

Title	ネットワーク状態を反映した経路選択によるQoS制御方の研究
Author(s)	木下, 雅博
Citation	
Issue Date	2001-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1442
Rights	
Description	Supervisor:日比野 靖, 情報科学研究科, 修士

修士論文

ネットワーク状態を反映した 経路選択による QoS 制御法の研究

指導教官 日比野靖 教授

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

木下雅博

2001年2月15日

要 旨

従来の経路選択では一つの判断基準で、最短経路の木から経路を選択していた。

本論文では複数の判断基準で複数の経路からネットワーク状態に応じた経路選択によって QoS 制御する方法を提案していく。

また、遠くのネットワーク状態を知るために、その情報収集と保持を行い、Stub Router にネットワーク状態を反映した経路を知らせる機能も持つ管理制御を行うユニットを分散する方式を提案するのだが、各々の管理制御ユニット間の通信がすばやく完了することを保証する Queue Management System も提案する。

これらの機能をモデル化してシミュレーションを行い本論文で提案した方式の有効性について評価し、問題点を明らかにする。

目次

1	はじめに	1
1.1	本研究の背景と目的	1
1.2	本論文の構成	3
2	サービス品質 (QoS) に求められる要件とシステムの概要	5
2.1	サービス品質 (QoS) に求められる要件	6
2.2	システムの概要	6
3	経路選択	8
3.1	従来の経路選択法	8
3.2	経路選択アルゴリズム	10
3.2.1	計測要素と評価方法	10
3.2.2	経路を確定するためのアルゴリズム	12
3.2.3	静的要素による経路選択	13
3.2.4	動的要素による経路選択	14
4	管理制御機構	15
4.1	管理制御ユニット間の通信機構	17
4.1.1	管理制御ユニットと Router との連携	18
4.2	管理制御モデル -フロー毎の制御-	20
4.3	管理制御モデル -状態に応じた一括制御-	20
5	Queue Management	21
5.1	Queuing モデル	21
6	実験結果	24
6.1	Queue Management	24

6.1.1	シミュレーション手法	24
6.1.2	目的	25
6.1.3	結果と考察	26
6.2	管理制御機構	31
6.2.1	シミュレーション手法	31
6.2.2	目的	33
6.2.3	結果と考察	33
6.3	経路選択	35
6.3.1	シミュレーション手法	35
6.3.2	前提条件と目的	37
6.3.3	結果と考察	38
6.4	システム全体の評価	51
6.4.1	シグナリングのトラヒックの遅延について	51
6.4.2	経路選択による QoS 制御について	51
7	おわりに	53
7.1	まとめ	53
7.2	今後の課題	54

第 1 章

はじめに

1.1 本研究の背景と目的

近年、高品位なデータ伝送の行える通信方式が求められている。特に実時間性の保証が求められており伝送系路上での遅延を減少させ、また遅延の揺らぎも最小限にすることが必要である。遅延を減少させるためにはノードでのサービス遅延の減少が必要で、サービス遅延を減少させれば結果的に遅延の揺らぎも減少されると考えられる。また、局所的な輻輳が発生しても、このことが Source 側で分からなければトラヒックの混んでいるネットワークに向けて更なるフローを流すことになり、ネットワークの輻輳を助長する。

インターネットやイントラネット上では性能の異なる様々なクライアントが様々なネットワーク環境で続されている。このようなクライアント間では、映像や音声、画像など様々なマルチメディアコンテンツをユーザ操作に対する応答（ユーザーインタラクション処理）や映像や画像の表示、音声の再生などが制作者及び使用者の意図した時間内で実行されること（実時間性の保証）が求められている。

上述してきたシステムを構成するために、本研究では、中継ノードの使用率及び、経路選択というポイントを取り上げ、経路選択によるトラヒックの分散と中継ノードの使用率によるネットワークの許容を越えるデータ伝送を拒否することによる関係を調べ、効果的な QoS 制御法の一つとしてとらえることにしている。

前述したユーザーインタラクション処理の応答時間などの実時間性を保証するために用いられる方法の一つに Adaptive 型というものがある。これは、アプリケーション側でバッファを用意しておき、常にバッファ内にデータをためることで到着するパケットの到

着間隔にばらつき（遅延の揺らぎ）があってもそれを吸収しようとする試みであるが、この方法だけでは十分な実時間性の保証が得られないのは過去の歴史から明らかである。なぜなら、バッファ内のデータを使い切ってしまうと次のデータが届かない場合、映像や音声では再生が止まり、画像では画像全体を表示させることができなくなってしまい、実時間性の保証が出来なくなる。

そこで、中継ノードの使用率を考慮した経路選択をして、宛先ノードまでの経路上の各中継ノードのサービス時間の和による遅延時間がその時点で最小となる経路を選ぶことで、全体の遅延時間を最小にすることで遅延の揺らぎの実時間性の保証に及ぼす可能性を低くすることを明らかにし、シミュレーションを行ってその評価を行う。

また、サービス時間の和と述べたが、サービス時間は二つに分けられる。つまり、待ち行列における待ち時間と伝搬遅延時間とパケットの転送時間である。

待ち行列における待ち時間とは各中継ノードにはそれぞれやってきたパケットを処理した後、そのパケットが転送されるまでバッファに溜められるのであるが、そのバッファに滞留している間の時間を指す。

次に、伝搬遅延時間¹の影響が開始するのは帯域幅が数十 Gbps、数 Tbps を越えたあたりから転送速度が光の速さよりも速くならないために影響が開始するが、それ以下の帯域幅の回線であれば伝搬遅延は無視できるほどの時間であるので本研究では伝搬遅延を除いた遅延時間を用いる。

また、本研究では逆方向からの伝送では最近では全 2 重である FastEthernet などは当たり前となってきているため本研究では逆方向からの伝送は反対側からの伝送によって妨げられないものと位置づける。

また、ネットワーク状態に応じた経路選択とは宛先ノードまでの中継ノードの使用率という動的に変動する判断基準を基に複数の経路が存在する場合、最適なものを選択する。また、これらの判断基準は各々のルータで情報を得て判断するのではなく、このための管理制御ユニットをおき負荷を分散させる効果を目論んでいる。

¹伝搬遅延時間とは、回線の中を電気信号が伝搬する時間で、絶縁体の材質やケーブルの構造にも依存するがだいたい 4.5 ~ 5.5ns/m である。この数値と光の速度との比を公称伝搬速度 (NVP: Nominal Velocity of Propagation) と称し、光の速度を 100 とするとケーブル対内を伝わる信号の速度は約 65 ~ 75 となっている。また、伝搬遅延は絶縁体の比誘電率とよりピッチの影響を受ける。遅延はおよそ絶縁体の比誘電率の平方根に比例し、誘電率の小さい材質を絶縁体とした方が遅延時間は短くなる。また、すりピッチにが細かくなるとそれだけ伝搬遅延が増加し、UTP などではカテゴリ 3 よりもカテゴリ 5 の方がすりピッチすりピッチが細かくなり伝搬遅延が増す。

インターネットやイントラネットで今日のようにWWWが普及した結果、今後さらに普及しその上で動く Web アプリケーションが増えるとなると、さらにより多くの映像や音声を扱ったコンテンツやそれらを統合したものが増えていくものと予想される。すなわち、ネットワーク上のリソースとして映画やテレビ番組やラジオなどを求めるようになるはずである。

そうなった場合、今よりさらに広帯域なネットワーク環境を使用できるようになるかもしれないが映像や音声などを扱ったコンテンツもより多くなり、また今よりさらにネットワークを利用する人が多くなればネットワークに接続した端末の数もさらに増加すると予想される。

そうなれば、本研究の目指すネットワーク状態に応じた経路選択によって遅延時間を最小とすることで遅延の揺らぎの影響を減らすことによる QoS 制御法の必要性は高いといえる。

次に上述してきた QoS 制御法を満たす QoS 制御システムを設計するために、必要とされる要件を明らかにしてシミュレーションを行う。

なお、シミュレーションではイントラネット環境を利用する QoS 制御法を想定し、下記の点を研究目的ととしている。

- イントラネット上での新しい QoS 制御法の利点を探り、ネットワーク状態に応じた経路選択をする QoS 制御法のシミュレーションを行い問題点を見だし、評価して有効性を確認する。

シミュレーションの規模としてイントラネットは複数の経路をもつ可能性が高く、導入コストも低くて、なおかつ導入において他の組織のネットワーク管理者と競技する必要もないため最適であると考えられる

1.2 本論文の構成

第2章では本研究で紹介する QoS 制御システムのモデルの概要について述べる。最初にサービス品質 (QoS) を制御するため、本研究の QoS 制御システムで使用する判断基準について紹介する。次に実際のモデルを紹介し、各機能がどのように連携してシステムを構築しているかの概要を示す。

第3章では管理制御ユニットで行う経路選択アルゴリズムについて述べる。最初に計測

する判断基準となる要素と評価方法について述べ、次に経路を確定するためのアルゴリズムについて説明し、その次に時間の経過によって状態変異が起こりにくい静的要素と逆に状態遷移が起こりやすい動的要素とに判断基準となる要素を分けてそれぞれの要素における経路選択アルゴリズムについて述べる。

第4章では本研究のQoS制御システムの機能の一つである管理制御ユニットについて述べる。最初に管理制御ユニット間で連携を取るための通信機構について説明し、次に経路選択を行うための管理制御の二つの違うモデルについて述べる。最初はフロー毎に経路選択を行ったモデルについて述べ、次にネットワーク状態に応じて一括してすべてのフローの経路選択を行うモデルについて述べて、それぞれのモデルの利点と欠点を洗い出す。

第5章では各中継ノードで待ち行列のサービスを行う際に行うスケジューリングアルゴリズムである本研究で提案するQueue Management Systemについて述べる。最初にモデルの紹介をし、次にシミュレーションの結果を掲載する。

第6章では実験結果の評価を行い、本研究で提案したいくつかの機構の有効性や問題点の洗い出しを行う。

第7章では、本研究のまとめを行い、今後の課題について述べる。

第 2 章

サービス品質 (QoS) に求められる要件とシステムの概要

インターネットに関連づけられて提供されるデフォルトのサービスは最善努力型 (best effort) の変化を受けやすいサービス応答として特徴づけられる。これはネットワーク内を流れるトラフィックによって発生する負荷が変化するとき、ネットワークの最善努力型のサービス応答もまた変化する。

このようなネットワークでは静的な特性を持つ判断基準で経路選択する試みだと QoS 制御は難しい。そこで、動的な要素を持つ判断基準を Path 毎に収集して計算をし、望ましいトラフィックの均衡を保つモデルを入れてきたトラフィックに適応し、分散させる必要がある。

そこで、本研究では以下の 3 つの機能を組み合わせた QoS 制御法を行う。

- 管理制御
- 経路選択
- Queue Management

これらの機能を使用してシステムを構成し、複数の経路が存在するならばトラフィックを分散する。また、管理制御を行い、接続要求に対して、ネットワーク状態に応じて要求される QoS を達成できない場合はアクセスを拒否し、次々と入ってくるトラフィックによってサービスしているトラフィックに対して提供している QoS が著しく悪化することを避けることが可能になると考えられる。

2.1 サービス品質 (QoS) に求められる要件

サービス品質 (QoS) を達成するために求められる要件は以下のである。

- 遅延時間の減少
- 遅延の揺らぎを少なくすること

本研究では上記の要件を満たす QoS 制御を行うシステムを提案する。

2.2 システムの概要

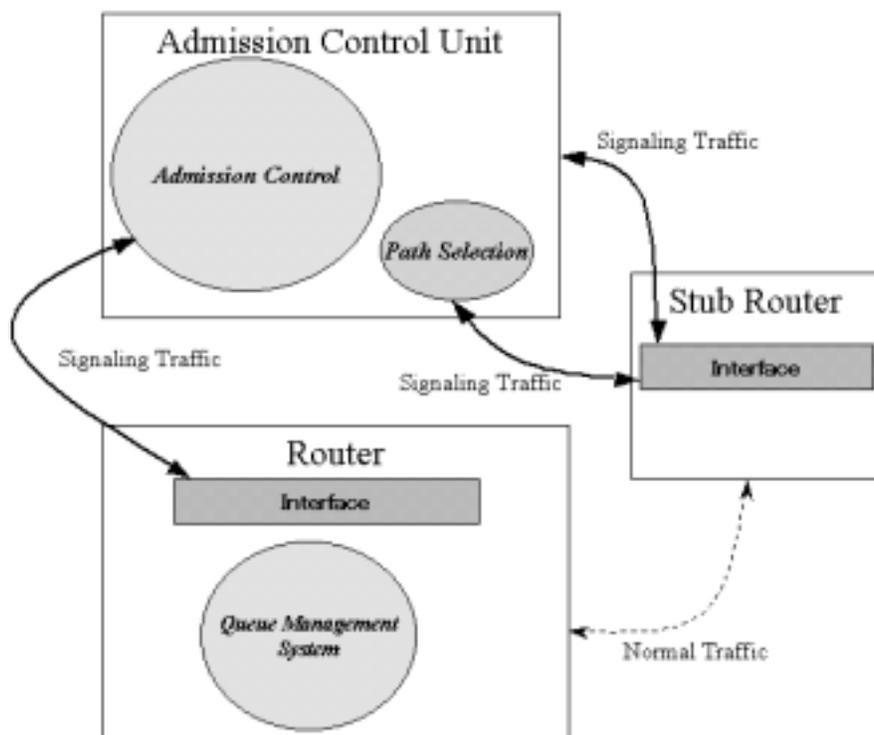


図 2.1: システムの構成図

図 2.1 の実線はシグナリングのトラフィックの流れを表している。点線は通常のトラフィックを表しており、四角はそれぞれ物理的な境界を表す。四角の中の楕円はそれぞれ機能を表している。

表し、四角の中の長方形は Admission Control Unit へのインターフェースを表す。

各部の機能の紹介をすると、図 2.1 の楕円の” Admission Control” は管理情報の収集、分類を行い、収集には中継ノードと管理制御ユニット間で通信を行う必要があるが、この部分もまとめて行う。” Path Selection” は” Admission Control” によって得られた情報を基に最初に後の章で説明する静的な要素の判断基準をもとに利用可能な経路を絞り、次に動的な要素の判断基準をもとに最終的な経路を決定し問い合わせ元のルータに応答を返す。

” Queue Management System” は管理制御ユニットと管理制御ユニットとの通信、及び管理制御ユニットと中継ノードや管理制御ユニットと問い合わせもとのノードとの間のシグナリングのトラヒックを通常のトラヒックの影響を出来るだけ受けないようにシグナリングのトラヒックの流量にあわせてたぶんだけ帯域を確保して通信を行えるような通信機構を提供する。

次に” Admission Control” と中継ノードとの通信について述べる。問い合わせ元のルータからのリクエストを受け取った管理制御ユニットは自分の管轄外の中継ノードについての情報を得るため、あらかじめ持っているデータを元にその中継ノードを管理している管理制御ユニットに問い合わせる。問い合わせを受けた管理制御ユニットは自分の管轄の中継ノードに問い合わせを行い、そのリクエストを受けた中継ノードは自らの情報を管理制御ユニットに返答し、返答を受けた管理制御ユニットはその情報を問い合わせもとの管理制御ユニットに送り返す。

各機能の具体的な詳細については後の章で述べることにする。

第 3 章

経路選択

3.1 従来の経路選択法

経路選択アルゴリズムについて説明する前に、その背景について簡単に説明する。多くのネットワークの内部トポロジーでは多くのポイント間で複数の経路が見つかる場合がある。伝統的な IP Forwarding の主な制限としてひとつの判断基準、可能な宛先ノードへの経路の最も短い経路のツリーというものがある。なぜなら、少し負荷のかかる代替経路というものは役に立たないと考えていたので、他の多くのネットワークの目的ノードに対して最も短い経路のツリー上に存在するルータには常に高い負荷がかかっていた。それらのルータは hot spot となり、たとえそのルータ自身は CQS(Classify, Queue and Schedule) アーキテクチャを持っていたとしても潜在的に十分なサービスの違いを提供するネットワークの能力に制限があった。[9]

hot spot のルータ上の平均負荷が上昇すると、ランダムなパケット損失と遅延の揺らぎが上昇する。従来はこの問題に対処するため、ルータをアップグレードして処理能力を向上させるか、もしくはトラヒックが代替経路に分裂するのを許可する追加的な Packet Forwarding 機構を使ってきた。[9]

確かに、安い、低中規模の 10Mbps もしくは 100Mbps といったような帯域幅をもつ回線しかないのならばこの方法は、導入しやすいという点で十分役に立つ。しかし、利用可能な技術を最大限に使ってなお、ルータが高い負荷平均を保っているのならばこの方法は役に立たない。[9]

そこで、本研究では伝統的な IP Packet Forwarding の主な制限をすべて取り払い、新たな視点から経路選択をすることにした。その一つは、複数の判断基準を持つことであり、もう一つは複数の経路をある程度の制限を付けつつ複数選択可能にすることである。

