

Title	リアルタイムトラヒックのための効率の良いスイッチング方式に関する研究
Author(s)	前田, 庄司
Citation	
Issue Date	2005-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1936
Rights	
Description	Supervisor:日比野 靖, 情報科学研究科, 修士

修 士 論 文

リアルタイムトラヒックのための
効率のよいスイッチング方式に関する研究

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

前田 庄司

2005年3月

修士論文

リアルタイムトラヒックのための
効率のよいスイッチング方式に関する研究

指導教官 日比野靖 教授

審査委員主査 日比野靖 教授
審査委員 田中清史 助教授
審査委員 丹康雄 助教授

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

310099 前田 庄司

提出年月: 2005 年 2 月

目次

第1章	序論	1
1.1	背景と目的	1
1.2	ラベル付き同期通信の提案	1
1.3	本論文の構成	2
第2章	ラベル付き同期通信	3
2.1	概観	3
2.2	データの転送	3
2.3	スイッチング方式	4
2.4	ルーティング方式	6
第3章	QoS 保証機構	8
3.1	サービスクラス	8
3.2	タイムスロットのスケジューリングポリシー	8
3.2.1	MVA	8
3.2.2	AVA	9
3.2.3	BEA	9
3.3	接続制御	9
3.4	プリエンプション	9
第4章	トラヒックモデル	10
4.1	リアルタイム型トラヒック	10
4.2	ベストエフォート型トラヒック	10
4.3	シミュレーション実験のトラヒックモデル	12
第5章	シミュレーション実験	14
5.1	各サービスクラス単独の場合	14
5.2	混合比が等しい場合	14
5.2.1	パケット単位での遅延	14
5.2.2	フレーム単位での遅延	17
5.3	混合比が異なる場合	20

5.3.1	MVA トラヒックが多い場合	20
5.3.2	MVA トラヒックが少ない場合	23
5.3.3	ベストエフォート型トラヒックが混在する場合	23
5.4	優先制御のみを行った場合との比較	28
第 6 章	結論	31
6.1	要旨	31
6.2	結び	32

第1章 序論

1.1 背景と目的

近年のブロードバンド環境の普及により、リアルタイムビデオストリーミングのようなトラフィックが急速に増加している。そして現在、非同期パケット通信ですべての通信を統合していこうという流れになっているが、このような方法では、帯域の保証や、要求する時間内に相手にデータを届ける保証が難しく、時間的制約が厳しい動画配信のような同期的データトラフィックを扱うには不向きである。そこで、非同期式ではなく、同期式の通信を基礎とすればこの QoS^1 に関する問題を解決することができる。しかし、同期式通信では、回線の利用効率が悪い。本研究では、今後増加していくことが予測されるリアルタイムトラフィックに適し、効率がよく、低コストなスイッチング方式を提案することを目的とする。

1.2 ラベル付き同期通信の提案

統計多重を行うラベルスイッチングは、様々なトラフィックに対して回線の利用効率はよいが、実時間性を伴うトラフィックの QoS 保証は困難である。一方、同期式の時分割多重は、実時間性の保証は可能だが、回線の利用率が悪く、コストが高い。そこで、回線交換による同期式通信・時分割多重を基礎とし、ラベルスイッチングの技術を併用することによって、非同期式通信の特徴も併せて持つラベル付き同期通信を提案する。

本研究で提案するラベル付き同期通信では、ラベル付きパケットを用い、その転送は所定のタイムスロットで行うものとし、このタイムスロットの割り当てにより転送レートを決める。そして、空いているタイムスロットにはベストエフォート型にパケットを割り当てる。これらを同期型パケット、非同期型パケットとして区別し、非同期型パケットのタイムスロットは、同期型パケットがそれを必要とする場合、転送を放棄し、同期型パケットに譲ることとする。同期型パケットのタイムスロットはコネクション確立時に全ノードを予約する集中制御方式とし、タイムスロットのスケジューリングは各々のノード間での分散制御とする。そして、本提案方式における QoS 保証機構を設計する。ここで特に問題となるのが、実時間性が求められ、かつビットレートが一定でない場合である。ラベルが付いているので、許容範囲であれば既に他のコネクションによって予約されたタ

¹Quality of Service : サービス品質

タイムスロットの空き部分に別の接続のデータが混在してもかまわない。

最後に、これらのシミュレーションを行い、QoSの保証や回線の使用率について評価を行う。

1.3 本論文の構成

本論文の各章では、

第1章：本研究に至った背景・研究の目的と、提案する方式の概要を述べる。

第2章：本提案方式の詳細を述べる。

第3章：本提案方式におけるQoSの保証機構について述べる。

第4章：本提案方式が対象とするトラフィックモデルについて述べる。

第5章：シミュレーション実験により、本提案方式の評価を行う。

第6章：全体をまとめ、結論を述べる。

第2章 ラベル付き同期通信

2.1 概観

本研究で提案するラベル付き同期通信の概観を、図 2.1 に示す。転送するデータとして、

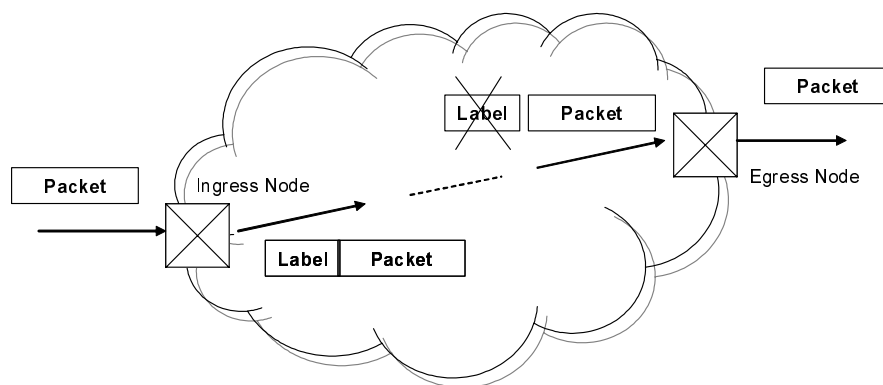


図 2.1: ラベル付き同期通信の概観

可変長パケットを扱う。ラベル付き同期通信網に入ったパケットは、入り口のノードでラベルがつけられ、そのラベルを参照することによって網内でスイッチングされ、出口のノードまで到達すると、ラベルを除去し、網を抜ける。

通常、同期通信ではタイムスロットの位置を固定してしまうが、これを動的に移動することができるように、同期・非同期パケット全てに対してラベルを付加する。

2.2 データの転送

転送されるデータの構造を、図 2.2 に示す。データの転送はフレームというデータ単位で行う。SDH¹/SONET²への適応が容易なように、SDH の 125[μ s] フレーム 8 つで、本提案方式における 1 フレーム (1 [ms]) を構成する。そして、そのフレームを、各々が 64 バイト (512 ビット) のタイムスロットに分割し、このタイムスロット単位で管理を行う。このタイムスロットの数は、物理回線の伝送速度に依存する。また、各フレームの最初の

¹Synchronous Digital Hierarchy : 同期デジタルハイラキ

²Synchronous Optical Network : 同期光通信網

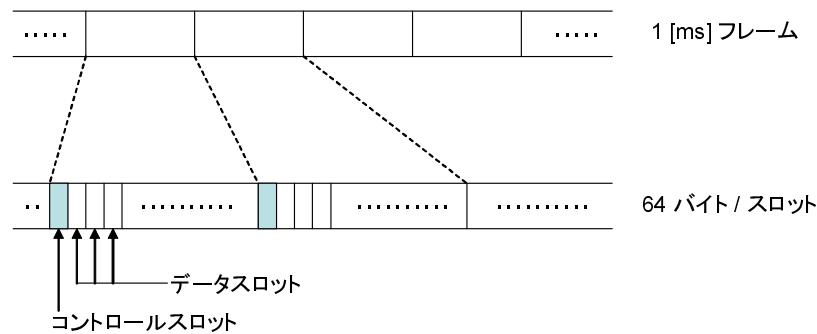


図 2.2: 転送されるデータの構造

タイムスロットは、シグナリングのための専用のタイムスロットとする。これにより、輻輳が起こったときでも、確実に制御パケットを送ることができる。

本提案方式は、STM-1/OC-3 以上のバックボーン回線への適用を想定する。STM-1/OC-3 における例を表 2.1 に示す。

表 2.1: STM-1/OC3 の例

物理回線	STM-1/OC-3 伝送速度: 155.52	単位 [Mbps])
毎秒のフレーム数	1,000	[フレーム]
1 フレーム	159	[Kbit]
1 フレーム (ホップ) の遅延	1	[ms]
1 タイムスロット	512	[bit]
フレーム毎のタイムスロット数	318	[スロット]
シグナリング帯域	500	[Kbps]

2.3 スイッチング方式

網内を流れるパケットには、図 2.3 に示されるようなラベルが付けられている。各フィールドの内容は、表 2.2 に示す。

そして、各ノードには、図 2.3 に示されるような構造のエントリをまとめたフォワーディングテーブルがあり、コネクション ID と、入出力ポートの組み合わせが書き込まれている。各フィールドの内容は表 2.3 に示す。ラベルに書かれたコネクション ID を元に、フォワーディングテーブルを参照して出力ポートを決定する。

また、最後にコネクション ID が参照されてからの時間を確認することによって、一定

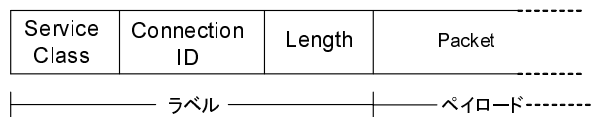


図 2.3: ラベルの構造

表 2.2: ラベルのフィールド

フィールド	内容
Service Class	サービスクラス (3 章で述べる) を示す - 11:MVA 10:AVA 01:BEA
Connection ID	コネクション ID
Length	長さを必要なタイムスロット数で表す

