる。

- 経路として設定されている回線が使用不可能に陥った場合、その経路へ向けられていた首振りの全てが機能しなくなり、パケットはルータから先に転送されなくなる。

4.6 本モデルの位置付け

4.6.1 経路制御機構との関わり

ここで、首振りシステムである OROCHI と、経路制御機構との関わりについて論じる。

経路制御機構とは、経路制御プロトコルに基づき、パケットの経路制御を行なう機構のことである。動的経路制御、静的経路制御とも、ネットワーク全体を通して、パケットが適切な経路を辿り、目的ノードへ到達させるための機構である。

首振りシステムは、経路制御機構とは異なる。首振りの処理を行なう前段階で経路表を参照するものの、次ノードへのインターフェイスを決定する際に、既存の経路制御機構のように、経路表に依存しない。また、首振り機構は、首振りを行なうルータの中で、完結する。

OROCHI と既存の経路制御機構の関係は、ルータ内で行なう処理の一つとして、既存の経路制御機構の不足した機能を OROCHI で補完する。この関係を図 4.2 に示す。

4.6.2 OSI 7階層モデルとの関わり

OSI 参照モデルに基づくと、経路の制御ではネットワーク層が隣接する上位層あるいは下位層からデータを受け取り、経路を決定して下位層へ渡す。ネットワーク層では、他の層の情報には関知しない。

一方 OROCHI では、パケット転送の過程において、上位層の情報をパケット転送に適用する。また、下位層のネットワークの条件を汲み取り、上位層の要求に適合するようにパケット転送を行なう。

OROCHI では、パケット転送はネットワーク層で行なうにも拘らず、他の層の情報を獲得し、パケット転送に反映させる。このことは、OSI で定義されたレイヤリングに反する。しかし、ネットワークでのより効率的なパケット転送を行なう場合、OSI のモデルの階層構造を越える処理を行なう方が現実的である。

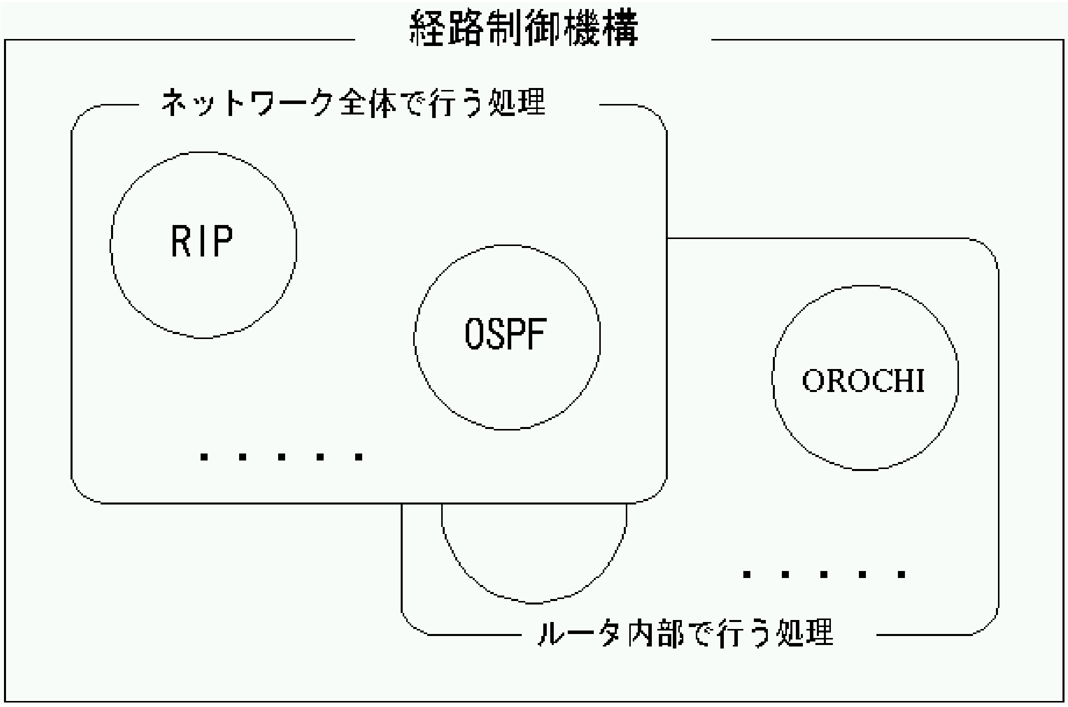

図 4.2: 経路制御機構における OROCHI の位置付け

4.7 様々なレイヤーの情報による首振り

既存の経路制御機構では、ネットワーク層の情報でしかパケット転送が行なわれなかった。しかし、本モデルでは、複数のレイヤーの情報をパケット転送に使用できる。レイヤーとパケット転送に使用できる情報の関係を、図 4.3 に示す。