複数の判断基準をもつことはすでに前章で述べたのでこの章では複数の経路を選択可能にする事について述べ、まず最初にそれを可能にする技術を簡単に紹介する。

実際に明示的に経路を制御して Packet Forwarding する技術にはいくつかある。そのいくつかを下記に並べる。

- IP-in-IP tunneling
- MPLS(Multiprotocol Label Switching)
- Source Routing

”IP-in-IP tunneling”とは IP パケットの中に別の IP パケットをカプセル化して伝送する技術である。トンネルの送信側で、トンネルされるべき IP パケットは別の IP パケットのペイロードに置かれる。トンネルの終端の識別子は tunneling パケットの IP ヘッダの Destination フィールドで定義される。tunneling パケットは、通常の IP Forwarding 技術を使ってトンネルの終端に向かって伝送される。tunneling パケットがトンネルの終端に到着したとき、トンネルの終端はペイロードに格納されている IP パケットを抽出し、後は一般の Interface に到着したパケットと同様に処理する。

ですが、”IP-in-IP tunneling”は経路を変更するためにはトンネルの送信側の設定を手動で変えて、別のノードと接続しなければならず、なおかつ、トンネルの終端側のノード側の設定も変更しなければならず、本研究で、明示的に経路を制御する技術としては役に立たないと言える。

次に MPLS(Multiprotocol Label Switching) ですが、簡単に説明すると L₃ のヘッダの前に Label と呼ばれるヘッダをおき、LSR(Label Switching Router) が、通常のルータがその L₃ ヘッダ内から宛先に関する Entry を見つけて Routing Table 内から検索し、Forwarding するのと同様に、Label から宛先に関する Entry を取り出し、Forwarding する。Label Switching では Label 交換のもとになる一つの Forwarding アルゴリズムを使い、パケット内で運ばれる Label は短く、Forwarding とリソース予約の両方のセマンティクスをもった固定長の構造化されていない Entity をもつ。また、Label Switching Forwarding は L₃ から上の層に関係なく交換できるため、IPX であろうと、IPv4 であろうと IPv6 であろうと Appletalk であろうと同じコンポーネントで Switching でき、Packet Forwarding を L₃ から分離した形となる。[4]

次に”Source Routing”であるが、これは IP ヘッダ内に Source Route Option を送信側でつけて伝送する。Source Routing には 2 種類ある。

- Strict source routing

- Loose source routing

前者の Strict source routing は送信側が IP データグラムが伝送される正確な経路を設定し、もし、ルータの隣接ノードに次の Forwarding されるべきノードがなければ ICMP の”source route failed”のエラーメッセージを伝送する。後者の Loose source routing は送信側が IP データグラムが伝送されなければならないノードの IP アドレスのリストを決めるが、IP データグラムはリスト内の二つのアドレス間の他のルータを通過していくことができる。[1]

本研究では、IP を使うことを前提としているので目的が同じならば手段を問う必要性は低いので MPLS でも Source Routing でもどちらを使ってもかまわないのだが、導入のしやすさからいえば Source Routing で、将来性を考えるとどのプロトコルでも扱える MPLS となると考えられる。本研究では、導入のしやすさから Source Routing をするものとして話を進める。

3.2 経路選択アルゴリズム

3.2.1 計測要素と評価方法

本研究では計測要素となるものの一覧を以下のようにした。

- 静的な要素
 - Hop 数
 - 利用回線の帯域幅
- 動的な要素
 - 各ノードの平均サービス時間
 - 各リンクの利用可能帯域幅

上述の計測要素について、動的な要素には賞味期限があり、データベースにデータを蓄えていた場合動的要素はトラヒックの状態に依存して変化する値であるので、実際に使用する時にはすでに違う値に大きく変わっている可能性が非常に高いため、接続要求を出す度に要求を出すか、非常に頻繁に値を計測する必要があると考えられる。

静的な要素はトラヒックの状態に依存して値が変化しないものであるため、時間が経過しても値は変化していない可能性が高く、変化すると考えられるのは、ノードに障害が起こったときのみである。このため、この値の計測は、動的要素の更新間隔と比べると非常

に長い期間で更新してもかまわないため、管理制御ユニット側から定期的に Polling を行うのと同時に、障害の起こったノードがまだ通信可能であるならば、障害の起こったノードから管理情報ユニットに障害の報告を行うことで、十分状態の計測を行えると考えられる。

例えば下記の表のように値をそれぞれ静的要素と動的要素を得ていたとする。

表 3.1: 静的な要素

Node	Hop	bandwidth
SR2	4	7
SR2	3	4
SR3	4	7
SR3	5	8
SR3	5	10
SR3	6	11
SR4	6	7
SR4	5	8
SR4	6	9

表 3.2: 動的な要素

Node	q_usage
SR2	0.3
SR2	0.4
SR3	0.3
SR3	0.2
SR3	0.1
SR4	0.8
SR4	0.3
SR4	0.7

上記の場合、これは管理制御ユニットの Stub Router1 に対する判断基準の表なのだが、例えば、SR1 から SR2 へトラフィックを流したい場合、表 3.2 より二つの経路が存在するこ

とが分かる。まず最初に、図 3.1 に書いてあるとおり静的な要素に対してどちらか経路を決定する。次に、その経路の動的な要素について判別を行う。

3.2.2 経路を確定するためのアルゴリズム

前節で述べた計量要素を判断基準として次の図のように経路を決定する。

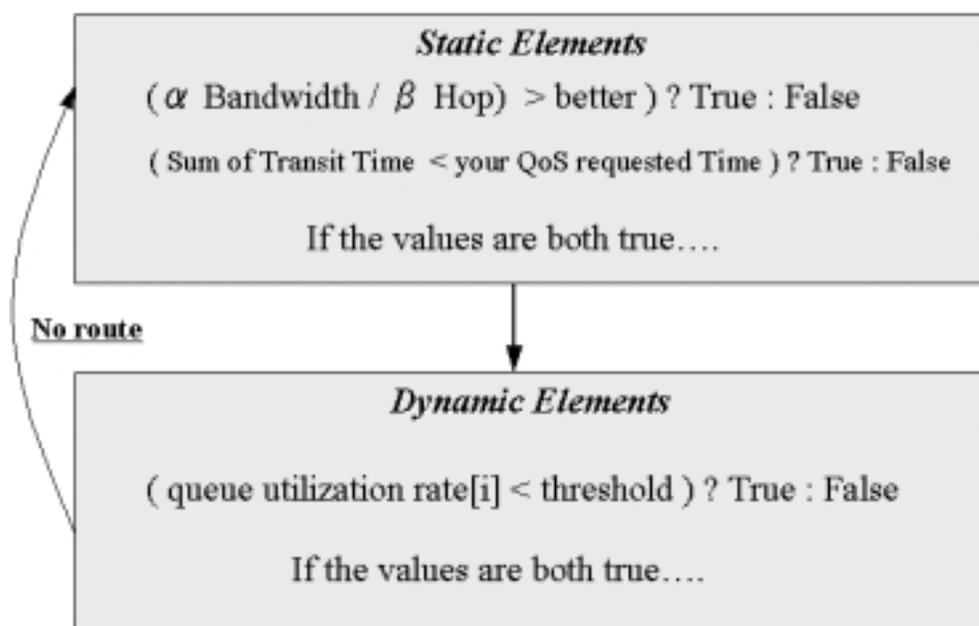


図 3.1: 経路選択アルゴリズム

まず、最初に比較するのは回線の伝搬遅延時間と宛先ホストの属する subnet router までの Hop 数である。

$$\alpha \times \frac{Bandwidth}{Hop} \quad (3.1)$$

ここで α は二つの判断基準である Bandwidth と Hop 数が同等の数的価値で評価できるようにどちらかが片方の値より圧倒的に大きい値である場合にそれを調整するためにしよう

する任意の実数値をとる係数である。上述の関係より、例えば、この場合は帯域幅の値も hop 数もほとんど桁的に同じ数であるので、 $\alpha = 1.0$ とすると表 4.1 よりそれぞれ、1.7、1.3 となり、表 3.3 の下から 6 番目の経路をとる方が、より快適な経路であると思われる。次に、転送遅延時間であるが、これは単純な目安として転送遅延時間の場合は、帯域幅の逆数の合計として比較してもよいので約 2.5 とし、要求している QoS より小さいならば True を返し、この経路について動的な要素を判断基準として要求にあうかどうか判断する。Core Router 及び Edge Router の使用率がすべて要求される QoS を達成するための閾値よりも低いかどうか調べて、すべて大丈夫ならば次の評価に移り、すべての経路の利用可能な帯域幅が要求される QoS を満たすかどうかを調べて大丈夫ならばこの経路に決定し、ここまででもし要求が満たされない経路であるならば、次の経路を同様に調べる。すべての到達可能な経路がすべて要求される QoS を満たさない場合、この接続要求は却下する。

3.2.3 静的要素による経路選択

まず最初に、なぜ静的な要素と動的な要素とあって、最初に静的な要素から評価したかの理由について述べる。これは、動的な要素、つまり中継ノードの使用率から評価した場合、たとえ回線が空いていたとしてもその経路がたくさんある経路の中で比較的帯域幅が広くて待ち行列の待ち時間が短くて QoS を満たせるかどうかということには直接結びつかないので、最初に経路を絞るのには適さないと考えられるためである。

次に各々の要素の判断基準の評価方法のアルゴリズムについて述べる。まず、図 3.2 より、最初に帯域幅と Hop 数を評価したがこれを最初に評価したのは、パケットがあちこち動く際に一番問題になる静的な要素の中で大きい割合を占める遅延時間で絞りこむほうが効率がいいのは明白だからである。

また、帯域幅と Hop 数を割ったのは Hop 数は少ないほど中継するノード数が減るので転送時間、待ち行列の待ち時間で遅延が増加する傾向が低くなり、帯域幅は広いほうがいいのでこの二つの値を割った値の大きい方は Hop 数が少なくて帯域幅が広い経路を選択する可能性が高くなるからである。

次に転送遅延時間と伝搬遅延時間の合計を取って要求される QoS を満たすかどうかを判別しているが、これは転送遅延時間に関してはどれくらいのパケット長であるかを事前に知ることは本研究の QoS アーキテクチャでは不可能であるので、基本単位として帯域幅の逆数を用いることにした。

また、転送遅延時間の合計から QoS を満たせるかどうかの評価を行ったのは、第 1 章で述べたように「実時間性」ということを目指すならば、時間で評価するのが最適であると考えられたからである。

3.2.4 動的要素による経路選択

動的要素の評価については、中継ノードの使用率をみて評価しているが、これは静的な要素から良いと思われる経路の状態を評価するのに使用している。なぜなら、中継ノードの使用率とはその経路の各回線の利用可能な帯域幅も同様に示しており、その経路の空き具合を調べるのに最適だと考えられるからである。

第 4 章

管理制御機構

管理制御機構を持つユニットを管理制御ユニット (Admission Control Unit) と呼ぶことにする。また、ここで本研究では管理制御とは何をすることを指すかということを確認にする。

本研究で管理制御する項目についてまとめると以下のようになる。

- 各中継ノードの使用率
- 物理的な回線の帯域幅
- Hop 数

各項目について採用の理由及び使用目的を説明する。

「各中継ノードの使用率」の採用の理由は、待ち行列の性質によるものが大きい。待ち行列はその長さに応じて使用率が 1 に限りなく近づく待ち行列の長さは無限大に発散する性質があり系内滞留時間が増す性質を持つ。本研究で基本となる M/M/1 の待ち行列のモデルを元にそのことについて簡単に説明する。

待ち行列理論において、Little の公式というものがある。

$$L = \lambda W \quad (4.1)$$

$$L_q = \lambda W_q \quad (4.2)$$

ここで、

$$L = \frac{\rho}{1 - \rho} \quad (4.3)$$
$$L_q = \sum_{n=1}^{\infty} (n-1)p_n = \sum_{n=1}^{\infty} np_n - \sum_{n=1}^{\infty} p_n$$

$$\begin{aligned}
&= L - (1 - p_0) = \frac{\rho}{1 - \rho} - \rho \\
&= \frac{\rho^2}{1 - \rho}
\end{aligned} \tag{4.4}$$

ここで、式 (3.2) の Little の公式より

$$\begin{aligned}
W_q &= \frac{1}{\lambda} L_q \\
&= \frac{1}{\lambda} \frac{\rho^2}{1 - \rho} \\
&= \frac{\lambda}{\mu(\mu - \lambda)} \\
&= \frac{\rho}{\mu - \lambda}
\end{aligned} \tag{4.5}$$

ここで、結果だけ述べると M/M/1 の待ち行列の場合の系内滞留パケット数の平均は

$$\frac{\rho}{1 - \rho} \tag{4.6}$$

となり、系内滞留時間の平均は

$$\frac{\rho}{\mu - \lambda} \tag{4.7}$$

となる。この式 (3.11) は待ち行列の典型的な性質をよく表しており、中継ノードの使用率 ρ が例えば、0.2 から 0.4 に増えた場合、待ち行列の長さは 0.25 から約 0.67 となり、使用率が 0.4 から 0.8 に増えた場合、待ち行列の長さは約 0.67 から 4 となり、前者は使用率が 2 倍になったのに対して待ち行列の長さは約 2.7 倍になり、それほど差はないが後者は、使用率が 2 倍になったのに対して約 6 倍にもなり、その差は急激に広がる。

このように使用率を無視してノード内に留まる時間（系内滞留時間）を制御しようとしても困難なことは明らかである。

このため、QoS 制御をする目的で中継ノードの使用率をその判断基準とするのは妥当であると言える。

次に利用可能な帯域幅であるが、実際に使用可能な帯域幅が分からなければいくら回線の物理的な帯域幅が 100Mbps でも混み合って使用率が例えば 50% ならば実際はそれぞれ 50Mbps しか使用できず、80Mbps の帯域幅が必要なものだった場合、十分な応答時間でサービスを提供出来ないからである。本研究ではそれぞれのフローがそれぞれ異なる細かい要求を求めてきて QoS 制御するところまでは残念ながら達していないが将来的には必要なものになりうると思われるが、これは中継ノードの使用率から計算すればでてる

値であるので本研究では利用はするが、使用率から計算して求めるものとする。

物理的な回線の帯域幅は待ち行列の使用率から具体的な帯域幅の値を計算するために必要とされるため、具体的な QoS の要求に対して答えるためにも必要な要素であると言え、判断基準として採用することは妥当であると言える。

Hop 数は、不必要に経路を迂回しないために必要であり、また中継するノードがすくなくなるほど各中継ノードでの転送時間が減少し、待ち時間も減少するため必要である。つまり、Hop 数で経由するノードの数がある程度制限しないと可能な限り広い帯域幅の回線を使う経路を選んでどこまでもトラヒックが拡散してその他のトラヒックの遅延時間に影響を与えることになって収拾がつかなくなる可能性もあり、また Hop 数に比例してそのトラヒックの到達ノードまでの転送時間が増加するため全体の遅延時間が延びる。このため可能な限り少ない Hop 数の経路を選ぶことでトラヒックの到達ノードまでの遅延時間は減少し、また他のトラヒックに影響を与えて遅延時間を増加させてしまう危険を最小限に押さえられると考えられるため Hop 数を判断基準として採用することは妥当であると言える。

4.1 管理制御ユニット間の通信機構

管理制御ユニット間の通信は大きく二つに分類される。一つは動的な要素の判断基準の取得に使われるもので、もう一つは静的な要素の判断基準の取得に使われる。ここで、動的な要素とはトラヒックの状態によって値が変化しやすいもので、静的な要素とはトラヒックの状態によって値が変化しないものを指す。

動的な要素はフローが発生する度に値を新たに取得する。これは、常に最新の情報でなくもし古い情報であるならばトラヒックの状態によって変化する要素はすでに違う値を持っている可能性が非常に高いため、実際にトラヒックが流れるときには QoS を満たすことができない可能性が高いと考えられるからである。逆に静的な要素はトラヒックの状態によって変化しない値であるので常に最新の情報を保つ必要性は非常に低い。なぜならば実際にトラヒックが流れるときにその値が変化している可能性が非常に低いからである。そのため、静的な要素は動的な要素に比べると非常に長い期間、有効性が保たれると考えられるのでなにか障害が起こったときにだけ情報が更新されればよいと考えられる。

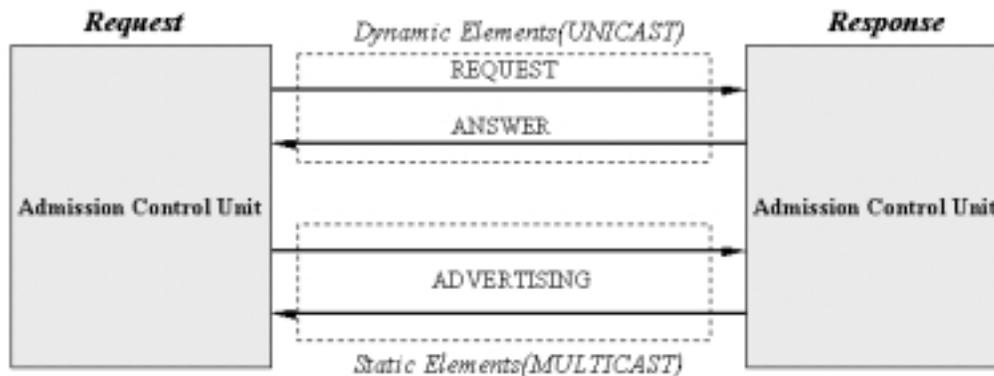


図 4.1: 管理制御ユニット間の通信

4.1.1 管理制御ユニットと Router との連携

図 4.2 が示すように、ここでも動的な要素の情報をルータから得るか、静的な要素の情報をルータから得るかで振る舞いが違う。図 4.2 の上側の点線で囲まれた部分は動的な要素をルータから得る手順を示しているが、これはまず最初に管理制御ユニットの方からルータに問い合わせがあって、初めてルータ側は自らの情報を収集し管理制御ユニット側に応答を返している。