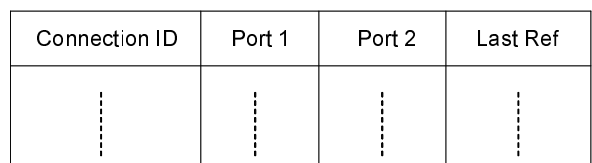


図 2.4: フォワーディングテーブル

表 2.3: フォワーディングテーブルのフィールド

フィールド	内容
Connection ID	コネクション ID
Port 1	入出力ポートの組み合わせ
Port 2	
Last Ref	最後に参照されてからの時間

時間参照されていない接続については、それを管理ノードに通知し、予約されたタイムスロットの開放を行う。

2.4 ルーティング方式

簡単なネットワークの例を図 2.4 に示す。

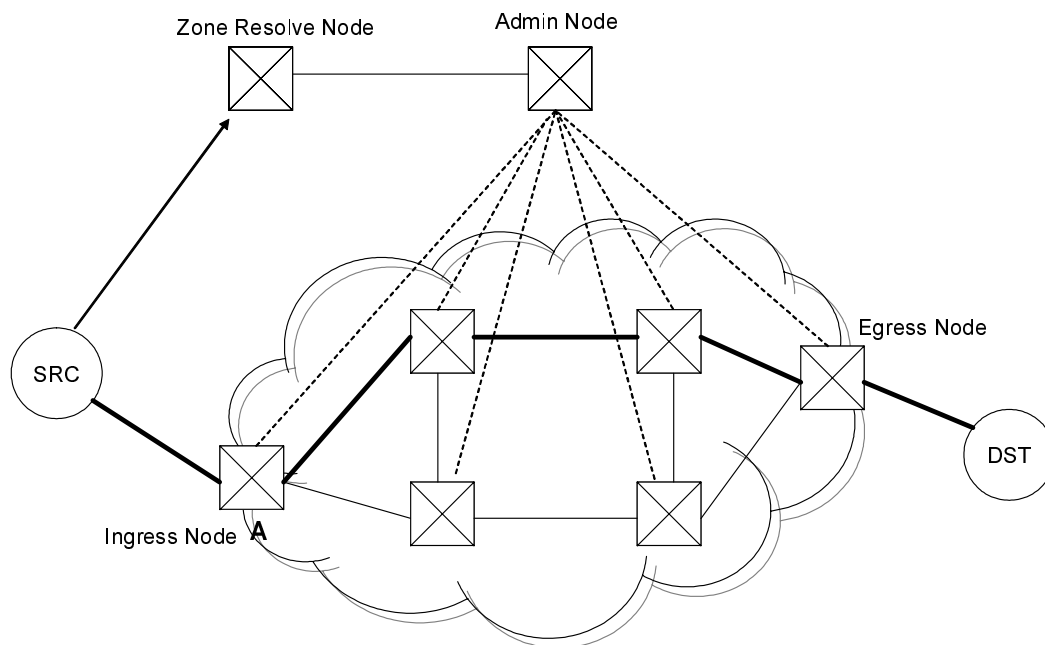


図 2.5: ルーティング

コネクションを確立するに当たっては、まず入り口ノードが管理ノードに対して要求を行う。管理ノードは、そのデータを転送するパスを決定し、そのコネクションに対して、コネクション ID を発行する。そのコネクション ID をラベルに書き込み、転送を開始する。

ここで、注意しなければならないのは、QoS を保証する必要があるトラヒックは、管理ノードの決定したパスを通すために、進入ノードが一定でなければならない。そのために、QoS 保証を必要とする場合は、ゾーン解決ノードに対して、自分のいるゾーンを明示して、どのノードを進入ノードとして選べばよいかを問い合わせる。そして、データの宛先 DST に対する IP ヘッダとは別に、進入ノード A に対する IP ヘッダを付け、確実にノード A を通るようになる。ノード A では、この IP ヘッダを除去し、ラベル付き同期通信網でのラベルをつけ、転送を開始する。

QoS を保証する必要のないトラヒック（ベストエフォート型トラヒック）については、各ノードから、全てのノードに到達するためのコネクション ID を 1 つ用意しておき、そのコネクション ID をラベルに書き込みイッチングを行う。基本的には、この固定ルート

で要求を処理するが、トラヒックの増加等の原因により、そのパスで要求を受け付けられなくなると、管理ノードは新たに追加のパスを決定し、各ノードに対して通知を行う。

管理ノードは、図 2.4 に示されるようなルーティングテーブルを持っており、これによって経路制御を行う。経路の決定では、なるべくトラヒックの少ない経路を選ぶものとし、これは、グラフ理論の最大フロー問題において、その解法が示されている。

Enable / Disable	SRC	DST	Route Info	Connection ID
⋮	⋮	⋮	⋮	⋮

図 2.6: ルーティングテーブル

各フィールドの内容は、表 2.4 のとおり。

表 2.4: ルーティングテーブルのフィールド

フィールド	内容
enable/disable	このコネクションが有効か無効か
SRC	入口ノード
DST	出口ノード
Route Info	通過するパス上のノード
Connection ID	コネクション ID

第3章 QoS保証機構

3.1 サービスクラス

要求される QoS 保証の度合いに応じて、以下のような 3 つのクラスに分類する。

Maximum Value Allocation (MVA) : 最大のビットレートで帯域の保証を行う。

Average Value Allocation (AVA) : 平均のビットレートで帯域の保証を行う。

Best Effort Allocation (BEA) : 保証は行わない。

クライアントアプリケーションは、接続要求を行う際に、どのクラスでの転送を要求するかを進入ノードに対して通知する。

図 3.1 に示されるように、各ノードは、スケジューリングユニットを持っており、各タイムスロットに書き込まれるデータを管理する。現在のフレームを送出している間に、次に送出するフレームの転送データの書き込みを行う。

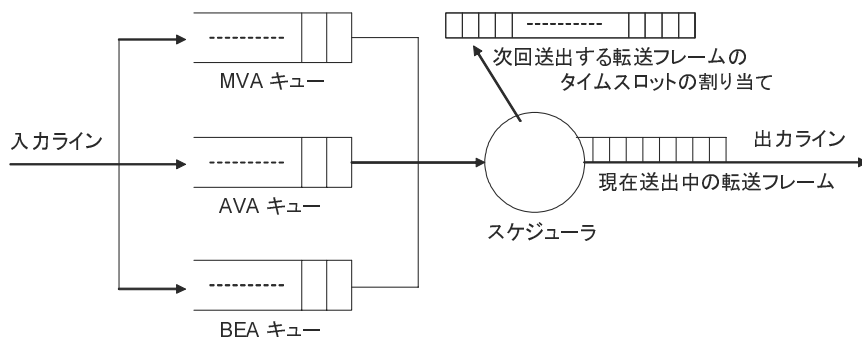


図 3.1: スケジューリングユニット

3.2 タイムスロットのスケジューリングポリシー

3.2.1 MVA

要求のあった最大のビットレートでタイムスロットの予約を行い、MVA によって保証されたデータが到着した場合は、全てのトラフィックに優先してこれをタイムスロットに書

き込む。データの転送に必要な数の空きスロットが無かった場合、MVAによって保証されたもの以外のデータが入る予定のタイムスロットが無いかを調べる。BEA・AVAの順に調べ、もし存在すればMVAで置き換え、AVA・BEAで既書き込まれたデータが存在すれば、AVA・BEAキューに戻す。

3.2.2 AVA

要求のあったビットレートの平均値でタイムスロットの予約を行い、AVAによって保証されたトラフィックは、MVAの次に優先される。MVAによって保証されたトラフィックが占有しているタイムスロット以外には優先して書き込むことができる。データの転送に必要な数の空きスロットが無かった場合、BEAトラフィックが利用しているタイムスロットがないか調べる。もし存在すれば、AVAでおきかえ、BEAで既書き込まれたデータが存在すれば、BEAキューに戻す。

3.2.3 BEA

QoS保証を必要としないベストエフォート型トラフィックは全てこのクラスで転送される。BEAトラフィックは、MVA、又はAVAによって保証されたトラフィックが予約していないタイムスロットを使用して転送される。

3.3 接続制御

クライアントからの要求は、まず管理ノードがその要求を受け付けてよいかどうかを判断する。判断の指標は、QoS保証を必要とするトラフィックであれば、その要求を満たすことのできる経路が存在するかを判断する。QoS保証を必要としないトラフィックは、網内にそのデータを流して他のコネクシオンに影響を与えるようであれば、網内へのパケットの進入をブロックする。これにより、ネットワークが能力にみあったデータの転送を行うことが可能となり、トラフィックがあふれてしまうような事態を未然に防ぐことができる。

3.4 プリエンプション

本提案方式では、同期フレーム転送を行っていることを利用して、パケットの優先制御に加え、先の節で述べたように、優先度がより上位のパケットを、既にスケジュールされている下位のパケットと置き換える、プリエンプションを行う。これにより、MVA・AVA保証を行ったトラフィックの品質をより良くすることができる。

第4章 トラヒックモデル

本章では、シミュレーションで採用する2種類のデータ、リアルタイム型ストリームデータ及びベストエフォート型パケットデータのトラヒックモデルを決定し、そのモデルについての説明をする。また、第5章で行うシミュレーション実験で使用するトラヒックモデルについても説明をする。

4.1 リアルタイム型トラヒック

リアルタイム型トラヒックとして、ビットレートが平均1 [Mbps] のMPEGのようなストリーミングデータを仮定し、このストリーミングの本数によって、リアルタイム型トラヒックが占める帯域を決定するものとする。このストリーミングデータの特徴は以下のとおり。

- ビットレートは平均1[Mbps] で、最大3[Mbps] 程度になる。
- 毎秒のフレーム数は平均30 フレーム/秒 (33[ms] 周期) とする。
- フレームの発生間隔は0 ~ 66[ms] の一様分布とする。
- GOP¹構造は、Iフレーム・Pフレームがあり、これらが1:2の割合で出現するものとする。
- 1フレームのサイズは、平均のフレームサイズを中心とした正規分布とする。
- IフレームとPフレームのフレームサイズの比は、5:1 とする。