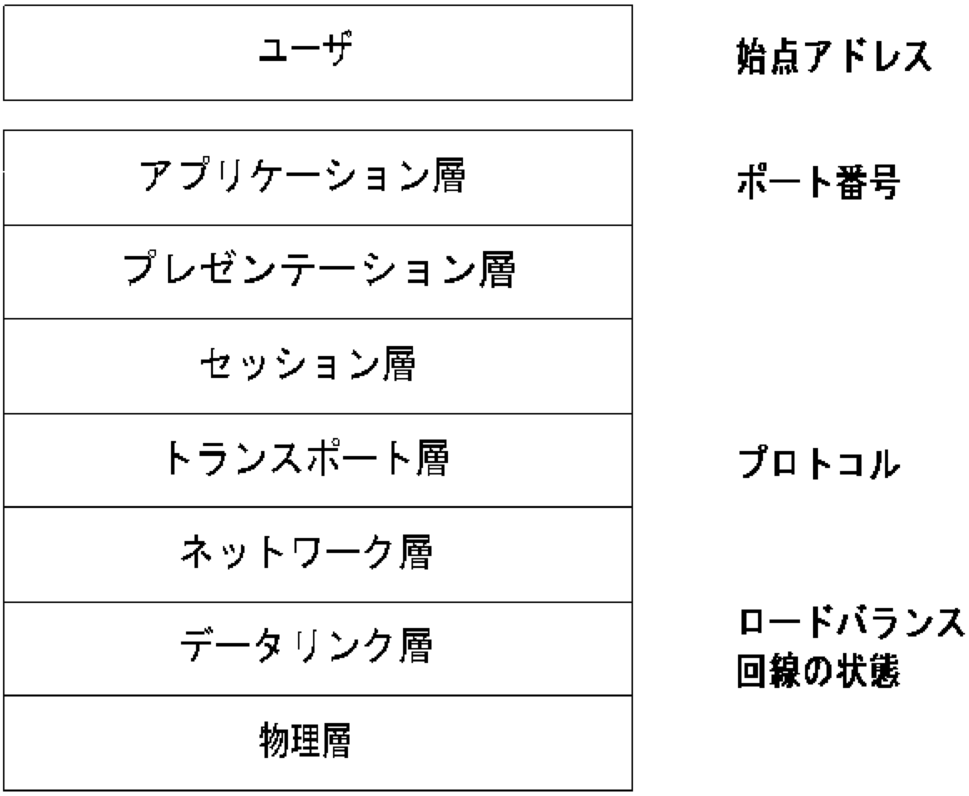

図 4.3: OSI7 階層モデルにおける首振りの各要素の位置付け

第5章　設計

本章では、第4章で提案した、管理者が自由度を持ってパケットの経路を制御できる統合的な首振りルータである、OROCHIの設計について述べる。

OROCHIは、首振りを統括する制御部と、実際に首振りのためのポリシー決定を行なう複数のモジュール部から成り立つ。これら全体についての設計と、個々の部位の設計について述べる。

5.1　設計基本方針

第4章で提案したOROCHIを、汎用性の高い首振りルータとして実現するため、以下の基本方針に基づいて設計する。

ネットワークトポロジーの前提

本機構は、隣接する2つのルータ間に複数の経路が存在することが、前提となる。2つのルータの間に複数のルータが存在した場合、ループが生じるなどの一般の経路制御のポリシーに反する現象が起こる可能性がある。しかし、既存の経路制御の枠組以外の情報によりパケットを転送することによって発生する問題は、運用上で回避するべきである。

パケットが持つ様々な要素による首振りの実現

ユーザが可能な限り任意にパケット転送を制御するために、パケットや回線状態などの様々な情報をもとに首振りを行なえるようにする。

拡張性

首振りの処理を行なう部分を、首振りを行なう要素ごとにモジュール化する。モジュール化することにより、ポリシーの追加による機

能の拡張が容易になる。また、このことにより、ユーザからの複数の要求を、独立して処理しパケット転送に反映することができる。

優先度の順位づけ

複数のポリシーを、同時にはパケット転送に反映できない場合が考えられる。ポリシーに対して優先度を決定できるようにし、優先度に応じてパケット転送を行なうよう設計する。

5.2 OROCHI の設計

OROCHI は、実際に首振りのための処理を行なうモジュールと、これら全体を監視し制御する制御部から成る。

制御部は、ユーザからの要求に基づいて、ポリシーを持った各振分部を呼び出す。呼び出しに対し、振分部は、各々が持つポリシーに基づいた値を、制御部に返す。この関係を、図 5.1 に示す。