次に、図??の下側の点線で囲まれた部分であるが、これは静的な要素をルータから得る手順を表しているが、Polling とは管理制御側が動的要素の問い合わせ間隔時間に比べると非常に長い期間の間隔をおいて管理制御ユニット側から異常がないかどうか確認するための手順である。その下側の Emergency とはルータ側が自らに障害が発生したとき、まだ通信可能であるならばその障害を管理制御側に通知するための手順である。この二つの手順を組み合わせることでルータ側に障害が発生した場合の検知がどちらか一方を採用するより検知できる。

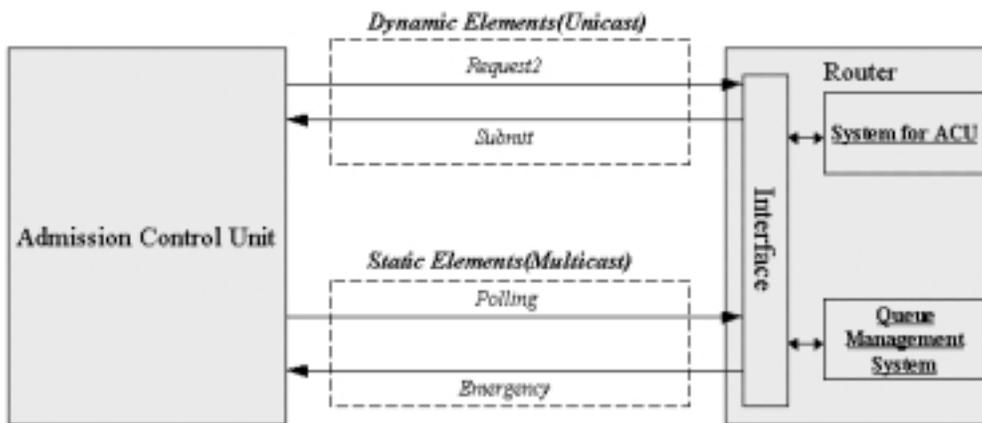


図 4.2: 管理制御ユニットと Router との連携

もし、他の管理制御ユニットまでの経路が一つしかなくて、故障した中継ノードを監視する管理制御ユニットが他の管理制御ユニットに直接中継ノードの障害を告知出来ない場合、その故障した中継ノードを通るトラヒックを流したい他の管理制御ユニットは故障した中継ノードの一つ手前の中継ノードから ICMP の network/host unreachable のエラーメッセージが返されるため、故障したノードの一つ手前のノードまでトラヒックが流れてしまうが、これを他の管理制御ユニットへ告知することでそれ以後の故障したノードを通る経路の先のノードへのトラヒックを遮断することが可能になる。

また、トラヒック状態に応じて動的要素の情報を得るモデルを次に述べる。

4.2 管理制御モデル -フロー毎の制御-

フロー毎の制御とはフロー毎にすべて管理制御ユニットに問い合わせを行って QoS を満たすことが出来るかどうかを行う QoS 制御である。この制御の特徴はすべてのフローにおいて QoS 制御できるが、反対にトラヒックが混み合っているときなどは管理制御ユニットにかかる負荷が大きくなりすぎてしまうという欠点を持つ。

4.3 管理制御モデル -状態に応じた一括制御-

状態に応じた一括制御とは第 3 章 2 節で紹介したフロー毎の制御とは対照的で、フロー毎に管理制御ユニット側に問い合わせる必要がない。つまり、管理制御ユニット側が定期的を取得してくる管理情報を元に変動する割合を元に通すフローと通さないフローを分ける。これは、フロー毎に管理情報を取得する必要がないため処理は単純になり、トラヒックが混み合っているときなどに管理制御ユニット側の負荷を和らげる効果をもつが、一方でたとえトラヒックが混み合っていなくても必ずある割合で通信できない場合が発生する。

第 5 章

Queue Management

待ち行列を持つルータからなるネットワークは、いくつもの待ち行列が連なったものだと考えられる。

これは、例えば最初にあるノードにパケットが到着したとして、そのノードから送出され、そのパケットがまた別のノードに到着したそのノードから去り次のノードに移る。このようにパケットがノード間のノードからノードへ次々と伝送されていくものを指す。しかし、必ずしも次々とノードを移っていく訳でもなく、同じノードに出たり入ったりする場合もあるし、同じノードを再び訪れるということもあり得る。本研究では以下のような特徴をもつものであるとしている。

- やってくるパケットの到着間隔時間の分布は負の指数分布に従う。
- サービス時間の分布も負の指数分布に従うが、パケットの到着間隔よりは短い。
- 各々のノードでサービスを完了したパケットが次のノードへいく確率は1。

5.1 Queuing モデル

本研究では、各々の中継ノードはパケットを二つのクラスに分けて待ち行列に格納する。それぞれの待ち行列は通常のパケットを格納する待ち行列とシグナリング用のパケットを格納する待ち行列とに分ける。それぞれの待ち行列は以下の図のようなスケジューリングを行い、各々の中継ノードから送出される。

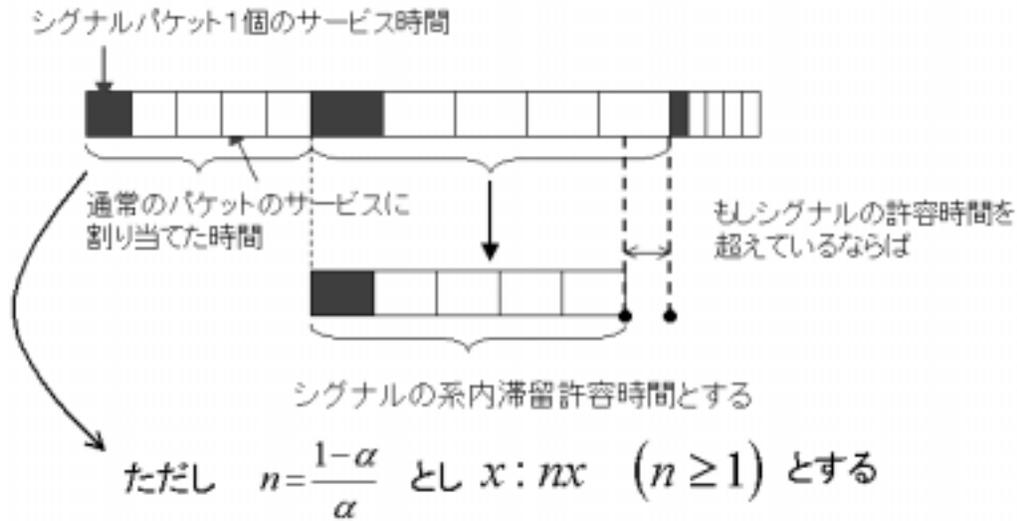


図 5.1: スケジューリングの概要

図 5.1 の長方形の横の長さは時間を表し、長方形全体の横の長さ是一个のシグナルの packets のサービス時間と一个のシグナルの packets のサービス時間に対して通常の packets に割り当てる時間の合計を表している。同図の α というのはトラフィック全体の packets の中に占めるシグナリングのトラフィックの packets の割合を示す。これは、最初、0 ではない例えば 0.3 という値を与えておいて、そこから packets が到着する度に α の平均値を下式のようにして取得し、スケジューリングする際に使用する。

$$\alpha_{avg} = 0.99 \times \alpha_{avg} + 0.01 \times \alpha_{tmp} \quad (5.1)$$

次に図 5.1 のように時間を割り当てるアルゴリズムについて述べる。

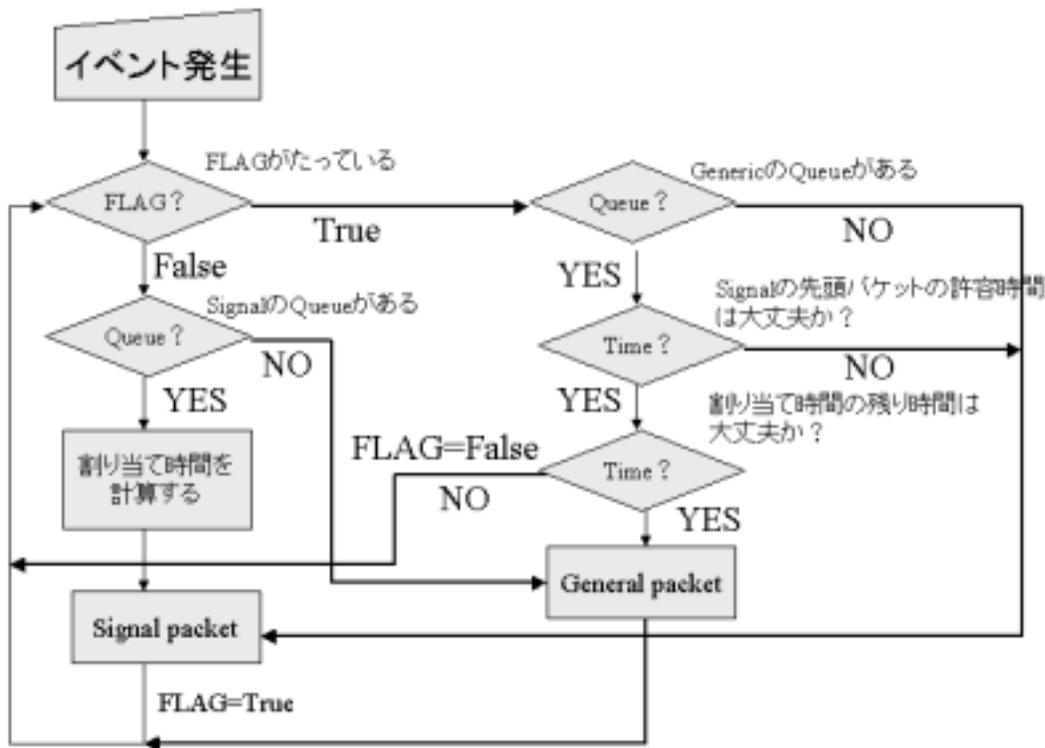


図 5.2: スケジューリングアルゴリズム

図 5.2 は目的としては、シグナリングのパケットの排出と通常のパケットの排出とをどちらもある程度制限しながらほんの少しだけ、シグナリングのトラヒックが優先されるようにした。なぜ、このような手法を取ったかというどちらかを優先してしまうと、必ず片方の待ち行列に入っているパケットの排出が滞ってしまい、結果、優先している方の待ち行列の待ち時間に比べて、優先していない方の待ち時間が、大きいときには10倍以上違ってしまふということになってしまい、性能の悪いスケジューリングとなってしまい、実際には使いものにならずにうまくいかないためである。

第 6 章

実験結果

6.1 Queue Management

6.1.1 シミュレーション手法

シミュレーション手法としては、図 6.6 のようなネットワークを想定し、各々の中継ノードに上述したスケジューリングアルゴリズム（以下、Queue Management System とする）を載せてそれぞれの中継ノードの待ち行列の長さを測定した。

また、負の指数分布にそった乱数を発生する関数には以下にものを使った。

```
int exponential_dist(float base, ushort_t *seed, float scale) {  
    return ( (-log( erand48(seed) ) / base ) * scale + 5.0 ) / 10;  
}
```

base は平均を表し、*seed* は乱数のシードである。乱数は [0:1] の範囲の一様分布の乱数を発生する。詳しくは `erand48(3C)` を参照してもらいたい。*scale* は発生する負の指数分布の乱数の桁を調節するため使用した。

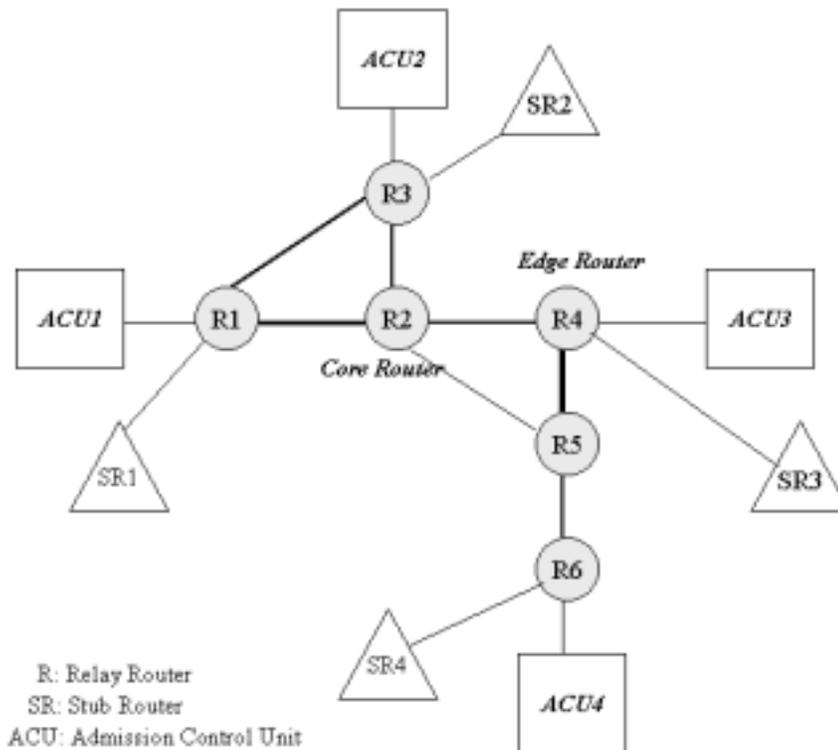


図 6.1: 想定した接続図

6.1.2 目的

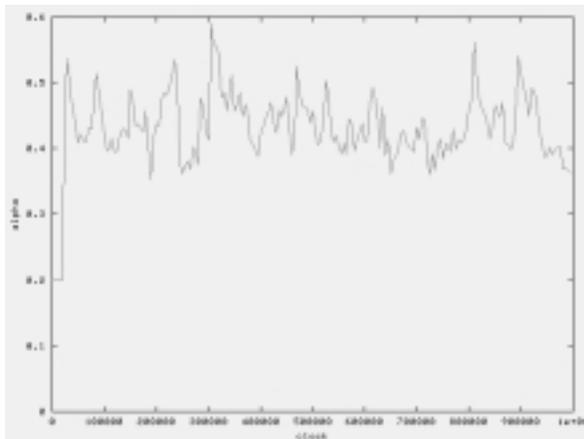
本研究で提案した Queue Management System についてシミュレーションを行った結果をまとめ、この方式の有効性を確かめる。

評価目標としては以下の点を上げた。

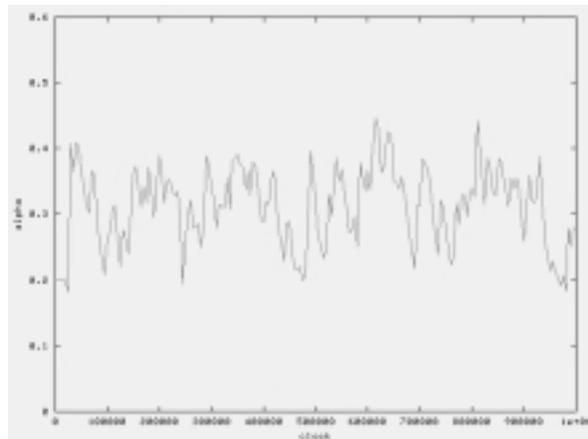
1. 全体のトラフィックに占めるシグナリングのトラフィックの割合である 値によって通常の packets に割り当てる時間を決めているがトラフィックの状態を反映した制御が出来ているか？
2. シグナリングの packets を優先しているが通常のトラフィックの packets のサービスが犠牲にされすぎているかどうか？
3. 提案した Queue Management System で目標としているおおよそ規定された時間内にシグナリングの packets をサービスできているか？

6.1.3 結果と考察

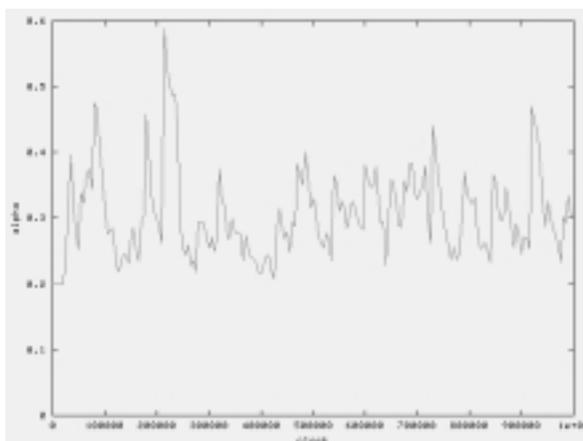
次に評価目標 1 についての有効性の確認をする。 はシグナリングのトラヒックの割合を表していることを前提としているのであるが、例えば図 6.2(e) をみると 200000clock 付近で急激にシグナリングのトラヒックが瞬間的に増加しているが図 6.10(e) をみても同様に 200000clock 付近で増加している。他の図を比べてみてもほとんどそのトラヒックの増減に対して追尾された動きを 値は示していることが分かり、本研究で 値はシグナリングのトラヒックの割合を示す信頼性の高い値であることが分かり、本研究で提案している Queue Management System で行われている通常のトラヒックに割り当てるサービスの割り当ても理論通りにうまく機能していることが分かり、その有効性が示された。



