優先制御ポリシーデリゲートによるトラフィック優先制御

環境情報学部 小畠 元
学籍番号 79895138

平成 14 年 2 月 17 日

概 要

本論文では階層的なネットワークにおいてトラフィック優先制御を行う手法を構築した．

トラフィック優先制御を構成する要素としてアドミッション制御，クラシファイ，シェーピングやドロッピング，およびパケットスケジューリングがあげられる．階層的なナットワークにおいては一般的に対外線がボトルネックになるため，ボトルネックに直接接するゲートウェイルータにおいてトラフィック優先制御を適用する．しかし，NAT や IPv4-v6 トランスレータを含むネットワークではこれら機構がトラフィック情報を変更してしまうため，クラシファイできないという問題がある．このような場合においてトラフィック優先制御は有効に機能しない．

本論文では対処方法としてトラフィック情報が変更される前にクラシファイする手法を導入した．また，キューイング技術の適用もトラフィック情報が変更される前とした．

本機構の動作概要を以下に示す．

1. ゲートウェイルータにおいて帯域を下位ルータ毎に分割する．
2. 分割した帯域を下位ルータへ通知する．
3. 下位ルータにおいてキューイング技術を適用し，全体でボトルネック回線を共有する．

本機構を用いる事により，NAT や IPv4-v6 トランスレータを含むネットワークにおいてトラフィック優先制御を有効に行う事が可能となった．

概 要

In this thesis, we study traffic management models and algorithms in hierarchical networks containing network address translators.

Traffic management consists of admission control, packet classification, packet shaping or dropping, and packet scheduling. A typical method of traffic management in a hierarchical network operates by implementing those mechanisms in a network gateway router, which is directly connected to the bottleneck link. However, in a hierarchical network, containing Network Address Translation mechanisms such as NAT and IPv4-v6 Translator, information on the flow of the data is concealed and the gateway router is unable to classify packets. Under these circumstances, traffic management performance would be highly inefficient.

The objective of this thesis is to classify packets before contained information is concealed by network address translators. We also applied queuing technologies on the non-gateway routers, which works as follows.

1. The gateway router determines the bandwidth usage for the non-gateway routers.
2. The gateway router delegates the bandwidth to the non-gateway routers.
3. Apply queuing technologies on the non-gateway routers to share the bottleneck resource.

Using this method, we have demonstrated the improvements of the traffic management system that can result from a network containing network address translators.

目 次

第 1 章　序論 **7**

1.1 はじめに 7
1.2 背景 8
1.3 研究概要 9
1.4 論文の構成 10

第 2 章　インターネットにおけるトラフィック優先制御機構 **11**

2.1 トラフィック優先制御の必要性 11
2.1.1 トラフィックの種類 11
2.1.2 ボトルネック 12
2.2 既存のトラフィック優先制御 14
2.2.1 トラフィック優先制御機構のモデル 14
2.2.2 キューイング技術 15
2.2.3 キューイング技術の適応 17
2.3 本章のまとめ 17

第 3 章　一般的末端ネットワークにおけるトラフィック優先制御とその問題点 **19**

3.1 トラフィック識別に失敗する場合 19
3.1.1 トラフィック情報を隠蔽化する機構 19
3.1.2 NAT などトランスレータを導入した末端ネットワーク 21
3.2 トラフィック優先制御に失敗する具体例 21
3.3 現状のトラフィック優先制御機構で対処不能な場合 22
3.3.1 一つのノードによる多数宛先に対するデータ送信 22
3.3.2 一般的なサービスに対する大量データ送信 22
3.4 問題点のまとめ 23

第 4 章　設計 **25**

4.1 解決方法のアプローチ 25
4.2 トラフィック識別手法 25
4.2.1 中央集権型識別手法 25
4.2.2 分散型識別手法 27
4.2.3 本研究で採用した識別手法 28
4.3 キューイング技術の適用 28
4.3.1 上位ルータによるキューイング 28
4.3.2 下位ルータによるキューイング 29
4.3.3 本研究で採用したキューイング方法 29

4.4 ボトルネック回線資源の状態感知 . . . 29
4.4.1 下位ルータ同士の通信による感知 . . . 30
4.4.2 SNMP を利用する場合 . . . 30
4.4.3 上位ルータから下位ルータへ回線状態を通知する場合 . . . 30
4.4.4 上位ルータから下位ルータ毎の回線資源上限を通知する場合 . . . 30
4.4.5 本研究で利用するボトルネック回線状態を感知する仕組み . . . 31
4.5 下位ルータでキューイングする際必要な制御 . . . 31
4.5.1 帯域の割り振り . . . 31
4.5.2 バーチャルライン . . . 31
4.5.3 帯域を割り当てるための要素 . . . 32
4.5.4 割り当ての計算 . . . 32
4.5.5 割り当てのタイミング . . . 33
4.5.6 割り当ての実行 . . . 33
4.6 ルータ同士の情報交換 . . . 33
4.7 全体像 . . . 33
4.8 必要な機構 . . . 33
4.8.1 関連研究 . . . 34

第 5 章 実装 **35**
5.1 開発環境 . . . 35
5.2 トラフィック優先制御を構成する機構 . . . 35
5.2.1 下位ルータ毎のトラフィック監視を行う機構 . . . 35
5.2.2 帯域の割り当て計算を行う機構 . . . 36
5.2.3 ルータ間通信を行う機構 . . . 36
5.2.4 ノード毎の帯域制御を行う機構 . . . 37
5.2.5 全体の流れ . . . 38
5.3 WIDE 合宿へ導入 . . . 38

第 6 章 評価 **40**
6.1 定性的評価 . . . 40
6.2 定量的評価 . . . 40
6.2.1 実験 1 . . . 41
6.2.2 実験 2 . . . 42
6.2.3 結果と考察 . . . 47

第 7 章 結論 **48**
7.1 達成した事 . . . 48
7.2 今後の課題 . . . 48
7.2.1 現状の問題点 . . . 48
7.2.2 不足している機能・調査 . . . 49

第 8 章 謝辞 **50**

付 録 A appendix **53**
A.1 SNMP 53
A.2 TCP/IP プロトコルスタック 53
A.3 IP ヘッダ (version 4) 53

図 目 次

1.1 インターネットにおけるデータ転送方式 9

2.1 FIFO Queue 12
2.2 特定のフローによる回線の占有 13
2.3 末端のネットワーク構成の一例 14
2.4 Priority Queueing 15
2.5 Class Based Queueing 16
2.6 Weighted Fair Queueing 16

3.1 NAT 20
3.2 IPv4-v6 トランスレータ 21
3.3 NAT や IPv4-v6 トランスレータを導入したネットワーク 22
3.4 特定フローによるボトルネックの占有 23
3.5 一つのノードによるボトルネック占有 24
3.6 WFQ の適用 24
3.7 CBQ の適用 24

4.1 トラフィック情報の変換対応表をボトルネックルータへ渡す 26
4.2 変更前の情報と変更後を組み合わせた表の作成 26
4.3 実時間をトリガーとした更新に生じる無駄 27
4.4 分散識別手法 28
4.5 バーチャルライン 31
4.6 下位ルータの優先度決定 32
4.7 全体像 34

5.1 本機構の流れ 38
5.2 WIDE 合宿ネットワークの構成 39

6.1 実験 1 の構成 41
6.2 実験 1.1:トラフィック量の移行 42
6.3 実験 1.1:回線使用率の変化 42
6.4 実験 1.2:トラフィック量の移行 43
6.5 実験 1.2:回線使用率の変化 43
6.6 実験 2 の構成 44
6.7 実験 2.1:トラフィック量の移行 44
6.8 実験 2.1:回線使用率の変化 45
6.9 実験 2.2:トラフィック量の移行 46

6.10 実験 2.2:回線使用率の変化 46

7.1 不公平が生じる場合 49

A.1 Simple Network Management Protocol 53
A.2 TCP/IP プロトコルスタック 53
A.3 IPv4 ヘッダ 54

表 目 次

5.1 開発環境 35
5.2 ルータ間通信に利用されるプロトコル 37
5.3 メッセージタイプ 37
5.4 メッセージタイプに対応する内容 37

6.1 実験に使用したノードの構成 40

第1章　序論

1.1　はじめに

インターネットを利用するユーザの増加と共に，ネットワークの質に対する要求は高まってきている．この要求を満たすために，近年インターネットで使用される回線は広帯域化している．これまでは，ラストワンマイルと呼ばれる ISP と顧客の各家庭を結ぶ回線の帯域が狭いことが問題であったが，ADSL や光ファイバなどの導入で問題は解消しつつある．ユーザが利用できる回線帯域が増加するにつれて，利用できるアプリケーションの幅が広がり，動画配信や音楽配信などをインターネットで楽しむユーザも多くなってきている．

しかし，1 人のユーザがネットワークに送出できるデータ量も多くなることによって引き起こされている問題がある．回線帯域の狭い部分を一部のユーザが占有することによって，他のユーザの通信に障害を起こしたり，場合によっては利用できなくなる．この回線帯域の狭い部分は，ネットワークのボトルネックと呼ばれている．ボトルネックの問題は以前より存在したが，近年のネットワークの広帯域化とユーザが利用する帯域格差の拡大が問題を軽視できないほどに大きくなものにした．ごく少数のユーザ，時には 1 人のユーザでもボトルネックに大量のデータを送出してしまいネットワークに障害が発生してしまう可能性がある．また，ボトルネックと成り得る部分を多くのユーザが共有しているときに，一部のユーザが突発的に多くのデータを流し込むと他のユーザの通信に障害が起こり得る．ユーザ一人一人が利用できる帯域が狭かったころは障害による不利益は均等に配分されたが，帯域格差の拡大に応じて不利益にも格差が生じる．これは，インターネットのアーキテクチャにおいては，ネットワークリソースが流れているパケット単位で均等に配分される為である．パケットそのものは利用者を特定しないので，リソースが利用者ごとに均等に配分されるとは限らない．

インターネットの交換機であるルータでは，パケットを一時的に蓄積するために通常一つの FIFO(First In First Out) キューが用意されている．この FIFO キューの特性により，パケットごとにリソースが均等に配分される．このキューの仕組みを置き換えると，より細やかなリソース配分を行うことができる．例えば，流れているトラフィックごとに複数のキューを用意し，それぞれのキューごとに均等にリソースを割り振る Fair Queue[4] のような仕組みが存在する．この仕組みを実現するために，データ送信者 (送り元ホスト) からデータ受信者 (宛先ホスト) までの一連のパケットの流れを区別する必要がある．本論文では，このパケットの流れを，特にフローと呼ぶ．

実際に，キューイング技術を利用するには各フローごとの情報を管理することが必要である．フローを特定する情報として，宛先 IP アドレスや送り元 IP アドレス，さらに宛先ポート番号や送り元ポート番号などが挙げられる．原則的にインターネットは End-to-End のアーキテクチャに基づいているので，フローの情報を正確に管理すれば，ホストに対して均等なリソースの配分が実現できる．

しかし，現状のインターネットでは，フローによる識別が直ちに通信しているホストの識別に結びつかない場合がある．それは，例えば NAT[1] などの中間ノードがフローの識別に使用する IP アドレスなどの情報を変更してしまう場合があるためである．

1.2 背景

ラストワンマイルとよばれる家庭と ISP(Internet Service Provider) とをつなぐ回線の帯域の広帯域化は長い間進歩が無かった．各家庭における通信用の回線は電話線のみであり，新たにデータ通信専用の回線である光ファイバーなどを導入するのにはコストが高く現実的でなかった．そこで各家庭にはじめから設置されている回線を利用し，更に広帯域化させる DSL(Digital Subscriber Line) などの技術が発達した．DSL は同じ物理回線である電話回線を利用しつつ従来の通信速度を大幅に改善した．他にもケーブルテレビ用の回線をデータ通信に利用するなどラストワンマイルの広帯域化が進んだ．またインターネットのバックボーンにおける回線は光多重化などの技術で同じく広帯域化された．

インターネットにおける回線の広帯域化が進むことでインターネットを利用した通信の可能性は広り，音声を利用したリアルタイム通信や，動画配信が可能となった．

インターネットの通信はパケット交換方式である．通信にはパケットと呼ばれるデータの区分に，宛先の情報やデータ全体のどの箇所にあたる区分なのかといった情報を付加した形式を用いる．パケットを交換して通信を行うインターネットにおける通信方式はパケット交換方式と呼ばれる．パケット交換方式の通信は，パケット転送を中継する中間ノードによるパケットのリレーで成り立っている．

一方，古くから利用されている電信電話通信は回線交換方式の通信である．回線交換方式の通信では，通信の開始から終了まで中継局において回線を確保し，通信を保護する．よって回線交換方式では，回線の占有に失敗したエンド ノードは，回線が開放されるまで通信を行えない．

回線交換方式では，中間ノードの機能が重要な役割を持つ．中継システムの障害はシステム全体の機能に大きな影響を与える．よって中間ノードに非常に高い信頼性が要求される．

パケット交換方式であるインターネットの中間ノードの処理は電信電話通信における中間ノードのそれに比べ単純である．中間ノードであるルータは次の中間ノードにパケットを渡すのみで通信の発生箇所から到達箇所までの到達性は一切感知しない．図 1.1 にパケット交換方式の通信を利用したデータ送信を示す．送り元 (Src) ノードから宛先 (Dst) ノードまである一連のデータを送信する際に，送り元ノードはそれを細かく分解し，パケットに変換して宛先ノードに送信する．

このように中間ノードでの処理が単純なためインターネットはその規模拡大が容易である事が利点である．また，パケット交換方式では，通常エンド ノードに回線を占有させない．従って同時にネットワーク資源を利用できるエンド ノードの数に制限がない．このようなパケット交換式ネットワークの特徴は，多数の利用者が同時にネットワーク資源を利用する広域コンピュータネットワークの特性に適している．

しかし，パケット交換方式では，中継システムによるサービスは回線交換方式のそれと比べ，低信頼性で提供される．ネットワークに流入するデータ量の総量が回線資源を越える事があり，パケットが中間ノードによって破棄される現象「パケットロス」が発生する可能性がある．従って，パケット交換型のネットワークにおいてデータの完全性を保証する，TCP/IP[5] などの通信プロトコルが開発され，発達した．

TCP(Transmission Control Protocol) はネットワーク資源が過度に競合し，パケットロスが頻繁に発生する状態「輻輳」を事前に予測し，エンド ノードの通信速度を抑制する機能 [6] を持つ．

しかし，現在のインターネットにおいてトラフィックは多様化している．P2P アプリケーション [7] や，VoD(Video On Demand)，VoIP(Voice over IP) などのリアルタイムアプリケーションの登場はインターネットにおけるトラフィックに編かを及ぼしている．中でもリアルタイムトラフィックには輻輳回避などトラフィック流量を抑制する仕組みを適用できない．従って，こういっ