このトラヒックの各フレームのサイズ分布を、図4.1に示す。

4.2 ベストエフォート型トラヒック

ベストエフォート型のトラヒックとしては、各パケットのサイズが平均1[Kbyte] のようなものを仮定し、この発生間隔によって、ベストエフォート型トラヒックの占める帯域を決定する。このパケットの特徴は以下のとおり。

¹Group of Picture

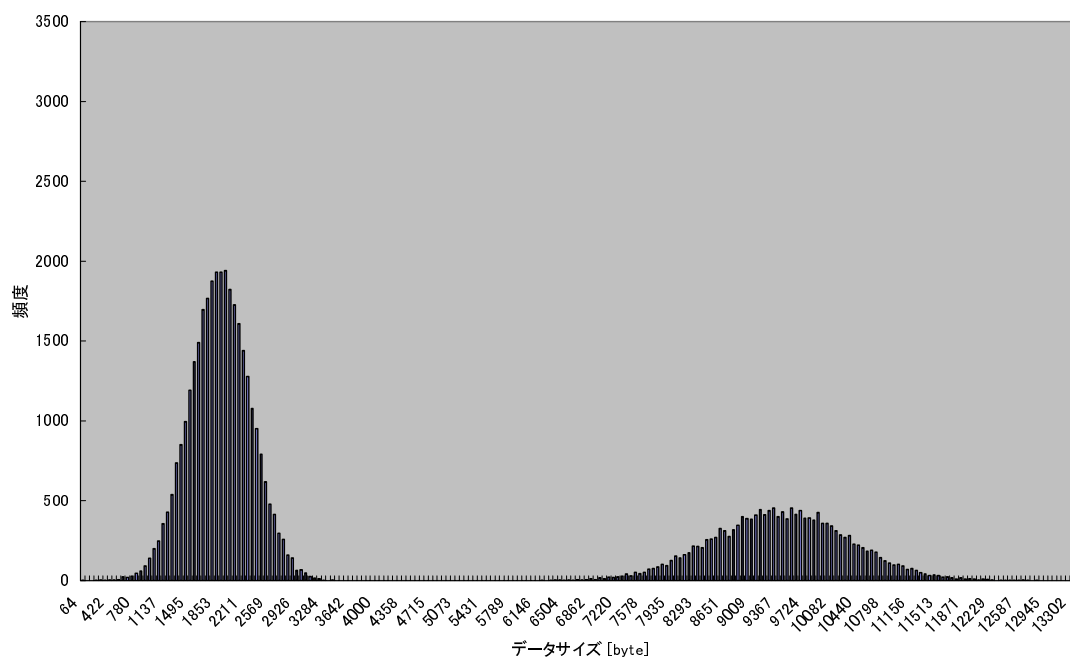


図 4.1: ストリーミングデータの (I,P) フレームサイズの分布 - サンプル数 : 50,000 フレーム

- パケットのサイズは平均 1[Kbyte] で、1 ~ 1.5[Kbyte] 付近のものが多くような分布とする。
 - このパケットの到着率は λ で、到着間隔は $1/\lambda$ の負の指数分布とする
- このトラフィックの各パケットのサイズの分布を、図 4.2 に示す。

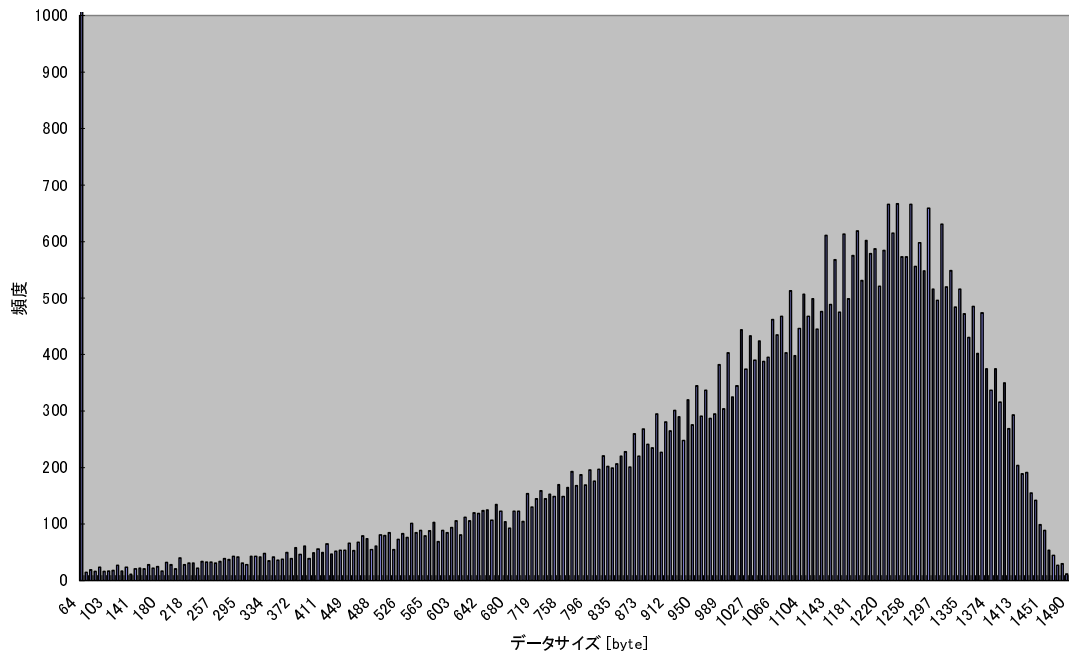


図 4.2: ベストエフォート型データの packet サイズの分布

4.3 シミュレーション実験のトラフィックモデル

シミュレーション実験では、

1. 各サービスクラスのを単体で流した場合
2. 各サービスクラスの要求する帯域が等しい場合
3. 各サービスクラスの要求する帯域が異なる場合

の 3 種類の場合について実験を行う。1,2 では、本提案方式の基本的な性質を明らかにする。2,3 において、異なるサービスクラスを要求するトラフィックが混在した場合について検討を行う。MVA での保証を要求するトラフィックの占める帯域が、AVA での保証を求めらるトラフィックの占める帯域と等しい場合、また MVA トラフィックの存在する帯域が極端に多い場合、少ない場合について、QoS がどの程度保証されるのかを検証する。

具体的には、表 4.1 に示すような割合で混合する。

表 4.1: トラヒックモデル

	MVA	AVA	BEA
1.	100%	-	-
	-	100%	-
	-	-	100%
2.	50%	50%	-
	33%	33%	33%
3.	80%	20%	-
	20%	80%	-
	72%	18%	10%
	18%	72%	10%

第5章 シミュレーション実験

本章では、第2章の表2.1で示した物理回線、STM-1/OC-3の例を元にシミュレーション実験を行う。第4章で示したトラヒックを、本提案方式の交換網に流した場合の、使用率0.1~0.9における画像フレーム、パケットの遅延をみることにより、本提案方式の評価を行う。遅延のゆらぎが、画像フレームの発生周期である33[ms]以内であれば、許容範囲内であるとする。ここで使用率とは、流れる全トラヒックが使用する平均帯域の、物理回線の全帯域に対する割合である。

シミュレーションにおいては、パケットが進入ノードから入力され、同期フレームで出力される様子を見ていくが、パス上の中継ノードでは、同期転送を行っているため、遅延のゆらぎが増加することはない。また、接続制御を行い、経路上の全ノードに対して帯域の予約を行っているため、中継ノードにおいてデータがあふれることも、基本的にはない。

5.1 各サービスクラス単独の場合

MVA又はAVAトラヒックを単体で流した場合のI,Pそれぞれのフレームの遅延時間の平均を図5.1に示す。使用率が0.9になっても、3転送フレーム程度の遅れで転送できている。

次に、パケットレベルでの遅延について、BEAトラヒックのみを流した場合のパケットの遅延を、図5.1に示す。こちらでも、2転送フレーム遅れ程度で転送ができている。フレーム転送を行っているため、FIFOキューで非同期転送を行う場合に比べ、使用率が低い状態でも、ある程度の遅延が発生しているが、使用率0.8程度まで遅延がほぼ一定になっており、ゆらぎが少ない。

5.2 混合比が等しい場合

5.2.1 パケット単位での遅延

MVAトラヒックとAVAトラヒックを1:1で混合した場合のパケットの遅延の平均、及び最大値を図5.2.1、図5.2.1に示す。平均の遅延では、MVA保証をしたもの、AVA保証をしたものいずれも許容範囲内におさまっている。しかし、最大値でみると、使用率0.9

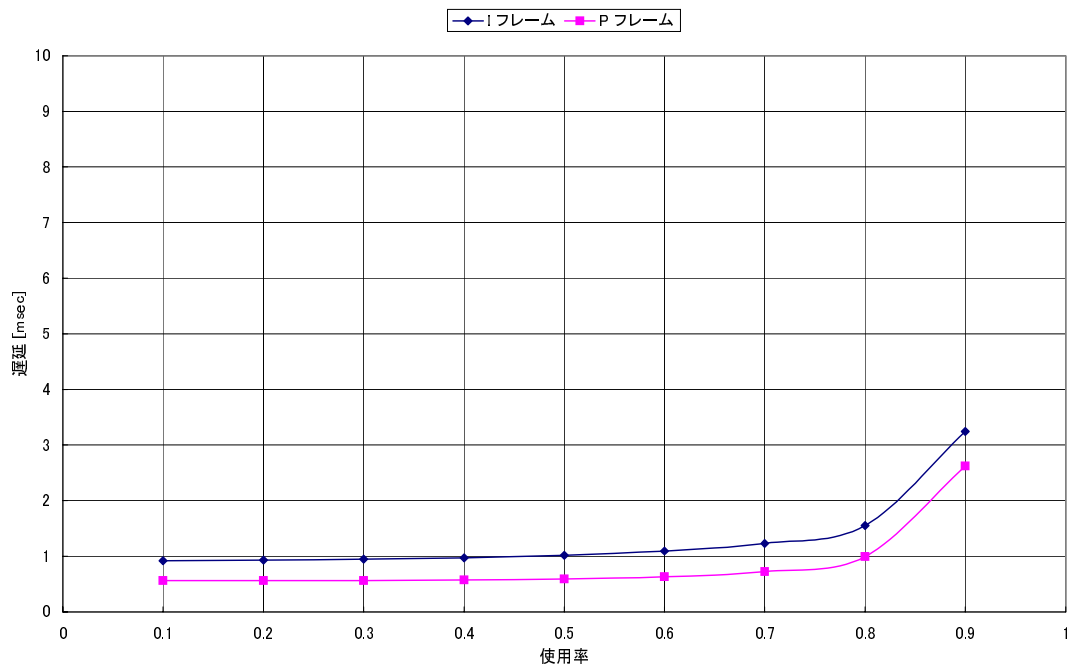


図 5.1: フレームの遅延 (MVA 又は AVA のみ)