(a) R1(A)



(b) R2(A)

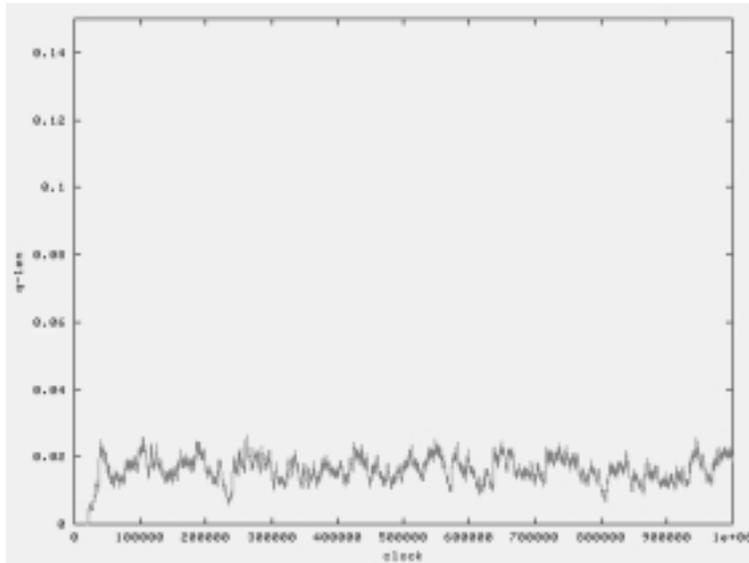


(c) R3(A)

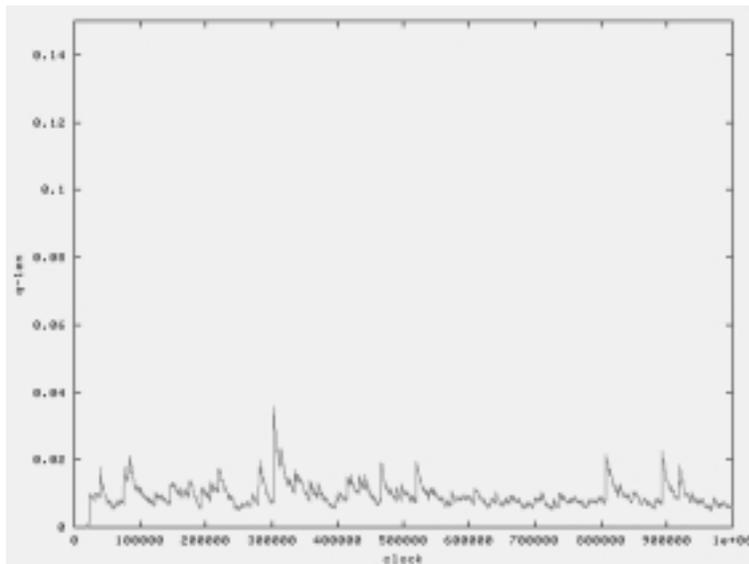
図 6.2: 各中継ノードの α 値の遷移

次に、評価2について確かめる。図 6.3 は R1 のシグナリング側の待ち行列の長さと同様のパケットを格納する側の待ち行列の長さを表している。

図 6.3 に示されているように二つの待ち行列のサービスが各中継ノードで行われている。



(a) 待ち行列の長さ (Normal)



(b) 待ち行列の長さ (Signal)

図 6.3: R1 における二つの待ち行列の長さ

二つの待ち行列は図 6.3 から、それぞれ待ち行列の長さが定常状態に至っており Queue Management System が期待通りに動作していることが分かるが、待ち行列を本研究で提案した Queue Management System を用いることで の値によってシグナリングの待ち行列の方を優先しているが、そのことによって通常の packets を格納する待ち行列の方のサービスが滞っていないことは図 6.4 の SR1 から SR2 の通常のトラヒックの遅延時間を見ることで達成されていると言える。

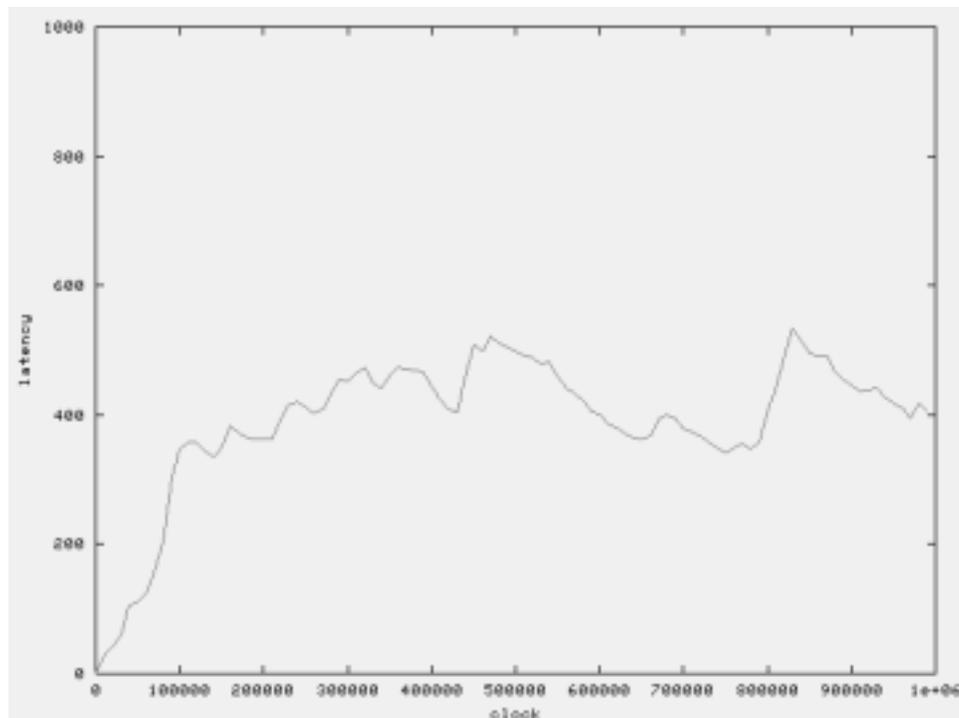


図 6.4: SR1 から SR2 への通常のトラヒックの遅延時間

次に、評価目標 3 についてシグナリングの packets の遅延時間について図 6.7 を参照していただきたい。また、ここでは第 6 章 2 節で紹介した場合について述べるため、各中継ノードの状態については図 6.10 から図 6.13 までを参照していただきたい。

なお、第 5 章で述べた Scheduling 方法でこの Queue Management System は各待ち行列に対してサービスを行うが、その際、その時点までの全体のトラヒックに占めるシグナリングのトラヒックの割合について調べている。その値を元にシグナルの packet 一個のサービス時間に対して、通常のトラヒックの方の packet のサービスに割り当てる時間を決定しているが、実はこの方式がうまくいかない場合というのは の値が 0 になってし

まった場合である。そうならないように最初にあらかじめある程度の値を与えておいて、最初にシグナリングの packets がいない場合にもある程度の時間は耐えられるようにしていた。もっとも τ が 0 になった場合の条件分岐において、0 ならば通常の packets の方に割り当てる時間をシグナリングの許容時間全体で割り当てることは可能である。本研究で提案した Scheduling アルゴリズムを使用すればたとえ、シグナリングの許容時間いっぱい割り当ててもシグナリングのトラヒックが来れば、制限時間内にサービスできるはずである。

まず、最初にシグナリングの packets のサービスをある程度優先していることは例えば図 6.10(b) と図 6.11(b) を参照するとよく分かる。シグナリングの packets を格納する待ち行列の長さや通常の packets を格納する待ち行列の長さを比べると明らかにシグナリングの packets を格納する待ち行列の長さの方が短い。これはシグナリングの方の packets のサービスの方が優先して行われていることを示している。また、本研究で提案した Queue Management System においてできるだけ均等にシグナリングの packets のサービスと通常の packets のサービスとが行われるようにしているが、これが実際にうまく機能していることは、図 6.10(c) と図 6.11(c) をみるとふたつの待ち行列の長さの比がほぼ一定であることから期待通りに機能していると言える。

また、図 6.7(a) を参照すると、ある程度混み合っている状態であっても packets のサービス時間が図 6.5 で示すように数十 clock であることを考えると、十分な値であると言える。

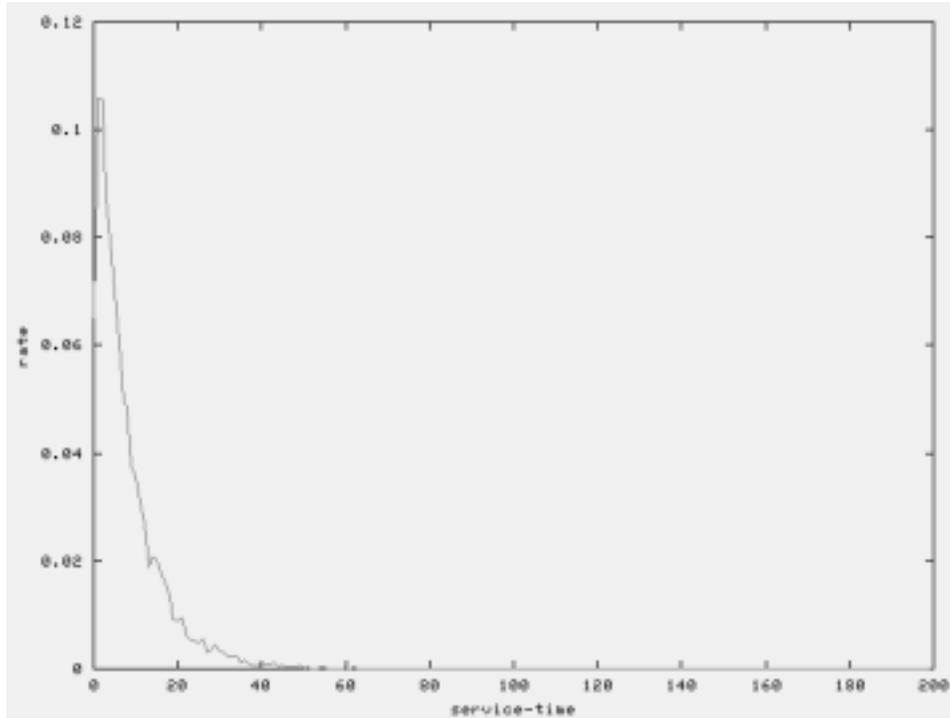


図 6.5: 管理制御ユニット1から発生するパケットのサービス時間の分布

6.2 管理制御機構

6.2.1 シミュレーション手法

本研究では次のようなネットワーク構成でシミュレーションを行った。

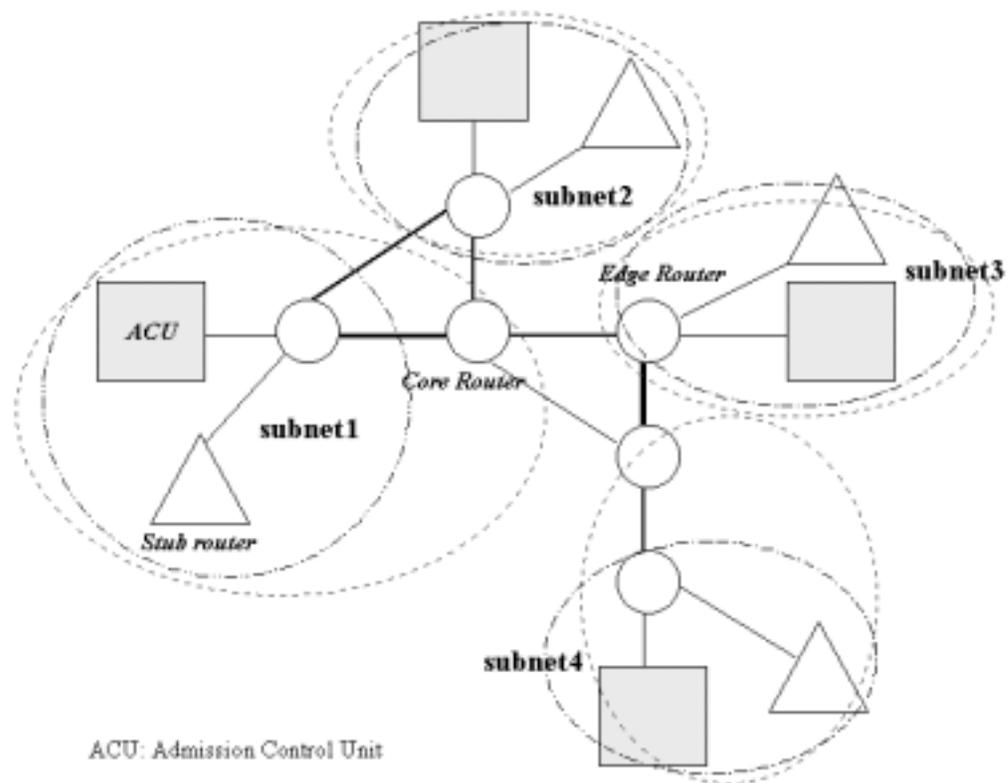


図 6.6: 想定した接続図

点線の楕円はそれぞれの管理制御ユニットが管理する管轄の範囲を表しており、長鎖線の楕円はこのイントラネットの subnet を表しており、それぞれ subnet[1-4] と呼ぶことにする。それぞれ図形を結んでいる実線は回線を表しており、太さは帯域幅を表している。

各々の stub router から管理制御ユニットに対して、接続要求を出し、その要求に従って管理制御ユニットは自分の管理していない中継ノードを管理している管理制御ユニットに問い合わせる。問い合わせを受けた管理制御ユニットが各々の中継ルータに問い合わせてその応答を問い合わせもとの管理制御ユニットに返すまでの遅延時間を測定した。

測定法方は stub router からの問い合わせを受けた最初の時間でタイムスタンプを押し、他の管理制御ユニットに問い合わせをするパケットにその情報を持たせ、それ以降のすべての処理でそのタイムスタンプの値を維持し、そのパケットが返った時間と比較し遅延時間を算出した。

6.2.2 目的

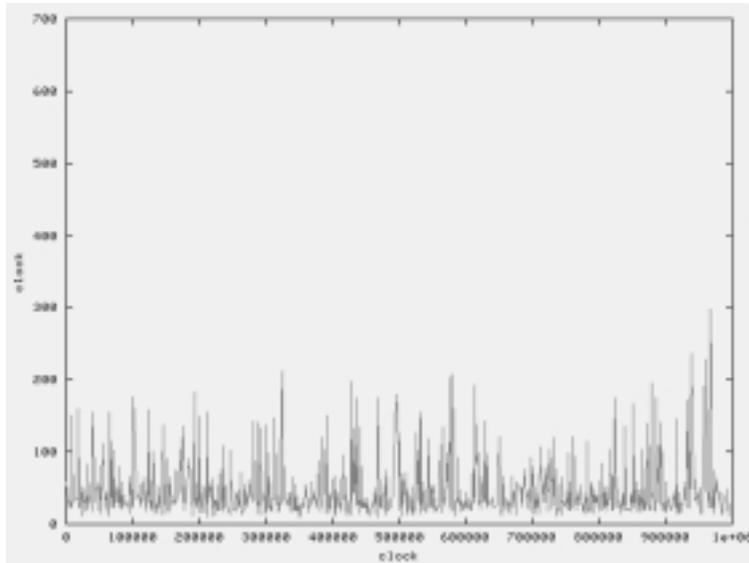
本研究で提案した管理制御における総合的な遅延時間についてのシミュレーションを行った結果をまとめ、この方式の有効性を確かめる。

評価目標としては以下の点を挙げた。

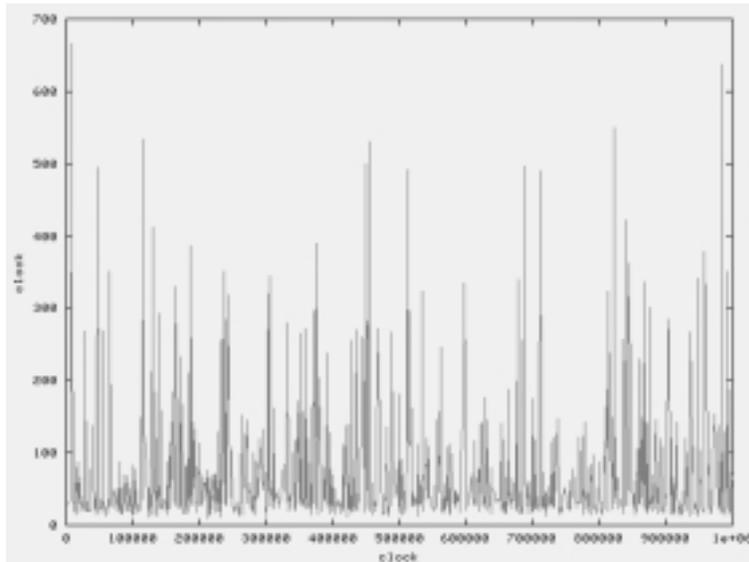
1. Stub Router からの問い合わせがあった場合の、応答時間がどれくらいの遅延で済むのか？
2. 各管理制御ユニットが自らの管理する Router の障害等があった場合の他の各管理制御ユニットへの告知にどれくらいの遅延が発生するのか？