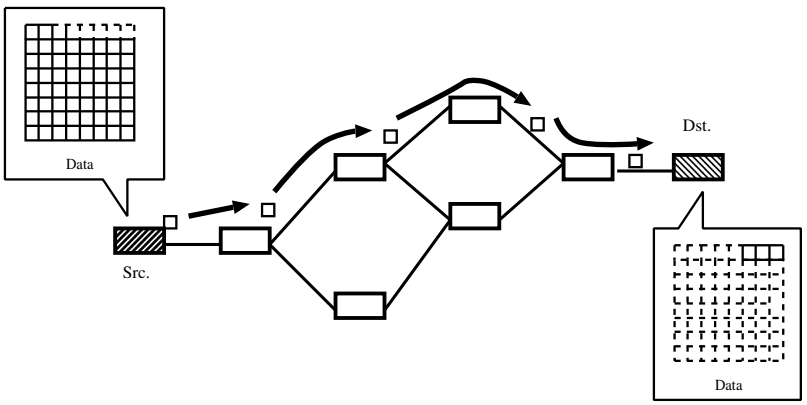

図 1.1: インターネットにおけるデータ転送方式

たトラフィックは輻輳回避を行わないため回線資源を占有する可能性がある．

回線資源の占有を防ぐ手段として，中間ノードによるトラフィック優先制御 [8] がある．中間ノードによるトラフィック優先制御では中間ノードのキューイング機構が各トラフィックの挙動を監視し，特定のトラフィックを優先的に転送することで回線資源の占有を防ぐ．また，優先的にトラフィックを転送する機構を持つため，トラフィック優先制御はインターネットにおける回線資源を予約する際利用される．VoIP や VoD トラフィックではパケットロスの発生を防止するためトラフィック優先制御機構が必要である．

しかし，こういったトラフィック優先制御技術は NAT や IPv4-v6 トランスレータなどのパケットの持つ情報を隠蔽化する機構の存在するネットワークにおいて，うまく機能しない．

1.3 研究概要

本研究は末端ネットワークにおいて，エンド ノード 毎に帯域保証を行うトラフィック優先制御機構の確立を行った．本研究では，あたらしいトラフィック優先制御技術に対する要求を整理し，次の 3 つの要求の解決を行った．

- トラフィック情報を隠蔽化する機構の存在するネットワークにおけるトラフィック優先制御に対する要求

  現在のインターネットでは NAT や IPv4-v6 トランスレータといったパケットの持つ情報を隠蔽化する機構を利用する場合がある．同機構を用いた場合，トラフィック優先制御が充分に機能しない．こういった場合に対応するトラフィック優先制御機構が必要である．

  本研究ではパケット情報を隠蔽化する機構の存在するネットワークにおいて機能するトラフィック優先制御機構を開発した．

- 最低限利用できる帯域を確保する要求

  エンド ノードは通常最低限利用できる帯域を知る事ができない．よって多様化したトラフィックをユーザがいかに利用するか，また利用できるかと言った判断ができない．最低限帯域保

証を行う機構が必要である．

本研究ではエンド ノードに最低帯域保証を行うトラフィック優先制御技術を実現した．

- ネットワーク管理者の手間をかけずトラフィック優先制御を行う要求

  一般的にトラフィック優先制御はネットワーク管理者が通信機器に設定を施すことで行われる．これはネットワーク管理に大きな負荷を与える．トラフィック優先制御を自動で行う機構が必要である．

  本研究ではネットワークの状態を検知し，それに対応しトラフィック優先制御機構の振舞を変化させる機構を構築した．

1.4 論文の構成

本論文は全 8 章で構成される．

第 1 章は序論である．第 2 章ではトラフィック優先制御の現状，第 3 章ではトラフィック優先制御の問題点について述べる．次にその解決方法の考察，およびその実装に付い第 4，5 章で述べ，第 6 章において，その正当性の評価について言及する．第 7 章で結論にいたり，最後に 8 章の謝辞で終る．

第2章　インターネットにおけるトラフィック優先制御機構

本章ではインターネットおけるトラフィック優先制御の必要性と，その機能について述べる．

2.1　トラフィック優先制御の必要性

インターネットにおいてトラフィック制御が必要な理由を示す．

2.1.1　トラフィックの種類

適応型トラフィック

本論文ではネットワークの状態を推測しながら，データ転送量を制御するトラフィックを適応型トラフィックと呼ぶ．

適応型トラフィックの代表的なものに TCP トラフィックがある．TCP はインターネットにおいて中間ノードによるデータ到達性の確認機能がないために発達したプロトコルである．

インターネット上では中間ノードの転送能力を超過するトラフィックの発生などを理由にパケットが到達せず，損失が生じる場合がある．中間ノードではパケット損失を検出しないため，TCP を用いた通信では，通信の開始ノード，終点ノードであるエンドノードがパケットを受け取るたびに確認を通信相手に返す．確認を受け取らない場合には，パケット損失が発生したと認知する仕組みである．パケット損失が生じた場合，エンドノードは当該パケットを再送する．またパケット損失を多く検出した場合，エンドノードは通信相手までの経路が現状通信するのに好ましくないと判断し，一定時間パケット送出を中止する．

この様な機構から適応型トラフィックはパケット送出量をネットワークの状態に合わせ動的に変更さえせるため，固定された送出量を保つことが難しい．また再送メカニズムをもつためパケットの順番がいれかわることがありうる．よってリアルタイム通信である音声通信やビデオ会議には適していない．これらの用途にはパケットの再送を行わない，転送量のコントロールを一切行わないプロトコルが用いられ，それは非適応型トラフィックと呼ばれる．

非適応型トラフィック

インターネットにおいてネットワークの状態を考慮せずに転送するトラフィックを非適応型トラフィックと呼ぶ．

代表的な非適応型トラフィックとして UDP(User Datagram Protocol)[9] を用いたトラフィックがあげられる．UDP は特徴としてパケットの大きさを指定でき，中間ノードの状態に関係なく送出可能である事があげられる．よって回線の混雑にかかわらず際限無くデータを送出できる．輻

輳を感知する仕組みが無いため，これらのトラフィックは非適応型トラフィックと呼ばれる．

UDP を用いたトラフィックは再送，パケットの順序を整えるといった作業が必要無いためプロトコルのオーバヘッドが小さい．パケットの再送がなく，転送量を通信中に変化させる事が無いため，音声通信などのリアルタイム通信に適している．

2.1.2 ボトルネック

パケットロスが頻繁に発生し，輻輳を起こす回線をボトルネックと呼ぶ．

インターネット上の回線は広帯域化した．しかし回線には帯域格差があり，トラフィックの集中が発生する事により，ボトルネックとなる回線の存在は解消されない．

以下にボトルネックにおいてパケットロスが発生する理由を示す．

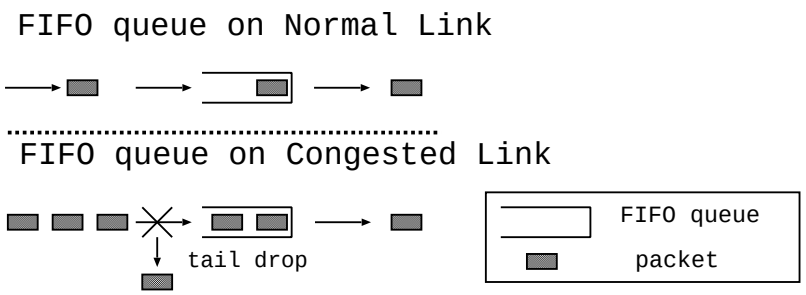

図 2.1: FIFO Queue

ネットワークインターフェースにおいてパケットを送受信する際，キューと呼ばれる一時的にパケットを保持するバッファを持つ．

ルータでは標準で各インターフェースの出口と入口に FIFO キューが用意される．FIFO キューは初めにキューに入力されたパケットを初めに送出する簡単な仕組みである．本論文ではインターフェースの入口にあるキューを入口キュー，出口にあるキューを出口キューと呼ぶ．

インターフェースの入口は物理回線の転送限界量に耐え得る設計がなされているため，パケットが入口キューにおけるパケット損失はない．しかし，出口の FIFO キューではトラフィックの偏り，物理回線の帯域格差がある場合，パケットがキューに入り切らずに tail drop する．この状態をパケットロスの発生と言う．tail drop とはキューの長さに限界があり，キューに入り切らなかったパケットが廃棄される事である．図 2.1 に FIFO キューにおいてパケットロスが発生する例を示す．通常中間ノードの転送能力を越えない場合には出口キューに入ったパケットはキューに溜ること無く転送される．しかし，中間ノードの転送能力を越えるトラフィックが発生した場合，出口キューからパケットが送出されるデータ量より入口キューに挿入されるデータ量が多いため，パケットロスが発生する．

特定のフローによる回線の占有

本論文ではフローを個別のエンドノードから発生する一連のパケットと定義する．

通常，適応型トラフィックならばボトルネックとなる回線を通るフローはその転送量を減少させる．他のフローも同等に転送速度を制御するため，ボトルネックの輻輳を回避することができる．

しかし，解消されないボトルネックでは特定のフローが回線を占有してしまうことがある．

以下に特定のフローがボトルネックを占有する場合を図 2.2 に示す．

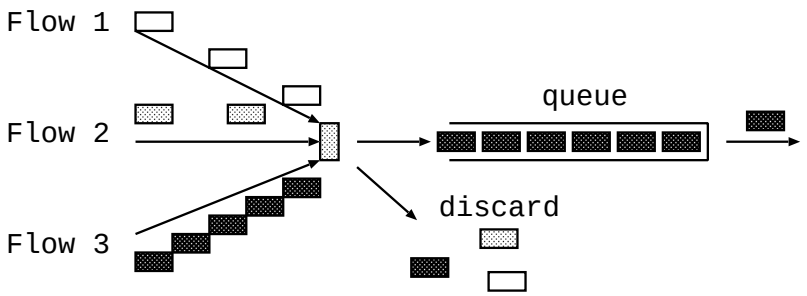

図 2.2: 特定のフローによる回線の占有

ルータは通常パケットレベルの転送処理を行い，フローの検知はしない．そのため出口キューが特定のフローに占有される場合がある．

図 2.2 中のフロー 1，2，3，がボトルネックを共有している．フロー 1，2 はパケット量が少量なのに対し 3 は大量で，出口キュー内を全てフロー 3 のパケットで埋めている．その結果ボトルネックにはフロー 3 のパケットしか流れない．TCP を用いた場合，自律的にエンド ノードが転送量を減少させる．しかし，UDP や他のトラフィックといった非適応型トラフィックは際限無く転送することが可能なため，回線を占有し，他のトラフィックを排除する．

DoS(Denial of Service) アタックはその一例である． DoS アタックは一般の利用者を装った特定のフローをサーバに送り付けることにより，サーバの過負荷を起こすかクラッシュさせ，他の利用者の要求に応えなくする攻撃を指す．DoS アタックはサーバだけでなく回線に過負荷を起こす要因であり，ボトルネックにおいてその帯域を占有する．

一般的にボトルネックとなりやすい場所

ラストワンマイルの広帯域化によりラストワンマイルはボトルネックではなくなった．現状ボトルネックとなっているのはラストワンマイルを束ねた上流の回線である．ISP においては，ISP のバックボーンとの接続を担う回線である．セグメントがいくつか集合し，一つの回線を共有するツリー型のネットワークにおいて対外線にあたる回線がボトルネックになりやすい．

同様の事が末端の組織ネットワークにもいえる．内部帯域は比較的安価で潤沢に用意できるのに比べ，対外線は高価で使用帯域に制限がある．企業や学校のネットワークが末端のネットワークの例である．図 2.3 にボトルネックとなる回線を共有する末端のネットワークを示す．

このようなネットワーク構成においてトラフィックがすべて適応型トラフィックならば特別な制御を必要としない．この場合，エンド ノードによるトラフィック制御によりボトルネックを公平に共有することができる．しかし現実にはインターネット上では UDP を用いたリアルタイム通信のトラフィックや動画配信を利用したトラフィックが考えられ，様々なトラフィックが混在している．これらのトラフィックに対しエンド ノードで独自にトラフィック制御する従来の方法は，遅延やパケット順序の入れ替わりが発生するため好ましくない．有効な手段としてエンド ノードではなく途中経路にあるルータによる通信品質保証があげられる．

末端ネットワークにおいてボトルネックとなる回線はの対外線である．末端ネットワークなど単一の対外線を共有する組織のツリー型ネットワークではボトルネックの対策を練らなければならない．

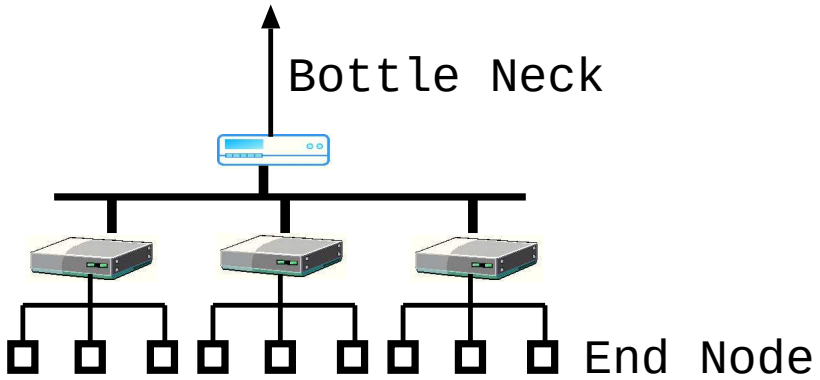

図 2.3: 末端のネットワーク構成の一例

2.2 既存のトラフィック優先制御

ボトルネックでは特定のトラフィックが回線を占有してしまい，他のトラフィックの通信品質を低下させる．この様な場合，例えば DoS アタックなどのボトルネックを占有するトラフィックに対し低い優先度をつけ，優先度の高い他のトラフィックを優先的に転送することで他のトラフィックを保護する．

以下にトラフィック優先制御機構のモデルと，トラフィック優先制御を実現するキューイング技術について述べる．

2.2.1 トラフィック優先制御機構のモデル

トラフィック優先制御は以下の 4 手順で行われる．

- アドミッション制御

  優先度の高いトラフィックに対しそれに見合った優先制御を行う充分な回線資源があるかどうかを判断する．不足している場合にはエラーを返す．

- クラシファイ/識別

  トラフィックの優先度を識別し，優先度別に分類する．IP ネットワークでは通常送り元 IP アドレス (source address)，宛先 IP アドレス (destination address)，送り元ポート番号 (source port)，宛先ポート番号 (destination port)，プロトコル (protocol)，計 5 つのうちいずれか，又はその組み合わせで分類される．

  他にはその検索コストを削減するために IP ヘッダの TOS フィールドを利用する方法がある．あらかじめ 5 つの組み合わせから 8 ビットの TOS フィールドにその優先度をマップしておくことで，ルータはトラフィックの分類を容易に行える．

- シェーピング/ポリシング

トラフィックのバーストを一定のレートに変える/規定以上の入力がある場合には破棄する．

- パケットスケジューリング

  優先度別にパケットの送出タイミングを設定する．

以上の手順を実現する手段としてキューイング技術がある．

2.2.2 キューイング技術

入口キューは回線の物理限界を越えたトラフィックを受けることがない．制御対象になりえるのは出口キューのみである．出口キューを制御するキューイング技術について述べる．
以下に代表的なキューイング技術をあげる．