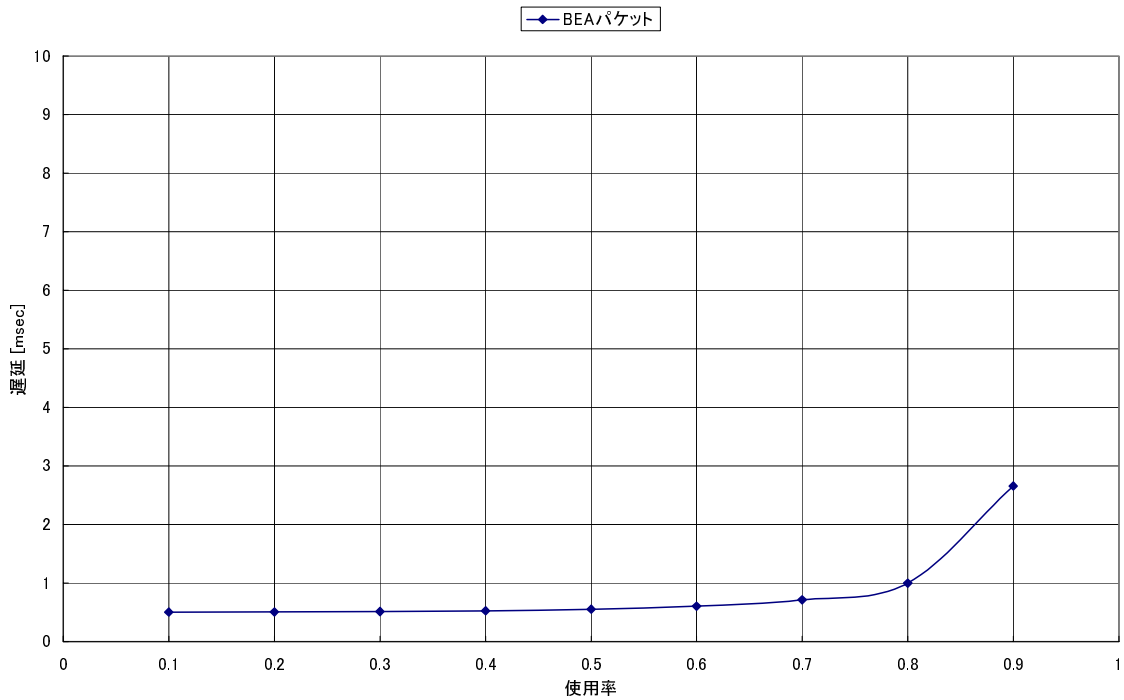


図 5.2: パケットの遅延 (BEA のみ)

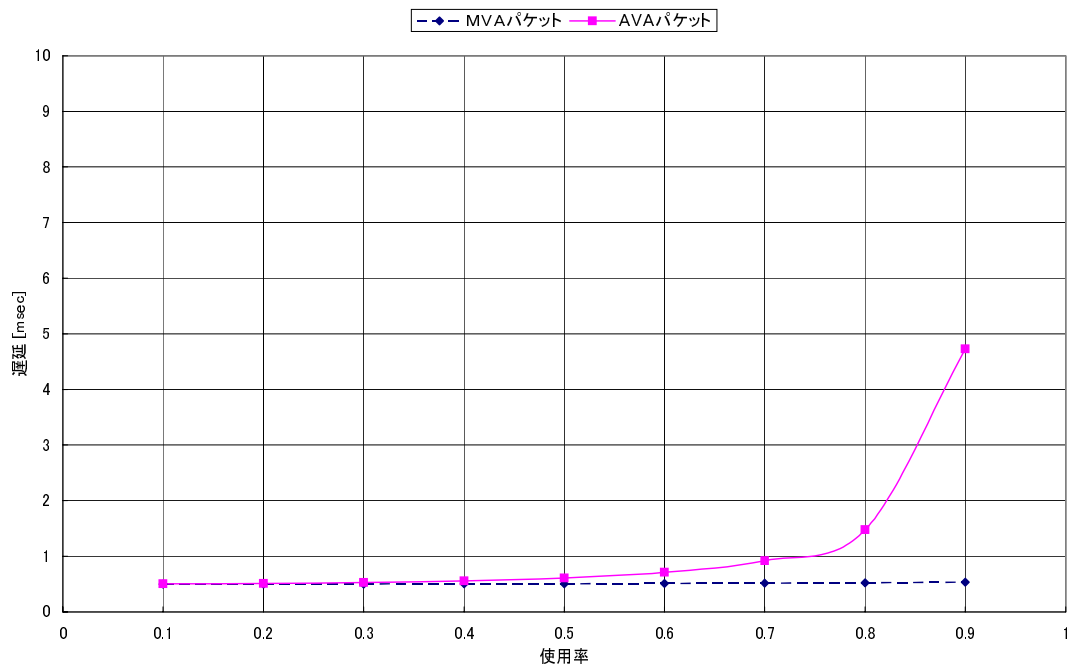


図 5.3: パケットの遅延の平均 (混合比 MVA:AVA = 1:1)

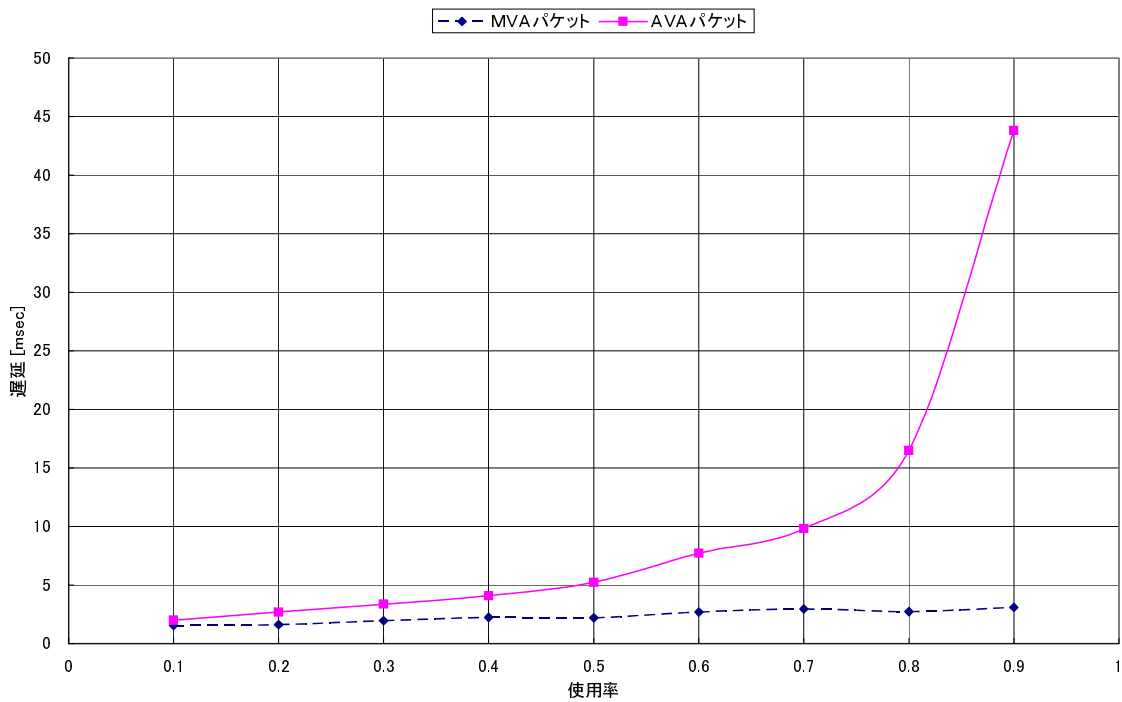


図 5.4: パケットの遅延の最大値 (混合比 MVA:AVA = 1:1)

において、AVA 保証をしたものの中に、許容範囲外の遅延があるパケットが存在する。これが画像フレームに与える影響についてみてみることにする。

5.2.2 フレーム単位での遅延

MVA トラヒックと AVA トラヒックを 1:1 で混合した場合のフレームの遅延の平均、及び最大値を図 5.2.2、図 5.2.2 に示す。まず、MVA トラヒックの特性は、全体的にほぼ 1

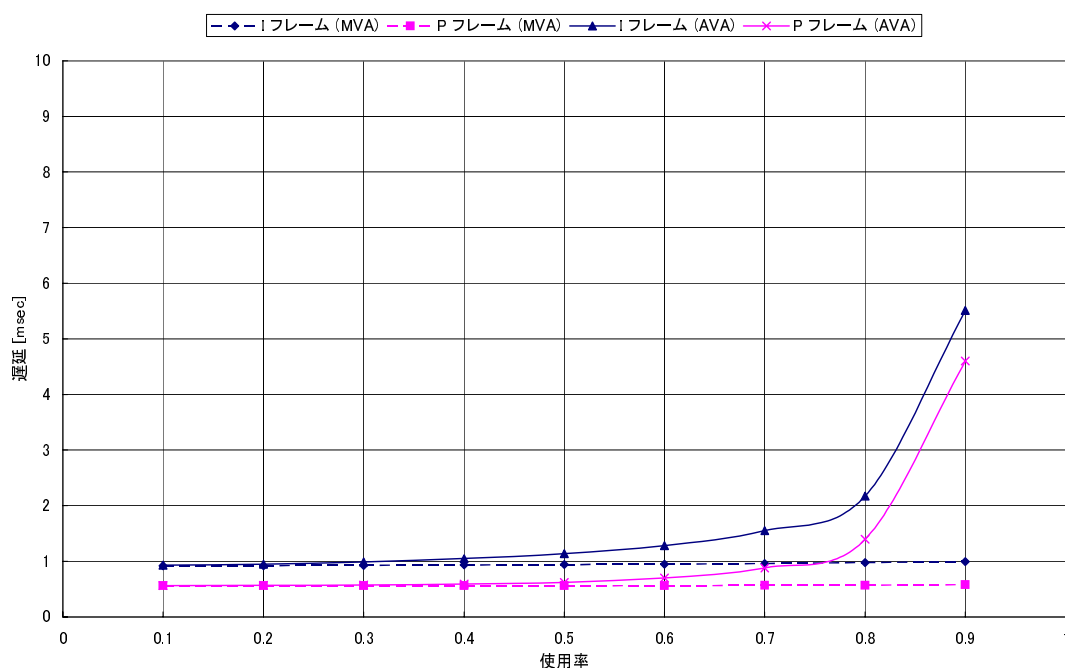


図 5.5: フレームの遅延の平均値 (混合比 MVA:AVA = 1:1)