6.2.3 結果と考察

評価 1 について、Stub Router からの問い合わせに対する応答時間の結果を下図に示す。



(a) ACU1 から ACU2 へのシグナルのパケットの遅延時間



(b) ACU4 から ACU3 へのシグナルのパケットの遅延時間

図 6.7: Stub router の問い合わせの応答時間

まず、図 6.7(a)、図 6.7(b) の両図を見比べて、平均遅延時間は多少異なるがこれは図 6.6 を見てもらえれば分かるのだが管理制御ユニット 1 から各管理制御ユニットへの HOP 数はそれぞれ管理制御ユニット 2、3、4 に対して、4、4、6 であるのに対して、管理制御ユニット 4 の方の各管理制御ユニット 1、2、3 までの HOP 数は 6、6、4 となり、少々ネットワーク的に遠い位置にあるためであると考えられる。

応答時間であるので第 3 章で述べたように、Stub Router から要求を受けたときに管理制御ユニットが各管理制御ユニットに対して問い合わせをし、問い合わせを受けた管理制御ユニットが各中継ノードに問い合わせをして中継ノードからの応答があったらその結果をもとの管理制御ユニットにたいして返すというものであるため、そのすべての遅延時間を合計した結果であるのに、図 6.7(a)、図 6.7(b) の両図を見て分かるようにそれほど時間はかかっていないように思われる。

6.3 経路選択

6.3.1 シミュレーション手法

図 6.6 と同様のネットワーク構成でシミュレーションを行った。

ここで、各々の経路に関する静的要素による判断基準の値を明らかにし、図 6.9 で示される経路を取ることを示す。

表 6.1: 判断基準の一覧

経路	帯域幅	Hop 数	B/H	Transit Time
SR1-R1-R2-R3-SR2	7	4	1.75	2.83
SR1-R1-R3-SR2	4	3	1.33	2.50
SR3-R4-R5-R6-SR4	8	4	2.00	2.75
SR3-R4-R2-R5-R6-SR4	7	5	1.40	4.00

よって表 6.1 から図 6.9 で示される経路は妥当である。

各々の subnetwork に属する stub router からそれぞれ同じ subnetwork に属する管理制

御ユニットに対して接続要求を行い、要求を受けた管理制御ユニットは中継ノードを管轄する各管理制御ユニットに問い合わせを行いその応答から得られた動的な要素の情報とすでに持っている静的な要素の情報とから経路を割り振り、通常のトラヒックを要求を出した stub router から宛先ノードの属する stub router に対して伝送する。それぞれの stub router に到着したとき、Source 側の stub router からどれくらいの時間で到着したかをパケット毎に記録し評価する。また、動的な要素つまり中継ノードの使用率は毎クロック測定し、下記に示すような式で平均を割り出し、必要なときにその値を抽出する。

$$U = (-L_q + \text{sqrt}(L_q^2 + 4L_q))/2 \quad (6.1)$$

また、それぞれの各回線の帯域幅は下図のように定めた。

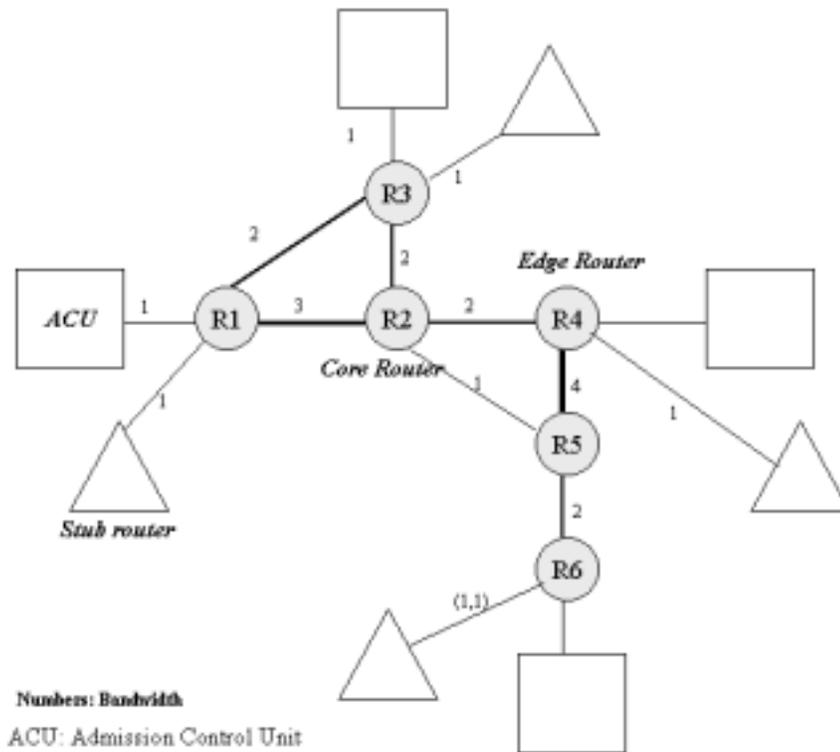


図 6.8: 各回線の帯域幅

また、図 6.8 の各パラメータの一覧表を載せる。

6.3.2 前提条件と目的

本研究で提案した経路選択による QoS 制御において、シミュレーションによってトラフィック分散を行ったが、シミュレーションでは実際の管理制御は行わず、管理制御ユニットにおいて本研究で提案している経路選択アルゴリズムを使って経路を設定した結果、どのように中継ノードの状態が変化したかの結果をまとめ、その有効性について考察する。

また、この実験における評価目標を下記に述べる。

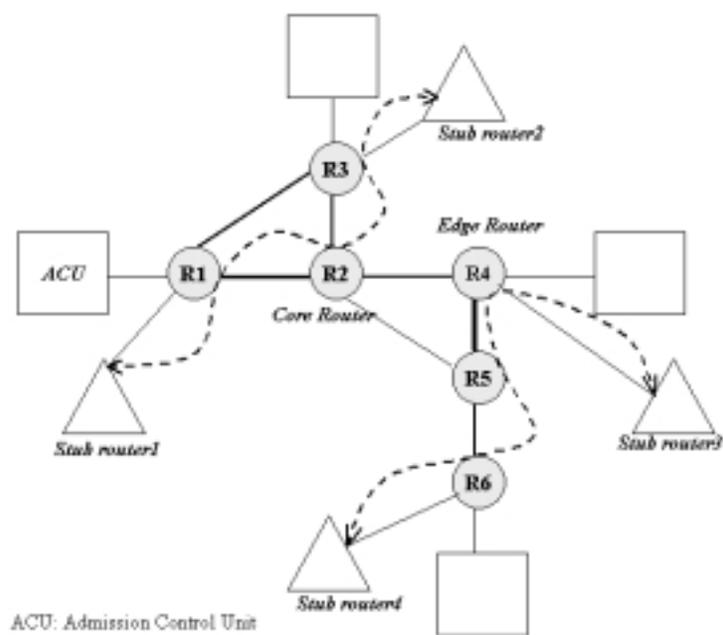
1. トラフィック分散を行った結果、各々の中継ノードの状態はどのように変化したか？

表 6.2: 帯域幅の一覧

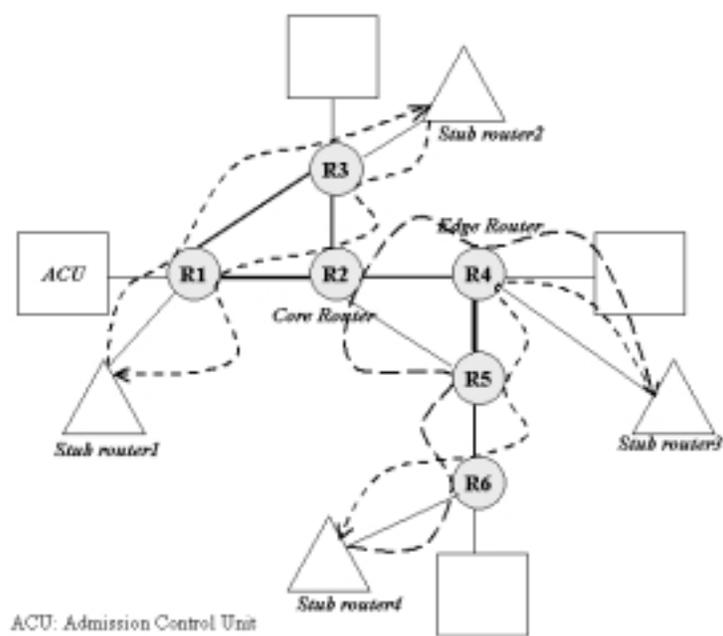
Node	bandwidth
ACU1-R1	1
ACU2-R3	1
ACU3-R4	1
ACU4-R6	1
SR1-R1	1
SR2-R3	1
SR3-R4	1
SR4-R6	1
R1-R2	3
R1-R3	2
R2-R3	2
R2-R4	2
R2-R5	1
R4-R5	4
R5-R6	2

6.3.3 結果と考察

まず、シミュレーションでを使用したネットワーク構成について示すが、A は経路選択アルゴリズムによって経路選択をしなかった場合の経路の場合のトラヒックの流れを表し、B は経路選択アルゴリズムによって、経路選択をした結果、トラヒック分散を行った場合のトラヒックの流れを表すとする。



(a) トラフィックの流れ (A)



(b) トラフィックの流れ (B)

図 6.9: ネットワークの接続図とトラフィックの流れ

また、Bの場合の経路を選択するに至った過程を第4章で紹介した経路選択アルゴリズムを使って導出してみる。

まず、Stub Router1側からStub router2へ至る経路とStub Router2からStub Router1へ至る経路の判断基準の表を静的な要素について表にまとめる。ただし、ここでは双方とも同じ表を参照することになるのでNodeをSR(Stub Router)としている。

表 6.3:

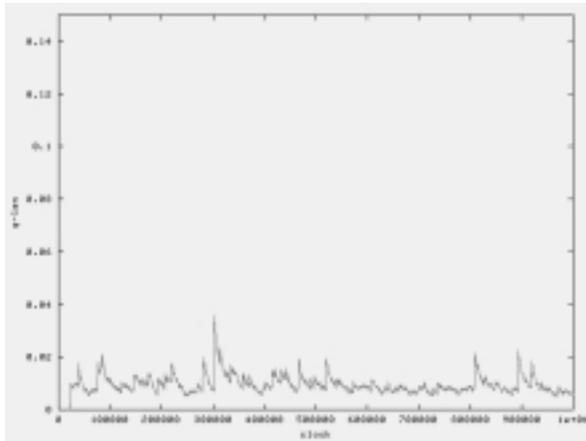
Node	Hop	bandwidth
SR1-SR2	3	4
SR2-SR1	4	7

まず、静的な要素の判断基準から経路を絞る。上の表を参照し、各々の経路について式(3.1)を用いて計算をする。ここでは簡単のため、 $\alpha = 1.0$ であるとする。それぞれの計算結果はそれぞれ約1.3と約1.75となる。次に各回線の帯域幅の逆数を取り、それぞれ2.5、2.8となり、とりあえず要求されたQoSを満たしているとする。と転送時間の短いことが期待されるR1-R3の回線を通る経路について評価をする。

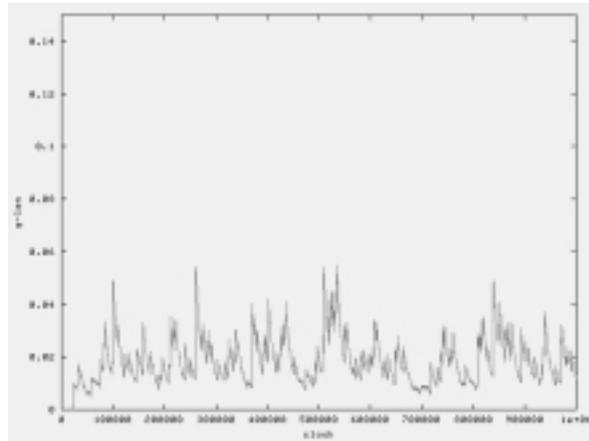
このため、図6.9(b)で使っている経路が選ばれ、次に各々の中継ノードの使用率を確かめることになるが、ここではどちらにしる、経路を分散した場合は必ず図6.9(b)で示される経路しか取りえないため、同じトラヒックで最初からトラヒック分散をしなかった場合とした場合とを比較し、その有効性について考察する。

次に各々のトラヒックの流れにおける中継ノードの状態について考察する。以下に各中継ノードにおける待ち行列の長さ及び使用率の結果の示し、さらに本研究で使用する全体のトラヒックに占めるシグナリングのトラヒックの割合を示すの値の遷移について各ノード毎に示す。

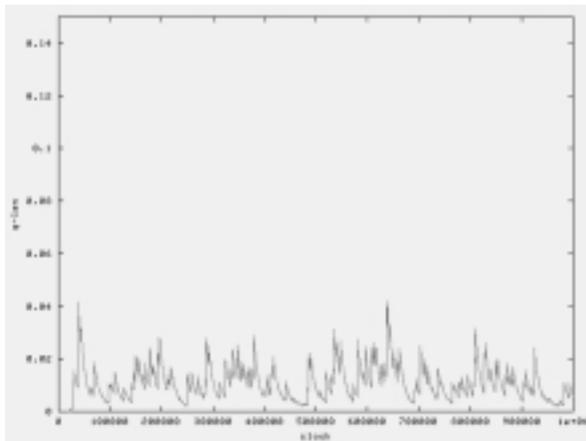
そこで、ここでは図6.9にあるStub Router1とStub Router2の間のトラヒックにおける各中継ノードの状態について取り上げる。各図の見出しの()内のA,Bは図6.9で示されているトラヒックのパターンを示している。



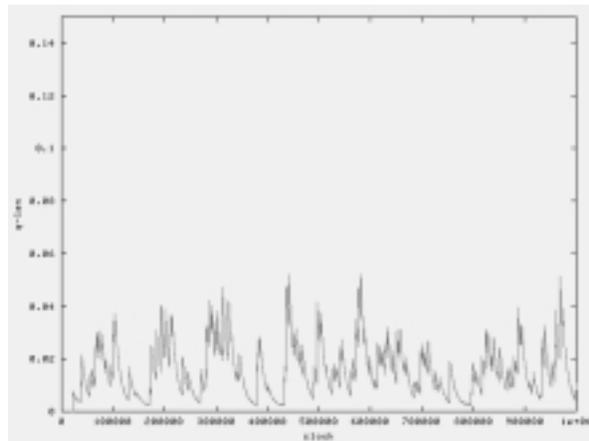
(a) R1(A)



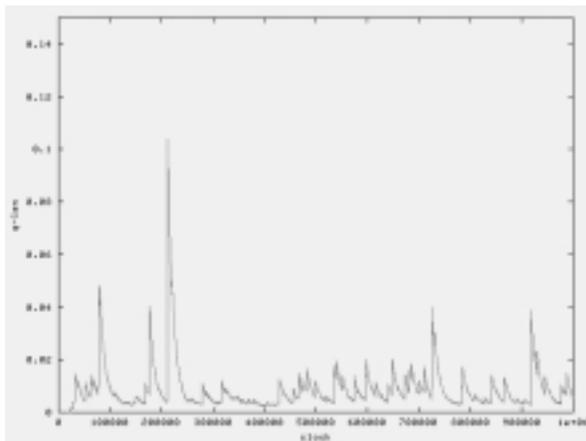
(b) R1(B)



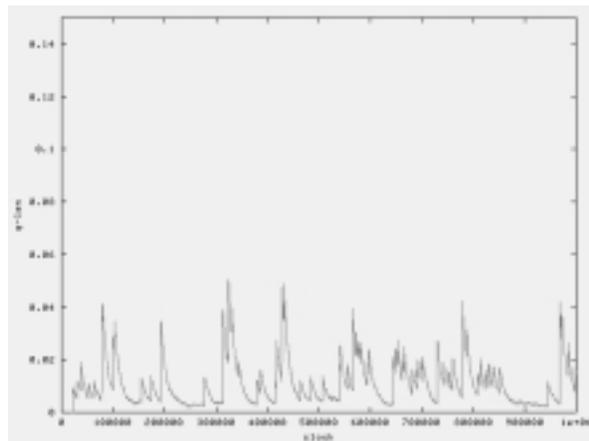
(c) R2(A)



(d) R2(B)

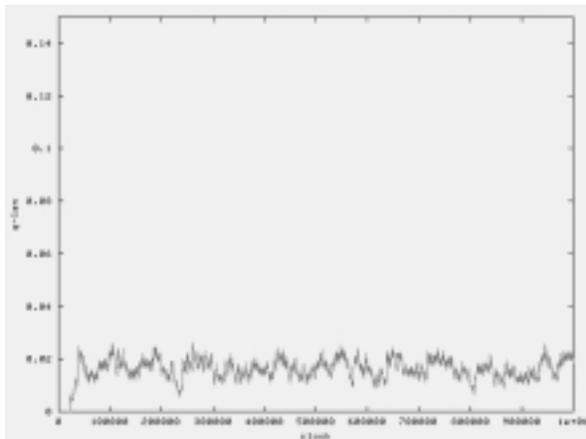


(e) R3(A)