- FIFO(First In First Out) Queueing

  標準でルータに実装されているキューである．キューに初めに入ったパケットが初めに送出される．単純だがパケットの種類にかかわらず転送するため，高速にパケットを転送することができるという利点がある．
  図 2.1 は FIFO キューの例である．

- Priority Queueing

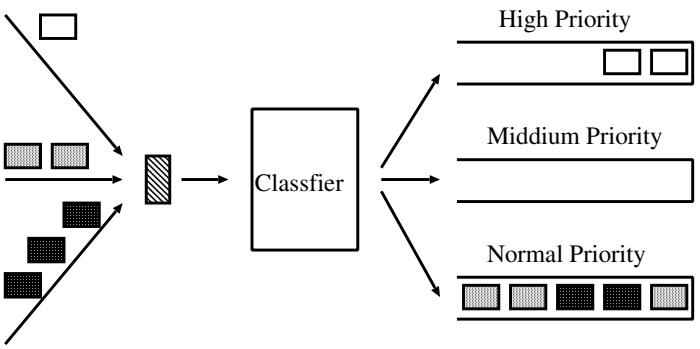

図 2.4: Priority Queueing

いくつかの優先度別に FIFO キューを用意する．優先度の高いキューから順番に送出され，優先度の高いキューが空になるまで優先度の低いキューからパケットが送出されることが無い．

図 2.4 に Priority Queueing の例を示す．入口キューから入って来たトラフィックに対しクラシファイアでパケットの優先度を判断し，優先度に対応する出口キューにパケットをいれる．

各パケットの優先度の判断を行うためルータに負荷がかかる．パケットスケジューリング上優先度の高いパケットがある限り他のトラフィックが送出されないため優先度の低いトラフィックが帯域資源を全く使えないという問題が発生する．

- Class Based Queueing[10]

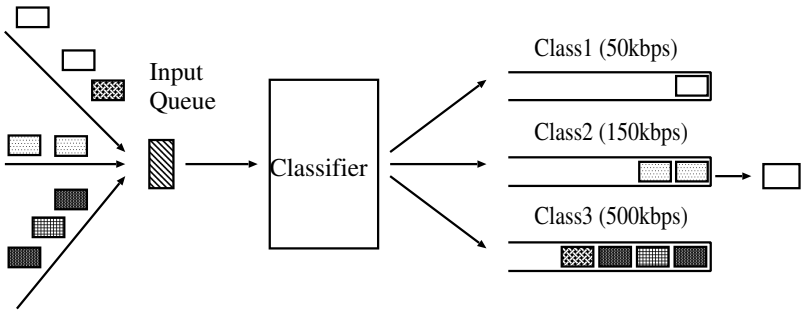

図 2.5: Class Based Queueing

トラフィックをいくつかのクラスに分類し，それらに対し FIFO キューを用意する．クラスごとにスループットを監視して、各クラスが占める帯域を調整する．

図 2.5 に CBQ の適用例を示す．クラシファイアはパケット毎にそのパケットがどのクラスに属するかを判断し，クラスに対応する出口キューにそのパケットを受け渡す．

CBQ では，クラス毎に最低限の帯域幅を確保するため Priority Queueing の様に優先度の低いトラフィックが枯渇する事はない．また回線の空きを検知した場合は割り当てられた帯域以上の帯域を使用することができる．Priority Queueing と同じくパケットの識別から優先度の判断という手順が必要なためルータへの負荷が増加する．

- Weighted Fair Queueing

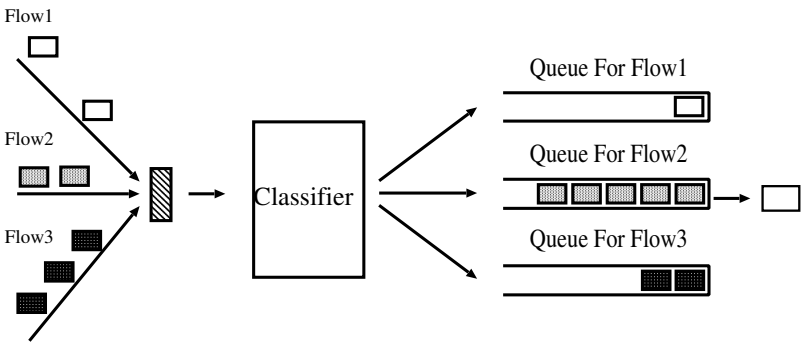

図 2.6: Weighted Fair Queueing

各フロー毎に独立した FIFO Queue を用意する．パケット送出タイミングに一貫性があるため非適応型トラフィックの影響を最小限に抑えることが可能である．WFQ では，キュー毎に重み付けをし，パケットスケジューリングすることで優先度の違うトラフィックに対し優先的に送出させる事も可能である．また WFQ では，使われていない帯域を他のキューへ公平分配することでボトルネックの帯域を有効活用する．
図 2.6 に WFQ の適用例を示す．フロー毎に出口キューを用意する WFQ では，クラシファイアが同じフローに属するキューを同じキューにいれる．

フロー毎にキューを用意する事がルータの多くの資源を使用してしまう事や，トラフィックをクラシファイするために 5 個のパラメータ全てを用いなければならず処理のオーバーヘッドが非常に大きい．よって実装ではフロー毎のキューではなく固定数のキューに割り振られるうえ，パラメータは部分的にだけ適用する．

2.2.3 キューイング技術の適応

キューイング技術を導入し，ボトルネックの利用を改善させる例を示す．

特定のフローの帯域を厳密に確保したい場合

ビデオ会議を二つの末端ネットワークを接続して行いたい場合があてはまる．パケットロスを発生させること無く，指定された時間にビデオ会議のデータを送信する事が可能な帯域幅を確保する必要がある．
こういった場合一つの解法として CBQ を導入する．ビデオ会議が行われる時間にだけ必要な帯域を確保できるキューを用意し，これを優先的に転送させる．

人による複雑な管理を必要とせず，帯域の占有を防ぎたい場合

特定のフローを保護する必要は無いが，特別な設定をせずにボトルネックの帯域占有を事前に防止する需要がある．
WFQ ではユニークなトラフィックを発見した場合に別のキューにそのパケットを送るので特別な設定は必要無く，全てのフローがボトルネックを使用できる環境を構築することができる．

実装例

実際に運用されているシステムの多くには FIFO キューのみ実装されている．数少ない実装例のうち，広く利用されているものに ALTQ(Alternate Queueing)[?]，代替キューイング機構がある．1997 年に公開され，以降多くの代表的なキューイング機構を利用できる研究用プラットフォームとして利用されている．

2.3 本章のまとめ

インターネットではトラフィックの集中，帯域格差などの理由からボトルネックとなる回線がなくならない．ボトルネックにおいてその回線資源を独占する様なトラフィックは他のトラフィッ

クの通信品質を著しく低下させる．このような場合にトラフィック優先制御が必要である．
トラフィック優先制御を実現する手段としてキューイング技術の利用がある．

第3章　一般的末端ネットワークにおけるトラフィック優先制御とその問題点

本章では既存のトラフィック優先制御機構について検討し，その問題点を明らかにする．

第 2 章で述べた一般的なネットワーク構成において正規トラフィックを圧迫する非適応型トラフィックを制限し，正規トラフィックを保護する需要がある．しかし現状のトラフィック優先制御機構ではこれを実現できない．トラフィック識別をできない場合があるからである．

以下にトラフィック識別のできない場合を示す．

3.1　トラフィック識別に失敗する場合

トラフィック優先制御を行う際，必要な手順としてトラフィックの識別がある．通常宛先 IP アドレス，宛先ポート番号，送り元 IP アドレス，送り元ポート番号プロトコルの 5 つのパラメータを利用する．これらパラメータから優先トラフィックと非優先トラフィックを識別し差別化するため，これらに変更を加えた場合に識別できない．変更を加える機構として NAT や IPv4-v6 トランスレータがあげられる．これらは送り元 IP アドレス，ポート番号の情報を隠蔽化し，ボトルネックでトラフィックの識別を部分的に不可能にする．

3.1.1　トラフィック情報を隠蔽化する機構

以下にトラフィック情報を隠蔽化する機構について述べる．

NAT(Network Address Translator)[11]

フローの始点と終点の IP アドレスを変換する仕組みで，一般的にはプライベート IP アドレスをグローバル IP アドレスに変換する，またはその逆を行う．

グローバル IP アドレスとは IPv4 アドレスの内，ユニークであることが保証される IP アドレスである．プライベート IP アドレスとは反対に誰でも利用できる IP アドレスで，外部から触れる事のできない内部ネットワークなどに使う．

NAT はグローバル IP アドレスを持ったネットワークとプライベート IP アドレスを持ったネットワークを相互通信可能にする仕組みで，IP アドレスの不足を補うために使用される場合がある．IPv4 アドレスは数に限界があることが明白で，末端のネットワークにノード数に対応した IP アドレスを取得可能とは限らない．よってアドレス不足を補う手法として用いられる．

NAT を導入したネットワークでは WAN 側からは一つの IP アドレスからパケットが届いている様にしか見れないため，WAN 側から LAN 側に接続する手段が限定する事が可能だ．よって内部ネットワークである LAN と広域ネットワーク WAN を分別し，セキュリティを高めるために導入されている場合も多い．

図 3.1 に NAT で行われる処理について示す．ノード 1，ノード 2，ノード 3 で構成されるネットワークで，ノード 1 はプライベート IP アドレスを持ち，ノード 3 はグローバル IP アドレスを持つ．通常ノード 2 は NAT を導入したルータ，NAT box と呼ばれる．ノード 1 からノード 3 にデータ転送を行う場合には，ノード 2 において送り元 IP アドレスおよびポート番号をノード 2 の情報に書き換える．

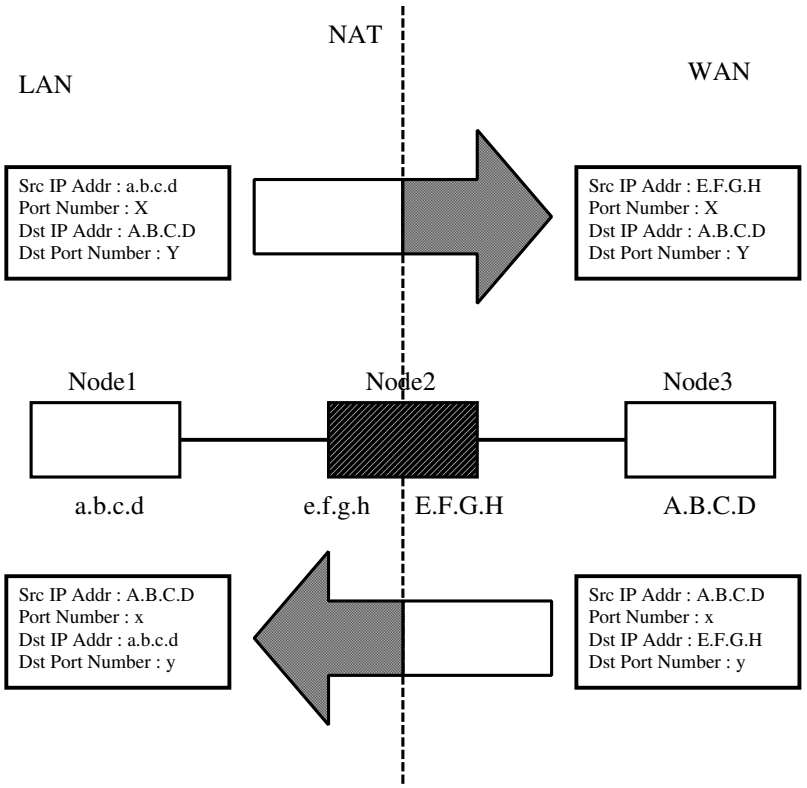

図 3.1: NAT

NAT の実装されたルータではフローを WAN 側，LAN 側に送出する際，アドレスの変更を行う．アドレスの変更はルータ上でアドレス変更表を参照し，対応するアドレス情報を入れ換える．

IPv4-v6 トランスレータ

NAT と同じくトラフィックの始点と終点の IP アドレスを変換する機能を指すが，IPv4 アドレスと IPv6 アドレスを変換するという点で違う．

現在主流である IPv4 はそのアドレスの枯渇が危惧されている．1995 年 12 月に行われた IETF では IP アドレスは 2018 年近辺に枯渇するだろうといわれていた．しかし，2000 年 12 月の行われた Global IPv6 Summit においてその予測は大幅に前倒しされ，2006 年には IPv4 ネットワークでユニークなアドレスを割り振る事が不可能になるという結論に至った．

家電製品や携帯電話などをインターネットに接続する需要のある昨今，潤沢なアドレス空間を持つ．IPv6 への移行は必要不可欠である．現在最も利用されている IPv4 アドレスと IPv6 アドレスでは形式が違うため直接通信することができない．よってそれらアドレスを変換して通信可能にする機構，IPv4-v6 トランスレータが必要である．

図 3.2 に IPv4-v6 トランスレータの存在するネットワークを示す．トランスレータは IPv4 アドレスを持つネットワークと，IPv6 を持つネットワークを相互通信可能とする．

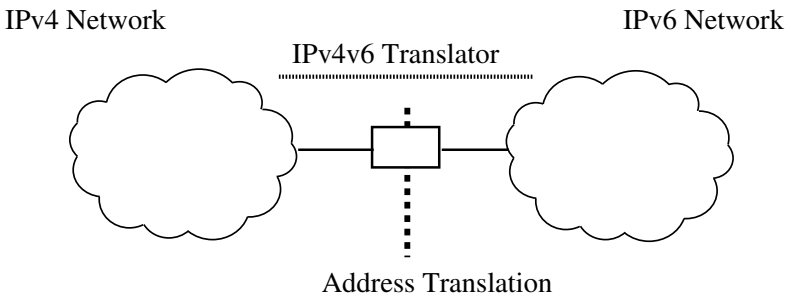

図 3.2: IPv4-v6 トランスレータ

3.1.2 NAT などトランスレータを導入した末端ネットワーク

一般的な末端ネットワークでは IP アドレスの制約，セキュリティなどの理由から NAT や IPv4-v6 トランスレータを導入している例が多い．またボトルネックとなる対外線を組織の部署などにセグメント分けされていることが考えられる．

3.2 トラフィック優先制御に失敗する具体例

第 2 章で述べた一般的な末端ネットワークは図 3.3 のようになる．この様なネットワークおいて内部ネットワークである上位ルータと下位ルータ間，および下位ルータ下のローカルネットワークは潤沢な帯域資源を用意される事が予想される．それに対し，外部とを繋ぐ回線は一般的に低帯域かつ高価格である．よって対外線はボトルネックとなる．図:generalnet ではボトルネック回線は up link として示されている回線になる．

明白な帯域格差のある回線を特定のフローが占有した場合，他のフローの通信品質を極端に悪化させる．一般的にボトルネックに直接面するルータにおいてトラフィック優先制御を行う．