転送フレームの遅れにとどまっている。これは品質を保証できていると評価できる。

AVA トラヒックについては、平均で見ると、使用率 0.9 においても 5~6 転送フレームの遅れで転送ができています。しかし、遅延の最大値をみると、使用率 0.8 程度までは許容範囲内に入っているが、それ以上の使用率になると、許容範囲をオーバーしてしまう画像フレームがでてくる。図 5.2.2 と図 5.2.2 に、使用率 0.9 におけるフレームの遅延の分布を示す。MVA 保証された I フレームの遅延はおおむね 2 転送フレーム以内におさまっている。AVA 保証された I フレームについても、ほぼ 5~6 転送フレーム以内である。しかし、許容範囲内におさまらなかった画像フレームが存在する。50,000 フレーム中どれくらいの割合で許容範囲を超えてしまうのかを表 5.1 に示す。使用率 0.9 の時、AVA 保証した I フレームが 37 フレーム、P フレームが 25 フレーム許容範囲外に出てしまっている。I フレームの場合は約 45 秒に 1 フレーム、P フレームの場合は約 66 秒に 1 フレーム許容範囲を超えるフレームが存在することになる。

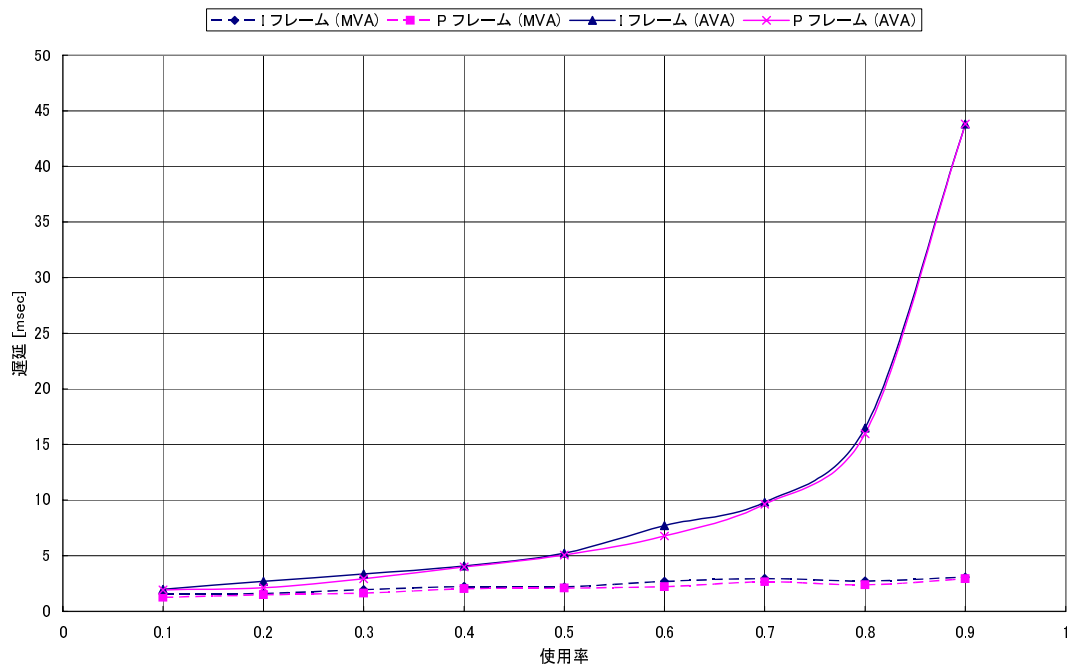


図 5.6: フレームの遅延の最大値 (混合比 MVA:AVA = 1:1)

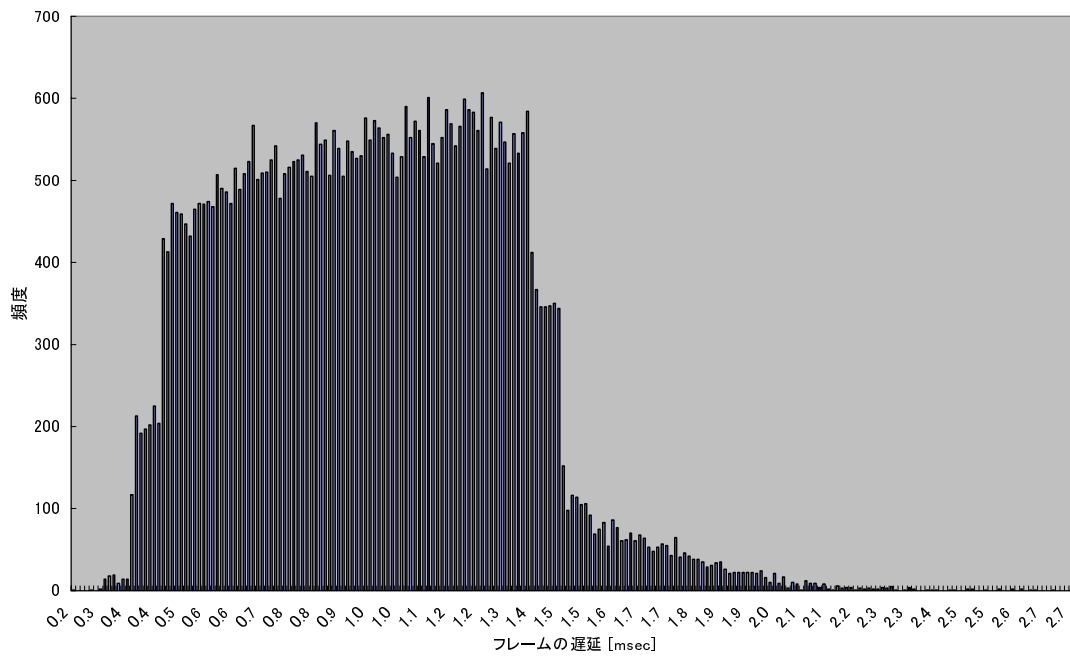


図 5.7: MVA 保証した I フレームの使用率 0.9 における遅延の分布 (混合比 MVA:AVA = 1:1)

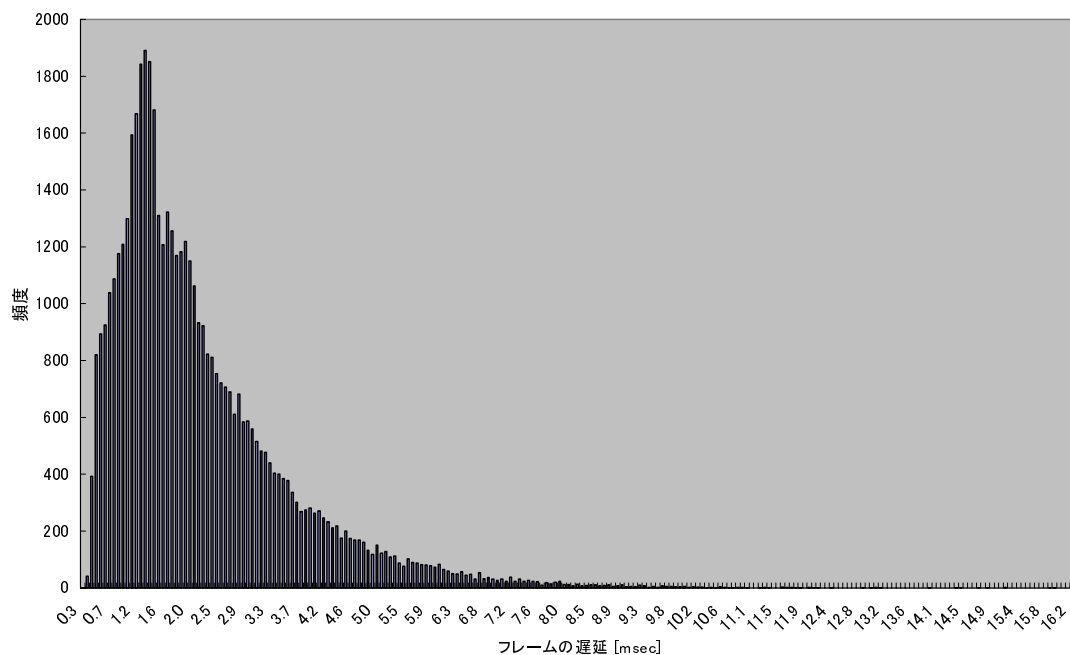


図 5.8: AVA 保証した I フレームの使用率 0.9 における遅延の分布 (混合比 MVA:AVA = 1:1)

表 5.1: 許容時間内に届かなかった画像フレームの数 - サンプル数 : 50,000 フレーム

使用率	MVA		AVA	
	I フレーム	P フレーム	I フレーム	P フレーム
0.1	0	0	0	0
0.2	0	0	0	0
0.3	0	0	0	0
0.4	0	0	0	0
0.5	0	0	0	0
0.6	0	0	0	0
0.7	0	0	0	0
0.8	0	0	0	0
0.9	0	0	37	25

