

平成 12 年度
学士学位論文

ネットワークトランスファ性能評価と
その高機能化に関する研究

A study on the high speed network transfer based on
Packet Assembly method and its performance
evaluation

1010373 浦西 慶規

指導教員 島村 和典

2001 年 2 月 5 日

高知工科大学 情報システム工学科

要 旨

ネットワークトランスファ性能評価と その高機能化に関する研究

浦西 慶規

大容量高速ネットワークの発展に伴い、ネットワークアプリケーションによるサービスは量だけでなくその種類も急激に増加している。さらに多くのサービスが考え出され、ネットワークに対する期待は非常に大きくなっている。

しかし IP はベストエフォート型の転送制御なので、大容量高速ネットワークとして回線容量を増加させるだけでは、アプリケーションごとに必要な Quality of Service を十分に満たすことができなくなってきた。QoS パラメータには Throughput や LossRate, Hop 数, Delay, Delay Jitter などがある。

本論文ではまず、行き先が同じであるパケットをルータでまとめて一つのパケットにし転送する方式を提案する。これによりネットワークを流れるパケット数を減らせることから、トラフィック転送時にルータのヘッダ処理の負荷軽減が可能となりより効率良くデータを転送することが可能になる。その具体的な方法としてルータによるパケットの組み立て (Packet Assembly) を提案した。

パケットをアセンブリすることにより発生する Delay は、リアルタイム系のトラフィックにかえて悪影響を与える懸念があるが、それに対する方策として、リアルタイム系トラフィックは優先的に転送する Priority Queuing と Packet Assembly を組合せた Priority Queuing with Assembly について述べる。本論文ではシミュレーションを行い、Packet Assembly がどのように QoS 向上につながるかを明らかにしている。

キーワード パケットアセンブリ, プライオリティキューイング, 実時間トラヒック, 負荷軽減, ルータ, 最大転送単位

Abstract

A study on the high speed network transfer based on Packet Assembly method and its performance evaluation

Yoshiki Uranishi

The networks are getting expanded and higher throughput transferable. And the Network Application Services have increased rapidly not only in quantity but also their types. Many network applications are being added, but current network technologies couldn't respond their requirements at all. But it is very difficult to build high capacity networks under satisfying the parameter of QoS using IP network technologies based on best effort policy. There are same QoS parameters of Throughput and LossRate, the number of Hop, Delay, DelayJitter, and so on. In this study, the packets which destinations are same are collected with a router connected into one packet and transferred. It becomes possible that data is transferred more efficiently because the load for the header processing in routers are reduced by decreasing the number of packets. Assembling of the packets on the router is proposed concretely. But the Delay produced by assembling packets can influence on delay sensitive traffic such as by real time applications. Priority Queueing with Packet Assembly is proposed for the concern. It is simulated and mentioned in this thesis how QoS is improved by PacketAssembly.

key words PacketAssembly PriorityQueueing RealtimeTraffic LoadReduction Router
MTU

目次

| | | |
|--------------|---|-----------|
| 第 1 章 | はじめに | 1 |
| 1.1 | 研究の背景 | 1 |
| 1.2 | 既存の技術, 研究 | 2 |
| 1.2.1 | MPLS | 2 |
| 1.2.2 | Fragmentation を使ったパケット合成 | 2 |
| 1.3 | 課題 | 2 |
| 1.3.1 | パケット毎のヘッダ処理によるルータの負荷 | 3 |
| 1.3.2 | MTU より遥かに小さいパケット | 4 |
| 1.4 | 研究の目的 | 5 |
| 第 2 章 | パケットアセンブリ | 6 |
| 2.1 | パケットアセンブリのシステム | 6 |
| 2.2 | パケットアセンブリの IP ヘッダ操作 | 7 |
| 2.3 | アセンブリパケットフォーマットの考察 | 9 |
| 2.4 | Priority Queuing with Assembly 方式 | 10 |
| 2.5 | パケットアセンブリの手順 | 11 |
| 2.5.1 | アセンブリキューイング | 11 |
| 2.5.2 | Priority Queuing with Assembly | 11 |
| 2.5.3 | アセンブリパケットの分離 | 12 |
| 第 3 章 | シミュレーション実験と結果 | 14 |
| 3.1 | ns によるシミュレーションの条件 | 14 |
| 3.2 | パケットサイズと帯域による遅延時間の違いのシミュレーション | 15 |
| 3.2.1 | シミュレーション実験 | 15 |

| | | |
|--------------|---|-----------|
| 3.2.2 | シミュレーション結果 | 16 |
| 3.2.3 | 考察 | 17 |
| 3.3 | パケットアセンブリシミュレーション | 17 |
| 3.3.1 | シミュレーション実験 | 17 |
| 3.3.2 | シミュレーション結果 | 18 |
| 3.3.3 | 考察 | 19 |
| 3.4 | Priority Queuing with Assembly シミュレーション | 19 |
| 3.4.1 | シミュレーション実験 | 19 |
| 3.4.2 | シミュレーション結果 | 20 |
| 3.4.3 | 考察 | 20 |
| 3.5 | むすび | 21 |
| 第 4 章 | 考察 | 23 |
| 4.1 | パケットアセンブリの必要性 | 23 |
| 4.2 | アセンブリするパケット | 24 |
| 4.3 | パケットアセンブリ時にエッジルータにかかる負荷 | 24 |
| 4.4 | アセンブリパケットの Loss | 24 |
| 4.5 | 基幹ネットワークルータの負荷が低い場合 | 25 |
| 第 5 章 | 今後の課題 | 26 |
| 5.1 | アセンブリのパケットフォーマットと負荷 | 26 |
| 5.2 | マルチフローアセンブリ | 26 |
| 5.3 | パケット転送スケジューラによるパケットアセンブリ | 27 |
| 5.4 | ルータ内遅延を含めたシミュレーション | 27 |
| 謝辞 | | 29 |
| 参考文献 | | 30 |

目次

| | | |
|-----|----------------------------------|----|
| 1.1 | Hop 数 [2] | 3 |
| 1.2 | パケットサイズの分布 [1] | 4 |
| 2.1 | ネットワーク構成 | 6 |
| 2.2 | IP ヘッダの構造 | 7 |
| 2.3 | 連結の操作 | 8 |
| 2.4 | パケットフォーマットの方式 | 9 |
| 2.5 | PriorityQueueingwithAssemblby | 10 |
| 2.6 | Packet Assemblby | 12 |
| 2.7 | Priority Queueing with Assemblby | 13 |
| 2.8 | アセンブリパケットの分離 | 13 |
| 3.1 | シミュレーション | 15 |
| 3.2 | 結果 UDP パケット遅延 (中央値) | 16 |
| 3.3 | 結果 UDP パケット遅延 (最大値) | 16 |
| 3.4 | シミュレーションモデル | 18 |
| 3.5 | パケットアセンブリによる実時間通信パケットへの影響 | 18 |
| 3.6 | シミュレーションモデル | 20 |
| 3.7 | priority Queueing | 21 |
| 3.8 | PriorityQueueing 導入比較 | 22 |
| 5.1 | scheduler によるアセンブリ | 27 |

表目次

| | |
|-------------------------|----|
| 1.1 IP の MTU | 5 |
| 2.1 フォーマットの考察 | 10 |

第 1 章

はじめに

1.1 研究の背景

大容量高速ネットワークの発展に伴い、ネットワークアプリケーションサービスは量だけでなくその種類も急激に増加している。さらに多くのサービスが考え出され、ネットワークに対する期待は非常に大きくなっている。

ただ大容量高速のネットワークとして回線容量を増やすだけでは、アプリケーションごとに必要な QoS 条件を十分に満たすことができなくなってきた。QoS パラメータには Throughput や LossRate, Hop 数, Delay, Delay Jitter などがある。インターネットユーザの急増によるネットワークへの負荷が大きくなり、これらのパラメータを十分に満たすことはさらに難しくなってきた。