図 3.3 に末端のネットワークにおいて非適応型フローが他のフローを侵害する例を示す．ボトルネックに直接面するルータ C，下位ルータ A，下位ルータ B で構成されるネットワークで各下位ルータ下に 2 台エンド ノードがある．エンド ノード 4 から発生するフロー 4 によりボトルネックが占有されている．結果ルータ C 上のアウトプットキューは tail drop を起こしフロー 1，2，3 の通信品質を低下させる．

通常，非適応型フローであるフロー 4 をルータ C で識別し，他のフローを優先的に処理する，又はフロー 4 に対し制裁を加え，意図的に通信品質を低下させることでフロー 4 が他のフローにあたえる影響を最小限にする．

非適応型トラフィックに制裁を加える方法は明らかに攻撃である，あるいは害があると判断できる場合には有効である．しかし実際には特定の害のあるトラフィックを識別する手法は確立されておらず，Fair Queueing や Weighted Fair Queueing を用い，フロー毎にキューを用意することによって対処する．

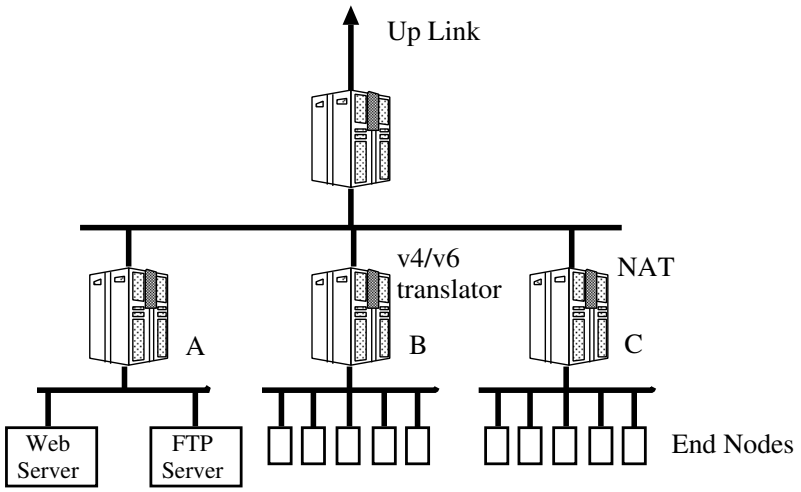

図 3.3: NAT や IPv4-v6 トランスレータを導入したネットワーク

3.3 現状のトラフィック優先制御機構で対処不能な場合

NAT や IPv4-v6 トランスレータの存在するネットワークにおいてボトルネックでのトラフィック優先制御は機能しない場合がある．NAT や IPv4-v6 トランスレータは送り元 IP アドレス，ポート番号といった情報を変更してパケットを転送するのでこれらの情報を必要とするトラフィックの識別に支障を来す．例として送り元 IP アドレスを識別子としてトラフィック優先制御を行う必要のある場合を示す．

3.3.1 一つのノードによる多数宛先に対するデータ送信

図 3.5 に一つのノードによる多数宛先に対するデータ送信を行う場合の例を示す．

ノード 1，2 の 2 台でボトルネックを共有している．ノード 2 は 10 種類の通信，フロー 2 からフローの送出を同時に行い，ノード 1 はフロー 1 を持つ．ボトルネックの帯域使用率が 100%に近い状況にあり，回線をノード 1 から発生するトラフィックが占有している．

この場合，一般的な解法として WFQ を導入する．WFQ ではフロー毎のキューを用意するのでフロー 1 用にキューが用意され回線の独占を防ぐことができる．図 3.6 に図 3.5 の中間ノードに WFQ を導入した際行われる処理を示す．

しかし，結果としてボトルネックの多くの帯域をノード 2 の通信に利用され，フロー 1 のスループットに対し悪影響を与えることにかわりがない．

3.3.2 一般的なサービスに対する大量データ送信

図 3.7 に一般的なサービスに対する大量データ送信を行う例を示す．

同じく 2 台のエンドノードがボトルネックを共有している．ノード 1，2 は共に同じサービスを利用し，ノード 2 はノード 1 と同じ宛先に対し，大量のデータを送信している．

宛先 IP アドレスとポート番号，プロトコルから判断するとこれらフローは同一のものとして扱われる．この場合これらのパラメータを利用した CBQ などのトラフィック優先制御は機能しない．

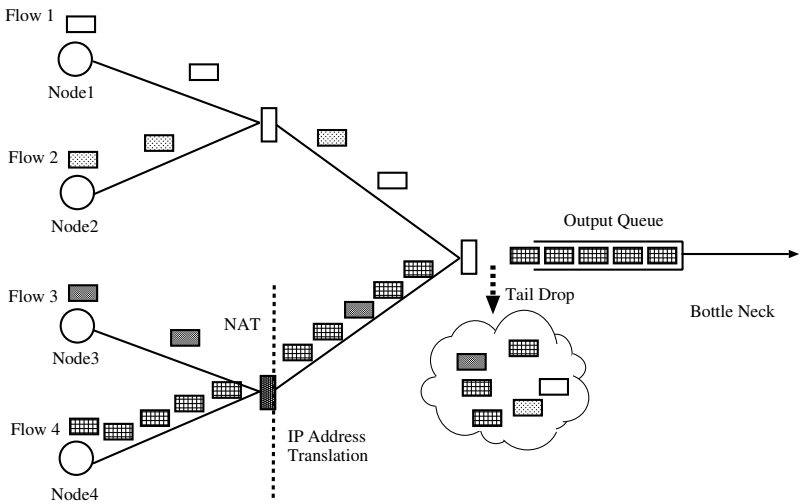

図 3.4: 特定フローによるボトルネックの占有

この様な場合，有効なトラフィック優先制御としては WFQ の導入が考えられる．しかし WFQ では実装する際，トラフィック識別に利用するパラメータを限定している．有力な実装例である ALTQ では宛先 IP アドレス毎のキューを用意することになっている．よってこれらトラフィックは全て同じ優先度と判断される．結果 WFQ では対処できない．

3.4 問題点のまとめ

一般的末端ネットワークでは対外線がボトルネックとなりやすい．ボトルネックでは非適応トラフィックが帯域を占有してしまう可能性があるためトラフィック優先制御を導入すべきである．しかし NAT や IPv4-v6 トランスレータといったパケットの情報を変更してしまう機構がトラフィック優先制御の支障となる．

NAT などのトランスレータを含む末端ネットワーク上で有効なトラフィック優先制御機構を作成する必要がある．

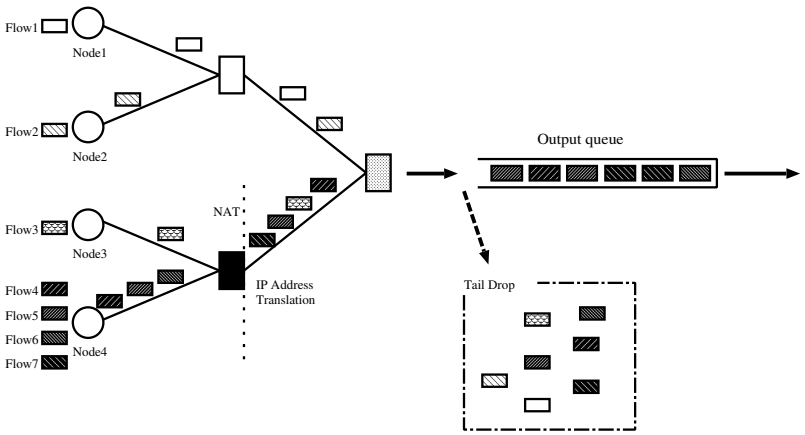

図 3.5: 一つのノードによるボトルネック占有

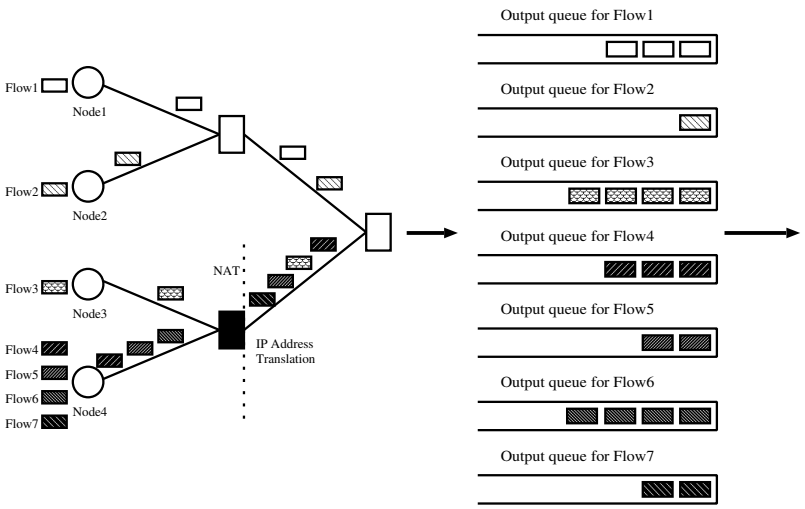

図 3.6: WFQ の適用

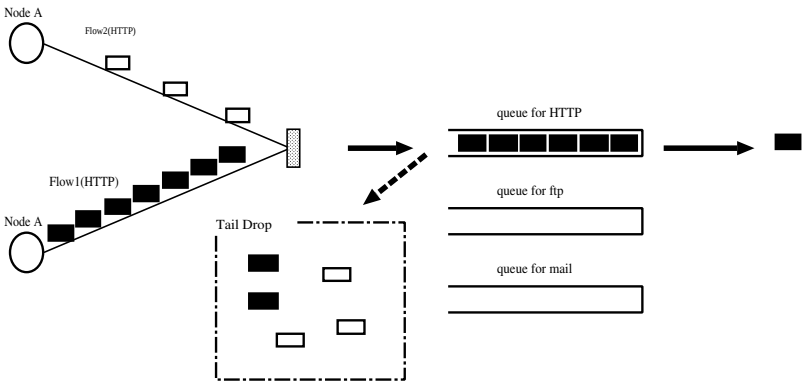

図 3.7: CBQ の適用

第4章　設計

本章では本論文で述べた問題点に対応する解決方法の設計について述べる．まず解決方法のアプローチについて述べ，その正当性について説明する．また解決方法に必要な機構とその仕組みについて説明する．

4.1　解決方法のアプローチ

一般的な末端ネットワークの構造を想定する．

ボトルネックに接するルータではなく，下位ルータである NAT や IPv4-v6 トランスレータでトラフィックの識別する．また各ノード毎の重み付けによるトラフィック優先制御を行う．結果情報が隠蔽化された環境において公平なトラフィック優先制御が可能となる．

4.2　トラフィック識別手法

送り元 IP アドレス，ポート番号を隠蔽化する NAT や IPv4-v6 トランスレータの存在するネットワークにおいて，トラフィックを識別する方法は 2 通り考えられる．識別をネットワークの 1 ノードで行う中央集権型識別手法，および下位ルータによる分散型識別手法である．

4.2.1　中央集権型識別手法

本研究では末端のネットワークにおいて，1 つのルータでトラフィックの識別を行う方法を中央集権型識別手法と呼ぶ．

従来のボトルネックルータによるトラフィック識別は，他のノードによる識別を行わない中央集権型識別手法である．NAT や IPv4-v6 トランスレータが変更したトラフィック情報を変更前の情報へ変換する方法は無い．よってなんらかの手法を用い，アドレストランスレータ下のトラフィック情報を採取し，ボトルネックルータに到達したトラフィックの情報とマッピングを行う必要がある．

アドレストランスレータはパケットの送り元 IP アドレスと送り元ポート番号の情報を変換してパケットを転送する．よって変更前と変更後の情報をマッピングできる対応表を格納しているノードはアドレストランスレータである下位ノードのみである．

トラフィック情報の変換対応表をボトルネックルータに渡す機構が必要である．同機構はアドレストランスレータ機構から変更対応表を抽出し，ボトルネックルータに送信する．図 4.1 は一般的末端ネットワークにおいて中央集権型識別手法を利用する流れを示す．

対応表を受理したボトルネックルータはトラフィック情報の表を二つ保持する．ボトルネックルータにおいて識別したトラフィック情報と下位ルータの変換対応表である．

次にボトルネックルータにおいて識別されたトラフィック情報から，送り元 IP アドレスがアド

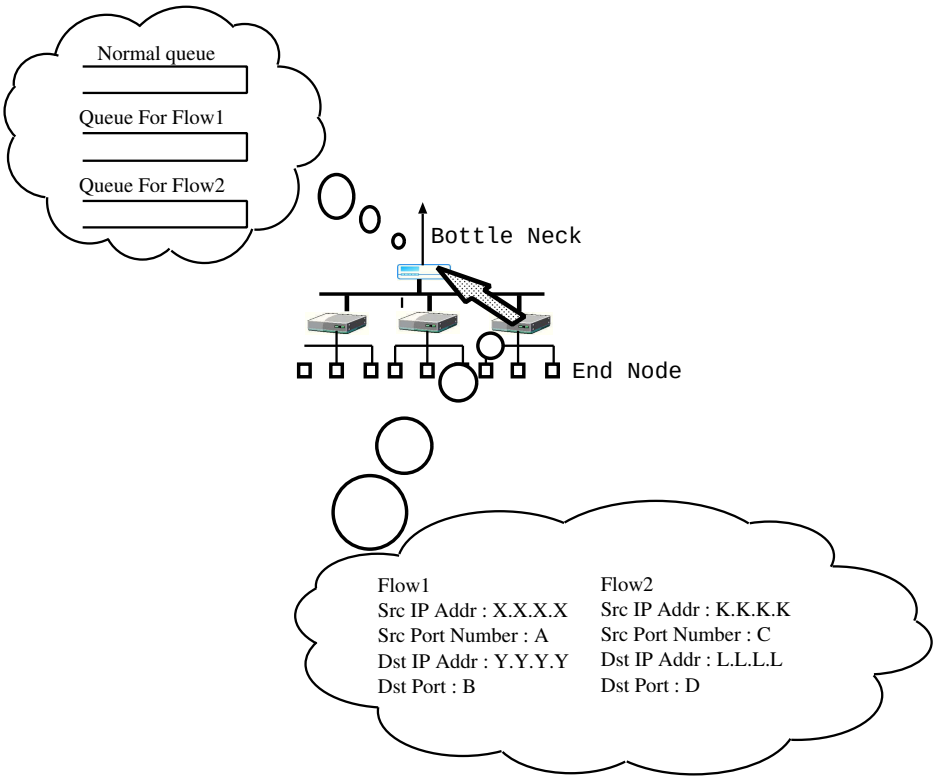

図 4.1: トラフィック情報の変換対応表をボトルネックルータへ渡す

レストランスレータである場合を抽出する．変換表を参照し，変換前の送り元 IP アドレスおよびポート番号をアドレストランスレータのそれらと交換し，新たにトラフィック情報の表を作成する．図 4.2 にトラフィック情報変更前と変更後の情報を組み合わせて表を作る例を示す．