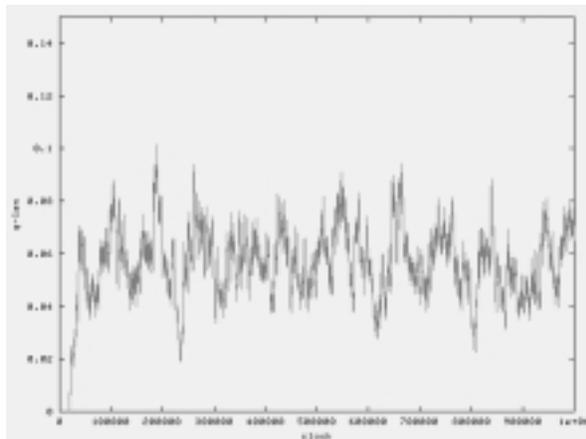


(f) R3(B)

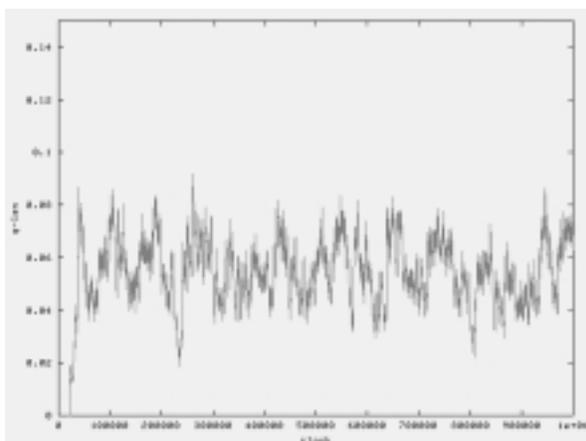
図 6.10: 各中継ノードの待ち行列の長さ (Signal)



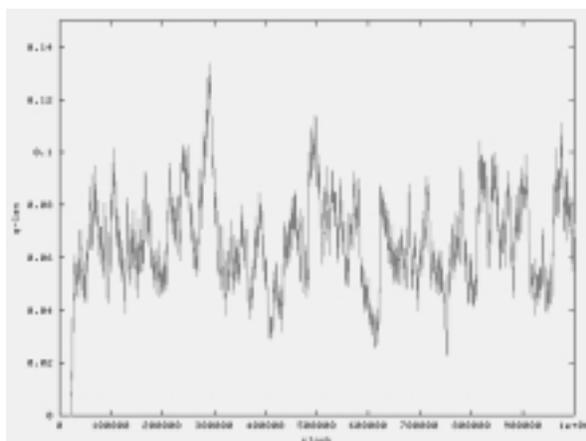
(a) R1(A)



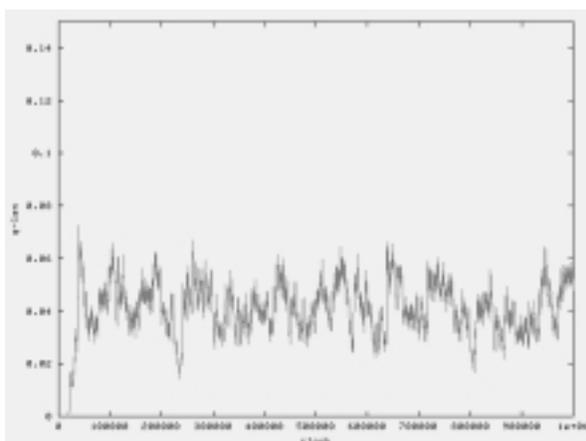
(b) R1(B)



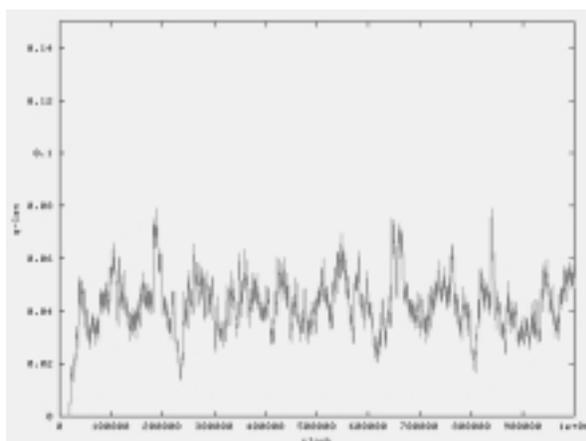
(c) R2(A)



(d) R2(B)

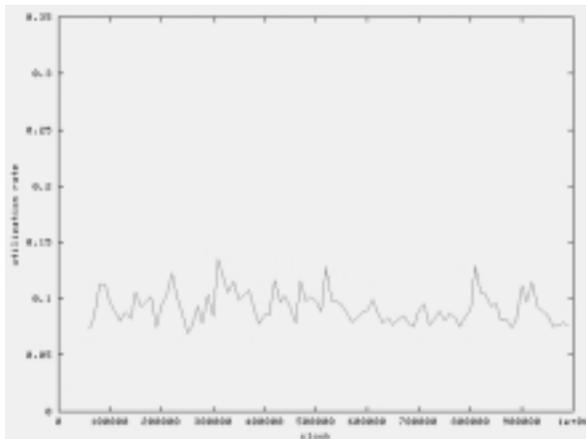


(e) R3(A)

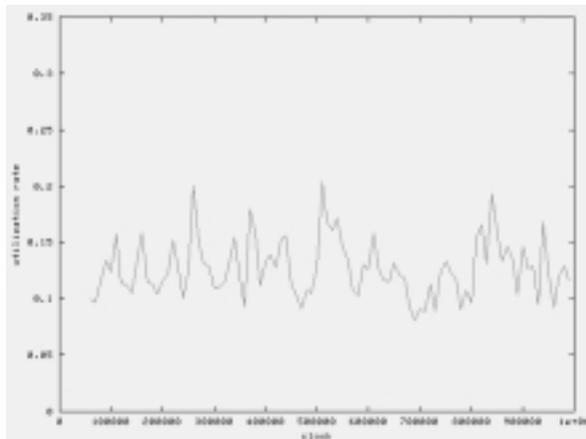


(f) R3(B)

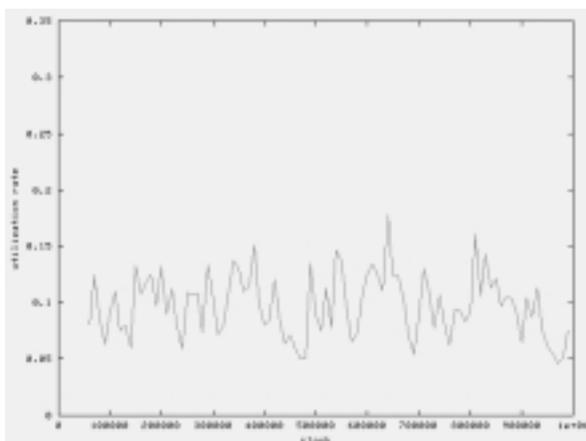
図 6.11: 各中継ノードの待ち行列の長さ (Normal)



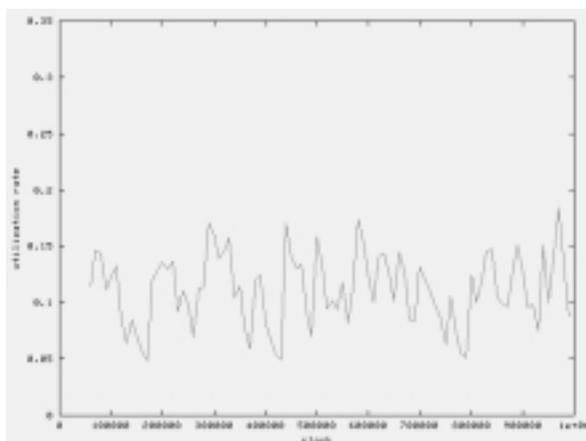
(a) R1(A)



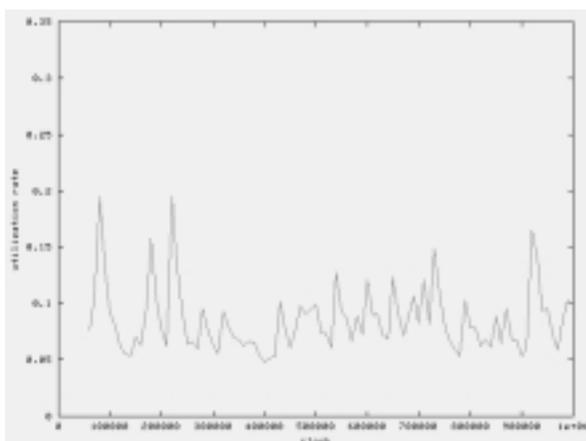
(b) R1(B)



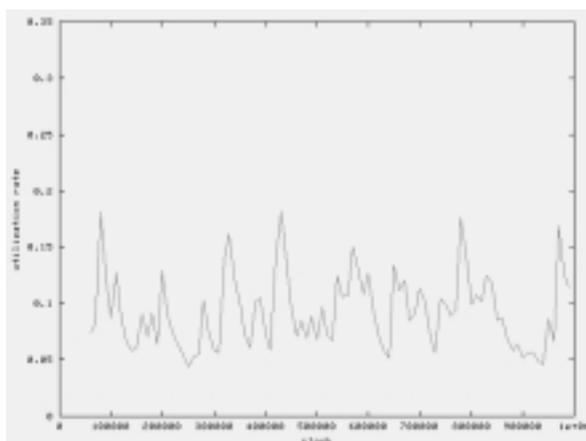
(c) R2(A)



(d) R2(B)

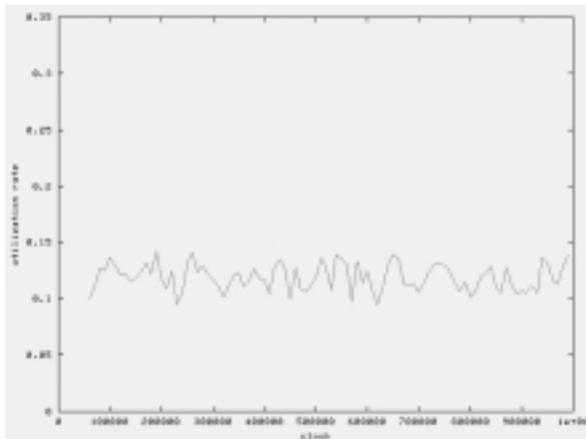


(e) R3(A)

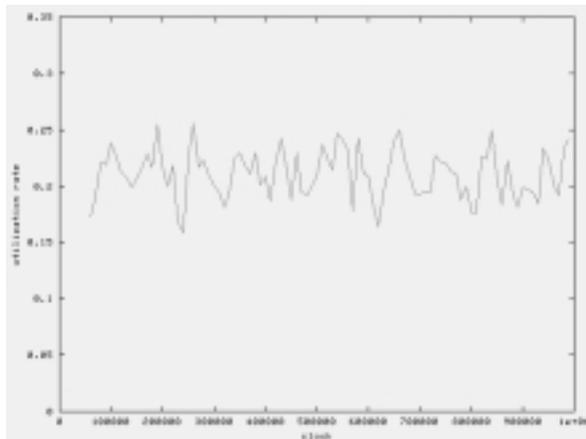


(f) R3(B)

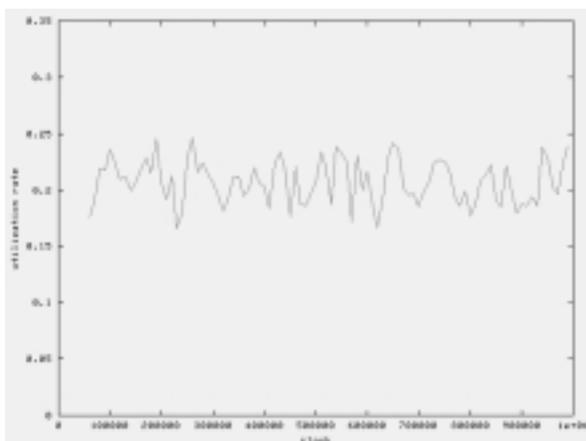
図 6.12: 各中継ノードの使用率 (Signal)



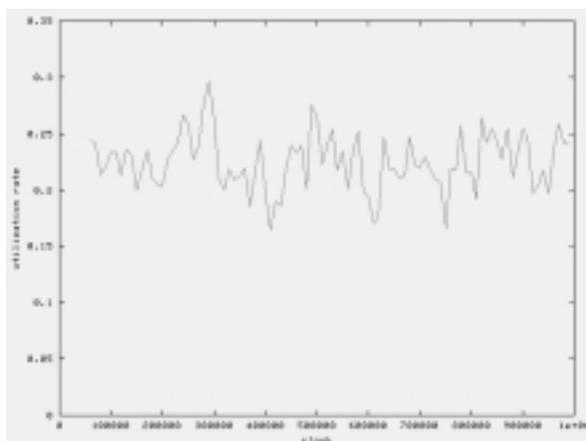
(a) R1(A)



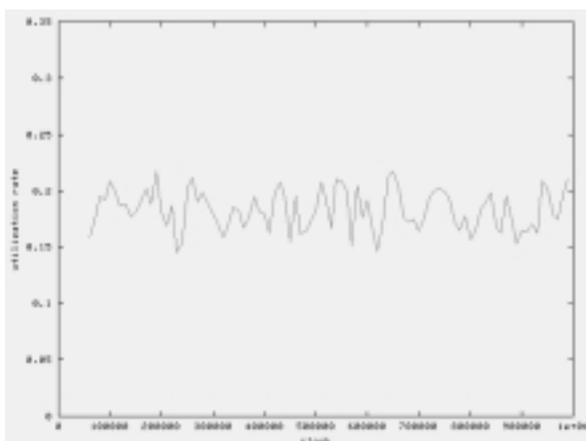
(b) R1(B)



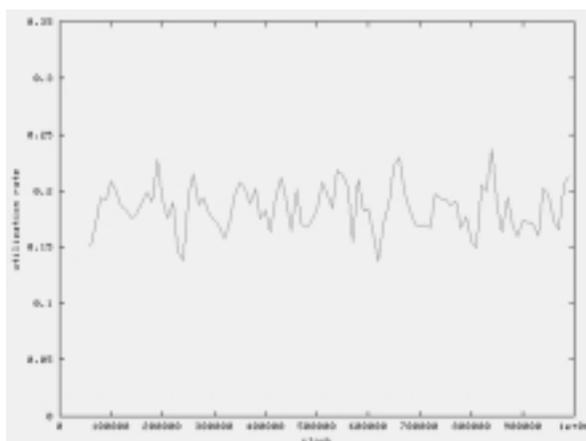
(c) R2(A)



(d) R2(B)



(e) R3(A)



(f) R3(B)

図 6.13: 各中継ノードの使用率 (Normal)

まず、各中継ノードのシグナルの待ち行列の長さについてそれぞれ図 6.10 のトラヒック分散をする前と後の各中継ノードの状態について考察し、トラヒック分散の効果を確認する。図 6.10 を参照すると、パターン A を通った場合とパターン B の場合とで、シグナルの待ち行列の長さについてはそれほど、差は見られないが若干、経路 A の方の待ち行列の方が長くなっていることが分かるが、パターン B の方はその待ち行列の長さに若干、それぞれの中継ノードの待ち行列の長さに乱れがあるが、パターン A の方の待ち行列の長さは、多少、経路 B よりもごくわずかに長めに見えるが、突出して平均の待ち行列の長さが長くなることもなく、どの中継ノードも安定して同じ程度の待ち行列の長さに収まっている。

つまり、どこかのノードで待ち行列の長さが長くなりすぎてしまうといくらそれ以外のノードの待ち行列の長さが平均的に短くてもそのノードで滞ってしまい、さらにトラヒックがそのまま増えると遅延時間は非常に長くなってしまうので、一部のノードだけ突出して待ち行列の長さが長くなってしまふことは、それだけ全体の遅延時間に影響を与える割合が大きいので、シグナルの packets 用の待ち行列の長さから評価した場合、トラヒック分散をした方が効率がよいと言える。

次に、各中継ノードの通常のトラヒックでの待ち行列の長さについて考察する。図 6.11 を参照すると、パターン A, B の R3 の待ち行列の長さにはそれほど差異は見られない。が、R2 についてはトラヒック分散をした B の方が若干突出した箇所がある。これは図??(b) のトラヒック分散をした場合の R1 の待ち行列の長さが、比較的長いためであると考えられる。一方、R1 についてであるが、これは明らかにトラヒック分散をしなかった方が待ち行列の長さが短い。同じトラヒックであるので全く同じタイミングで packets が発生し、同じ packet 長なのだが、これは図?? の全体のトラヒックに占めるシグナリングのトラヒックの割合を示している図の図??(a) と図??(b) を参照すると目算で約 1 割程度、トラヒック分散をしなかった方の R1 のシグナリングのトラヒックの割合が多いことが分かり、通常のトラヒックの割合が少なかったからだということが分かるが、なぜこのような現象が起こったかを考えると逆に考えると、トラヒック分散を行った方が通常のトラヒックが多かったと考えられる。これは SR1 から SR2 へトラヒックを流すとトラヒック分散をしなかった場合に比べて早い時間で SR2 に packets が到着する。packets が到着すると SR2 は Ack を返すため、そのトラヒックが SR2 から SR1 にむけて流れるが、Ack が到着する前に、ランダム時間経過した後、さらに packets を投げるのだが、その間隔が短かったため、Ack が到着する前にさらに packets を送出してしまい、R1 に到着した頃、Ack が R1 に次々と到着し、結果として R1 の通常のトラヒックの割合が高くなったためであると考えられる。