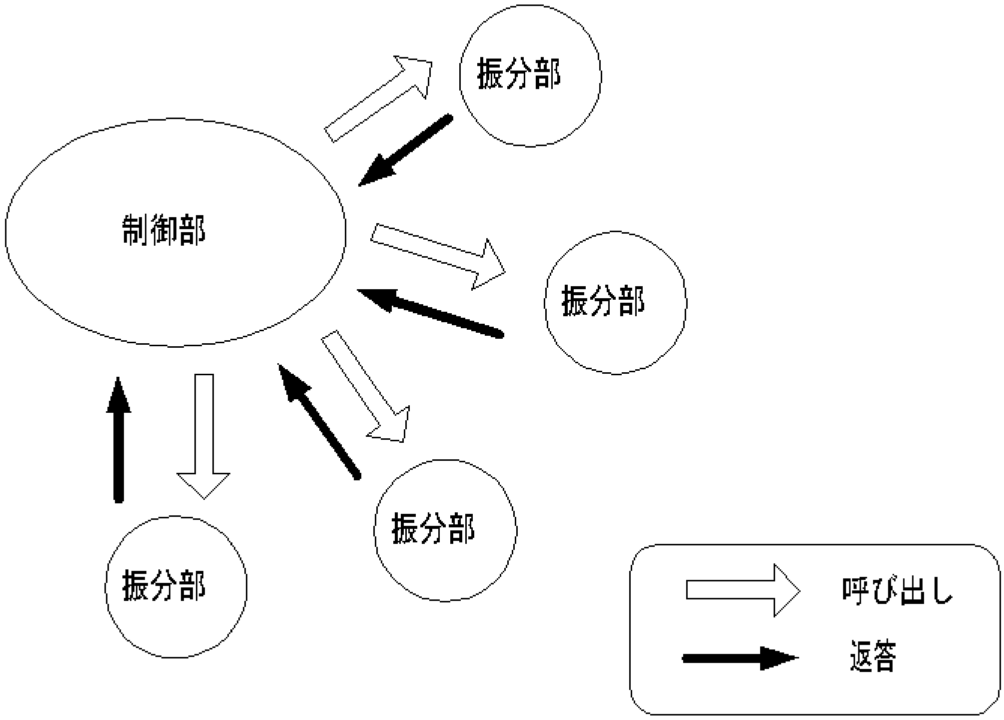

図 5.1: 制御部と振分部の関係

各モジュールの機能を組み合わせることにより、多様なポリシーによるパケット転送の制御が可能となる。

5.3　制御部

OROCHI 制御部は、各振分部を統括し、パケット転送の優先度づけを行なう。首振りの対象となるパケットを受信した時、振分部を呼び出す。制御部が行なう処理を、実行する順に挙げる。

1. ユーザが記述したパケット転送を行なう条件を読み込む
2. ポリシーを設定する
3. 回線の状態を検知するモジュールを呼び出す
4. 要求に沿ったモジュールを呼び出す
5. モジュールから送信先インターフェイスを受け取る
6. 優先度の順位づけを行なう
7. 送信のための処理に移す

OROCHI の内部で行なわれる一連の処理の流れを、図 5.2 に示す。

5.4　各振分部

実際に首振りを行なうための処理は、各モジュールが行なう。パケットに対する処理を行なうために基にする情報により、モジュールは複数のタイプに分かれる。本論文で OROCHI に設計を行なうモジュールのタイプは、3 つである。モジュールの種類と処理を、以下に挙げる。

タイプA ユーザからの要求

- 始点アドレス
- アプリケーション
- トランスポート層プロトコル

タイプB 回線の状態

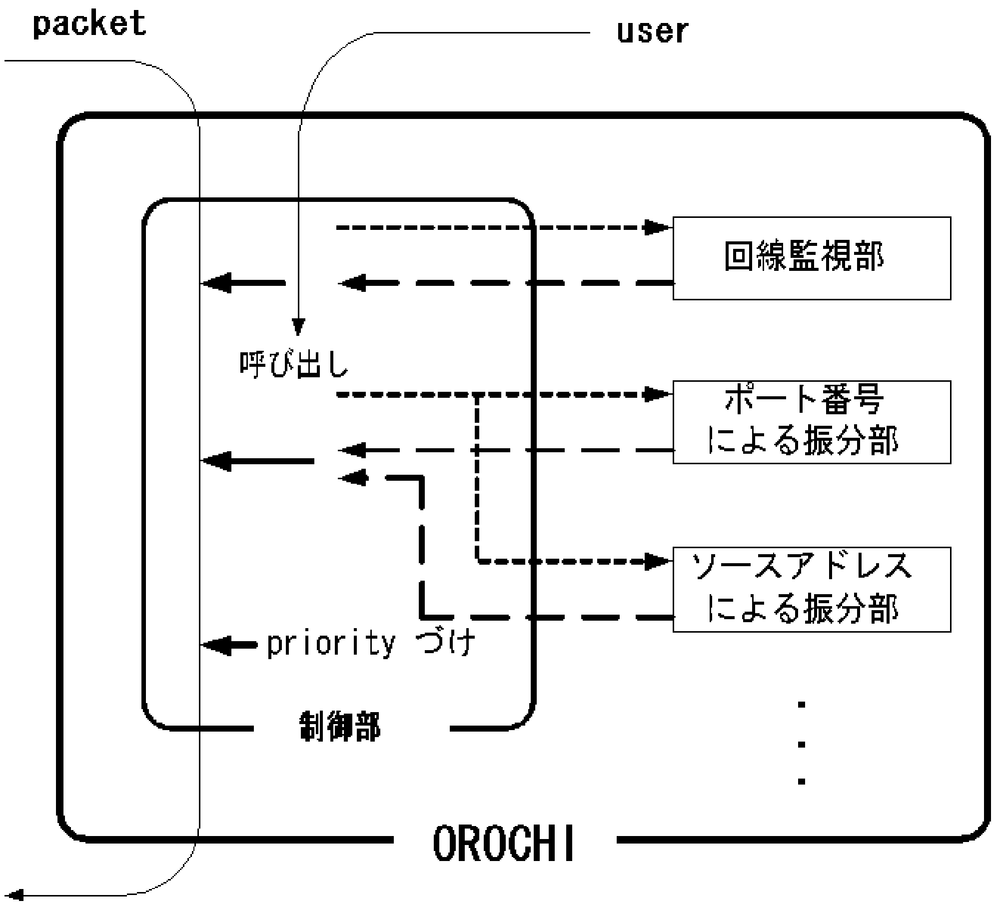

図 5.2: OROCHI 内部の処理の流れ

- 回線の状態

タイプ C　ユーザからの要求、回線の状態の両方

- ロードバランス

この関係を、図 5.3 に示す。

ポリシーがユーザからの要求に依るモジュールがタイプ A、ユーザからの要求と回線の状態の両方を考慮するモジュールがタイプ B、回線の状態のみに依存するものがタイプ C である。