	Dst. IP Addr.	Dst. Port Num	Src. IP Addr.	Src. Port Num	Proto.		Orig. IP Addr.	Orig. Port Num
Flow 1	X.X.X.X	80	a.a.a.a	8011	TCP		A.A.A.A	2080
Flow 2	Y.Y.Y.Y	22	b.b.b.b	8045	TCP		B.B.B.B	6059
Flow 3	Z.Z.Z.Z	25	c.c.c.c	8069	UDP		C.C.C.C	3024

図 4.2: 変更前の情報と変更後を組み合わせた表の作成

ボトルネックルータでトラフィックを識別する中央集権型手法は複雑である．従ってボトルネックルータに負荷がかかる事が短所となる．通常 NAT などのトランスレータはトラフィック情報の変更時，各トラフィックに比類の無いポート番号を指定する．よってトラフィック情報の変換対応表を新規トラフィック発生時にボトルネックルータへ送信する事が期待される．仮に新規トラフィックの発生をトラフィック情報の更新のトリガーとする場合，トラフィック数に比例した回数の更新をしなければならない．多数の更新は帯域資源，CPU 時間などの計算機資源を浪費する．

現実的な解法として実時間をトリガーとした更新を行う．ある一定期間毎にトラフィック情報を更新する．この解法はトラフィックが恒常的発生する場合において効果を発揮する．しかしトラフィックの転送時間は様々で 1 秒に満たない場合もある．図 4.3 にルータによる処理が無駄にな

る場合を示す．フロー 1，2，3 の中央集権型識別手法を用いて識別する．フロー 1 の情報を下位ルータから摂取する前にフロー 1 は終了している．

この様な場合，トラフィック情報の更新時には既に更新されたトラフィックは完了している可能性があり，ルータによる処理に無駄が生じる．フロー 1 に対するトラフィック制御は実際に機能しない．

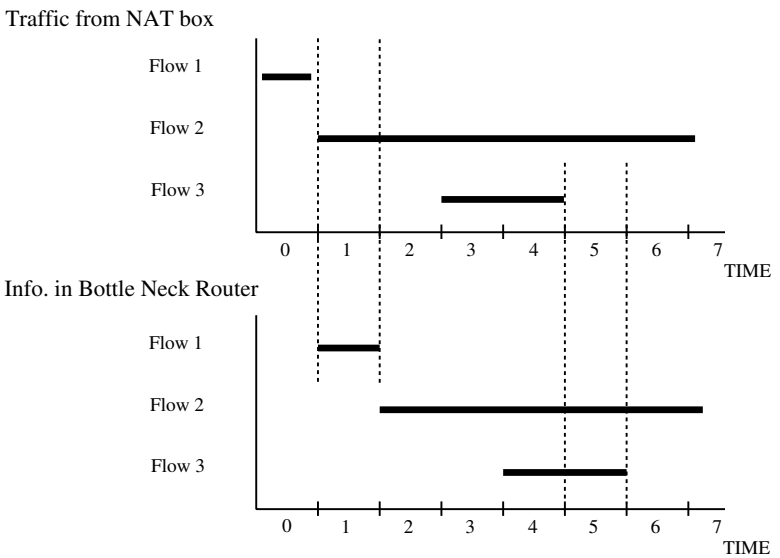

図 4.3: 実時間をトリガーとした更新に生じる無駄

また，トラフィック情報の変更対応表の更新にはルータ同士の通信が必要不可欠である．その際利用される制御パケットはトラフィック識別の対象となるパケットと区別がつかない．そのため，制御パケットを識別対象にふくまれないように，識別機構に仕様変更を加える必要がある．

識別されたトラフィック情報をトラフィック優先制御機構に適用するのが容易であると言う長所がある．

NAT ルータや IPv4-v6 トランスレータはトラフィックの識別を行い，トラフィック情報の変更を行っている．そのため上位ルータ上でトラフィックの識別を行う事はルータに無駄な処理を行わせる事となる．

4.2.2 分散型識別手法

個々の NAT ルータや IPv4-v6 トランスレータルータにおいてトラフィックの識別を行う手法を分散型識別手法と呼ぶ．従来のトラフィック識別はボトルネックルータのみで行われていたため，下位ルータはその機能を持たない．よって同機能を下位ルータに設置する必要がある．

下位ルータを通過するトラフィックに情報を変換されたパケットは無い．よってトラフィック識別を困難にする要素が無い．

図 4.4 に下位ルータによるトラフィック識別を示す．

この手法は既存のトラフィック識別機構を用いるため，特別な設計が必要無いという利点がある．

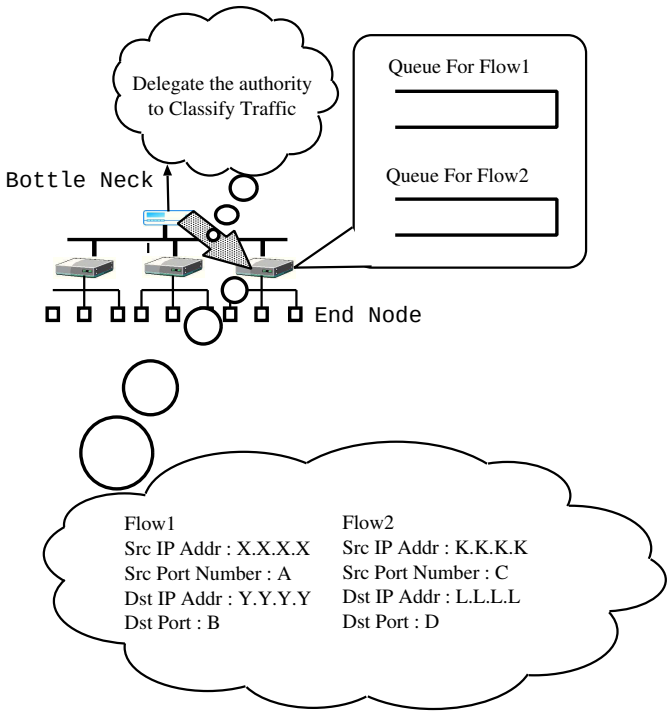

図 4.4: 分散識別手法

4.2.3 本研究で採用した識別手法

中央集権型識別機構はトラフィック識別に必要な情報をボトルネックルータに集結させるため，CPU 資源と回線資源の無駄が生じる．また下位ルータである NAT や IPv4-v6 トランスレータは既にトラフィックの識別を行っているため，更に上位ルータで行う必要は無い．

本研究ではトラフィック識別を行う際，障害の少ない分散型識別手法を採用した．

4.3 キューイング技術の適用

トラフィック優先制御に必要な手順であるトラフィック識別を行った後に，キューイング技術を適用する．キューイングは上位ルータで行う手法，および下位ルータで行う手法がある．

4.3.1 上位ルータによるキューイング

下位ルータで識別されたトラフィック情報を基に上位ルータでキューイングを行う．識別されたトラフィック情報は上位ルータに転送される必要がある．

通常のキューイングでは入口キューから入力されたパケットが classifier を通り，識別される．しかし，上位ルータで識別する事が不可能なため，下位ルータで識別された情報を利用し，classifier が機能する．下位ルータからトラフィック情報を上位ルータに転送する際，タイムラグが生じる．図 4.3 にタイムラグによっておこる無駄が示されている．

またネットワークの規模が大きくなる場合にトラフィック情報が肥大化すると言う問題が生じ

る．仮に 250 個のエンド ノードを擁するネットワークであった場合，各エンド ノードが 2 種類の通信を同時に行ったならば管理しなければ成らないトラフィック情報は 500 個に膨れ上がる．特定のトラフィック情報を抽出する場合にも下位ルータから転送された情報からマッチングする必要がある．多数のフローの状態を把握しつつキューイングを行う事はルータにとって大きな負荷でしかない．

上位ルータによるキューイングは，ボトルネックに直接面するルータであるため，ボトルネック回線資源の状態を把握するのに最適であるという利点がある．

4.3.2 下位ルータによるキューイング

下位ルータで識別した情報を基に下位ルータ上にキューイング技術を適用する．上位ルータにそれを適用する場合と事なり，タイムラグが発生することがない．

また下位ルータでは上位ルータと比較して包容するエンド ノード数が少ない．よってルータに与える負荷は相対的に低い．

一般的にボトルネックルータでトラフィック優先制御が行われる．これはボトルネックの状態を直接観測する手段としてボトルネックルータ上の出口キューの状態を観測する事が最適だからである．下位ルータではボトルネック回線資源の状態を把握する事ができない事が短所である．

また下位ルータは上位ルータと違い，複数台ある．知る術の無いボトルネック回線資源を複数台の下位ルータで共有するのは難しい．

4.3.3 本研究で採用したキューイング方法

下位ルータによるキューイングはボトルネックの回線資源を察知する術がない事が欠点である．しかし，上位ルータによるキューイングではその処理に大きな負荷がかかる．また下位ルータで把握したトラフィック情報を上位ルータに転送する際にタイムラグが生じてしまう．よって本研究ではボトルネックにおける回線資源を下位ルータに通知し，下位ルータによるキューイングを行う事が最適と判断し，本手法を採用した．

4.4 ボトルネック回線資源の状態感知

下位ルータでトラフィック優先制御を行うには，上位ルータから下位ルータへボトルネックの状態を伝達する必要がある．またツリー状のネットワークでは下位ルータが複数台存在するので，下位ルータ同士でボトルネック資源を分割する仕組みを用意する必要がある．ボトルネック資源を分割するには，利用されていない回線資源を把握しなければならない．

回線資源として分配可能なパラメータは帯域である．よって上位ルータから帯域を割り振り，下位ルータに伝える事で，下位ルータが使用できる帯域を感知する事ができる．下位ルータによるトラフィック優先制御に必要なボトルネックの帯域を感知する手法は 4 通りある．下位ルータ同士の通信による感知，SNMP(Simple Network Management Protocol)[12] を利用する方法，上位ルータから下位ルータへ回線状態を通知する方法，および上位ルータから下位ルータ毎の回線資源上限を通知する場合である．

4.4.1 下位ルータ同士の通信による感知

ボトルネック回線資源を下位ルータに分割する場合，下位ルータ同士で通信し，必要な情報を交換する事で帯域の割合を決定する手法がある．

下位ルータ上でボトルネック側インターフェースにおける出口キューを監視する．使用している帯域幅の情報を下位ルータ同士で交換し，ボトルネック帯域資源の状態を判断する．

ボトルネックの帯域上限から他ルータの帯域使用量，自ルータの帯域使用量の合計値を差し引いた値を使用可能な帯域資源と判断する．

ボトルネックの回線使用率が低く，パケットロスの発生しにくい環境である場合には使用可能帯域を把握することが可能である．しかし，トラフィック優先制御を必要とするボトルネックの帯域資源をほぼ完全に使用している場合，または下位ルータの帯域使用量合計がボトルネックの帯域上限を越えた場合に，各下位ルータで利用可能な帯域幅を分散して計算しなければならず，不整合が生じる場合がある．トラフィック識別の場合と同じく，時間軸に合わせた情報更新しかできないためである．

また，ボトルネックを通過しないトラフィックを例外に処理する事が困難であるという欠点がある．

4.4.2 SNMP を利用する場合

SNMP はネットワークから通信機器を管理するためのプロトコルである．SNMP を利用する事でボトルネック回線資源の状態を把握することができる．

SNMP では下位ルータがボトルネックルータの回線資源の状態をネットワーク経由で検知する．

SNMP は宛先 IP アドレス毎やポート番号，送り元 IP アドレス別にトラフィック量を取得する場合においては最適である．しかし，下位ルータ毎のトラフィック量を抽出するのが難しいという問題点がある．

4.4.3 上位ルータから下位ルータへ回線状態を通知する場合

ボトルネックに面する上位ルータでボトルネックの回線状態を把握し，下位ルータに通知する．

上位ルータの入口キューで下位ルータ毎のトラフィック量を監視する．出口キューでは実際にボトルネックを通過するトラフィックしか観測できないためパケットロスしたトラフィックを含めた監視ができない．

上位ルータから各下位ルータへ使用状況を通知する場合，その後に下位ルータ同士で使用可能帯域を譲り合う仕組みが必要である．

帯域の利用状況は上位ルータで観測されるため，時間によって不整合が生じることは無い．しかし使用可能帯域の決定は各ルータの判断にまかされるため，ここに不整合が生じる可能性がある．

4.4.4 上位ルータから下位ルータ毎の回線資源上限を通知する場合

上位ルータによるトラフィック量の監視を行い，下位ルータ毎の帯域上限を設定する方法である．下位ルータ同士の通信が必要ないため，使用可能帯域の決定に不整合が生じる事が無い．

また帯域上限を利用するキューイング技術に適用するのが容易であるという利点がある．

4.4.5　本研究で利用するボトルネック回線状態を感知する仕組み

下位ルータにおいてキューイング技術を適用する際，上位ルータから使用可能帯域を直接通知される方法では使用可能帯域の決定に不整合が生じる可能性がない．

本研究では上位ルータから下位ルータ毎の回線帯域の使用上限を通知する仕組みを導入する．

4.5　下位ルータでキューイングする際必要な制御

ルータ毎に分配可能な回線資源として帯域があげられる．よって下位ルータへ使用可能帯域を優先制御ポリシとして伝達する．本研究では以上の行為を優先制御ポリシのデリゲートと定義する．

4.5.1　帯域の割り振り

ボトルネックの帯域を下位ルータで共有する際，ボトルネックの帯域限界以上の値を下位ルータへ伝達する事は無意味である．また下位ルータ毎，ネットワークセグメント毎に優先度は異なる事が考えられる．従ってボトルネックの帯域限界以下の帯域幅でさらに下位ルータの優先度により帯域の割り当てを決定する．この手法は仮想帯域 (バーチャルライン) を下位ルータに提供する事になる．

4.5.2　バーチャルライン

ボトルネックにおいて仮想的に割り振られた帯域幅を持つことにより，下位ルータはバーチャルラインを持つ事になる．バーチャルラインとは仮想回線を指す．

図 4.5 にバーチャルラインを示す．

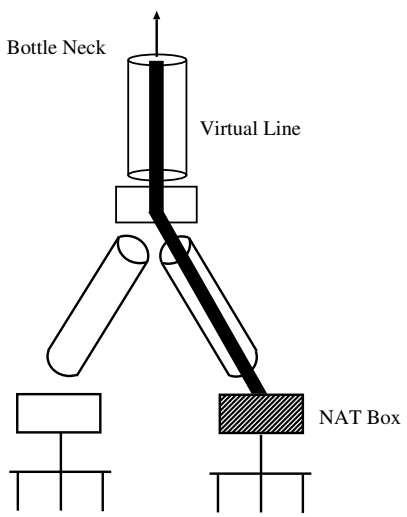

図 4.5: バーチャルライン

バーチャルラインの欠点としてボトルネックの帯域を有効的に利用できないという問題がある．本来ボトルネックルータにおいてトラフィックの優先制御を行うことでボトルネック帯域の限界

まで利用することが可能だった．しかしボトルネックではない下位ルータにバーチャルラインを割り当てることで下位ルータは割り当てられた以上のトラフィックは破棄される．これによって他の下位ルータの利用率が割り当てられた値に達しない場合にはボトルネックの利用率が 100%に達することがないという問題が生じる．