5.3 混合比が異なる場合

ここでは、MVA トラヒックが AVA トラヒックに与える影響について検討する。前節では、MVA トラヒックと AVA トラヒックの混合比が等しい場合についてシミュレーションを行ったが、以下では MVA トラヒックが極端に多い場合、少ない場合のそれぞれについてシミュレーションを行う。

5.3.1 MVA トラヒックが多い場合

MVA トラヒックと AVA トラヒックの混合比が、8:2 の場合の画像フレームの遅延の平均と最大値を図 5.3.1、図 5.3.1 に示す。平均値で見れば、MVA 保証したものについては、

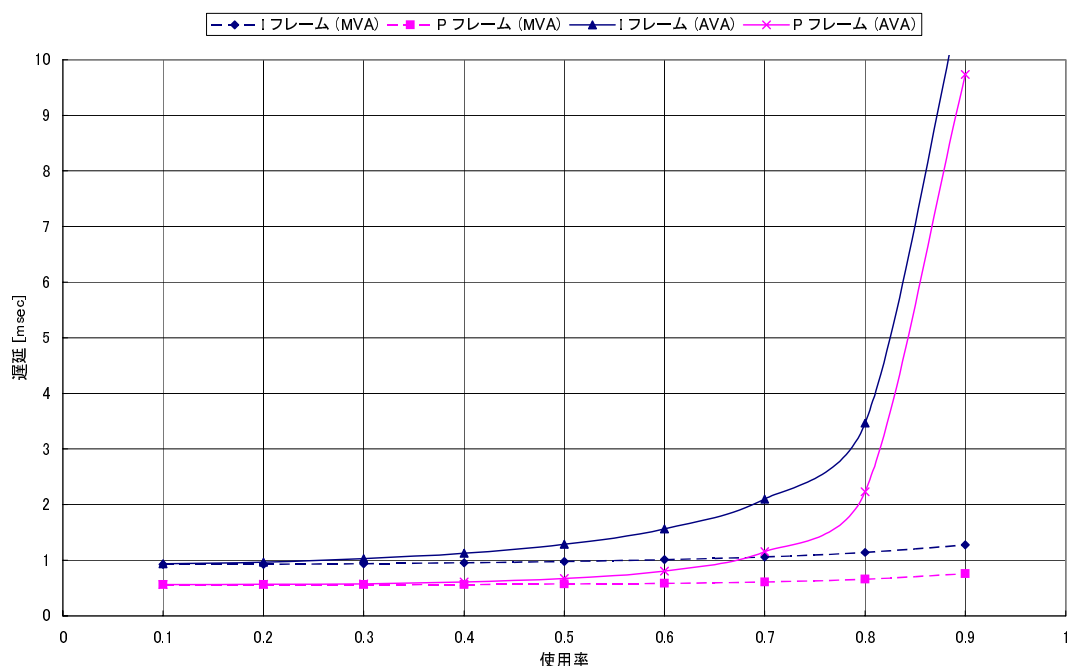


図 5.9: フレームの遅延の平均値 (混合比 MVA:AVA = 8:2)

1 転送フレーム程度の遅れで転送できており、AVA 保証したものに関しても、混合比 1:1 のときの遅延よりは大きくなっているものの、許容範囲内におさまっている。しかし、最大値で見ると、AVA トラヒックに関しては、使用率 0.78 あたりから、遅延が許容範囲内におさまらない画像フレームが出てくる。これは、MVA トラヒックのほうが AVA トラヒックよりも優先度が高いために、MVA トラヒックの量が増加すると、MVA 保証された画像フレームの packets をのせるために、AVA 保証された画像フレームが犠牲になってしまっていることを示している。使用率 0.9 の時の MVA・AVA トラヒックそれぞれの I フレームの遅延の分布を、図 5.3.1、図 5.3.1 に示す。MVA 保証したものに関しては、ほぼ

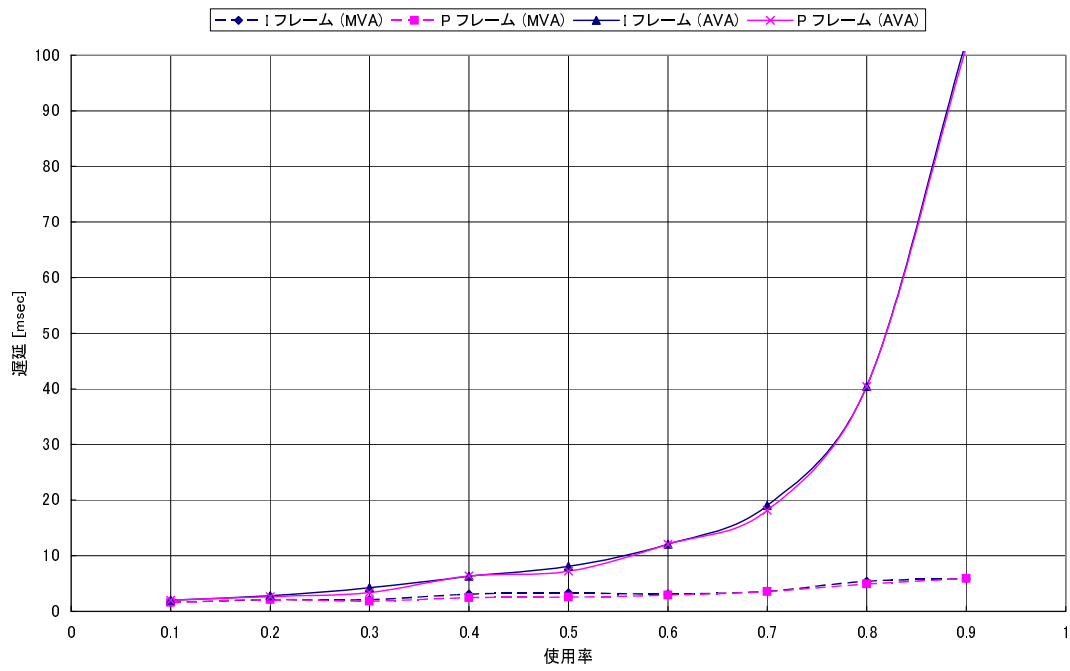


図 5.10: フレームの遅延の最大値 (混合比 MVA:AVA = 8:2)

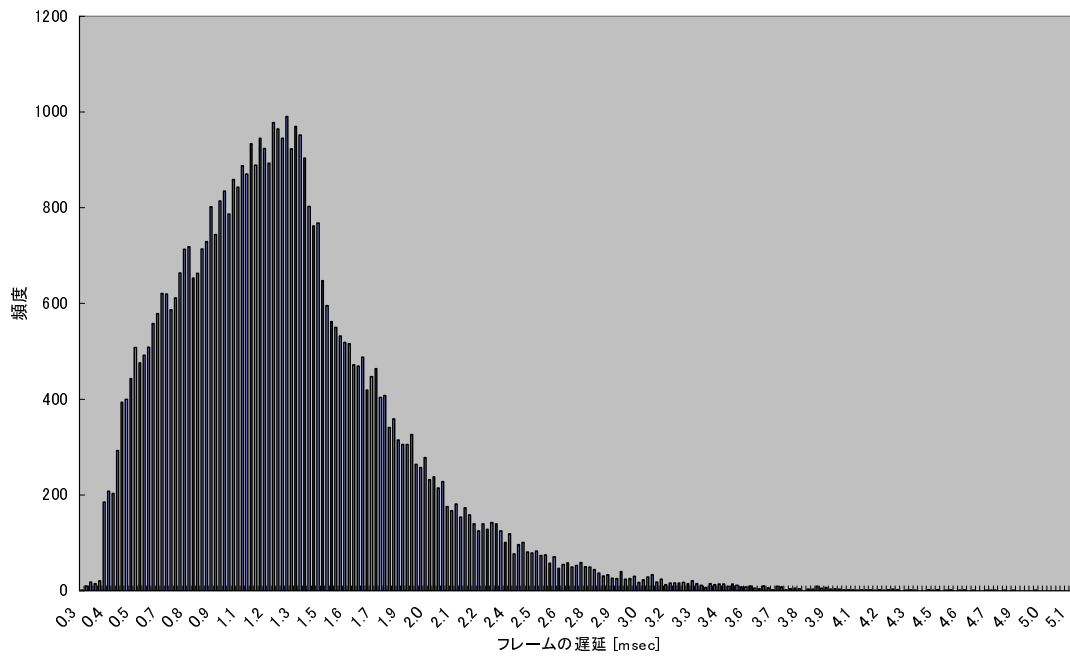


図 5.11: MVA 保証したIフレームの使用率 0.9 における遅延の分布 (混合比 MVA:AVA = 8:2)

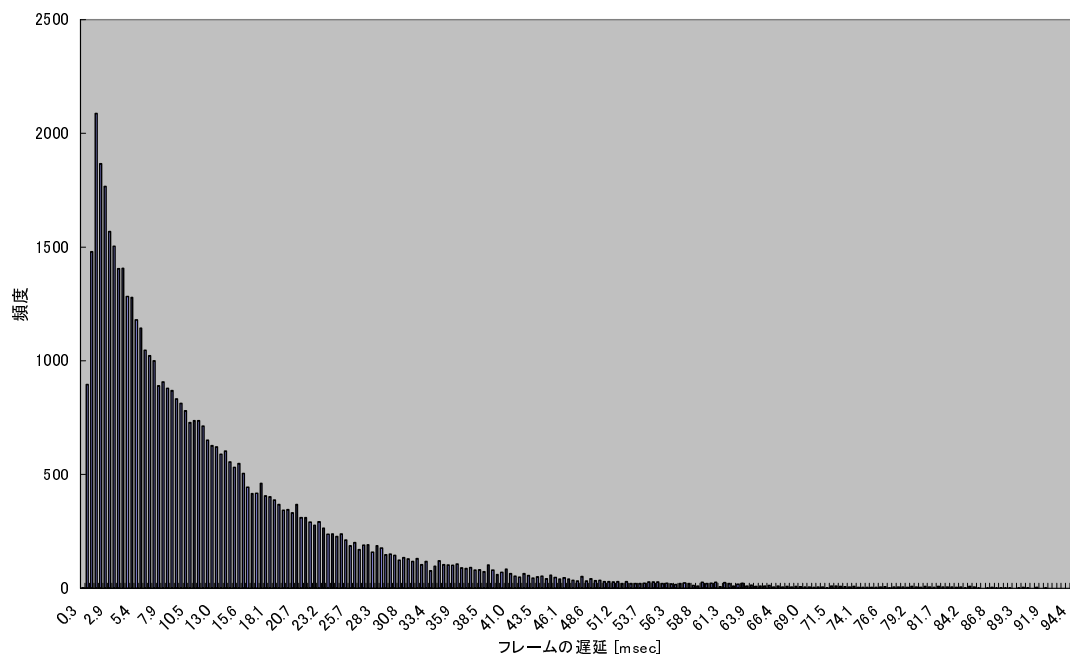


図 5.12: AVA 保証した I フレームの使用率 0.9 における遅延の分布 (混合比 MVA:AVA = 8:2)