これら 3 つのタイプのモジュールの設計について述べる。

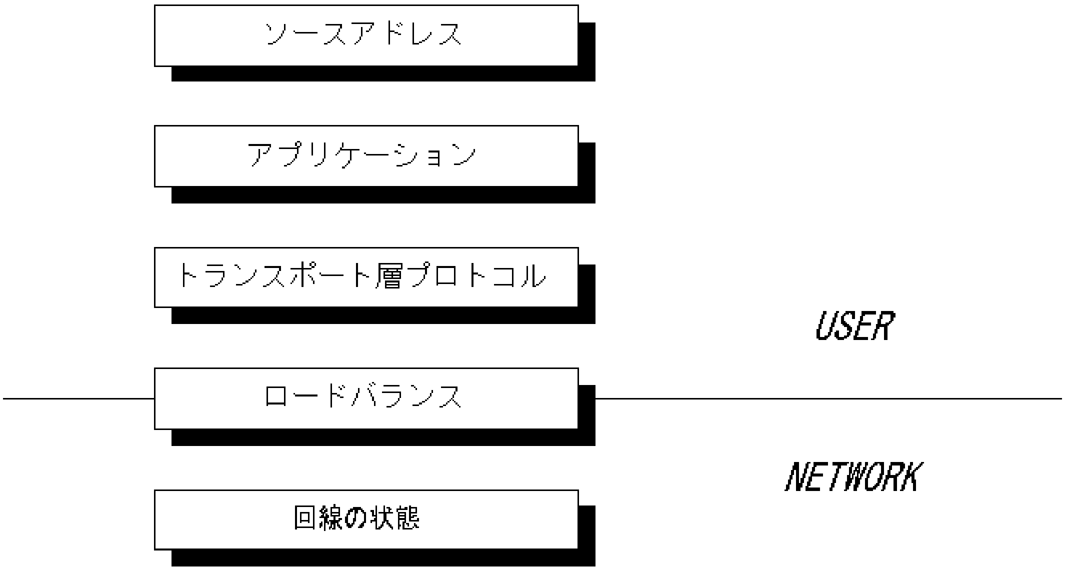

図 5.3: 各モジュールとユーザ、ネットワークの関係

5.4.1 タイプ A

始点アドレス、終点ポート番号、トランスポート層プロトコルの各振分部が、タイプ A のポリシーをもつモジュールに該当する。制御部からの呼び出しに応じ、ポリシーに基づいた振分のための処理を行なう。この結果得られる経路へのインターフェイスを、制御部へ返す。

5.4.2 タイプ B

回線の状態検知による振分部が、このモジュールに該当する。このモジュールは、ルータで首振りの対象となる回線へのインターフェイスを監視する。このモジュールは、制御部が首振りの対象となるパケットを受信するたびに呼ばれる。回線に障害が発生し、ネットワークの経路として機能しなくなったことを検知した場合、制御部への返答の際に通知する。また、障害の復旧を認識した場合にも、制御部へ通知する。回線の状態に異常がなければ、そのことを通知する。

何らかの要因によりネットワークを構成する回線に障害が発生した場合の状況を、図 5.4 に示す。

ユーザのポリシーにより、首振りルータ R1 から R2 への経路が link

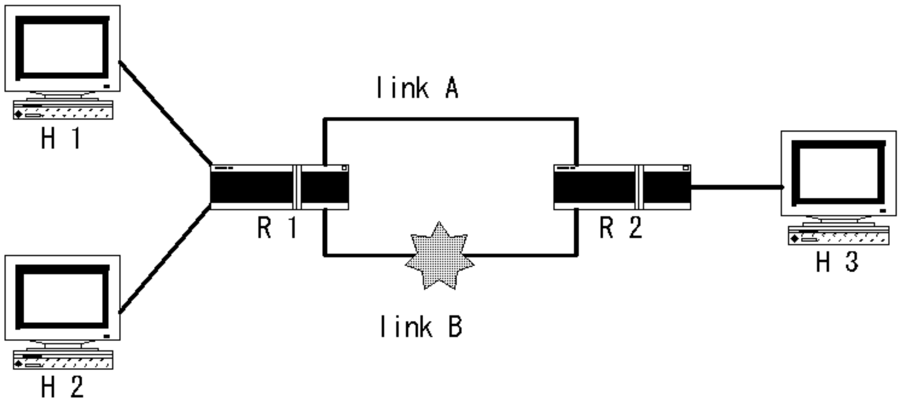

図 5.4: あるリンクに障害が発生した場合

B に向けられていたとする。link B に障害が発生した場合、link A が使用可能であるにも関わらず、ホスト H1、H2 から H3 に向かうパケットは H3 に到達できない。タイプ C のモジュールで障害を検知することにより、この問題を解決できる。