バーチャルラインの欠点を補うため使用率に応じて下位ルータ毎に帯域の借り入れを行う必要がある．

4.5.3 帯域を割り当てるための要素

帯域は下位ルータ毎の優先度で決定される．下位ルータ毎の優先度は収容するエンド ノードの重み付けにより決定する．下位ノードで集計されたエンド ノードの重み付けを優先度として上位ルータに通知し，通知された優先度とボトルネックの帯域限界から上位ルータは下位ルータの帯域割当を計算する．割り当てを決定する要素としてボトルネックの帯域と合計されたエンド ノードの優先度，およびその割合を利用する．

図 4.6 に下位ルータ毎の優先度決定の流れを示す．下位ルータはそれぞれのセグ メントに存在するエンド ノードの優先度を合計し，上位ルータに報告する．上位ルータは報告された優先度から各下位ルータに割り当てる帯域を決定する．

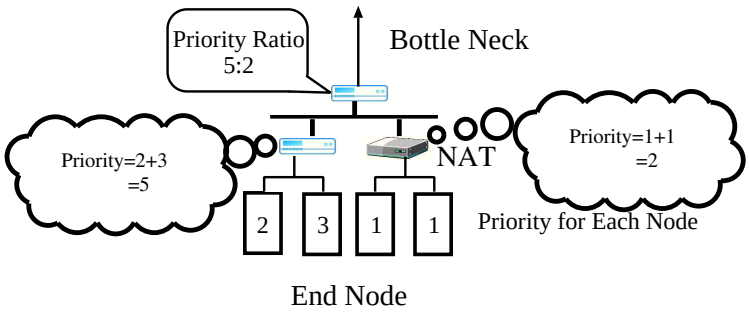

図 4.6: 下位ルータの優先度決定

4.5.4 割り当ての計算

ボトルネックの帯域を各下位ルータにおける優先度の割合で分割する．しかしトラフィック量が少ないバーチャルラインを作る事は回線帯域を無駄にしてしまうので上位ルータは分割を決定しても下位ルータにそれを通知するべきではない．優先度の割合で決定された帯域は保証されるべき帯域幅ではあるが，常にこの値が上限となるべきではない．

よって計算された値を目安値とし，越えた場合に借り入れ，少ない場合には貸し出しのできる仕組みにする．そのためになんらかのトリガーが必要である．そして各下位ルータ毎の帯域割り振りを決定するためのトリガーは下位ルータ毎のトラフィック量になる．トリガーが発生しない限り帯域制御を行わない．また目安値以下の割り当ては算出されないために，ボトルネックの有効活用と下位ルータ毎の帯域保証を行うことが可能となる．

4.5.5　割り当てのタイミング

トラフィック量をトリガーとする場合，そのタイミングを決定する必要がある．下位ルータ毎の流量を取得するタイミングは短時間であるほどより細かで柔軟な制御を行うことが可能である．しかしあまりに短い間隔で変更を加えることでシステムへの負荷が上昇するほか，隣接ルータとの情報のやりとりに使われるトラフィックが多くなることが懸念される．よってこれらの要素を考慮した最適なタイミングを模索する．

4.5.6　割り当ての実行

上位ルータから通知された帯域の割り当てをボトルネックにもつバーチャルラインだと仮定し，使用帯域を制限する仕組みが必要である．こういった帯域の制御にはキューイング技術を適用する．またキューイング技術を各ノードの重み付けに対応させる必要がある．

トラフィック量をトリガーとして帯域制限が決定され，その値は変化する．従って動的にキューイングのパラメータを変更する必要がある．

WFQ ではフロー毎に重み付けを行いトラフィック優先制御を行うことが理論的に可能な事を証明した．しかし実装ではトラフィック識別に用いるパラメータは部分的になり，一般的に用いられるのは宛先 IP アドレスである．宛先 IP アドレスではノード毎の優先度を扱えないため送りもと IP アドレス毎の制御を行うキューイング技術を用いる．また WFQ では使用帯域上限を物理回線の限界に設定するため本研究には適用できない．

4.6　ルータ同士の情報交換

ネットワークの増設やノード数の変更，各ノードの優先度の変更などがあることから，下位ルータは各ノードの優先度を定期的に上位ルータへ報告する必要がある．上位ルータは下位ルータへ帯域制限を通達する仕組みを持つ．ルータ同士でトラフィック優先制御に必要な情報を交換する．

4.7　全体像

図 4.7 に本機構の機能と，その流れを示す．

図中のトラフィックを監視しているルータが上位ルータ，キューイング技術を適用しているルータが下位ルータである．

上位ルータは下位ルータから発生するトラフィックを監視し，下位ルータから送られた優先度から各ルータへの帯域制限を判断して帯域制御命令を出す．その際，目安値と実際の制限値をきりわけて使うことでボトルネックの帯域を有効に使い，かつ最低限の帯域保証を行うことが可能なバーチャルラインを構築する．

4.8　必要な機構

ルータ同士で通信し，帯域を分割する手法を用いて下位ルータにおいてトラフィック優先制御をおこなう．これらを実現する際必要な機構を述べる．

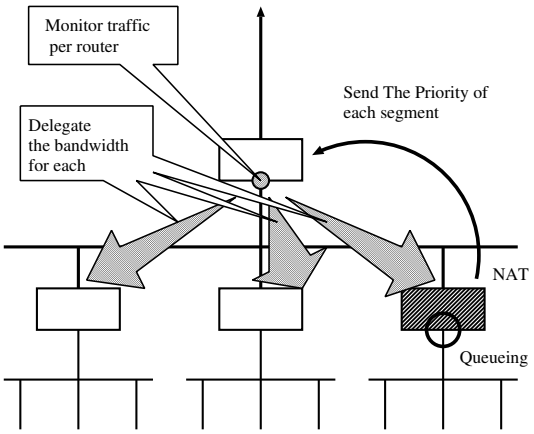

図 4.7: 全体像

- 上位ルータ
  各下位ルータ毎のトラフィックの監視を行う機構
  帯域の割り当て計算を行う機構
  下位ルータへ帯域制限を伝達する機構
  下位ルータから優先度を受理する機構

- 下位ルータ
  上位ルータから帯域制限を受け取る機構
  上位ルータへ優先度を伝達する機構
  ノード毎の帯域制御を行う機構
  キューイングパラメータを変更する機構

4.8.1 関連研究

本機構とおなじ環境において，中央集権型識別手法を用いた研究に kuma-ayame project[13][14] がある．

kuma-ayame project では NAT や IPv4-v6 トランスレータを含むネットワークにおいて，トラフィック優先制御を行う事を可能とした．手法として上位にフロー毎の優先度を管理するノードを用意し，当ノードにおいてトラフィックの識別情報から対応する優先度をボトルネックルータに通知する方法を用いた．

第5章　実装

本章ではNATやIPv4-v6トランスレータの存在するネットワークにおけるトラフィック優先制御機構の設計に基づいた実装について述べる．

5.1　開発環境

表5.1の環境で開発を行った．

表 5.1: 開発環境

CPU	Intel Pentium III 1GHz
OS	FreeBSD-4.4-release[15]
言語	C言語
使用ライブラリ	libpcap,libaltq
ルーティング機構	Zebra0.96a OSPFd
キューイング機構	ALTQd

以下に実装した機構を述べる

5.2　トラフィック優先制御を構成する機構

5.2.1　下位ルータ毎のトラフィック監視を行う機構

上位ルータにおいてボトルネック側のインターフェースを上流インターフェース，下位ルータ側を下流インターフェースとする．下流インターフェースにおいて下位ルータ毎のトラフィック量を監視する機構の実装は二通り行った．

一つはlibpcapを用い，レイヤ2の情報であるEthernet Addressによるトラフィックの識別からIPヘッダ以下のパケットサイズを換算する．libpcapはパケットをユーザの監視できる形で取り込むためのライブラリである．libpcapを使用する事でOSが提供するデータリンク層のパケット取り込み機構を隠蔽化できる．しかしtcpdumpやaguri[17]などで用いられているlibpcapは利用にかかる負荷が高いうえ，全てのパケットを確実に監視できるとは限らない．

もう一方はlibaltqを利用し，altqデーモンの保持しているスループット値を取り出す方法である．altqにおいてキューイングに必要な数値であるため，計算機に与える負荷はlibpcapを用いる場合と比較して低い．ルータ毎の流量を観測するため，下位ルータにおいてIPヘッダのTOSフィールドにルータ独自の値を記述し，上位ルータに転送する．

適切な値が分からないため，双方共に流量を測定する間隔を秒単位で変更できる様にした．

5.2.2 帯域の割り当て計算を行う機構

帯域の割り当てには二つの値を用いた．割り当ての目安値である帯域保証値および実際の帯域制御に用いられる帯域制限値である．

- 帯域保証値
  ボトルネックの帯域 (B) を下位ルータ (child router) における優先度合計 $(Prio_n)$ の割合で計算し，各下位ノードの帯域保証値 (w_n) を算出する．

$$w_n = B\frac{Prio_n}{Prio_0 + Prio_1 \ldots + Prio_N} \qquad N \text{ in number of child router} \tag{5.1}$$

- 帯域制限値

  下位ルータのいずれかが帯域保証値 (w_n) 以上のトラフィック (t_n) を発生している場合に帯域制限値 (l_n) を計算する．

  帯域制限値はボトルネックの帯域 (B) から帯域保証値 (w_n) にトラフィック量が達していないルータの利用帯域 (t_n) を差し引いた帯域幅となる．

$$l_n = B - \sum_{i=0}^{N} t_i \qquad \text{where } i \text{ is not } n \tag{5.2}$$

  複数台の下位ルータのトラフィック量が帯域保証値を越えた場合には人数の割合により空いた帯域を振り分ける．$Prio_nX$ は帯域保証値を越えるトラフィックを発生させているルータの優先度を示す．

$$l_n1 = (B - \sum_{i=0}^{N} t_i \qquad \text{where } t_i \text{ is less than } w_i)\frac{Prio_n1}{Prio_n1 + Prio_n2 \ldots + Prio_nX} \tag{5.3}$$

5.2.3 ルータ間通信を行う機構

ルータ間の通信は以下の 4 種類がある．

- 上位ルータにおいて下位ルータへ帯域制限を伝達
- 上位ルータにおいて下位ルータから優先度を受理
- 下位ルータにおいて上位ルータから帯域制限を受理
- 下位ルータにおいて上位ルータへ優先度を伝達

これらの通信に決まったプロトコルを設定した．交換されている値は帯域制限と優先度の 2 点で，両者共に数値で情報を保持するため同じフォーマットのメッセージを適用した．

メッセージフォーマットを表 5.2 に示す．

Type にはメッセージタイプが入り，上位ルータによる下位ルータの優先度を聞きに行くメッセージ (prio_req)，下位ルータによる上位ルータへの優先度報告 (prio_rep)，そして帯域制限メッセージ (restru_req) の 3 種類がある．Body には prio_msg の場合には空にし，prio_rep の場合は優先度，restru_req の場合には帯域制限数がバイト単位で入力される．

表 5.2: ルータ間通信に利用されるプロトコル

1 2 3 4	5 6 7 8	1 2 3 4	5 6 7 8
Type		Body	

表 5.3: メッセージタイプ

Type	
0x01	prio$_r eq$
0x02	prio$_r ep$
0x03	restru$_r eq$

表 5.4: メッセージタイプに対応する内容

Type	Body
0x01	- Enpty -
0x02	Priority in Int
0x03	Bandwidth limitation in byte

5.2.4 ノード毎の帯域制御を行う機構

送り元 IP アドレス毎のキューを用意する手法を用いる．手法は WFQ に似ているが WFQ はそれ単一では帯域上限を設定するできないので別のキューイング技術である HFSC(Hierachical First Service Curve)[16] を WFQ と同様に利用した．各エンドノード毎に帯域幅を設定するため，あらかじめ下位ルータと同セグメントに位置するエンドノードの IP アドレスを識別子として下位ルータに設定しておく必要がある．

バーチャルリンクの帯域を動的に変化させるために各下位ルータは帯域上限を受け取った後に各エンドノードの優先度からその使用帯域幅を計算する．計算方法は上位ノードによる下位ノードへの帯域割当と同じである．

各ノードの優先度の決定

各ノードの優先度は通常を 1 として計算する．こうすることで各ノードは公平に扱われ，非適応型トラフィックにボトルネックの帯域を占有されることがない．優先制御はあくまでノード毎に行われる．

arp table を抽出し，自インターフェースの情報を排除する事でエンドノードの数を監視した．

各ノードのトラフィック監視

各ノードのトラフィック監視を行う機構は aguri を利用した．定期的に各ノードのトラフィック量を観測した．

5.2.5　全体の流れ

図 5.1 で全体の流れを示す．

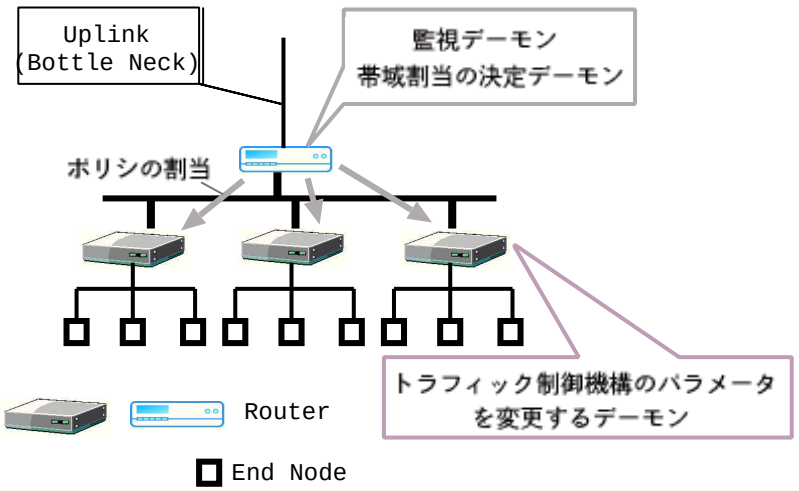

図 5.1: 本機構の流れ

5.3　WIDE 合宿へ導入

2001 年 9 月 12 日に行われた WIDE 合宿において本システムを導入し，実験を行った．WIDE 合宿のネットワークは図 5.2 の様に構成された．

1.5Mbps の対外線を 4 個のセグメントで共有している．セグメントはそれぞれ Terminal，Preinary，BoF，Wireless とよばれ，合宿地における部屋毎および無線セグメントに切り分けた．各セグメントにはエンド ノードが複数常に存在し，時間によりそのエンド ノード数が大きく変動するネットワークである．

また，ボトルネック以外の回線は無線を除き，100Mbps という環境で，ボトルネックルータと下位ルータ，下位ルータとエンド ノードの接続に 100Mbps スイッチングハブを使用した．

エンド ノード毎の優先度を 1 とし，平等な帯域制御をエンド ノードに提供した．

帯域制御のトリガーとなる使用帯域の監視は 5 秒おきに上位ルータで起動する様設定した．prio_req，restru_req を同じトリガーで下位ルータに送信するため帯域制御最長では 5 秒おきに発生する．