3 転送フレーム以内の遅延にとどまっている。AVA 保証したものに関しては許容範囲内におさまってはいるものの、許容範囲外になってしまっているものも少なくない。どのくらいの割合で許容範囲を超えてしまう画像フレームがあるかを表 5.2 に示す。使用率 0.7 ま

表 5.2: 許容時間内に届かなかった画像フレームの数 - サンプル数：50,000 フレーム

使用率	MVA		AVA	
	I フレーム	P フレーム	I フレーム	P フレーム
0.1	0	0	0	0
0.2	0	0	0	0
0.3	0	0	0	0
0.4	0	0	0	0
0.5	0	0	0	0
0.6	0	0	0	0
0.7	0	0	0	0
0.8	0	0	1	0
0.9	0	0	2638	2480

では MVA・AVA 共に品質は保たれているといえる。使用率 0.8 では、AVA 保証した I フレームが許容範囲を超えてしまっているが、50,000 フレーム（約 27～28 分）に 1 回 1 フレーム欠落することを許容するならば問題のない範囲である。使用率 0.9 においては、全体のうちの約 5% のフレームが許容範囲を超えてしまっている。

5.3.2 MVA トラヒックが少ない場合

MVA トラヒックと AVA トラヒックの混合比が、2:8 の場合の画像フレームの遅延の平均と最大値を図 5.3.2、図 5.3.2 に示す。MVA トラヒックが少ないため、AVA トラヒックの帯域をあまり圧迫しないため、よい特性が得られている。遅延の最大値でも、使用率 0.9 において許容値を超えていない。MVA トラヒックが 2 割程度であれば、使用率が 0.9 になったとしても AVA トラヒックの品質を保つことができている。

5.3.3 ベストエフォート型トラヒックが混在する場合

先に行ったシミュレーションにおける MVA トラヒックと AVA トラヒックの比率を保ったまま、図 5.3.3 に示すように、流れる全トラヒックのうちの 1 割程度をベストエフォート型トラヒックにする。BEA トラヒックを混ぜることにより、AVA トラヒックの予約帯域を越えたものが重なった時等に、BEA の帯域を使用してこれを救済することができる。

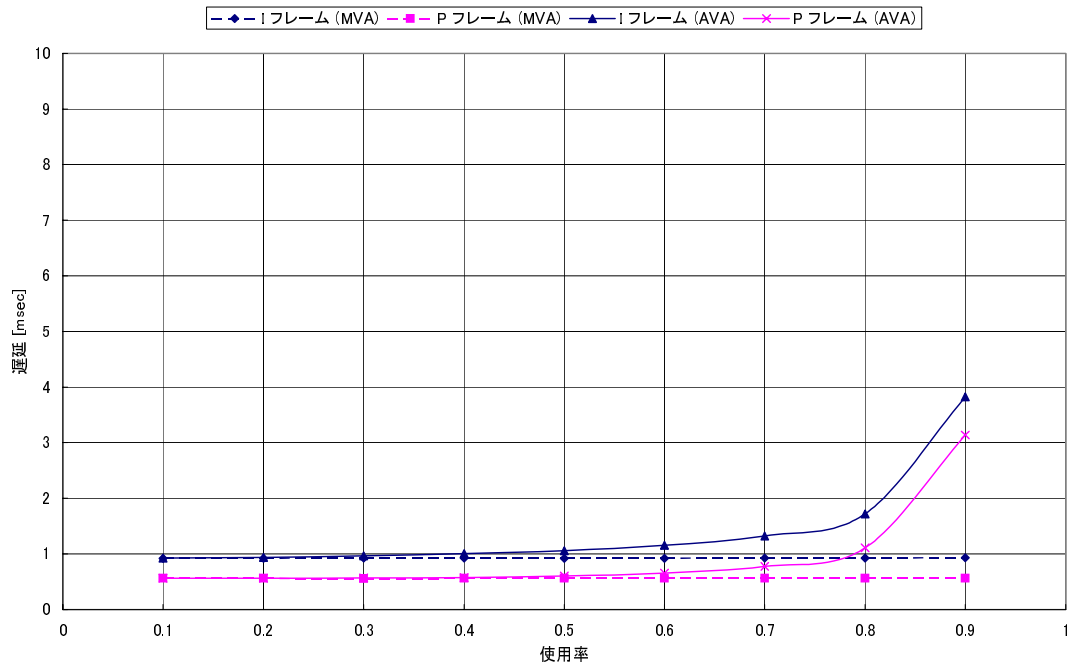


図 5.13: フレームの遅延の平均値 (混合比 MVA:AVA = 2:8)

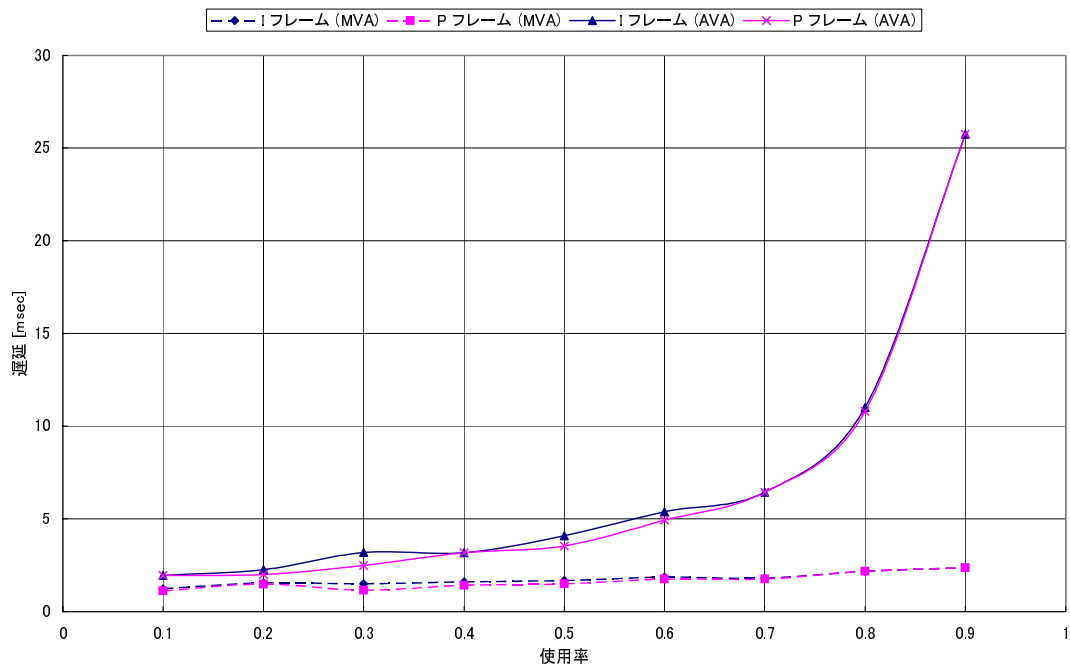


図 5.14: フレームの遅延の最大値 (混合比 MVA:AVA = 2:8)

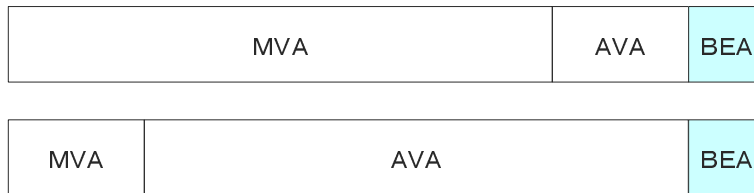


図 5.15: BEA パケット帯域の配置

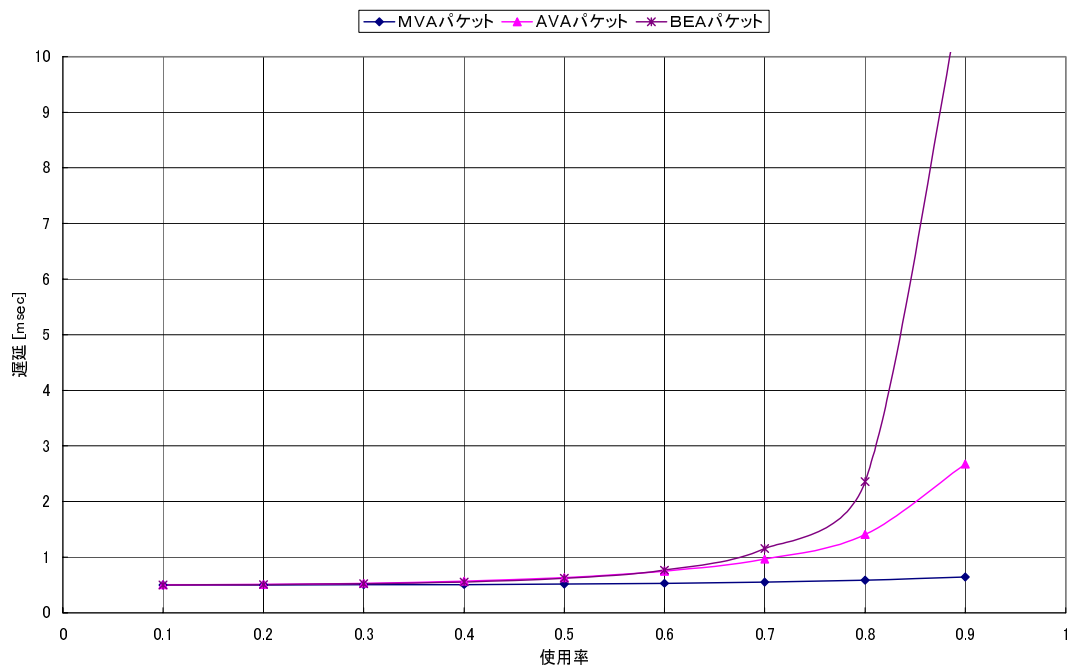


図 5.16: パケットの遅延の平均値 (混合比 MVA:AVA:BEA = 7.2:1.8:1)