IP トラフィックは今までの音声通信のトラフィックを越え、その増加勢いは爆発的なものである。次世代ネットワークは IP プロトコルによる統合化が進むと考えられている。現在の IP はベストエフォートポリシーに基づいているのでアプリケーションに応じたトラフィックの制御（優先制御など）を行っていない。限られた帯域の中ですべてのトラフィックを平等に扱っていたのではトラフィックの混雑時に通信の品質は下がってしまう。そこでアプリケーションの特性に応じたトラフィックの制御をするパケット転送方式を提案する。

1.2 既存の技術, 研究

1.2.1 MPLS

パケット転送の負荷軽減の技術として MPLS (Multi-Protocol Label Switch) がある。これはパケット転送情報として、IP アドレスより下位レイヤの情報 (ラベル) を利用し、ルータの負荷を軽減するものである。

欠点としては、一つずつのパケットの処理の負荷軽減はできても PerPacket で処理負荷がかかっていることに変わりはないことや MPLS に対応するエッジルータ、コアルータが必要なことなどがある。

1.2.2 Fragmentation を使ったパケット合成

パケットを合成するという意味で Packet Assembly と似た考え方として、Transparent or intra-network fragmentation がある。これは MTU が小さいリンクでフラグメントを行い、MTU が大きいリンクになればすぐに合成するという方法である。これによりそれぞれのルータ間で適したパケットサイズに変更し効率の良い転送ができること考えられる。

しかし繰り返し Fragment, Defragment を行うことによりそのつど Delay が生じたり、その処理によりルータに多くのプロセッサ負荷がかかる。また転送の最初の HOP の MTU よりもパケットサイズが大きくなることができないという欠点がある。

パケットの Defragment については、経路 MTU 探索により fragment されたパケットがネットワークを流れることはほとんどないため行えないことがわかった。

1.3 課題

ネットワーク QoS の保証をするための方法としては、一つは限りなく無限に近い帯域を用意して混雑なしのネットワークを構築する方法ともう一つ限られた帯域の中でそれぞれのアプリケーションの必要に応じたパケット転送法を用いることにより効率よく転送する方法が

考えられる。

1.3.1 パケット毎のヘッダ処理によるルータの負荷

転送したいデータの量が多くなるにつれて、決められたネットワークの帯域の中でいかに IP パケットをより高速に大量に処理するかが今後のネットワークの重要な課題となる。IP トラフィックをルーティングする際に IP パケットごとにヘッダを見て処理していたのでは高速に転送できない。

ルーティングの負荷は per-bit よりも per-packet で決まる。per-packet で行われる処理としては、ルーティングの決定、パケットヘッダの処理、チェックサム計算、入出力キュー間の移動がある。よってルータのスループットは最大サイズのパケットが送られ時に最大となる。

過大なトラフィックによりルータへの負荷が増大すると各ルータ毎でパケット転送が遅れて大きな遅延を与えたりまたは処理しきれずそのまま破棄されるということが起こる。またホップ数の増加により各ノード毎のヘッダ処理による負荷がかかると、各ノード毎の遅延が積み重なって大きな遅延となってしまう。ホップ数は図 1.1 のようになっており、平均 16 ホップで現在も増加の傾向にある。

Hop Distance Distribution

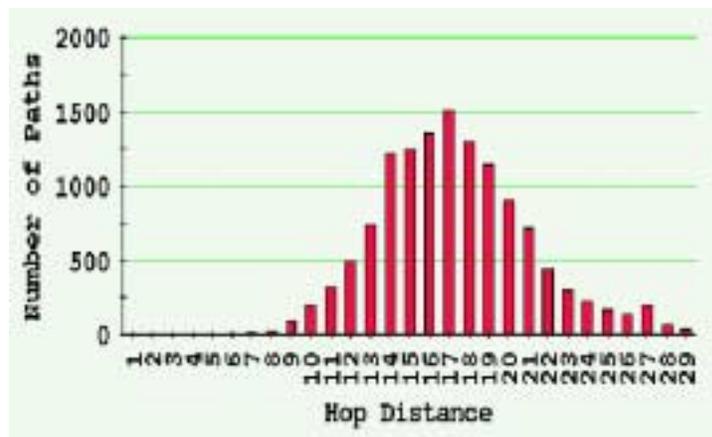


図 1.1 Hop 数 [2]

1.3.2 MTU より遥かに小さいパケット

図 1.2 は MCI の vBNS トラフィックの調査によるものである。1500Bytes 以下のパケット

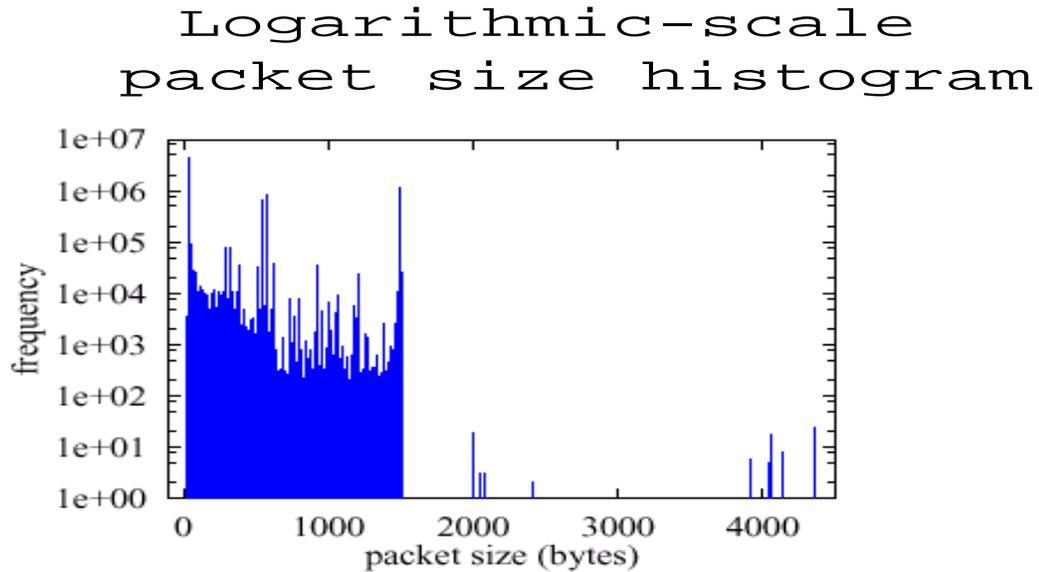


図 1.2 パケットサイズの分布 [1]

がほぼ 100%のトラフィックを占めている。また 44Byte 以下のパケットがほぼ半数である。

MTU (最大転送単位) はネットワークの転送方により扱えるパケットの最大サイズを各ネットワークによって定められている。各ネットワークの MTU は表 1.1 に示す。これらの MTU より遥かに小さいサイズのパケットがルーティングされて各ノードを次々に経由して流れることは、効率がよくない。

このようなパケットが転送されている原因として、MTU の小さいネットワークから MTU の大きいネットワークへパケットが転送されるときそのままの小さいパケットで転送されていること、また機器の調整によって MTU より遥かに小さいパケットに分割されてネットワークへ転送されていることがある。

表 1.1 IP の MTU

| ネットワーク | MTU(Bytes) |
|-------------|------------|
| IP の最大 MTU | 65535 |
| IP over ATM | 9180 |
| トークンバス | 8166 |
| トークンリング | 4464 |
| FDDI | 4352 |
| イーサネット | 1500 |
| PPP | 1500 |
| IP の最小MTU | 68 |

1.4 研究の目的

本研究の目的は、近年のトラフィック増大に対応するため、限られた帯域の中で、それぞれのアプリケーションに適応したトラフィック制御を行うことにより、効率のよい転送を行いネットワーク QoS を向上させることである。

基幹ネットワークにおいては、パケットを個々に処理するのではなく行き先ごとにまとめて転送する。基幹ネットワークルータの IP パケット処理負荷軽減をのために、MTU の小さいネットワークから MTU の大きいネットワークへパケットを転送する際、MTU の大きくなるルータで行き先が同じである複数のパケットを一つにまとめて転送する方式（パケットアセンブリ）を提案する.[3]