以降トラフィック監視の間隔を 5 秒おきを標準とし，実装した．また，WIDE 合宿での実装について論文 [18] を執筆した．

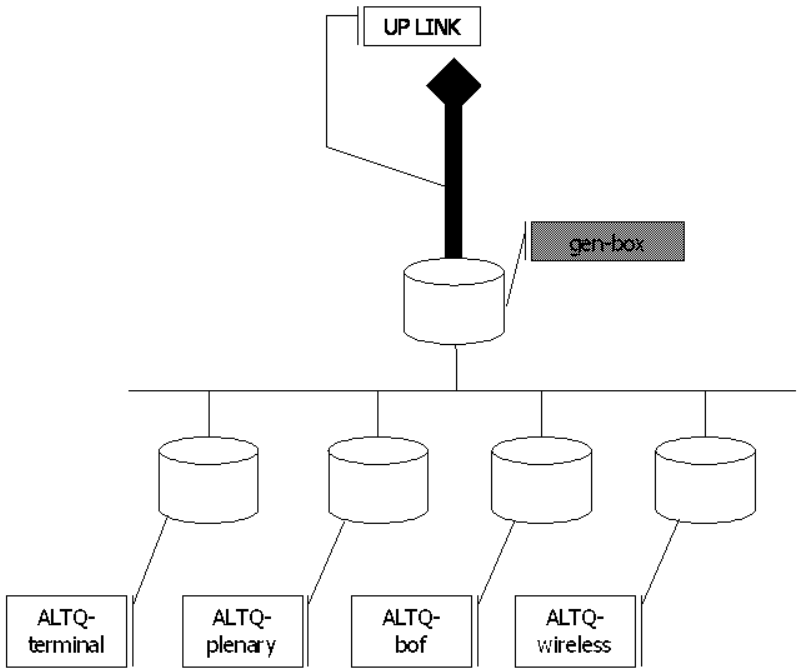

図 5.2: WIDE 合宿ネットワークの構成

第6章　評価

本章では実装を行った優先制御ポリシデリゲートによるトラフィック優先制御機構について実験および評価を示す．評価項目を以下に示す

- 各ノードの帯域を保証できているか
- NAT などの影響をうけない制御ができているか
- ボトルネックの帯域を有効に利用できているか

6.1　定性的評価

本機構は下位ルータのパケット転送処理に影響されない．よって従来実現不可能な処理だった，IPv4-v6 トランスレータ，NAT などトラフィックの情報を隠蔽化する機構の存在するネットワークにおいてトラフィック優先制御を行う事を可能とした．

ノード毎の帯域保証という新しい方法を提案した．

ルータ同士で協調しトラフィック優先制御する仕組みを実現した．

6.2　定量的評価

実験には表 6.1 の環境を利用した．それぞれ OS には FreeBSD4.4-release を利用し，ALTQ patch を適用したカーネルを用意した．ネットワークインターフェースは 100Mbps に統一した．

エンドノードには UDP を CBR(Constant Bit Rate) で転送するソフトウェアを用意し，シェーピングに altqd を利用した．

名称	CPU	メモリ
Router 1	Intel PentiumIII 1GHz	128MB
Router 2	Intel PentiumIII 600MHz	128MB
Router 3	Intel PentiumIII 650MHz	194MB
End node 1	Intel PentiumIII 600MHz	128MB
End node 2	Crusoe 550MHz	128MB
End node 3	Intel PentiumII 450MHz	128MB
End node 4	Crusoe 550MHz	128MB
End node 5	Intel PentiumIII 1GHz	256MB

表 6.1: 実験に使用したノードの構成

6.2.1 実験 1

エンド ノードを下位ルータにそれぞれ一台設置し，本機構を適用する．実験 1.1 では，ボトルネックを占有するトラフィックとそうでないトラフィックを発生させ，本機構が示す挙動を検証する．実験 1.2 では，二つのボトルネックを占有する可能性のあるトラフィックを発生さる．本機構が帯域配分をいかに行うかの検証を行う．

実験 1.1

実験環境を図 6.1 に示す．1.5Mbps の帯域であるボトルネック回線を共有するネットワークで，データをエンド ノード 1，3 からエンド ノード 5 に向けて送信する．通信は全て片方向の CBR トラフィックである．

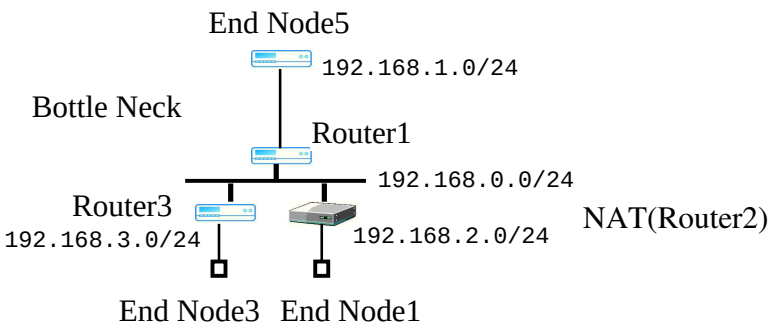

図 6.1: 実験 1 の構成

まず優先度の同じエンド ノード 1，3 からそれぞれ 300Kbps のフロー A，3Mbps のフロー B のトラフィックを発生させる．エンド ノード 5 にてトラフィック量を監視する．またボトルネックの回線利用率を観測した．

下位ルータにおける優先度はルータ 2，3 で同じである．よってフロー A は保護され，フロー B は破棄されることが予想される．

図 6.2 に結果を示す．X 軸は秒数，Y 軸はフローのビットレート (Kbps) を指している．およそ実験開始から 100 秒で本機構を導入した．本機構導入以前 (1) ではフロー A はおよそ 50Kbps の帯域，フロー B は 1.4Mbps の帯域を利用している．フロー A は本システム導入前には通信品質が低下している (2)．

本機構導入後にフロー A はおよそ 300Kbps 均一のスループットとなった．(3) 最低保証帯域幅はエンド ノードにそれぞれ 750Kbps となるため，フロー A は保護される．しかし，フロー B はそのスループットがおよそ 650Kbps となった．結果最低保証帯域幅以下の帯域利用となり，ボトルネックの利用率は図 6.3 の様に変移した．本機構導入前 (1) と導入後 (2) で回線利用率は低下した．

実験 1.2

実験 1.1 と同じネットワーク構成 (図 6.1) においてエンド ノード 1，3 からそれぞれ 3Mbps のトラフィックであるフロー A，B を発生させる．

エンド ノード 5 においてトラフィック量を監視し，ボトルネックの回線利用率を観測した．

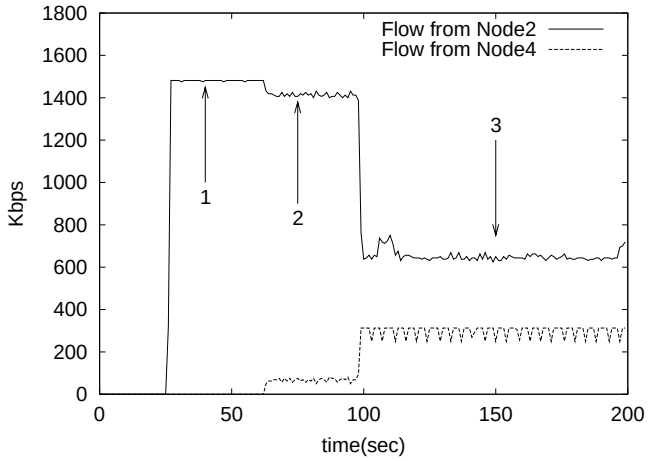

図 6.2: 実験 1.1:トラフィック量の移行

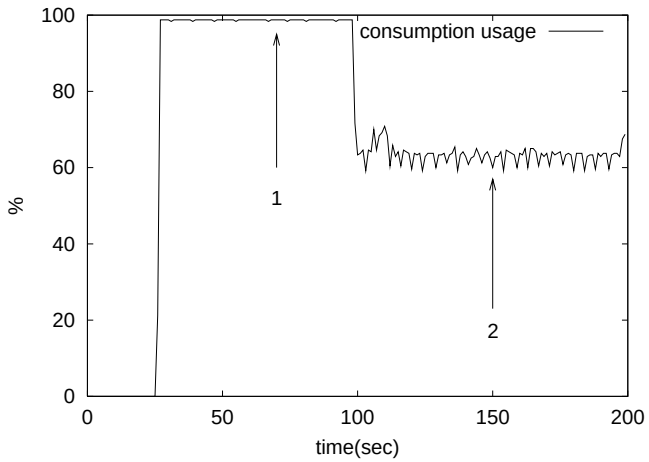

図 6.3: 実験 1.1:回線使用率の変化

図 6.4 にフロー A，B のトラフィック量の移行を示す．

(2) ではどちらも 3Mbps のフローだが，フロー A の占める割合が大きいことがわかる．FIFO キューのみの利用ではこの様な結果となる．(3) では本機構の効果で，フロー A，B は均等に帯域を利用している．

ボトルネックの回線利用率を図 6.5 に示す．

(2) で利用率が低下しているのがわかる．

6.2.2 実験 2

エンドノードを下位ルータにそれぞれ 2 台，または 1 台設置し，本機構を適用する．エンドノード数の変化による優先度の違いから発生する差を観測する．

実験 2.1 では同じ優先度のセグメントに帯域割り当ての正当性，を検証する．実験 2.2 では優先

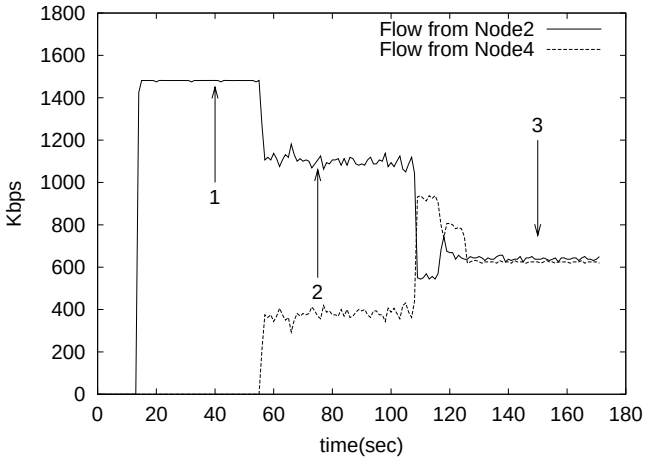

図 6.4: 実験 1.2:トラフィック量の移行

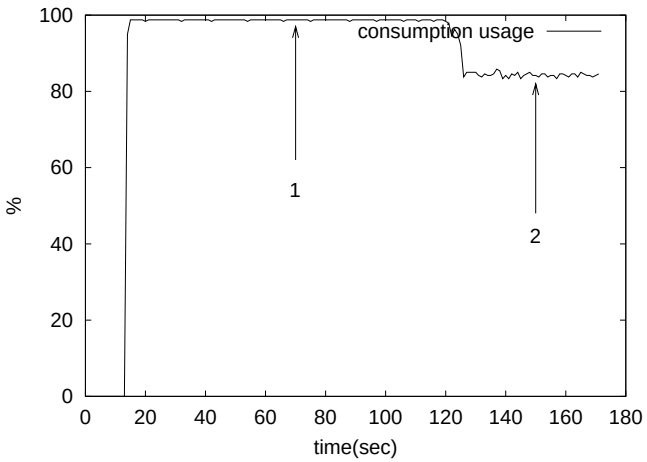

図 6.5: 実験 1.2:回線使用率の変化

度の違うセグメントに対する帯域割り当てについて検証を行う．

実験 2.1

エンドノードをそれぞれ 2 台，各セグメントに設置し，エンドノード 3 は実際にトラフィックを発生させない．その他 3 台のエンドノードはそれぞれ 1 Mbps のトラフィックを発生させる．トラフィックはそれぞれフロー A，B，C と呼ぶ．

エンドノード 5 においてトラフィック量の監視，およびボトルネックの回線利用率を観測した．

下位ルータ毎の優先度が同じ場合，エンドノード 4 に与えられる利用可能帯域は 1，2 のそれに比べ 2 倍に成るはずである．

図 6.7 にフロー A，B，C のトラフィック量の移行を示す．

本機構導入前の (1)(2) では帯域の利用に差異がみられる．本機構導入後の (3) ではルータ 3 下

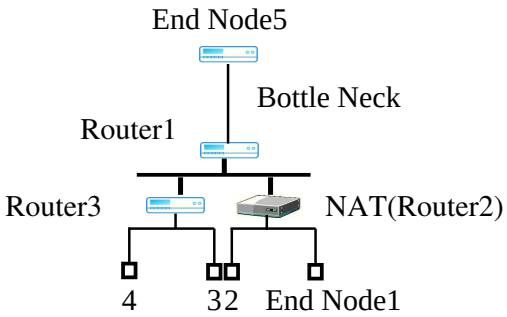

図 6.6: 実験 2 の構成

のノードから発生したフロー A，B の帯域利用幅はフロー C の半分ほどになっている．
ボトルネックの回線利用率を図 6.8 に示す．
回線利用率は以前の実験に比べ低下幅を小さく保てた．

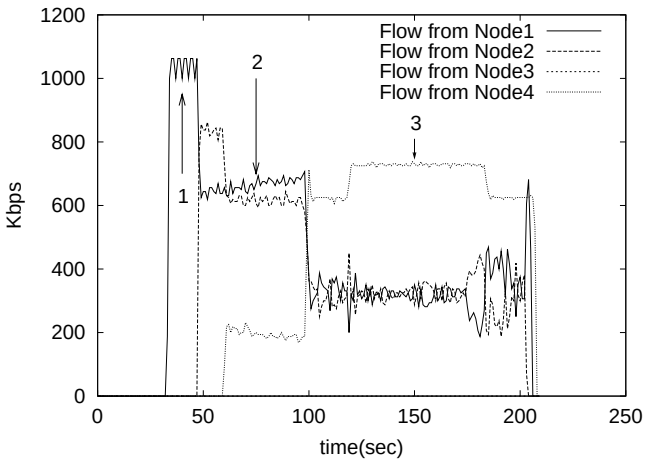

図 6.7: 実験 2.1:トラフィック量の移行

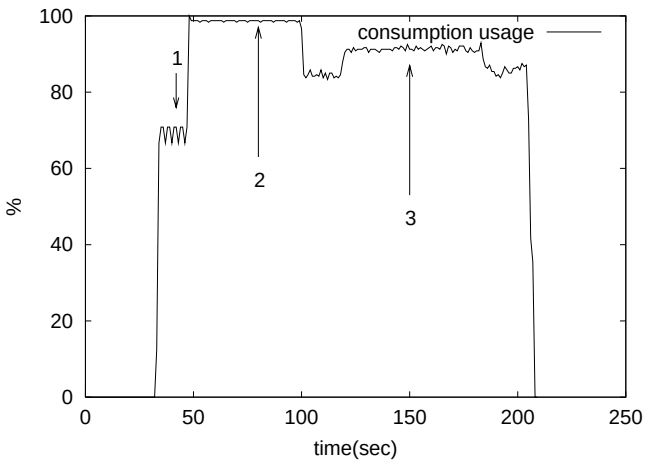

図 6.8: 実験 2.1:回線使用率の変化

実験 2.2

実験 2.1 と同じ環境で，エンド ノード 3 を取り除いた．よって下位ノード毎の優先度は 1 対 2 となる．トラフィックはそれぞれフロー A，B，C と呼び，それぞれ 1 Mbps の帯域を使用する．