5.4.3 タイプ C

ロードバランスによる振分部が、このモジュールに該当する。このモジュールは、ユーザからの情報と、回線のトラフィックの状態の、両方の情報に基づいてパケットの回線ごとの送信量の制御を行なう。

5.5 首振りの優先度

ユーザは、OROCHI に対して一度に複数の要求を記述できる。この時、ネットワークの状態と、ユーザからの要求の両方を満足することができない可能性もある。

首振りを行なうための要素は、首振りの対象となっている回線の状態、回線使用率、ユーザからの要求の 3 つである。

これらの要求、条件が衝突を起こさないようにパケットの転送を制御するためには、転送を行なうポリシーに優先度の順位づけをする必要がある。このため、OROCHI では次に述べるように首振りの各要素の優先

度を決定する。

まず、回線が経路として使用可能であるか否かについて考える。回線がパケット転送のための経路として使用不能であった場合、ロードバランスを考慮したり、ユーザの要求を受け入れる余地はない。従って、回線の状態を検知することを、最も高い優先度とする。

次に、回線使用率を監視しパケット転送の制御に反映させることと、ユーザからの要求を反映させる、2つの間の優先度を考える。回線使用率が100% に近い時に、ユーザからの要求を受け入れて回線容量を超えるパケットを流そうとしてしまうことは、非効率的である。よって、回線のロードバランスを考慮することを、ユーザからの要求よりも優先させる。

従って、3つの要素の優先度は、回線の状態、ロードバランス、ユーザからの要求、の順になる。

要求内容における、内容ごとについての優先度づけは、それぞれの要求を該当する各モジュールに割り振り、それぞれが首振りの処理を行なう。その結果が制御部に返ってきた時点で、優先度を順位づけする。この要求内容についての優先度を決定するポリシーは、ユーザからの設定によるものとする。

5.6 首振りを必要としない場合

首振りを必要としない時、OROCHI によって他の経路制御プロトコルの動作に影響を与えてはならない。OROCHI は、首振りを必要としない時、パケットに対して一切の処理を行なわない。

パケットの流れとしては、経路表を検索した後に OROCHI 制御部に処理が移る。このため、OROCHI で処理が何も行なわれなかった場合、経路表の検索結果がパケット転送に反映される。

5.7 ネットワークトポロジーの変化への対応

本モデルは、図3.2 に示したように、ルータ間の回線が2つである場合を想定して設計を行なった。しかし、ルータ間の回線の数を n 本とした場合、実際の動作は n に依存しない。

図3.1 に示したものが、n=1 であるネットワークトポロジーである。ルータ間の経路が1つの場合ロードバランス、回線の状態検知に

よる振り分けは、意味をなさない。

5.8 ネットワークでのループの回避

首振りルータは、経路制御プロトコルを考慮せずにパケット転送を行なう。このため、首振りルータを含むネットワークで、首振りルータと、他のルータとの間でループが起こる。図5.5を用いて、ループが起こる例を述べる。

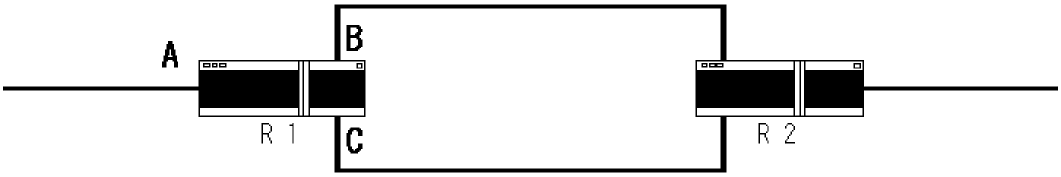

図5.5: ルーティングループが起こるネットワークトポロジー

ルータR 1 が首振りルータである。A、B、C は、R 1 の各回線へのインターフェイスを表す。この場合、首振りとは、R2へのインターフェイスであるBまたはCを選択することである。

例えば、ポリシーに基づき、telnetのパケットはCから送信されるように設定されている場合を考える。この時、Bから受信されたtelnetのパケットは、無条件でCから送信されてしまい、R1とR2の間でループが起こる。

このことを回避するためには、経路表を検索して、次ホップがR2である場合のみ首振りを行なわなければならない。このために、経路表を検索した後にOROCHI制御部に処理を移す。

5.9 機能の拡張性

OROCHIでは、振分部のモジュールを追加することによって、新たなポリシーによる首振りが可能となる。振分部は、制御部からの呼び出しに対して返り値を返す、という規則に従うことで、モジュールの追加が容易になる。

第6章　実装

本章では、第5章の設計に基づいた、OROCHIの実装について述べる。

6.1　実装環境

OROCHIの実装は、FreeBSD 3.2-RELEASE上で、IPv4を対象として行なった。

6.2　全体の流れ

OROCHIは、カーネル内のパケット転送の際に経路を決定するプログラムである、ip_output.cに変更を加えることにより実装した。

各モジュールは、OROCHIの制御部から呼び出す。

各モジュールは、制御部とパケットやリンクの状態の情報を交換し、首振りのための処理を行なう。最終的には、制御部が取りまとめてパケットの転送先経路へのインターフェイスを決定する。