つまり、トラヒックを分散して到達時間に差が現れると Ack を返すが Ack が返ってくる前にデータグラムを送出するようなもの場合、時間差によってトラヒックが集中する Hot Spot が生じてしまう可能性が高くなる。

各中継ノードの待ち行列の長さからトラヒック分散をした場合としない場合を比較して、トラヒック分散の有効性を評価した場合、シグナリングのトラヒックについては比較的緩やかな変化であるため有効性が高いと言えるが、通常のトラヒックを流した場合、時間差によって Hot Spot が生じてしまい、結果としてトラヒック分散をしない方が状態がよくなっている。経路選択によって QoS 制御をするならば、そのトラヒックで使用するプロトコルに制限を付けなければならないことになり、QoS 制御を行うための判断基準に要求されたトラヒックの性質を示すパラメータも含めた方がよいのかもしれない。

次に、各中継ノードの使用率からトラヒック分散の有効性について考察する。使用率は式 (3.2) で示すように待ち行列の長さから算出して得た値であるのだが、各ノードの到着するパケットの数とサービスされるパケットの数の比 ($\frac{\lambda}{\mu}$) であるので各ノードの状態がよく分かる。

本研究の場合、各ノードのふたつの待ち行列の長さからそれぞれ使用率を出し、各待ち行列の状態がよく分かるようにして問題点を洗い出してみる。まず、通常のパケットを格納する側の待ち行列について図 6.13(a),(b) を参照すると目算で約 2 倍違うように見えるが、これは先ほどの時間差による Hot Spot の出現ということで理由がつく。使用率がトラヒック分散をしない場合と下場合をそれぞれ約 0.1、0.2 ととらえると、トラヒック分散をしなかった場合のパケット到着間隔にくらべてトラヒック分散をした場合はその半分の時間で、約 2 倍のトラヒックが集まっているということが分かる。

その他の各中継ノードについては、図 6.13(c) ~ (f) までを参照すると、それほどトラヒックの流量に大きな差は生じていない事が分かる。つまり、使用率によって各ノードの状態は分かっても、使用率がわずかに 0.1 と 0.2 という比較的低い使用率でしかもわずかな差でしかないのに大きな遅延を生み出していることがわかり、一律に規定値を設けて制御するだけでは要求される QoS を満たすことは難しいことが分かる。もちろん、使用率が 0.7 とか 0.9 に達してしまえばほとんど待ち時間が非常に長くなってしまい遅延時間が長くなり、制御は必要であるが、たとえ小さな値の使用率であってもそのわずかな差のトラヒック量は簡単に 2 倍、3 倍に膨らんでしまうことも考慮して使用率を使う必要があり、さらなる経路選択アルゴリズムの改良が求められる。

だが、ここで単純に通常のトラヒックだけを流した場合の R1,R2,R3 のトラヒック分散

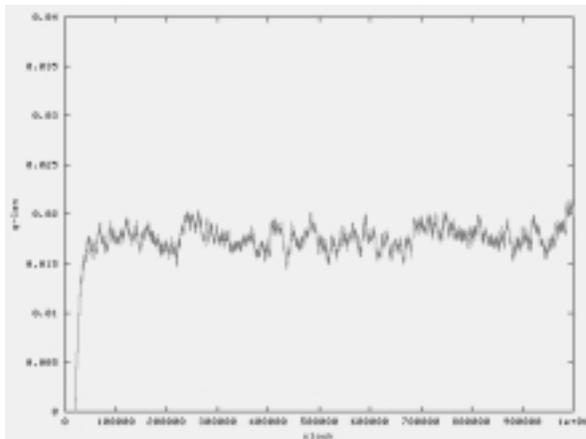
前と後の待ち行列の長さを調べてみる。しかし、ここでは Ack は返さないようにして通常のトラフィックを流すことにし、また分散しない場合、すべてのトラフィックが R2 を経由するようにし、それぞれの経路は R1-R2-R3 と R4-R2-R5-R6 の二つの経路を使用した。トラフィック分散するときは、SR1 から SR2 へのトラフィックは R1-R3 の経路を通り、SR3 から SR4 へのトラフィックは R4-R5 の経路を通るようにした。また、この実験はトラフィック分散すること自体の有効性を探るためだけなのであえて、先ほどの条件を変え、R2 にトラフィックが集中するため処理能力は上がらないので R2 への各回線の帯域幅を 10 とした。また、その他の条件について下記に示す。

表 6.4: シミュレーションを行った時の各パラメータ (分散前)

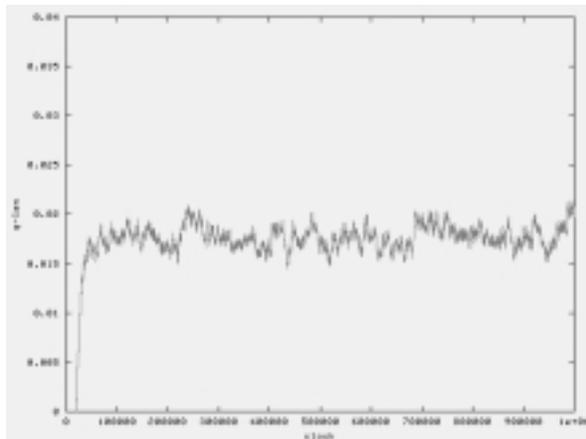
ノード	到着率	サービス率
SR1	0.276	1.00
SR2	0.276	1.00
SR3	0.276	1.00
SR4	0.277	1.00
中継ノード	平均待ち行列長	使用率
R1	0.405	0.465
R2	0.944	0.608
R3	0.316	0.426
R4	0.318	0.427
R5	0.475	0.492
R6	0.439	0.478

表 6.5: シミュレーションを行った時の各パラメータ (分散後)

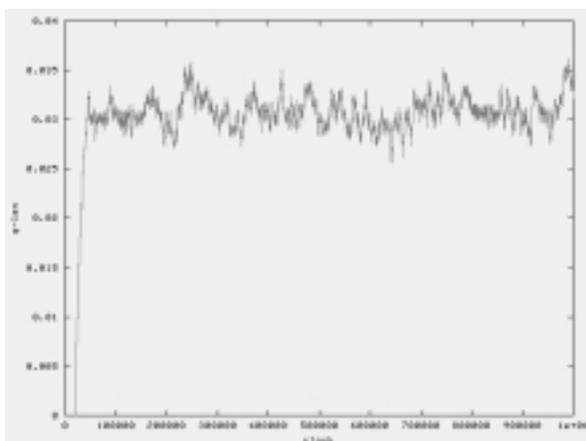
ノード	到着率	サービス率
SR1	0.276	1.00
SR2	0.276	1.00
SR3	0.276	1.00
SR4	0.277	1.00
中継ノード	平均待ち行列長	使用率
R1	0.444	0.480
R2	0.511	0.504
R3	0.345	0.440
R4	0.374	0.452
R5	0.554	0.517
R6	0.524	0.508



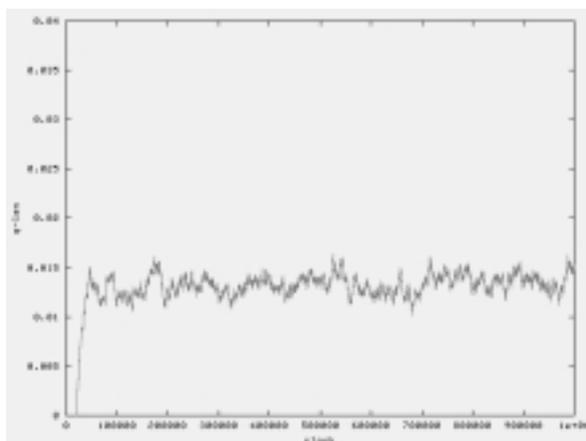
(a) R1(A)



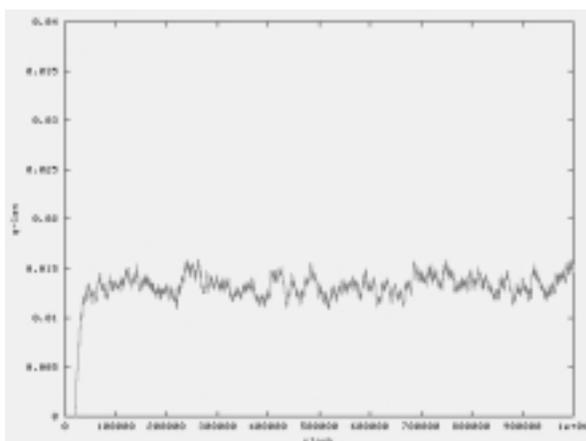
(b) R1(B)



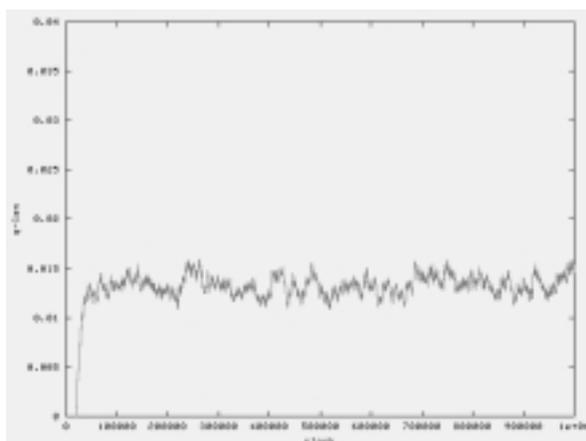
(c) R2(A)



(d) R2(B)

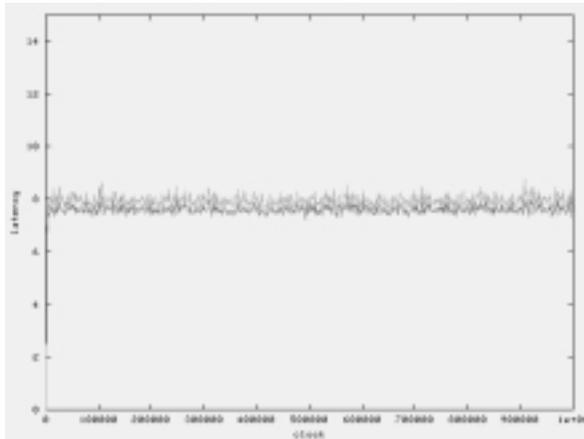


(e) R3(A)

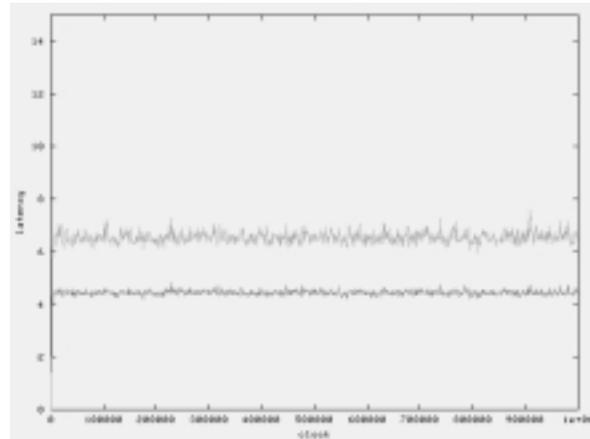


(f) R3(B)

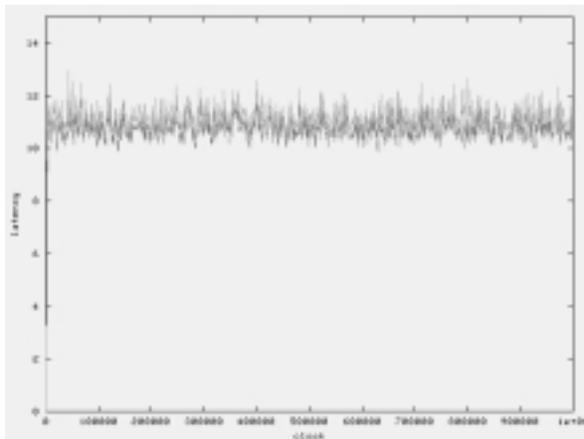
図 6.14: 通常のトラフィックのみを流した各中継ノードの待ち行列長



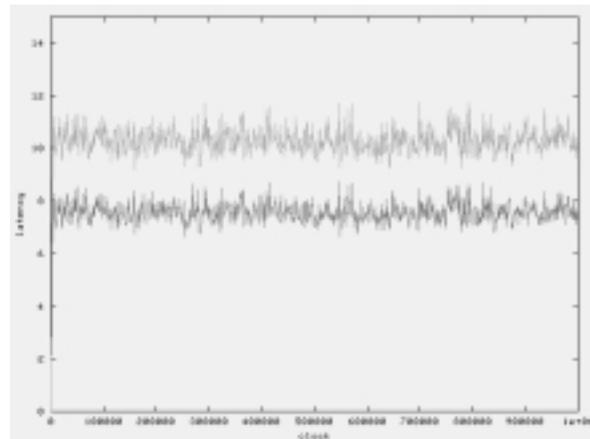
(a) SR2-SR1



(b) SR1-SR2



(c) SR4-SR3



(d) SR3-SR4

図 6.15: 通常のトラフィックのみを流した場合の遅延時間

だが、ここで図 6.14 をみてみると R1,R3 のトラフィック分散前とトラフィック分散後の状態を見比べてみるとほとんど代わりないが、R2 の状態が大きく異なる。

図 6.4 と図 6.5 よりだいたいおなじぐらいのトラフィック量が発生しているものと考えたと分散前の R2 の待ち行列長 0.944 が分散後 0.511 になったのは SR1 から SR2 へのトラフィックと SR3 から SR4 へのトラフィックが R2 を流れなくなったと考えたと上の R2 以外の待ち行列長の平均は 0.391 であるので、 $\frac{0.391}{2} \times 2 = 0.391$ となり、多少測定値の方が大きいもののトラフィック分散の効果が出ているといえる。

つまり、上記の図 6.14 と表よりトラフィック分散だけ考えた場合、SR1 から SR2 への

トラヒックの遅延時間は R2 を通らなくなった分、待ち時間が減り遅延時間が短くなり、SR2 から SR1 へのトラヒックは R2 の待ち時間が短くなったぶんだけ遅延時間が短くなり、結果として両方とも遅延時間が減少したことを示し、経路選択によるトラヒック分散をすることの有効性は十分に示せたと言える。

6.4 システム全体の評価

6.4.1 シグナリングのトラヒックの遅延について

第 6 章 1 節,2 節,3 節と各機能についての評価を行ってきたが、本研究で目指す目標の一つである、シグナリングの packets を制限時間内にサービスすることで管理制御を分散して行う際の通信コストが通常のトラヒックに影響を受けずに伝送されることは達成されたといえる。

また、Queue Management System によって行う Scheduling も図 6.10、図 6.11 を見る限り期待された機能は達成していると言える。また、その結果、シグナリングのトラヒックの遅延時間もほとんど通常のトラヒックの影響を受けずに伝送されることが確認された。

6.4.2 経路選択による QoS 制御について

本研究で提案した経路選択アルゴリズムで用いた判断基準だけでは十分な QoS 制御は出来ないことが分かった。QoS 制御を行うための新たな要素の洗い出しとともに、中継ノードの使用率の低い値での遷移におけるトラヒックの状態変化を考慮した経路選択アルゴリズムが必要であると考えられる。

また、シミュレーションによって時間差による Hot Spot 出現の可能性を証明できたが、このことは逆手に取ると遅延差を利用してうまく交互にトラヒックを SR1 と SR2 で制御できれば経路上は交わるノード上であってもトラヒックの負荷を押さえることが出来る可能性があることが分かり、トラヒック分散をする有効性の可能性は示せたと言える。

さらに、相手側の応答を無くしてトラヒックを流した場合、R1 及び R3 の待ち行列の長さにはほとんど分散した場合としない場合とで変化はなく、R1、R3 での遅延時間の短縮はないが、R2 では SR1 から SR2 へのトラヒックと SR3 から SR4 へのトラヒック

がR2を通らなくなったおかげで待ち行列の長さが短くなり、結果として、SR1からSR2へのトラヒックの遅延時間は短くなり、さらにSR2からSR1へのトラヒックの遅延時間はR2の待ち時間が短くなったぶんだけ遅延時間が減少していることが分かり、十分にトラヒック分散をすることの利点は示せたと言える。

しかし、R2を通るトラヒックの遅延時間はR2の待ち時間の減少によるものであるものでR2でほとんど待ち行列にパケットが貯まることが無ければ、R2を通る経路を流れるトラヒックの遅延時間はほとんど減少しない可能性があるが、トラヒックが混み合っているときにはその効果は大きいと言える。

第 7 章

おわりに

7.1 まとめ

本研究では、ネットワーク状態を反映した経路選択による QoS 制御をするにあたって以下のように考えた。

- 管理制御機構: 管理制御を行うことでネットワーク状態を考慮した経路選択が行えるように考える。
- 経路選択: QoS 制御をするために必要な要素を判断基準とする QoS 制御のための経路選択を考える。
- Queue Management: 管理制御を行う際のシグナリングのトラヒックを通常のトラヒックの影響を受けずに伝送できるための Scheduling を考える。

上記の考えをふまえて、経路選択による QoS 制御の必要性、達成すべき問題等を要件としてまとめ、そのシステムモデルのシミュレーションを行い、評価を行った。

本研究の成果を以下にまとめる。

- Queue Management System によってシグナリングのパケットが他のトラヒックの影響を受けずに伝送されることを達成した。
- Queue Management System によってシグナリングのパケットを優先しつつも通常のパケットの方の待ち時間が無限に長くないスケジューリングを達成した。