エンド ノード 5 において，トラフィック流量，およびボトルネックの回線使用率を観測した．

図 6.9 にフロー A，B，C のトラフィック量の移行を示す．

本機構導入前と後にトラフィック量にぶれが出た．これはトラフィック制御機構が有効に機能していなかったと思われる．

ボトルネックの回線利用率を図 6.10 に示す．

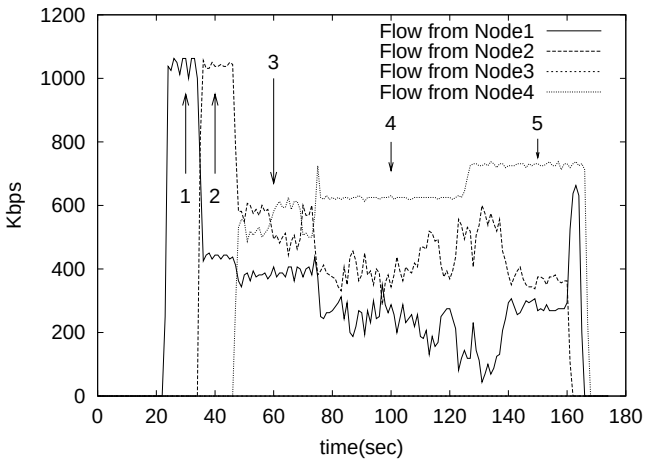

図 6.9: 実験 2.2:トラフィック量の移行

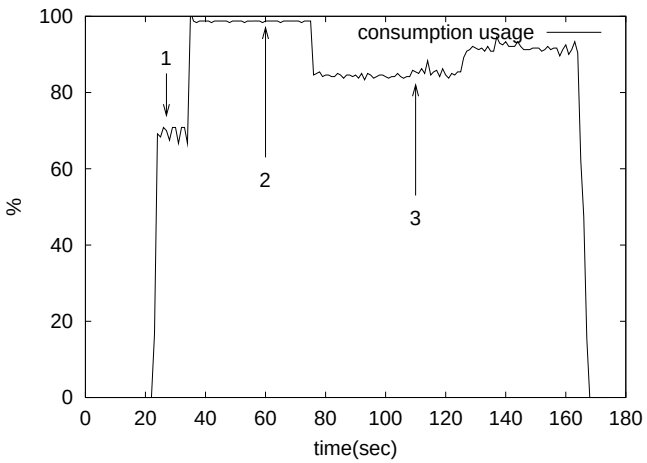

図 6.10: 実験 2.2:回線使用率の変化

6.2.3 結果と考察

実験 1.1 においてボトルネックを占有するトラフィックから他トラフィックを保護できることがわかった．

実験 1.2 において優先度の同じセグメントにあるエンドノードは同じ帯域をわりあてられ，平等に帯域資源を利用できることがわかった．

実験 2.1 では優先度の同じセグメント内でノード毎に帯域資源を公平に割り振るれる事がわかった．

実験 2.2 において優先度の違うセグメントに優先度に対応した帯域資源を割り振れる事がわかった．

本機構を導入することで回線利用率は，実験 1，2 の両方において低下する事がわかった．

以上の結果より，本機構を利用することで，各エンドノードに対し，最低帯域保証を提供できる事がわかった．

第7章　結論

本章では本研究の成果を明らかにし，今後解決すべき課題，および発展の方向性について述べる．

7.1　達成した事

本節で本研究で達成した事を述べる．

本研究ではトラフィックの情報を隠蔽化してしまう機構が存在するネットワークでのトラフィック制御機構の構築に成功した．これにより NAT や IPv4-v6 トランスレータの存在するネットワークにおいて，それらを意識すること無くトラフィック優先制御を行うことが可能となった．

本研究では，解決方法としてエンド ノードの優先度別に，ボトルネック上の帯域資源を割り振る方法を採用した．下位ノードから優先度の合計を上位ノードに伝搬し，上位ノードがその優先度から各下位ノードの帯域資源の割り振りを決定する．各下位ノードはエンド ノードの優先度を保持する．下位ノードは各エンド ノードの優先度，エンド ノードのトラフィック量，および割り当てられて帯域幅から，エンド ノード毎の帯域制御を行う．

本機構は複数ルータによる分散トラフィック優先制御を行う仕組みとなった．また，本機構は優先度が動的にかわる環境にも対応できる．従って本研究では動的に帯域制御を行う，中間ノードによる分散トラフィック制御を行う機構の作成に成功した．

7.2　今後の課題

以下に現時点での本機構における問題点を列挙し，今後の発展について述べる．

7.2.1　現状の問題点

不公平性

優先制御ポリシーをデリゲートし，エンド ノードに直接面するルータにおいてトラフィック優先制御を行う本機構はエンド ノード毎の公平性を保つ様設計したが場合によって不公平な状況を作る．

次に本機構を用いて際いかに不公平な環境が構成されるかを示す．

図 7.1 はセグメント毎に極端にノード数の異なるネットワークを示している．ボトルネックである $L1$ を二階層から成るネットワーク全体で共有している．下位ルータ 1，2，にはそれぞれ 2 台と 10 台のエンド ノードが接続されている．

本機構では各下位ルータの優先度は含有するエンド ノードに対応する優先度の合計で決定される．優先度により下位ルータの帯域制限値の計算が行われるため，優先度の高い下位ルータに割り振られる帯域は必然的に大きくなる．すると各ノードの優先度が標準の 1 である場合，ノード数の多いセグメントにはより広帯域のバーチャルラインが設置される事になる．各セグメントか

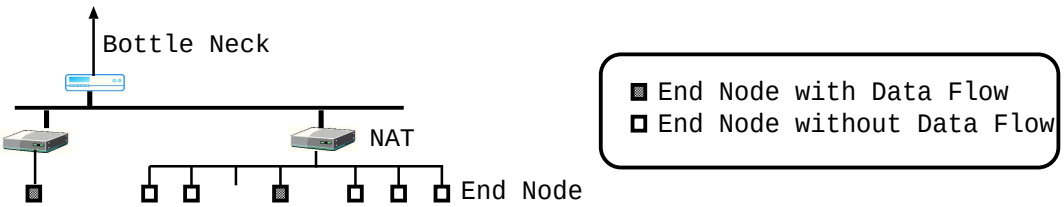

図 7.1: 不公平が生じる場合

ら 1 台づつがトラフィックを発生させている場合，各エンド ノードは帯域制限値までの帯域使用を許可される．よって広帯域のバーチャルラインを提供されたセグメントに位置するエンド ノードの使用できる帯域は低帯域のバーチャルラインをあたえられたセグメント下のエンド ノードと比較して約 10 倍利用できる事になる．結果優先度合計が低いセグメントに対し不公平な処理を行う．

回線利用率

本研究の目的の一つとしてエンド ノード毎の帯域制御がある．本機構を利用しない場合の帯域制御は最低帯域保証値を設定する方法と非適応型トラフィックに罰則を与え，他を保護する方法しかない．適応型トラフィックを優先的に転送する方法で，回線の利用率は向上する．

よって本機構ではエンド ノード毎に最低帯域保証をおこなった．またトラフィック量の多いエンド ノードには帯域上限を設定し，他ノードの保証された帯域を保護した．結果最低帯域の保証は行うことができたが，帯域上限がうまく機能せず，必要以上に帯域を制限してしまった．(図 6.2) それに伴って，ボトルネックの利用率 (図 6.3) が低下した．

帯域上限を施行する機能に障害があったものと思われる．

7.2.2 不足している機能・調査

エンド ノード別の優先度付け

現状の実装では各ノードの優先度を 1 としたトラフィック優先制御を行った．ノード毎に異なる優先度を設定する事ができない．

ノード毎に異なる優先度を設定するには下位ノードにおいて優先度の表を用意し，それをエンド ノードと比較し，優先度を抽出する仕組みが必要である．

適応型トラフィックを用いた実験

本研究では非適応型トラフィックである UDP を用いた実験を行った．転送量をエンド ノード側で制御する適応型トラフィックはパケットロスに敏感に反応し，使用帯域が大きく変化することが考えられる．

ノード毎の帯域制御が与える影響を評価する必要がある．

第8章　謝辞

本研究を進めるにあたり，御指導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，同学部助教授楠本博之博士，中村修博士，南政樹専任講師に感謝します．また，絶えず御指導と助言を頂きました慶應義塾大学政策・メディア研究科博士課程の杉浦一徳氏，小川晃通氏，今泉英明氏に感謝します．

そして，本研究を進めて行く上で様々な励ましと御助言をして下さった慶應義塾大学政策・メディア研究科修士課程の穎原桂二郎氏，川喜田佑介氏，有賀征爾氏，かいざみん氏，吉田信生氏，ash 氏，同大学学部の久松剛氏ならびに同大学の徳田・村井・楠本・中村・南合同研究室の諸氏に感謝します．

参考文献

[1] P.Srisuresh and M.Holdrege. *IP Network Address Translator(NAT) Terminology and Considerations*, August 1999. Request for Comments: 2663.

[2] Jun-ichiro Hagino K. Yamamoto. *An IPv6-to-IPv4 transport relay translator*, May 2000. Work in Progress, draft-ietf-ngtrans-tcpudp-relay-01.txt.

[3] J.D. Case, M. Fedor, M.L. Schoffstall, and C. Davin. *Simple Network Management Protocol (SNMP)*, May 1990. RFC 1157.

[4] A.Demers S.Keshav and S.Shenker. *Analysis and Sumulation of a Fair Queueing Algorithm*, 1990. Internetworking: Research and Experience, Vol.1.

[5] J. Postel. *Transmission Control Protocol*, September 1981. RFC 793.

[6] M. Allman, V. Paxson, and W. Stevens. *TCP Congestion Control*, April 1999. RFC 2581.

[7] I.Stoica R.Morris D.Karger M.Frans Kaashoek and H.Balakrishnan. *Scalable Peer-to-peer Lookup Service for Internet Applications*, August 2001. SIGCOM.

[8] P.Ferguson and G.Huston. インターネット *QoS*, May 2000. オーム社.

[9] J. Postel. *User Datagram Protocol*, August 1980. RFC 768.

[10] S. Floyd, V. Jacobson, and L. Zhang. *Link-sharing and Resource Management Models for Packet Networks*, August 1995. IEEE/ACM Transaction on Networking,Vol.3 no.4,pp.365-386.

[11] P. Srisuresh and M. Holdrege. *IP Network Address Translator (NAT) Terminology and Considerations*, August 1999. RFC 2663.

[12] J.D. Case, M. Fedor, M.L. Schoffstall, and C. Davin. *Simple Network Management Protocol (SNMP)*, April 1989. RFC 1098.

[13] Naoto Morishima, Akimichi Ogawa, Keijiro Ehara, and Youki Kadobayashi. Field-trial of dynamic sla in diffserv-capable network. In J.Crowcroft, J.Roberts, and M.I.Smirnov, editors, *Quality of Future Internet Services*, number 1922 in Lecture Notes in Computer Science, pages 117–128, Berlin, Germary, September 25-26 2000.

[14] 森島 直人, 小川 晃通, 纐原 桂二郎, 染川 隆司, and 山口 英. 動的資源予約システムの運用実験. インターネットコンファレンス *2000* 論文集, November 2000.

[15] *FreeBSD Project*. http://www.freebsd.org.

[16] Ion Stoica Hui Zhang and T. S. Eugene Ng. *A Hierarchical Fair Service Curve Algorithm for Link-Sharing, Real-Time and Priority Service*, April 1997. SIGCOM.

[17] K.Cho R.Kaizaki and A.Kato. *Aguri: An Aggregation-Based Traffic Profiler*, September 2001. QoFIS2001(Qualitiy of Future Internet Services).

[18] G.Kobatake A.Ogawa and K.Sugiura. *Delegation of Priority Control Policy*, September 2001. 情報処理学会台 63 回全国大会論文集.

付 録A　appendix

A.1　SNMP

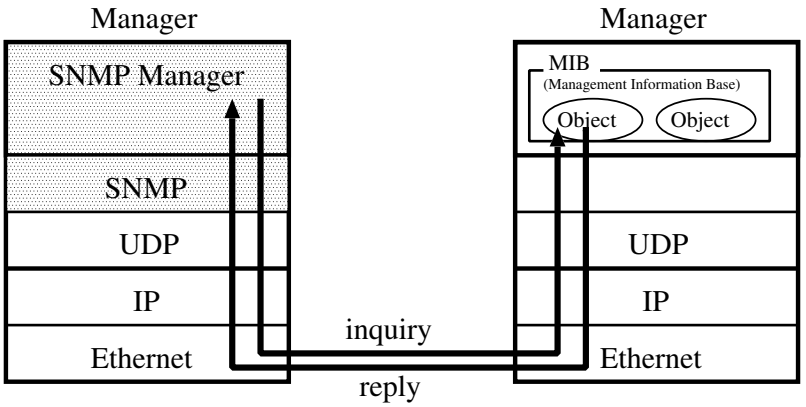

図 A.1: Simple Network Management Protocol

A.2　TCP/IPプロトコルスタック

Application Layer	E-mail,Web browser
Transport Layer	TCP,UDP
Network Layer	IP,ICMP,IGMP
Link Layer	device driver,network interface

図 A.2: TCP/IP プロトコルスタック

A.3　IPヘッダ (version 4)

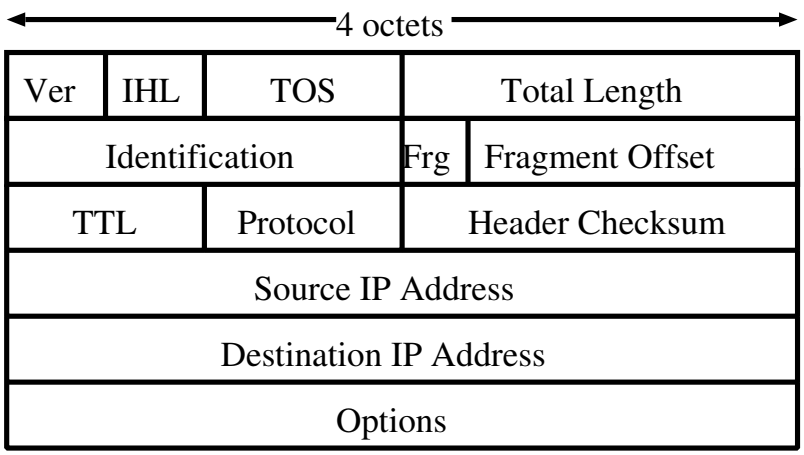
4 octets
Ver
IHL
TOS
Total Length
Identification
Frg
Fragment Offset
TTL
Protocol
Header Checksum
Source IP Address
Destination IP Address
Options

図 A.3: IPv4 ヘッダ