パケットがまとめられた分だけヘッダ処理の回数を削減でき、基幹ネットワークではパケット毎の処理の負荷がなくなり効率良くデータ転送が行える。

第 2 章

パケットアセンブリ

2.1 パケットアセンブリのシステム

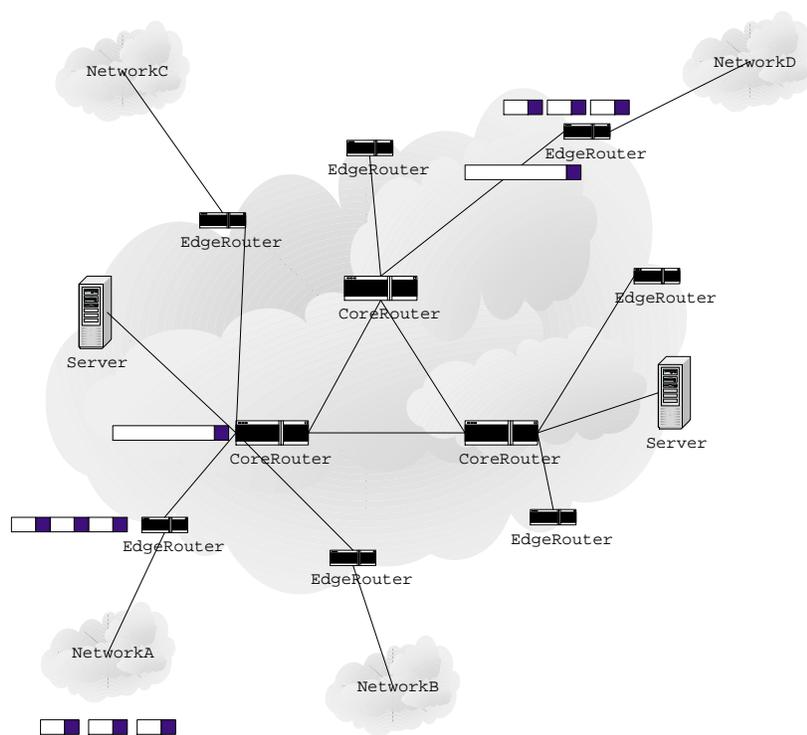


図 2.1 ネットワーク構成

パケットアセンブリは基幹ネットワークの端で一度アセンブリすれば、それ以降の途中のルータでは通常の IP として転送でき、再びエッジルータへ到着したときに分離すればよい。よって基幹ネットワークのエッジルータにのみ実装すればパケットアセンブリシステムが構築できる。

アクセスネットワークから基幹ネットワークへパケットが転送される際にエッジルータで行

き先の同じパケットを一つにまとめるパケットアセンブリを行う。アセンブリされたパケットは基幹ネットワークルータを効率良く転送される。その後、基幹ネットワークからでるときはエッジルータで分離されアセンブリ前の元の状態にパケットにもどされてから次のネットワークへ転送される。

例として図 2.1 ようにネットワーク A から転送された 3 つのパケットはエッジルータで一つのパケットにまとめてアセンブリされ、基幹ネットワークに転送される。このとき基幹ネットワーク内のそれぞれのルータではヘッダ処理が 3 分の 1 でデータを転送することが可能である。基幹ネットワークを出てネットワーク D へパケットを転送する際には、ネットワーク D はそのままのアセンブリされたパケットを受け取ることができないので、エッジルータで元どりのパケットに分離してからネットワーク D へ転送される。

2.2 パケットアセンブリの IP ヘッダ操作

図 2.2 に IP ヘッダの構造を示す。ルータの IP 処理部では、到着した IP パケットの大き

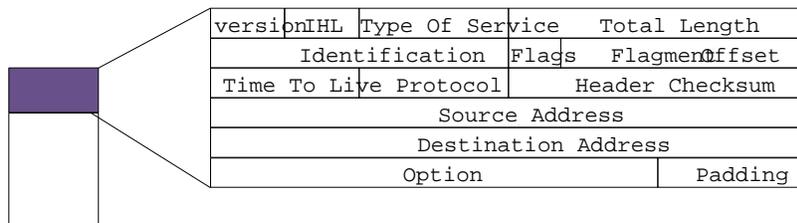


図 2.2 IP ヘッダの構造

さは、IP ヘッダの Total Length で見分けられている。よって DestinationAddress が同一である IP パケットを連結し、先頭の IP パケットのヘッダ部の Total Length を連結した長さ分だけ増加させる。またデータが書き変えられるので、ヘッダにエラーがないかを確認するための Header Checksum を再計算が必要である。これにより次のホップのルータでは、後続のパケットは先頭 IP パケットのデータ部になり、一つのパケットとしてルーティングされる。図 2.3 を例にすると、まず Destination Address が同一である 2 つの IP パケットを連結させる。この 2 つのパケットサイズはそれぞれ 40Bytes であり、連結させたサイズは

2.2 パケットアセンブリの IP ヘッダ操作

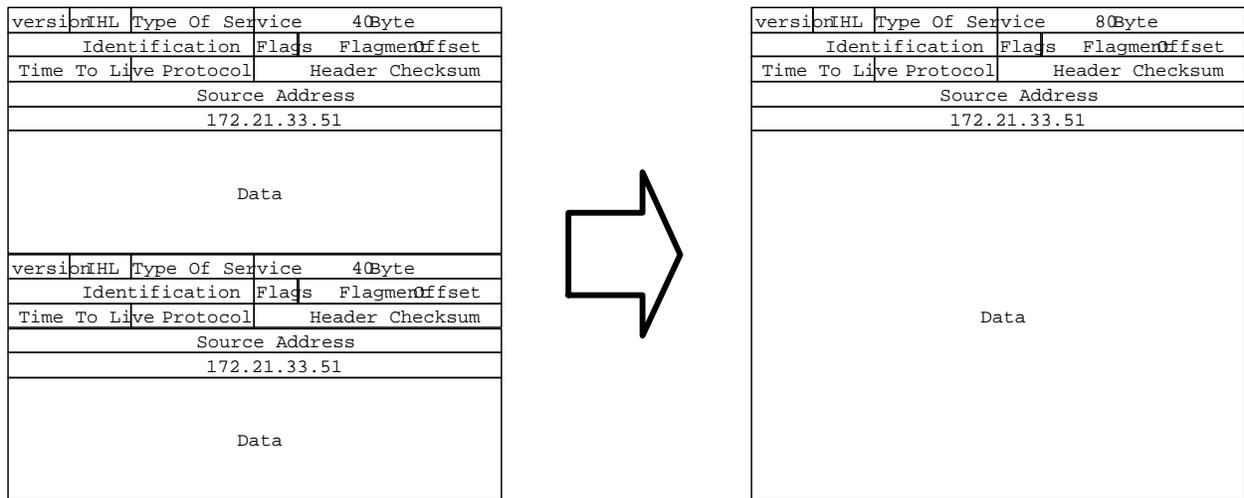


図 2.3 連結の操作

80Bytes である。よって先頭の IP ヘッダの Total Length にサイズ 80Byte という値を代入することで1つのパケットとなる。ここで問題となるのは、先頭 IP ヘッダの Total Length を書き換えたことにより元のパケットのサイズを失ったことになる。この場合パケットをどこで分離すれば良いかが分からなくなってしまう。そこで、先頭 IP ヘッダの後ろにパケットアセンブリヘッダを追加するか、IP ヘッダにはオプション部分を存在させることができる。この領域に元のヘッダの Total Length の値を代入すれば先頭パケットのサイズを失うことはない。Time To Live は、パケットがネットワークの経路のループのために永久に転送され続けることを防ぐ。アセンブリされたパケットの先頭以外のヘッダは、Time To Live を減らされることのない。このまま分離してしまうとホップ数が不正な値になってしまう。よって、分離時に基幹ネットワークで Hop した分だけ Time To Live を減算する必要がある。