6.3　パケット転送の処理の流れ

OROCHIを実装したルータにおける、パケット転送の処理の流れについて説明する。

ネットワークインターフェイスが、データリンク層からパケットを受信すると、キューイングを行なう。この後、受信したパケットが自ルータ宛のものか、他へ転送するべきものかを判断し、自ルータ宛ならトランスポート層へ送る。転送するパケットであるなら、経路表を検索し、OROCHI制御部へ処理が移る。OROCHI内部で首振りのための処理が行なわれた後、送信のための処理として、パケットの終点アドレスを抽出して経路表を検索し、送信するインターフェイスを決定する。最後に、

ネットワークインターフェイスからデータリンク層にパケットを送る。

このカーネル内のネットワーク層での処理の一連の流れを、フローチャートとして、実際に処理を行なう関数とともに図6.1に示す。

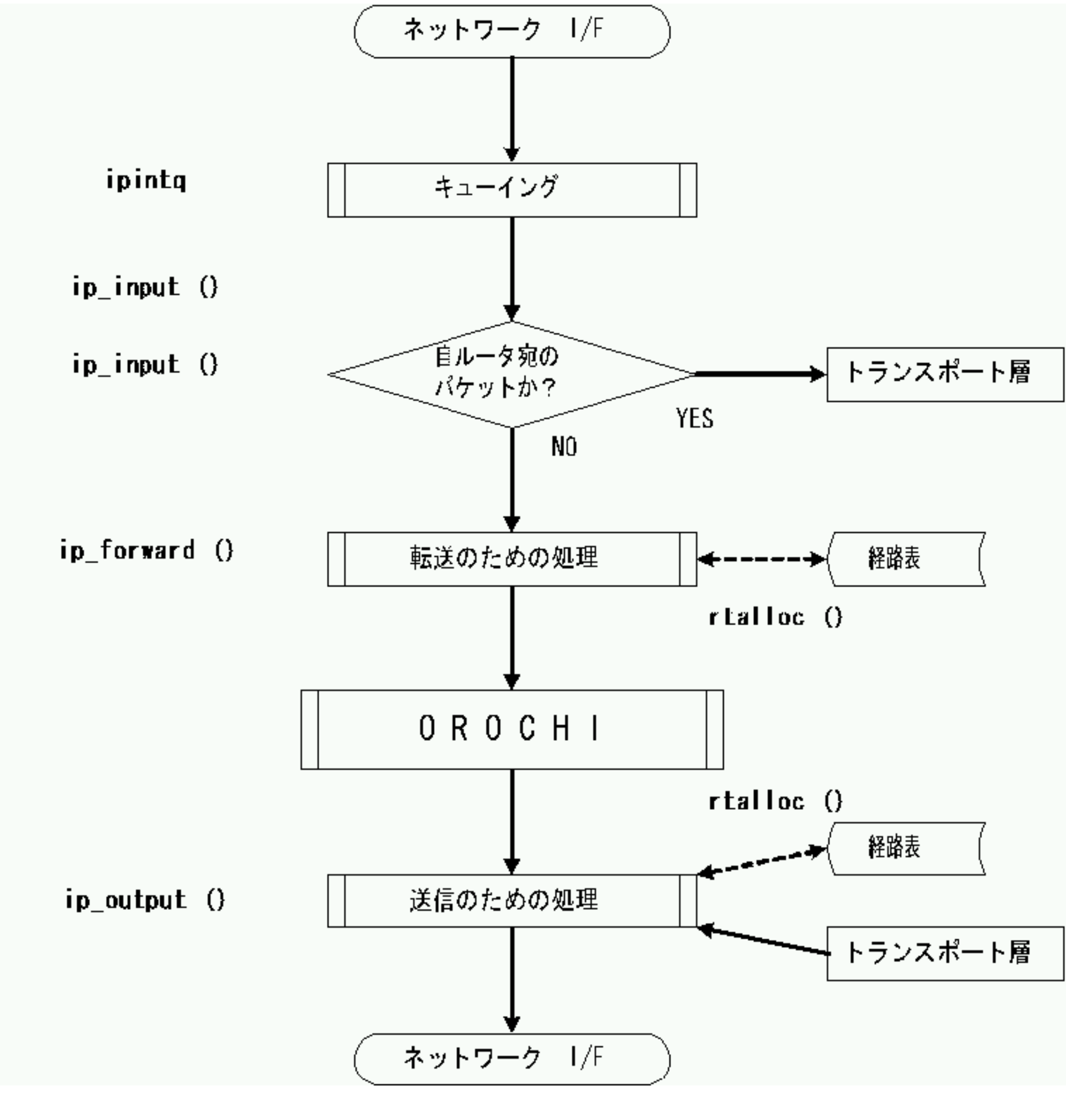

図 6.1: カーネル内でのパケット転送のための処理の流れ

表 6.1: 首振りルータを含むネットワークの IP アドレスの例

node	I/F	IP address
R1	a	10.1.0.1
R1	b	10.2.0.2
R1	c	10.3.0.3
R2	d	10.2.0.4
R2	e	10.3.0.5
R2	f	10.4.0.6