- 経路選択においてトラフィックを分散することで同じ経路で両端から流れるトラフィックの遅延時間が両方とも短くなることが示せ、トラフィック分散をすることの有効性を示せた。

7.2 今後の課題

システム評価によって明らかになった問題点を明らかにし、以下の点を考慮していくことで経路選択による QoS 制御がより一層確実なものになると考えられる。

1. 判断基準の動的な要素として中継ノードの使用率のみを用いたがそれでは QoS 制御を行うには不十分であった。プロトコルも意識した経路選択によって QoS 制御をする必要がある。
2. 中継ノードの使用率が低い値で遷移するときでもトラフィックの状態変化を意識した管理制御が必要である。
3. より多様なネットワーク構成でシミュレーションを行い、新たな問題点を発見する。
4. 遅延差を利用してトラフィック制御を行い経路が変わるノードがあっても負荷を軽減させる制御法

謝辞

本研究を行うにあたり、日頃御指導を賜りました日比野靖教授に心より感謝申し上げます。また、日頃から本研究に関して有益な助言をいただきました落水研究室の猪俣敦夫氏、篠田研究室の宇田仁氏、ならびに日比野研究室の皆様には深く感謝致します。

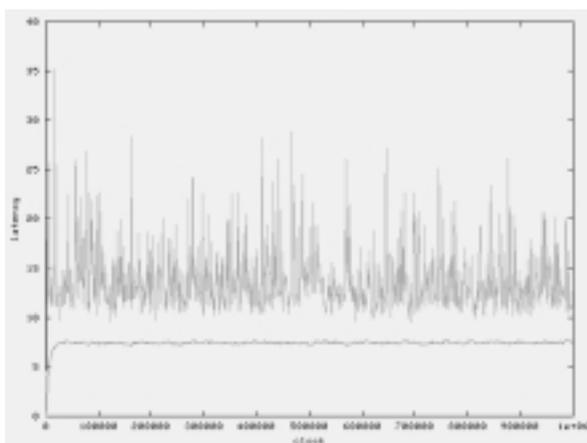
参考文献

- [1] W.Richard Stevens *TCP/IP Illustrated, Volume1* Addison-Wesley Longman, Inc.
- [2] Grenville Armitage *Quality of Service in IP Networks* Macmillan Technical Publishing
- [3] Uyles Black *IP routing : RIP, OSPF,BGP,PNNI, and routing protocols* Prentice-Hall, Inc.
- [4] Bruce Davie and Yakov Rekhter *MPLS Technology and Applications* MORGAN KAUFMANN PUBLISHERS
- [5] Donald Gross and Carl M. Harris *Queueing Theory* THIRD EDITION John Wily & Sons, Inc.
- [6] W. リチャード・スティーブンス著 篠田陽一訳 *UNIX ネットワークプログラミング 第2版 Vol.1* 株式会社トッパン 株式会社ピアソン・エデュケーション
- [7] G. Apostolopoulos, D. Williams, IBM, S. Kamat, Lucent, R. Guerin, UPenn, A. Orda, Technion, T. Przygienda, Siara Systems *RFC2676 QoS Routing Mechanisms and OSPF Extensions* The Internet Society
- [8] Daniel Reininger, Dipankar Raychaudhuri, and Maximilion Ott, NEC USA,Inc. *A Dynamic Quality of Service Framework for Video in Broadband Networks* IEEE Network November/December 1998
- [9] G. Huston, Telstra *RFC2990 Next Steps for the IP QoS Architecture* The Internet Society
- [10] S. Blake, Torrent Networking Technologies, D. Black, EMC Corporation, M. Carlson, Sun Microsystems, E. Davies, Nortel UK, Z. Wang, Bell Labs Lucent Technologies,

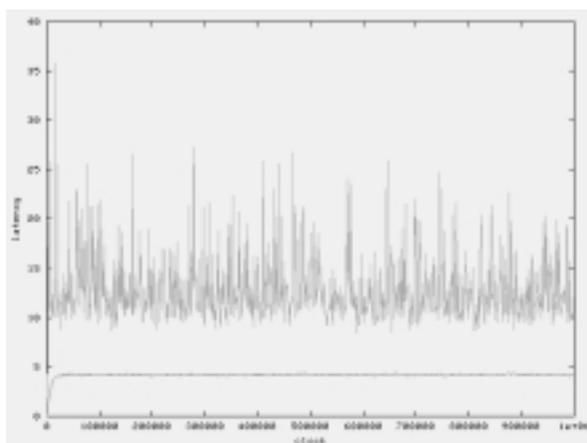
W. Weiss, Lucent Technologies *RFC2475 An Architecture for Dirrerentiated Services*
The Internet Society

- [11] M. Eder, Nokia, S.Nag *RFC3052 Service Management Architectures Issues and Review* The Internet Society

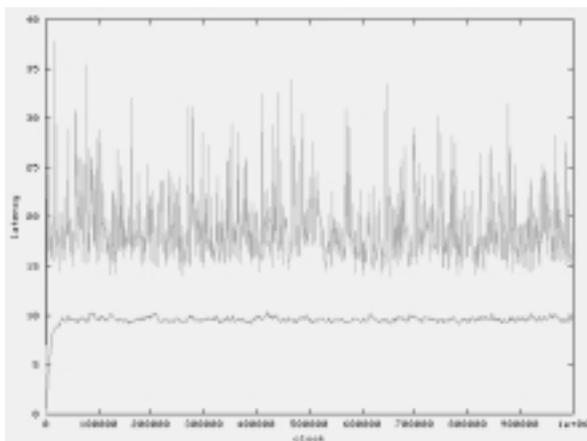
付録



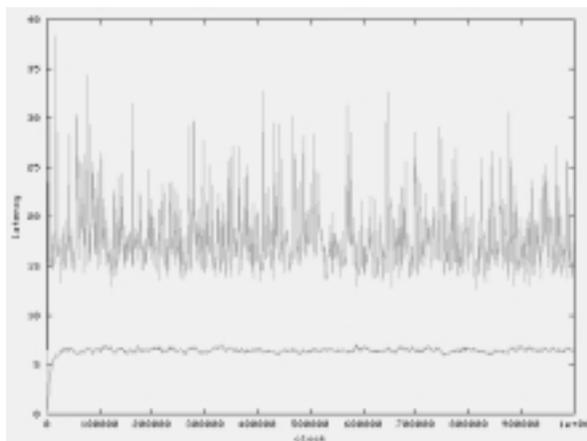
(a) SR2-SR1



(b) SR1-SR2



(c) SR4-SR3



(d) SR3-SR4

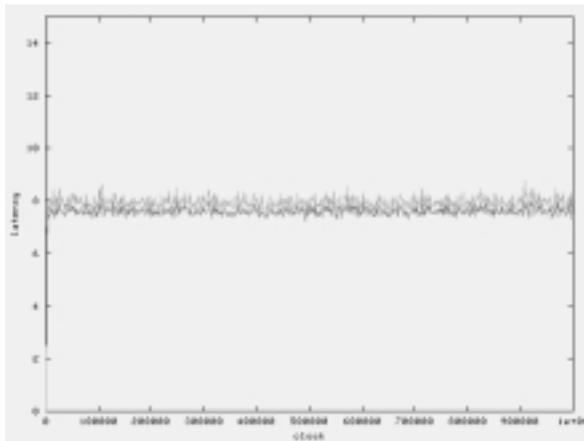
図 7.1: 通常のトラフィックのみを流した場合の遅延時間 (使用率 -高-)

表 7.1: シミュレーションを行った時の各パラメータ (分散前)

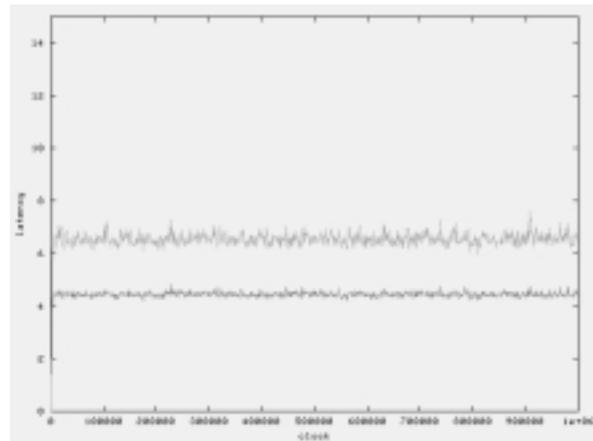
ノード	到着率	サービス率
SR1	0.286	1.00
SR2	0.287	1.00
SR3	0.287	1.00
SR4	0.287	1.00
中継ノード	平均待ち行列長	使用率
R1	0.885	0.597
R2	5.53	0.865
R3	0.700	0.557
R4	0.707	0.559
R5	1.35	0.668
R6	1.18	0.647
トラヒック	平均遅延時間	
SR1-SR2	10.7	
SR2-SR1	12.7	
SR3-SR4	17.4	
SR4-SR3	17.4	

表 7.2: シミュレーションを行った時の各パラメータ (分散後)

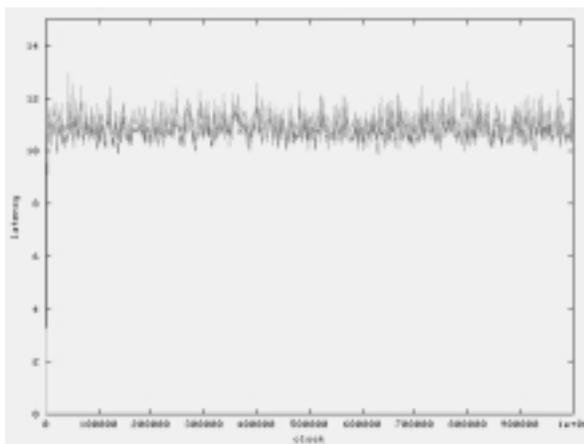
ノード	到着率	サービス率
SR1	0.286	1.00
SR2	0.287	1.00
SR3	0.287	1.00
SR4	0.287	1.00
中継ノード	平均待ち行列長	使用率
R1	0.885	0.598
R2	0.999	0.618
R3	0.706	0.558
R4	0.773	0.574
R5	1.44	0.679
R6	1.18	0.646
トラヒック	平均遅延時間	
SR1-SR2	4.23	
SR2-SR1	7.53	
SR3-SR4	6.55	
SR4-SR3	9.68	



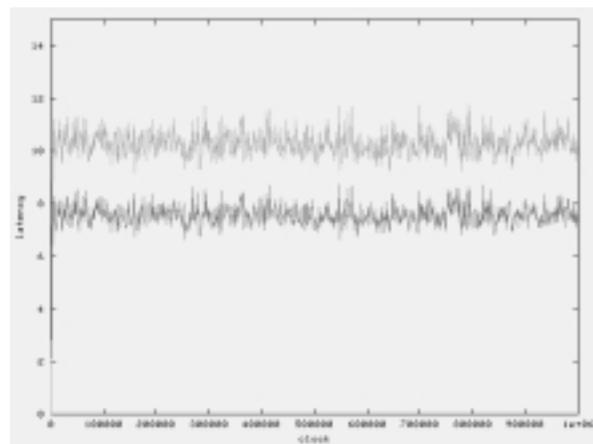
(a) SR2-SR1



(b) SR1-SR2



(c) SR4-SR3



(d) SR3-SR4

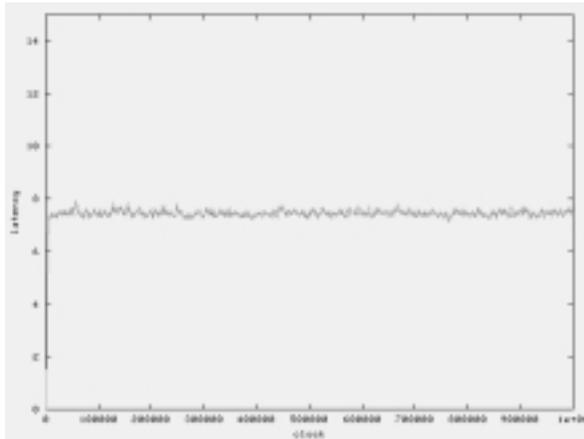
図 7.2: 通常のトラフィックのみを流した場合の遅延時間 (使用率 -中-)

表 7.3: シミュレーションを行った時の各パラメータ (分散前)

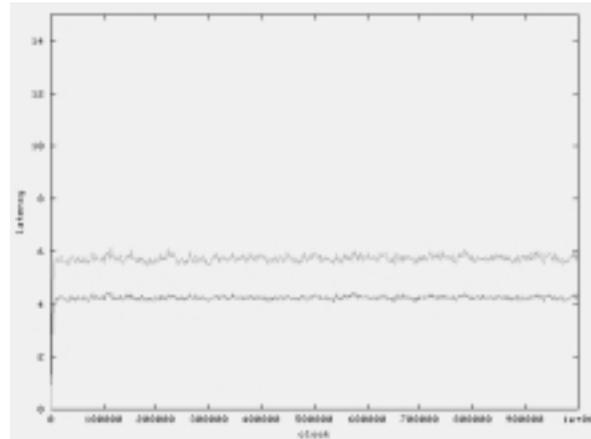
ノード	到着率	サービス率
SR1	0.276	1.00
SR2	0.276	1.00
SR3	0.276	1.00
SR4	0.277	1.00
中継ノード	平均待ち行列長	使用率
R1	0.405	0.465
R2	0.944	0.608
R3	0.316	0.426
R4	0.318	0.427
R5	0.475	0.492
R6	0.439	0.478
トラヒック	平均遅延時間	
SR1-SR2	6.56	
SR2-SR1	8.04	
SR3-SR4	10.2	
SR4-SR3	12.0	

表 7.4: シミュレーションを行った時の各パラメータ (分散後)

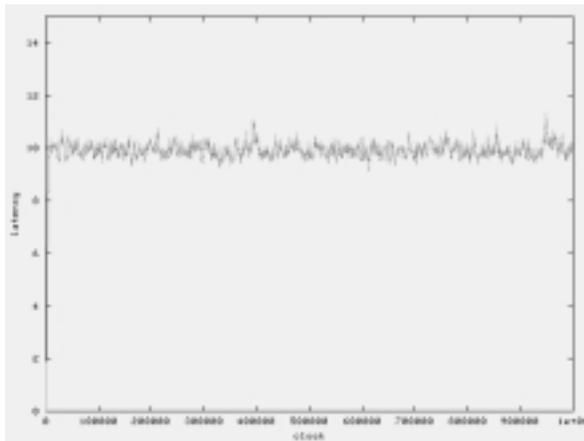
ノード	到着率	サービス率
SR1	0.276	1.00
SR2	0.276	1.00
SR3	0.276	1.00
SR4	0.277	1.00
中継ノード	平均待ち行列長	使用率
R1	0.444	0.480
R2	0.511	0.504
R3	0.345	0.440
R4	0.374	0.452
R5	0.554	0.517
R6	0.524	0.508
トラヒック	平均遅延時間	
SR1-SR2	4.39	
SR2-SR1	7.85	
SR3-SR4	7.56	
SR4-SR3	11.5	



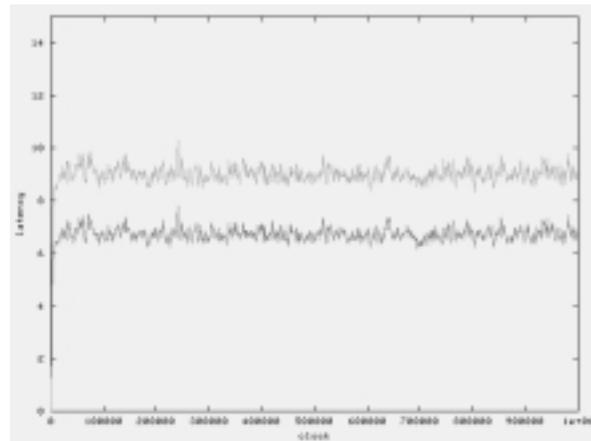
(a) SR2-SR1



(b) SR1-SR2



(c) SR4-SR3



(d) SR3-SR4

図 7.3: 通常のトラフィックのみを流した場合の遅延時間 (使用率 -低-)

表 7.5: シミュレーションを行った時の各パラメータ (分散前)

ノード	到着率	サービス率
SR1	0.234	1.00
SR2	0.234	1.00
SR3	0.234	1.00
SR4	0.235	1.00
中継ノード	平均待ち行列長	使用率
R1	0.186	0.348
R2	0.366	0.449
R3	0.140	0.311
R4	0.138	0.308
R5	0.193	0.353
R6	0.181	0.345
トラヒック	平均遅延時間	
SR1-SR2	5.73	
SR2-SR1	7.58	
SR3-SR4	8.84	
SR4-SR3	10.2	

表 7.6: シミュレーションを行った時の各パラメータ (分散後)

ノード	到着率	サービス率
SR1	0.234	1.00
SR2	0.234	1.00
SR3	0.234	1.00
SR4	0.235	1.00
中継ノード	平均待ち行列長	使用率
R1	0.186	0.348
R2	0.193	0.354
R3	0.141	0.311
R4	0.144	0.314
R5	0.193	0.353
R6	0.179	0.343
トラヒック	平均遅延時間	
SR1-SR2	4.25	
SR2-SR1	7.53	
SR3-SR4	6.61	
SR4-SR3	10.1	