2.3 アセンブリパケットフォーマットの考察

パケットアセンブリはヘッダを変更または取り除くことを考えているので、帯域利用効率が高くなると考えられる。アセンブリパケットのフォーマットとして図 2.4 4つの方式を例にあげる。

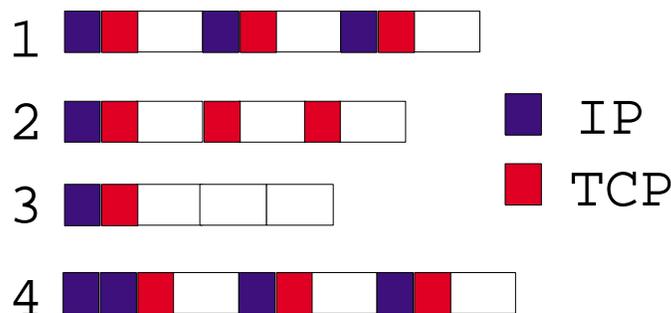


図 2.4 パケットフォーマットの方式

- 方式 1 ; ヘッダの削除は行わずそのままパケットをつなぎ合わせる方式である。ヘッダの操作は先頭のみで、Length の変更とアセンブリを行ったことを示す印をつけることである。このためプロセッサ負荷の小さいアセンブリが可能であると考えられるが、ヘッダの削除を行わないので帯域利用効率の向上はできていない。
- 方式 2 ; IP ヘッダを削除してパケットをつなぎ合わせる方式である。IP ヘッダを削除する毎に Length の記憶を行っていく。アセンブリ時の IP ヘッダの削除や、分離時の IP ヘッダの付与に負荷がかかる。IP ヘッダを削除した分帯域を有効に使用することができる。
- 方式 3 ; IP ヘッダおよび TCP ヘッダを削除してパケットをつなぎ合わせる方式である。帯域の使用効率は高いが IP ヘッダや TCP ヘッダの削除や、付与に負荷がかかる。また TCP ヘッダは、先頭のヘッダのコピーでは、元のヘッダの情報を失ってしまう。
- 方式 4 ; 1 の方式と同様にヘッダ削除無であるが、パケットをつなぎ合わせた後その上からアセンブリヘッダ (IP ヘッダ) を追加する方式である。IP ヘッダを付与する分帯域利用効率は低下するが、アセンブリ、分離の負荷はほとんどかからない。

表 2.1 フォーマットの考察

| / | プロセッサ負荷 | データ量 | ヘッダの再現性 | 適用範囲 |
|------|---------|---------------|----------|--------------|
| 方式 1 | 低い | 変化無し | 容易 | すべての IP パケット |
| 方式 2 | 高い | IP ヘッダ 削除 | 容易 | すべての IP |
| 方式 3 | 高い | IP,TCP ヘッダ 削除 | 難(もどせない) | TCP パケットのみ |
| 方式 4 | 低い | IP ヘッダ 追加 | 容易 | すべての IP パケット |

表 2.1 のフォーマットについては考察の段階である. これ以降に出てくるパケットアセンブリシミュレーションについては, 方式 1 のフォーマットによるシミュレーションである.

2.4 Priority Queuing with Assembly 方式

本方式は図 2.5 のようにアセンブリを行わない優先キューとアセンブリを行う非優先キューの 2 つのキューを用いる. Queuing 時に Delay センシティブなパケットは優先キューに入れることにより低遅延な転送が可能となる.

それに対して Delay ロバストなパケットは非優先キューでキューイングの待ち時間を利用してアセンブリを行う. これを Priority Queuing with Assembly(PQwA) と呼ぶ.

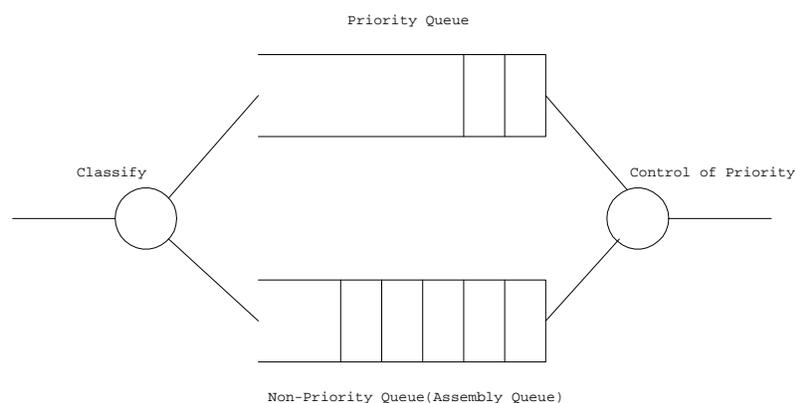


図 2.5 PriorityQueueingwithAssemblby

2.5 パケットアセンブリの手順

パケットアセンブリはキューイング時に行う。その手順を以下に示す。リアルタイムパケット、ノンリアルタイムパケットの例としてそれぞれ UDP パケット、TCP パケットを用いる。

2.5.1 アセンブリキューイング

アクセスネットワークから基幹ネットワークへパケットが転送される時でキューイングされる際、以下の手順で行う。また図 2.6 にそのフローを示す。

1. 到着したパケットはヘッダの PacketType を見る。UDP パケットであった場合は通常通りのキューイングを行う。
2. TCP パケットであった場合そのパケットのアドレスと、それまでにキューイングされていたパケットの中で、同一のアドレスのパケットがあるかどうかを検索する。同一のアドレスのパケットが無ければそのままキューイングする。
3. 同一のアドレスのパケットがあった場合、そのパケットが 6 回以上アセンブリされたパケットならばアセンブリは行えないのでそのままキューイングする。
4. 5 回以下であればヘッダにアセンブリの印をつける。
5. アセンブリ回数を 1 増加させる。
6. アセンブリした分のパケットの長さを変更しキューイングする。

2.5.2 Priority Queuing with Assembly

Priority Queuing with Assembly の場合は、優先パケット (UDP パケット) が到着したとき同じキューにキューイングするのではなく優先キューにキューイングする。これにより優先パケットの遅延を抑えることができる。図 2.7 にそのフローを示す。

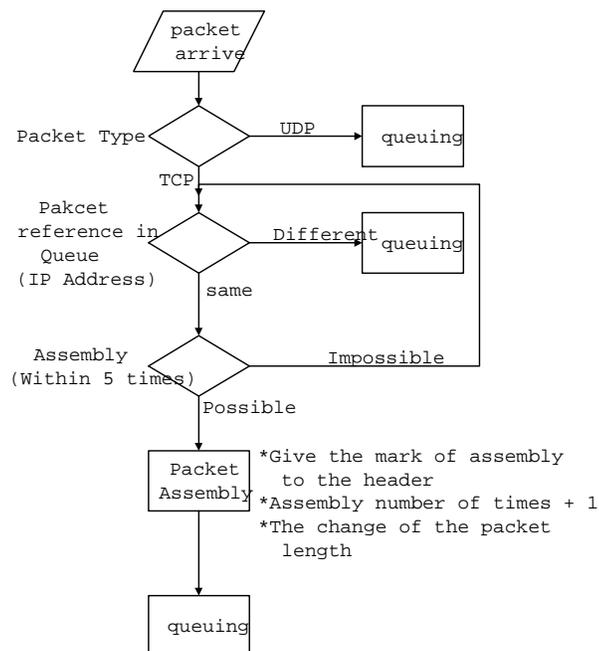


図 2.6 Packet Assemblby

2.5.3 アセンブリパケットの分離

基幹ネットワークからアクセスネットワークへパケットが転送される時でキューから取り出される際、以下の手順で行う。また図 2.8 にそのフローを示す。

1. 到着したパケットがアセンブリされたパケットかどうかを判別する。
2. アセンブリされたパケットならば何回アセンブリされたパケットかを調べる。
3. アセンブリ回数を 1 減らす。
4. パケットを分割する。
5. パケットの Length を変更する。
6. 分割されたパケットをキューイングする。