6.4 OROCHI 制御部の実装

パケットの転送先を決定するために、経路表を検索した後で、OROCHI の制御部をの処理を行なう。

各モジュールからの返答に対して優先度づけの処理を行なったあと、首振りの要求に従ったパケットを転送する処理を行なう。この後、OROCHI を抜けてパケット送信のための処理に移る。

ip_output.c の中にある、ip_output 関数で、rtalloc により経路表を検索した後に、OROCHI の制御部の実装である orochi 関数を呼び出す。

制御部で設定する項目を、図 6.2 の例を用いて述べる。このネットワークは、表 6.1 に示す IP アドレスを持つ。ルータ R 1 を、首振りルータとする。R1 の制御部では、次に挙げる項目を設定する。

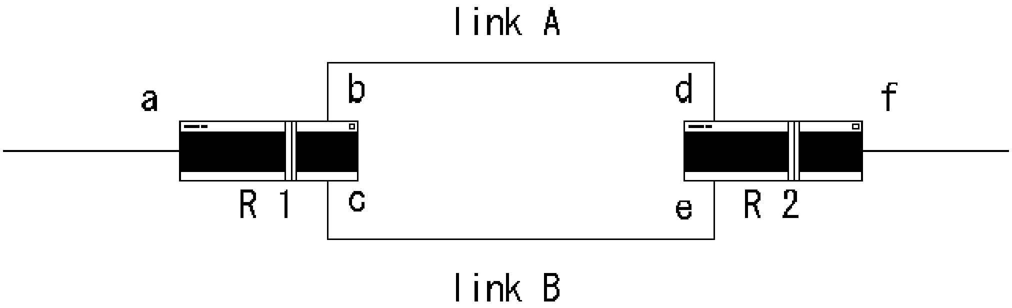

図 6.2: 首振りルータを含むネットワークの例

- 目的ノードへの経路の送信インターフェイスの IP アドレス

- パケット転送のためのポリシー
- 転送先ノードの IP アドレス

6.5 モジュール部の実装

各ポリシーに基づいた転送の処理は、それぞれのモジュールで行なう。各モジュールの実装について述べる。

6.5.1 ポート番号による振り分け部

受信パケットの、IP ヘッダの終点ポート番号を抽出する。ユーザから要求されたポート番号と合致した場合、指定されたインターフェイスからパケットを送信するよう処理を行なう。

6.5.2 始点アドレスによる振分部

受信パケットの、IP ヘッダの始点アドレスを抽出する。ユーザから要求された始点アドレスと合致した場合、指定されたインターフェイスからパケットが送信するよう処理を行なう。

6.5.3 トランスポート層のプロトコルによる振分部

受信パケットの、IP ヘッダのプロトコルフィールドを抽出する。ユーザから要求されたトランスポート層プロトコルと合致した場合、転送先のルータの IP アドレスを、指定されたものに書き換える。これにより、指定されたインターフェイスからパケットが送信される。

ポート番号は、カーネル内の netinet/in.h [9] に定義されている全てのものが有効である。

6.5.4 ロードバランスによる振分部

首振りの対象となるリンクの、トラフィックの状態を監視する。トラフィックが、ユーザが設定した回線使用率に達した場合、新たにルータに流入してくるトラフィックは、別の経路に向ける。

6.5.5 回線の状態による振分部

このモジュールは、制御部から呼び出されるのではなく、独立したプロセスとして働く。回線での障害の発生および復旧の検知は、ネットワーク層で行なう。

6.6 オプション機能

OROCHIでは、首振りの機能とは別に、2つのオプション機能を搭載する。これらの機能を備えることにより、より柔軟性のある経路の選択をすることが可能となる。

6.6.1 特定の経路を無視する

一つは、ある特定の経路を無視することである。これは、パケット転送の際、ネットワーク上にはその経路が存在しないかの様に見せかけ、首振りのための経路の選択肢から外す。