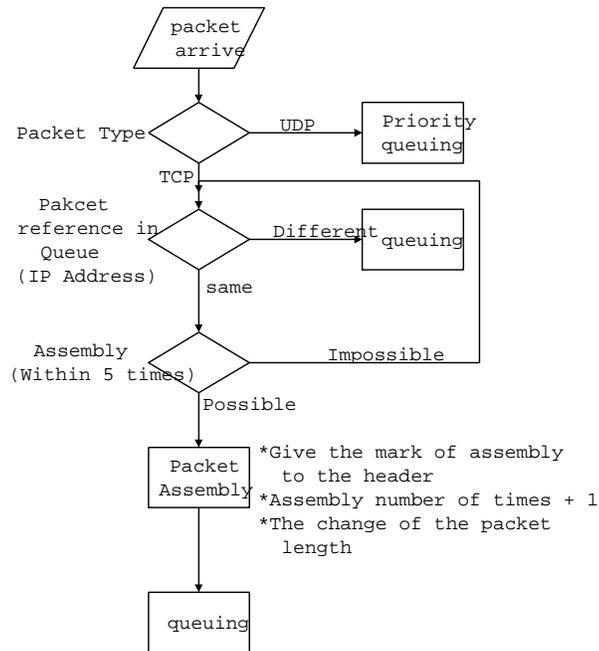


図 2.7 Priority Queuing with Assemblby

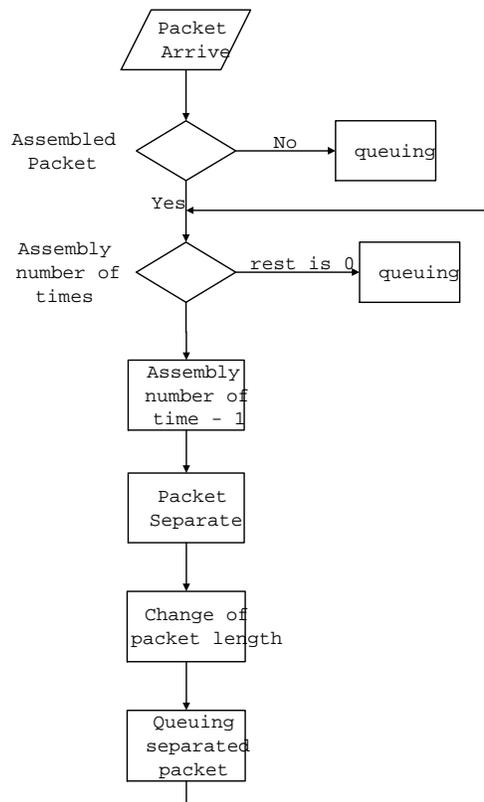


図 2.8 アセンブリパケットの分離

第 3 章

シミュレーション実験と結果

2 章の提案方式につき, その有効性についての基本検証を行った. その検証は ns(Network Simulator[7]) によるシミュレーション実験とした.

3.1 ns によるシミュレーションの条件

ns のルータ内遅延はキューイングとフォワーディングによってのみ起こる. ルータ内遅延の以外の遅延要素としてはノードノード間の LinkDelay がありこれは固定値である. ルーティングのヘッダ処理の遅延時間はその測定機能がない.

今回はヘッダ処理にかかる負荷や遅延時間は, 他のフォワーディングやキューイングなどの遅延に比べて, 極めて小さい値であると考えてそれらの遅延時間については, 測定結果に影響しないことを前提とする. また ns のシミュレーションでは, パケットのデータ部の値はシミュレートされていないので, パケットアセンブリでパケットを合成する際, パケットをつなぎ合わせるのではなく先頭のパケットのサイズを変更し, 他のパケットが一つのパケットにまとめられたことにしている. よって後続のアセンブリされるパケットのデータは基幹ネットワーク転送時には, 転送されていない. 分離時には元どおりになっているためデータ損失による通信上の影響は無い. ns は, パケットアセンブリをすることによるパケット転送の順序の入れ替わりによる遅延の

3.2 パケットサイズと帯域による遅延時間の違いのシミュレーション

3.2.1 シミュレーション実験

MTU の許す範囲でアセンブルを行うが、遅延時間や揺らぎの値に厳しいパケットには、アセンブリの処理は行わずに転送する。アセンブリによりパケット転送は効率化できるが、アセンブリされて長くなったパケットの転送に時間がかかることで短いパケットに遅延や揺らぎを起こす可能性がある。この遅延や揺らぎへの影響はバンド幅にも依存する。

そこで提案するパケットのアセンブリが行われる基幹ネットワークでは、長いパケットによる短いパケットの遅延や揺らぎは、バンド幅の広い場合、少ないことをシミュレーションで確かめた.[4]

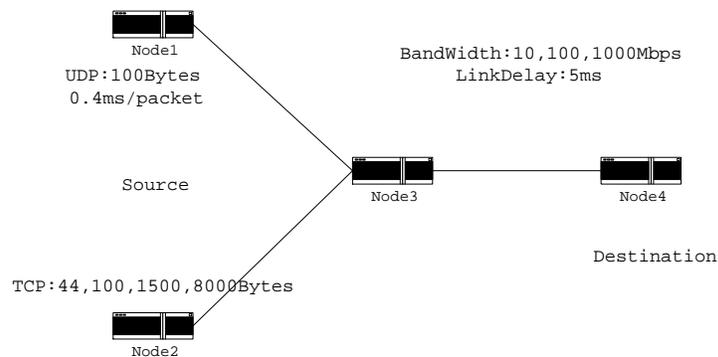


図 3.1 シミュレーション

図 3.1 のネットワークで Node1 から (VoIP パケットを想定して) UDP のパケットサイズ 44 バイトを等間隔で 0.4ms ごとに、Node2 から TCP のパケットサイズを 44, 100, 1500, 8000 バイトと変化させそれぞれ Node3 を通って Node4 ノードまで送信する。

各 TCP パケットサイズでネットワークのバンド幅を 10 , 100, 1000Mbps と変化させた場合の UDP パケットの遅延をシミュレータで測定した。

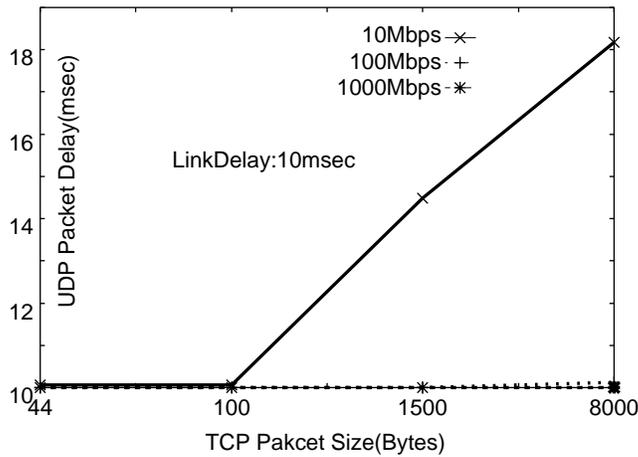


図 3.2 結果 UDP パケット遅延 (中央値)

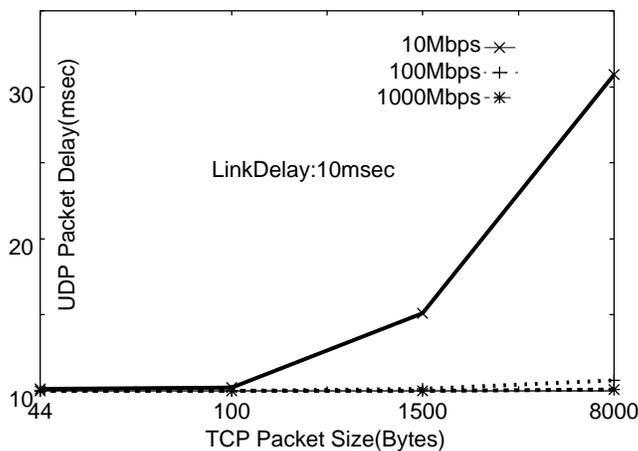


図 3.3 結果 UDP パケット遅延 (最大値)

3.2.2 シミュレーション結果

バンド幅 10Mbps, 100Mbps, 1000Mbps での UDP パケットの遅延の中央値を図 3.2 に、遅延の最大値の比較を図 3.3 に示す。いずれの図も縦軸に UDP パケットの遅延時間、横軸は TCP パケットサイズをとっている。アセンブリされた長いパケット (TCP パケット) が流れていてもバンド幅が広くなるに従いその影響が小さくなっていくことがわかる。

TCP パケットサイズが 8000 バイトの長いときでもバンド幅が 1000Mbps になると UDP パケットの遅延は 10.001ms ~ 10.065ms (このうち 10ms は Link Delay) で、そのときの揺らぎは最大でも 64 μ s と小さい。

3.2.3 考察

パケットのアセンブリを行う上で短いパケット疎通への悪影響が考えられたのでシミュレーションにより検証した。この実験によりバンド幅の広い基幹ネットワークではアセンブリした長いパケットを流しても、Delay センシティブなパケットに与える悪影響は軽微である。1 パケットのフォーワーディング遅延時間と帯域の関係は、

$$Delay = \frac{PacketSize}{BandWidth} \quad (3.1)$$

であり例えば狭帯域 (10Mbps) の場合は、

$$\frac{PacketSize}{BandWidth} = \frac{8000Bytes * 8bit}{10Mbps} = 0.0064 = 6.4ms \quad (3.2)$$

広帯域 (1000Mbps) の場合は

$$\frac{PacketSize}{BandWidth} = \frac{8000Bytes * 8bit}{1000Mbps} = 0.000064 = 0.064ms = 64 \mu s \quad (3.3)$$

となり帯域によって遅延時間は大きく変わることがわかる。

3.3 パケットアセンブリシミュレーション

3.3.1 シミュレーション実験

図 3.4 のモデルにおいて、送信端末 10 端末のうち一つの端末から UDP パケットをアセンブリなし、PacketSize 44Bytes、送出間隔 5ms で転送、また残りの 9 端末からは TCP パケットをアセンブリあり、PacketSize 1500Bytes で転送する。

9 端末からそれぞれ転送された TCP パケットは、E1 ノードにおいてアセンブリされいくつかのノードを経由して E2 ノードで分離された後、それぞれの受信端末へパケットが転送される。残りの 1 端末から転送された UDP パケットは E1 ノードにおいてアセンブリされない。

9 端末からの TCP パケット送信によってエッジノードにかかる負荷は平均約 48Mbps である。またアセンブリ時にはキューの中に 30 から 40 パケットがエッジノードでスタックされている状態である。

3.3 パケットアセンブリシミュレーション

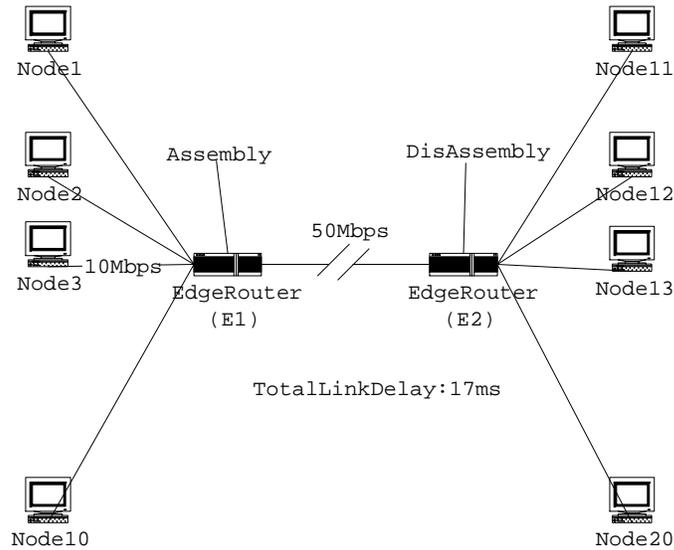


図 3.4 シミュレーションモデル

3.3.2 シミュレーション結果

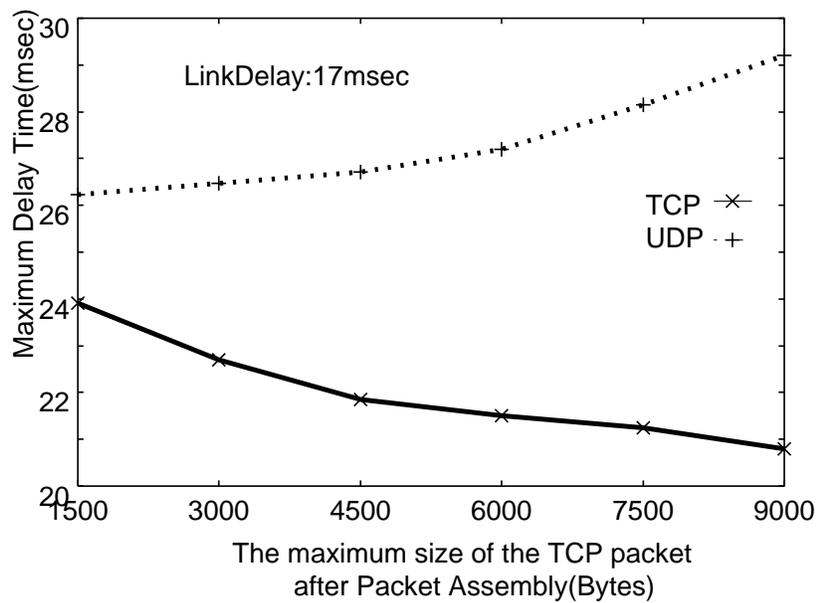


図 3.5 パケットアセンブリによる実時間通信パケットへの影響

図 3.5 にパケットアセンブリを行ったときの TCP, UDP パケットのそれぞれの遅延を示す。縦軸は UDP パケットの最大遅延時間, 横軸は TCP のアセンブリ最大パケットサイズを示す。

パケットをアセンブリするたびにキューの最後尾にキューイングする仕組みから、TCP パケットの最大アセンブリサイズを大きくするほど、TCP パケットはキューイングされる時間が長くなるため遅延が大きくなる。逆に UDP パケットは、TCP パケットのアセンブリによる順序の入れ替えで優先されることになった。よって TCP パケットの最大サイズを大きくするほど UDP の遅延は小さくなっていく。

3.3.3 考察

TCP パケットの遅延は、パケットアセンブリを行わないときと比べて、最大アセンブリサイズを 9000Bytes まで上げたときでも、その差は約 3ms である。これは TCP の通信の許容範囲内であると考えられる。しかし TCP パケットのアセンブリの最大サイズを大きくすればすればするほど UDP パケットの遅延が小さくなっているが、これは UDP パケットの転送される割合が少なかったためであることが考えられる。よって UDP パケットの割合が多くなれば大きな遅延を引き起こす可能性は否定できない。また負荷のかかり具合によって TCP パケットのアセンブリが大きくなっていっているため、アセンブリ最大サイズを 9000Bytes 以上に上げてシミュレーションしてみる必要もある。

3.4 Priority Queuing with Assembly シミュレーション

3.4.1 シミュレーション実験

Priority Queuing を用いることにより、UDP パケットのトラヒックを他のトラヒックより E 1 ノードで優先制御する Priority Queuing with Assembly のシミュレーションを行った。3.3 章の図 3.4 のパケットアセンブリシミュレーションと同様のモデルである図 3.6 において実験を行った。[5]