この機能は、ネットワークとして動いているが信頼性が低い、という回線を首振りの対象から外し、ネットワーク全体として、より高い信頼性を得るために効果的である。

本研究では、パケット転送の際に、経路制御プロトコルには依存しない。このため、経路制御プロトコルによって決まる経路のコストは、一切参照しない。

6.6.2 特定のトランスポート層パケットを転送しない

もう一つは、特定のトランスポートプロトコルのパケットを転送しない、という機能である。これにより、不要なデータグラムを、遮断できる。

ユーザから指定されたトランスポート層プロトコルのパケットを、OROCHIを搭載したルータで破棄する。

第7章　評価

本章では、本研究で提案した OROCHI の性能を検証し、評価する。

7.1　評価環境

評価のための実験なネットワークを構築し、評価を行なった。ネットワークトポロジーを、図 7.1 に示す。

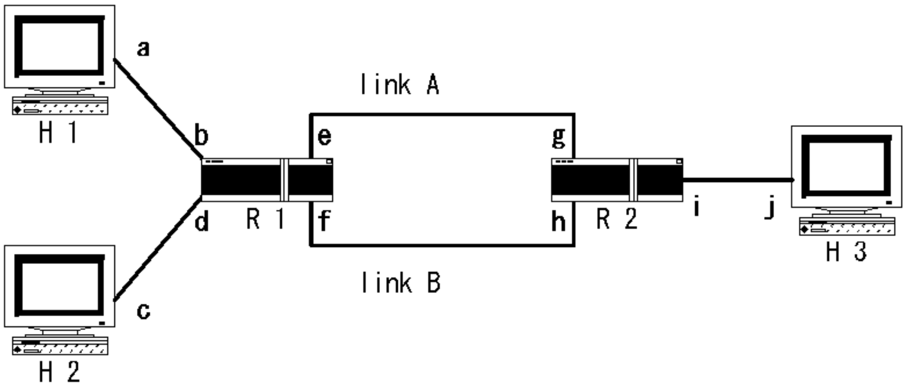

図 7.1: 評価を行なったネットワークトポロジー

各ノードが持つインターフェイス a ~ j に、ネットワークアドレスの異なるアドレスを割り振った。これにより、疑似的に、異なるネットワークを跨いだトポロジーを形成した。

ルータ R1 を首振りルータに設定し、ホスト H1、H2 から H3 に向けてパケットを送信した。ルータ R1、R2 の各インターフェイスでパケットを計測し、ルータ R1 によって首振りが行なわれていることを確認した。

表 7.1: 関連研究との比較 :機構の統合的な機能について

system	拡張性	policy の組合せ	首振り以外の機能
SuMiRe	×	△	○
froute	×	△	×
STAR	×	×	×
OROCHI	○	○	○

7.2 定性的評価

7.2.1 統合的な評価

機構が持つ性能について、関連研究との比較を行なった。3.3 で述べたように、SuMiRe は NAT とパケットフィルタリングの機能を持っている。OROCHI の 『他の機能」とは、回線の状態を検知することである。

評価項目は、以下の通りである。

拡張性
首振りのためのポリシーを、新たに追加できるなど、機構としての拡張性があるか

ポリシーの組合せ
複数のポリシーを、同時にパケット転送に反映できるか

首振り以外の機能
首振り以外の機能を備えているか

比較の結果を、表 7.1 に示す。

各機構の性能について述べる。

SuMiRe
始点アドレスと終点ポート番号の組合せとして、フローによる識別が行なえる。しかし、それ以外の組合せは、できない。首振りルータの他に、NAT、パケットフィルタリングの機能を持つ。

froute
SuMiRe と同じく、フローによる識別が可能である。

STAR

始点アドレスによる振り分けのみである。

OROCHI

モジュールの追加により、新しいポリシーを反映した首振りが可能となる。

7.2.2 首振りが可能なポリシーの評価

首振りを行なうことができるポリシーについて、関連研究との性能の比較を行なった。

この結果を、表7.2に示す。

表7.2: 関連研究との比較:首振りのポリシーについて

system	src address	port number	protocol	load barance	flow
SuMiRe	○	○	×	×	○
froute	○	○	×	×	○
STAR	○	×	×	×	×
OROCHI	○	○	○	○	×

7.3 評価のまとめ

本研究では、パケットごとに制御を行なう。よって、パケットの一連の流れを意味するフローごとの制御はできない。しかし、パケットごとに制御することにより、より柔軟性のあるパケット転送を実現できた。また、既存の首振りルータでは完全には実現されていないポリシーの併用や、機構としての拡張性では、本研究で提案したモデルが優れていることが分かった。

謝辞

本研究を行なう機会を頂き、御指導を頂きました慶應義塾大学環境情報学部教授の村井純博士徳田英幸博士に感謝致します。また、絶えず御助言と御指導を頂きました同学部の楠本博之博士と中村修博士に深い感謝の意を表します。

本研究を進めるにあたり、貴重な御意見を頂きました慶應義塾大学政策メディア研究科の西田視磨氏、小原泰弘氏、杉本信太氏をはじめとする多くの諸先輩、同輩の方々に感謝致します。

参考文献

[1] R. Braden. Requirements for Internet Hosts - Communication Layers. RFC1122, IETF, October 1994.

[2] G.Malkin. RIP Version 2. RFC 2453, IETF, November 1998.

[3] J.Moy. OSPF Version 2. RFC 2328, IETF, April 1998.

[4] R.Braden, L.Zhang. Resource ReSerVation Protocol(RSVP). RFC 2209, IETF, September 1997.

[5] 藤澤慎一, SuMiRe-0.52. http://www2.kame.net/˜ fujisawa/, February 1999.

[6] K. Egevang, P. Francis. The IP Network Address Translator (NAT). RFC 1631, IETF, May 1994.

[7] 小巻賢二郎, 所真理雄. フローを考慮した経路制御機構. 情報ネットワーク研究会, 電子情報通信学会, May 1998.

[8] 鍛冶武志, 小林和真, 山口英, 尾家祐二. ソース IP アドレスを考慮した経路制御システムの提案 地域 IX における経路制御問題の解決. 情報処理学会分散システム運用技術研究会, 1997.

[9] J.Reynolds, J.Postel. Assigned Numbers. RFC 1700, IETF, October 1994.