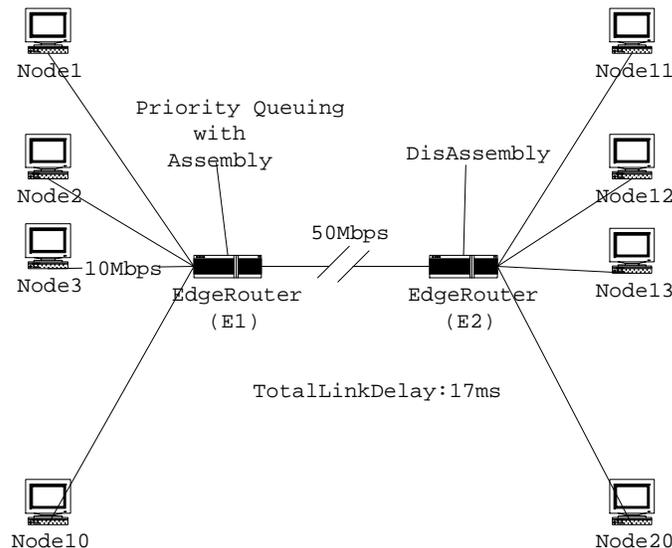


図 3.6 シミュレーションモデル

3.4.2 シミュレーション結果

図 3.7 に Priority Queuing with Assembly を行ったときの TCP, UDP パケットのそれぞれの遅延を示す。また図 3.8 に Priority Queuing with Assembly の導入前後の比較を示す。それぞれ、縦軸は UDP パケットの最大遅延時間、横軸は TCP のアセンブリ最大パケットサイズを示す。

E1 ノードでのアセンブリの際に、Priority Queuing with Assembly(PQwA) の導入前後で比較の実験を行った。図 3 の結果はアセンブリ後のサイズの最大値をパラメータに UDP, TCP の遅延を示す。TCP パケットは PQwA を導入前後でほぼ同一で、最大アセンブリサイズが大きくなるほど遅延も大きくなる結果となった。UDP パケットは PQwA を導入した時はしなかった時より遅延が抑えられ、9000Bytes まで TCP パケットをアセンブリした時でも最大で約 18ms の遅延にとどめられていることがわかった。

3.4.3 考察

Priority Queuing を導入すれば、TCP パケットの遅延についてはほとんど変化が無いが、UDP パケットの遅延の値がかなり低くすることができる。また最大アセンブリサイズを変

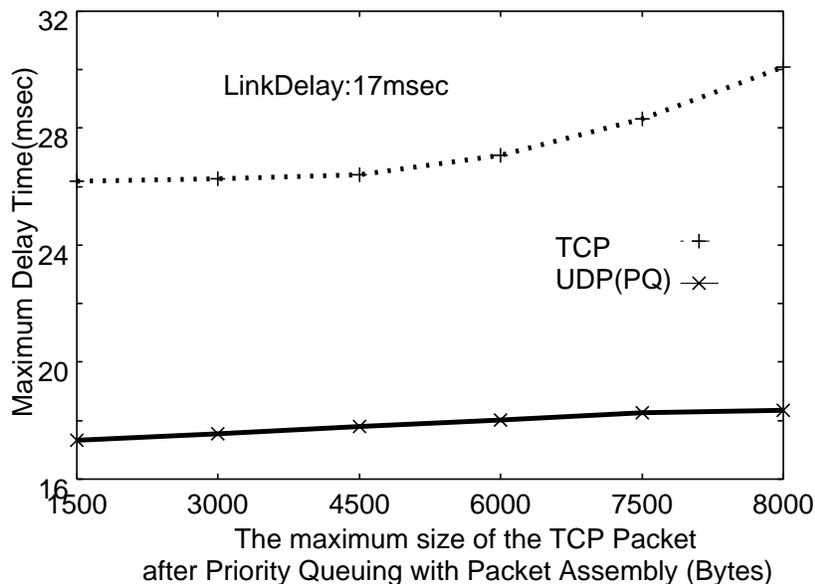


図 3.7 priority Queuing

化させていっても,Priority Queuing 導入前では約 4ms の遅延の変動があったが,その変動が約 1ms とほとんど無く安定させることができた.

3.5 むすび

今回のパケットアセンブリは,パケットの転送待ちによるキューイング時間を利用してアセンブリを行った. よってある程度の高い負荷がかからなければキューの中に同一の行き先のパケットがあることがなくなりアセンブリができない. アセンブリするエッジノードでは,大きな負荷がかかるとは考えにくい. 大きな負荷がかかるとした場合,アセンブリを行っていたのでは逆にエッジノードでのプロセッサ負荷が大きくなりそこがボトルネックになってしまう.

またパケットアセンブリノードにパケットが到着したときに,そのノードにキューイングされているすべてのパケットの中から一致しているアドレスを検索する方式をとっていた. ルータ内遅延をシミュレーションしなかったことからこれについての影響は結果として出てこなかったが,パケットアセンブリを実装するためには効率のパケット検索法を用いること

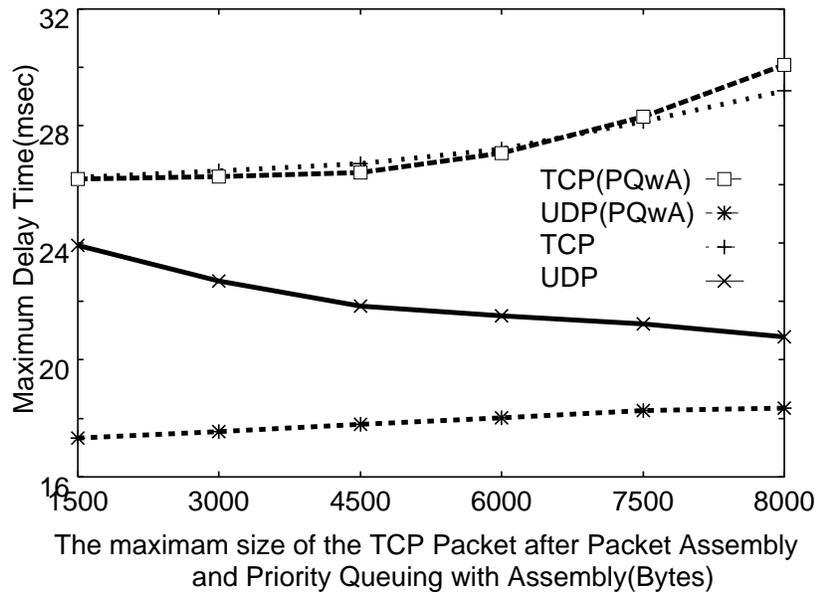


図 3.8 PriorityQueuing 導入比較

は重要である。

アセンブリシミュレーションで転送したトラヒックについては、パケットサイズや転送タイミング、負荷が一定であった。パケットアセンブリはトラヒックによる影響を大きく受けると考えられるので、これらの値を変化させた時のアセンブリされた回数やそれによる Delay について考える必要がある。

第 4 章

考察

4.1 パケットアセンブリの必要性

エラー率の違いは、MTU の異なるネットワークが存在する一つの理由である。パケットアセンブリは、MTU の大きいネットワークで MTU よりも遥かに小さいパケットが流れそれによって起こる負荷を軽減するものである。光ファイバが普及して端末間でエラーがほとんど起こらないことが保証されれば、すべてのネットワークで MTU が大きくなれば、MTU サイズに着目したパケットアセンブリができなくなる。

しかし転送速度の面から考えると、基幹ネットワークと比較して、低速なアクセスネットワークで大きなパケットを投げることは、短パケットに分割して通信しているリアルタイムアプリケーションに大きな影響を与えてしまう。大容量化が進んでアクセスネットワークの転送が高速になっても、リアルタイムパケットの Delay は可能な限り小さいほうが良い。

転送速度の低いアクセスネットワークでは、制限されたパケットサイズで転送して Delay を防ぎ、高速の基幹ネットワークでは、大きなパケットを投げて負荷軽減を行う必要がある。この機能を実現できるのがパケットアセンブリである。

またモバイル端末による通信が発達することを考えると、電波を使った通信ではエラー率が高くデータを確実に伝えることが保証できないので、MTU は小さく設定される。無線から有線に変わるポイントでパケットアセンブリを行う必要が出てくる。

4.2 アセンブリするパケット

現在のトラヒックの種類で、TCP パケットの数は約 8 割を占めている。これらすべてのパケットをアセンブリすることになれば、ルータの負荷問題の大きな改善が可能である。

しかし近年のリアルタイムパケットを使った通信の発達により、アセンブリが可能であるパケットの比率が低くなっていくことも考えられる。リアルタイムパケットでもその度合いにより、アセンブリを行っていく方向で考えることも必要になる可能性がある。

4.3 パケットアセンブリ時にエッジルータにかかる負荷

エッジルータにパケットアセンブリの処理を持たせることは、そのルータに負荷がかかりそこをボトルネックになる危険がある。特にアセンブリパケットの分離時には、ルータ到着した一つのアセンブリパケットの入力に対して、分離後複数のパケットを出力するのでアセンブリした回数分だけ負荷がかかる。

これにはまず、エッジルータでの負荷は、トラヒックが集まってくるコアルータの負荷よりも小さいので、アセンブリによってエッジルータが過負荷になることはないと考えることができる。次にパケットアセンブリのフォーマットを選ぶ際、できるだけ処理のステップ数が少なくアセンブリ、分離のできる負荷のかからないフォーマットを選ぶことが考えられる。またアセンブリパケットの検索やパケット合成のアルゴリズムについての最適化することがある。さらにハードウェアでパケットアセンブリの処理を行うことにより高速化を図っていく。エッジルータだけを高機能化すれば良いので、残りの基幹ネットワークルータにネットワーク構築時に大きな負担がかからなくてすむという考え方もある。

4.4 アセンブリパケットの Loss

パケットをアセンブリして長いパケットにすると、そのパケットにエラーが起こったときに影響が大きくなってしまふということが考えられる。

これに対してはまずパケットアセンブリを行う基幹ネットワークの光ファイバでのエラー

レートはほとんど無いといえる。またパケットアセンブリを行わないときにルータの処理能力の不足でパケットが破棄されてロスが起こる場合でも、パケットアセンブリを行ってれば逆にロスが防げるという結果になる。

4.5 基幹ネットワークルータの負荷が低い場合

ネットワークトラヒックは時間帯によって大きく変化する。基幹ネットワークのトラヒックの負荷が高いときには、パケットアセンブリを行うことによって次のホップから基幹ネットワークルータの負荷が軽減されることになり、エッジルータで多少の手間をかける意味が出てくる。しかし、トラヒックが少ないときにアセンブリを行うのは意味がない。エッジルータでアセンブリを行う時間や負荷が無駄になる。そこでエッジルータにトラヒックを監視させアセンブリを行うかを判断することが考えられる。

現在のデータ転送では、ネットワーク経路のどこで負荷がかかっているのかを特定し、それを避けることは難しい。トラヒックの混雑を制御用のパケットでルータ同士が通信し、その情報を交換することで負荷を回避することが可能であるが、制御信号を転送することでさらにネットワークに負荷がかかってしまう。

そこでエッジルータで delay や throughput を観測することにより、それらの値が悪化したときにパケットアセンブリを行うようにすればよい。これは負荷のかかっている場所を特定する必要がなく、ネットワークの負荷に対応できる。しかし Delay や Throughput の値は頻繁に変化するものであるため、切り替えの閾値の調節が重要である。

第 5 章

今後の課題

5.1 アセンブリのパケットフォーマットと負荷

今回のパケットアセンブリは ns のシミュレーション条件よりヘッダ削除なしでシミュレーションを行った。オーバーヘッドの大きいパケットについてはヘッダも削除してアセンブリするほうがより効率の良い転送を行うことができる。しかし削除しすぎても、削除や復元にかかる負荷が大きくなりすぎてエッジルータがボトルネックとなってしまう。今後はパケットフォーマットの違いによるヘッダ処理を考えたシミュレーションを行っていく。

5.2 マルチフローアセンブリ

今回はアドレスが完全に一致したパケット同士でアセンブリを行った。しかし基幹ネットワークでのアセンブリを行う際に、アドレスが完全に一致していなくても行き先が同一のドメインであれば（アドレスの途中までが同じ）基幹ネットワーク内のルートは同じである。これを考えると別のフローのパケット同士でのアセンブリが可能となる。

TCP の Ack パケットはバースト的に流れてこないためアセンブリできなかったが、小さいパケットの中で Ack パケットはトラヒックのかなりの部分を占めるのでアセンブリの効果が、マルチフローアセンブリを行えば TCP の Ack をアセンブリできると考えられる。Ack パケットのオーバーヘッドはパケットの約半分と大きいため IP ヘッダを削除する方式と組み合わせることができれば、帯域利用効率の大幅な向上にもなる。

5.3 パケット転送スケジューラによるパケットアセンブリ

今回のアセンブリではエッジルータにある程度負荷がかからなければアセンブリが行われなかった。エッジルータで負荷がかかっていなくても基幹ネットワーク内のルータで大きな負荷がかかっていることも考えられる。よってエッジルータで負荷がかからなくてもアセンブリできるようにキューで一定期間パケットを待機させてアセンブリを行う方法についての研究を進めていく。図 5.1 にその構造を示す。

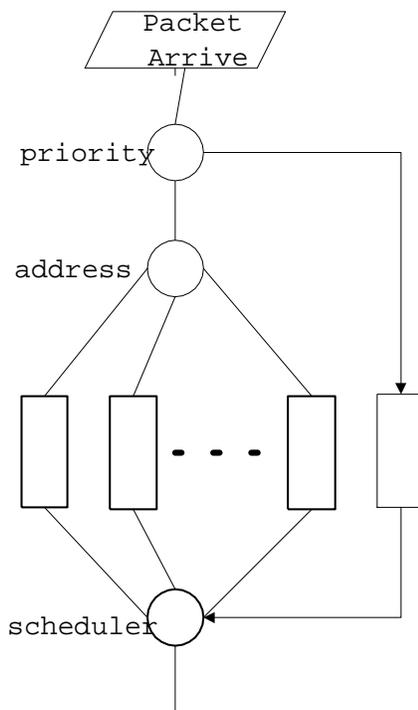


図 5.1 scheduler によるアセンブリ

5.4 ルータ内遅延を含めたシミュレーション

今回はヘッダ処理にかかる負荷や遅延時間は、他のフォワーディングやキューイングなどの遅延に比べて、極めて小さい値であると考えてシミュレーションを行ってきた。しかしパケットアセンブリはルータのヘッダ処理の負荷軽減を目的としているため、ルータ内遅延を含めずにシミュレーションを行ってもパケットアセンブリによってどの程度 QoS が改善さ

れたかを値として出すことができなかった。

しかし商用のシミュレータである OPNET ではルータ内遅延をシミュレーションしている
ので、負荷を考えた測定ができればパケットアセンブリによる QoS の改善について調査する
ことができると考えられる。

謝辞

本研究を行うに際し、貴重な御助言や多大なる御指導を頂いた高知工科大学工学部情報システム工学科の島村 和典 教授, ならびに通信・放送機構高知トラヒックリサーチセンターの神田 敏克氏に深く感謝いたします。また, 本研究室院生 中平 拓司氏ならびに本研究室学部生の皆様に感謝いたします。さらに在学中, 親切なる御助言を頂いた岩田 誠 助教授, 福本 昌弘 講師をはじめ諸先生方に心より感謝いたします。

参考文献

- [1] <http://www.vbns.net:8080/presentations/papers/MCItraffic.pdf>
- [2] <http://www.darpa.mil/ito/research/ngi/supernet.html>
- [3] 神田, 島村; パケットサイズ制御によるルータでのデータ転送効率向上に関する考察, 信学技報, SSE2000-114, IN2000-65, CS2000-45(2000)
- [4] 浦西, 神田, 島村; パケットアセンブリによるトラフィックへの影響に関する一考察, 電子情報通信学会通信ソサイエティ大会, B-11-19, 一般講演 (2000)
- [5] 浦西, 神田, 島村; パケットアセンブリにおけるプライオリティキューイングを利用した実時間トラフィックの性能向上, 電子情報通信学会総合大会, 一般講演 (投稿中)2001
- [6] GLOBECOM IEEE Global Communications Conference; <http://www.comsoc.org/confs/globecom/index.html>(2001/2/5 現在)
- [7] ns;<http://www.isi.edu/nsnam/ns/>(2001/2/5 現在)