図 5.3.3 に各サービスクラスのパケットの遅延の平均値を示す。品質を保証しなければならぬトラフィックが減少した分、先ほどのBEAトラフィックがなかった時よりはAVAトラフィックの品質が向上している。AVAトラフィックの品質は、MVAトラフィックの影響を受けて変化するが、BEAトラフィックの影響を受けず、品質が保たれているといえる。そして、フレームの遅延も、図 5.3.3 に示されるように少ない値となっている。そして、使用率 0.9 でのAVAトラフィックの遅延は大幅に改善されており、BEAトラフィックを混ぜることによって、高い使用率でもAVAトラフィックの品質を保つことができる。

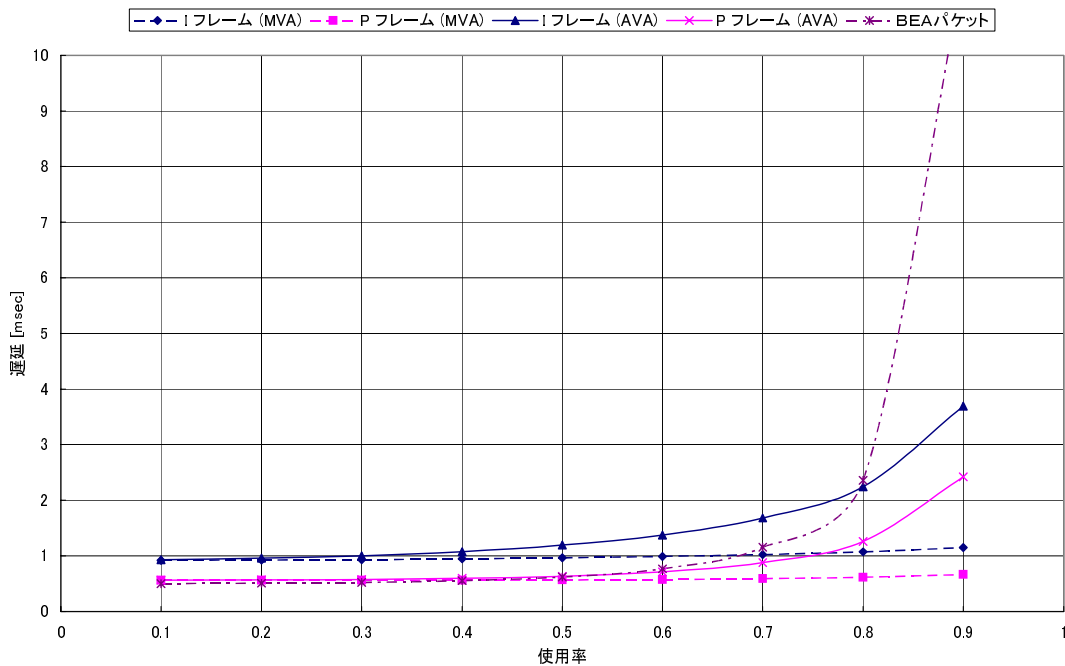


図 5.17: 画像フレームとパケットの遅延の平均値 (混合比 MVA:AVA:BEA = 7.2:1.8:1)

MVAトラフィックとAVAトラフィックの混合比率が逆の場合のパケット、フレームの遅延の平均値を図 5.3.3、図 5.3.3 に示す。BEAトラフィックは、MVA・AVAトラフィックの品質を保証するため、先ほどの場合と同様に使用率が0.9程度になるとかなり犠牲になる。しかし、MVAトラフィックの遅延は、使用率0.1~0.9にわたってほぼ一定であり、1転送フレーム以内に転送が完了している。AVAトラフィックについても、使用率0.9において、2転送フレーム以内に転送が完了しており、高い品質を保っていることが分かる。

ここで、MVAトラフィックの混合比率が高い場合と低い場合のBEAトラフィックのパケットの遅延の最大値を比較してみる。平均値をみればどちらもそれほど特性が変わらないのだが、最大値をみると、図 5.3.3 に示されるように、使用率が高い状態においては、MVAトラフィックの混合比率が高い場合の方が、遅延の最大値が小さくなっている。これは、MVAトラフィックの占める割合が少なく、残りをAVAトラフィックで埋めてしまっている場合よりも、MVAトラフィックの占める割合が多く、残りをAVAトラフィックで埋めきれない場合

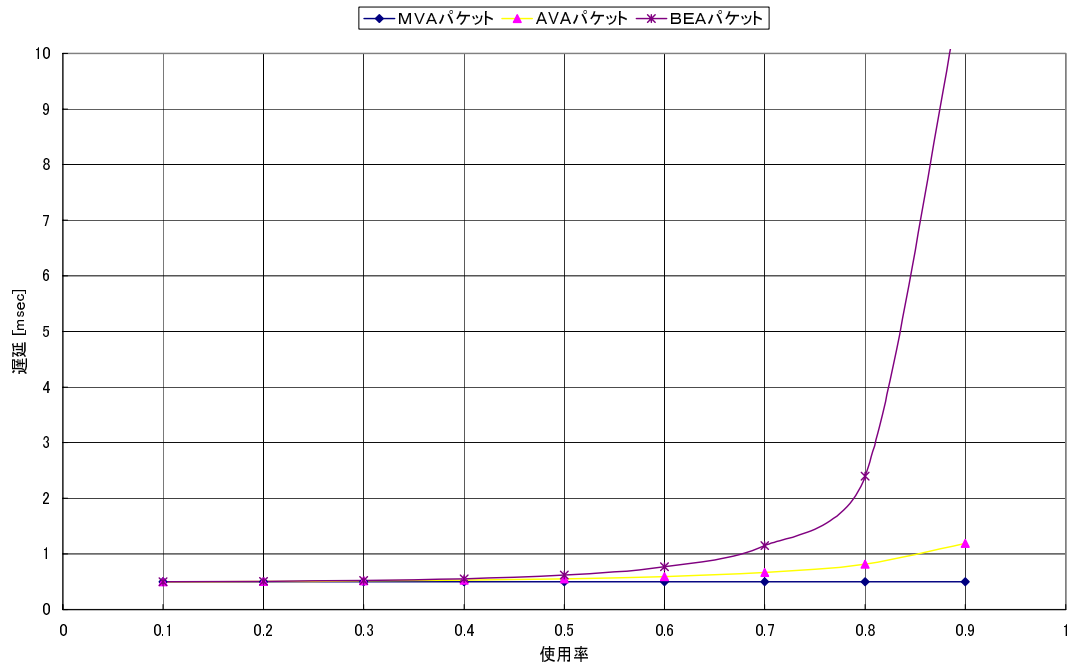


図 5.18: パケットの遅延の平均値 (混合比 MVA:AVA:BEA = 1.8:7.2:1)

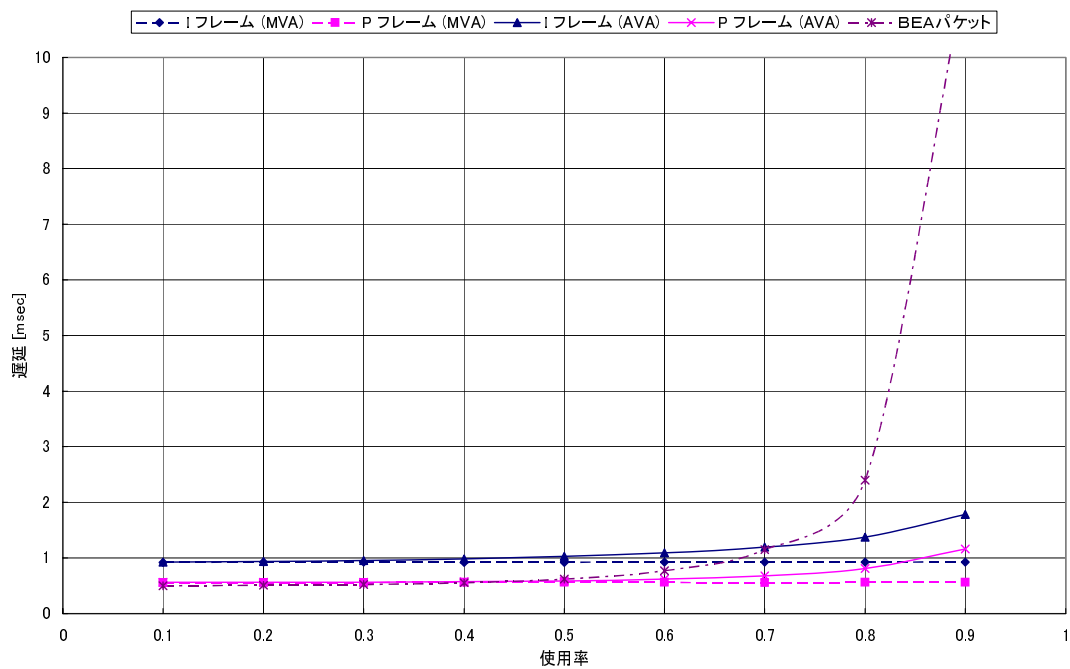


図 5.19: フレームとパケットの遅延の平均値 (混合比 MVA:AVA:BEA = 1.8:7.2:1)

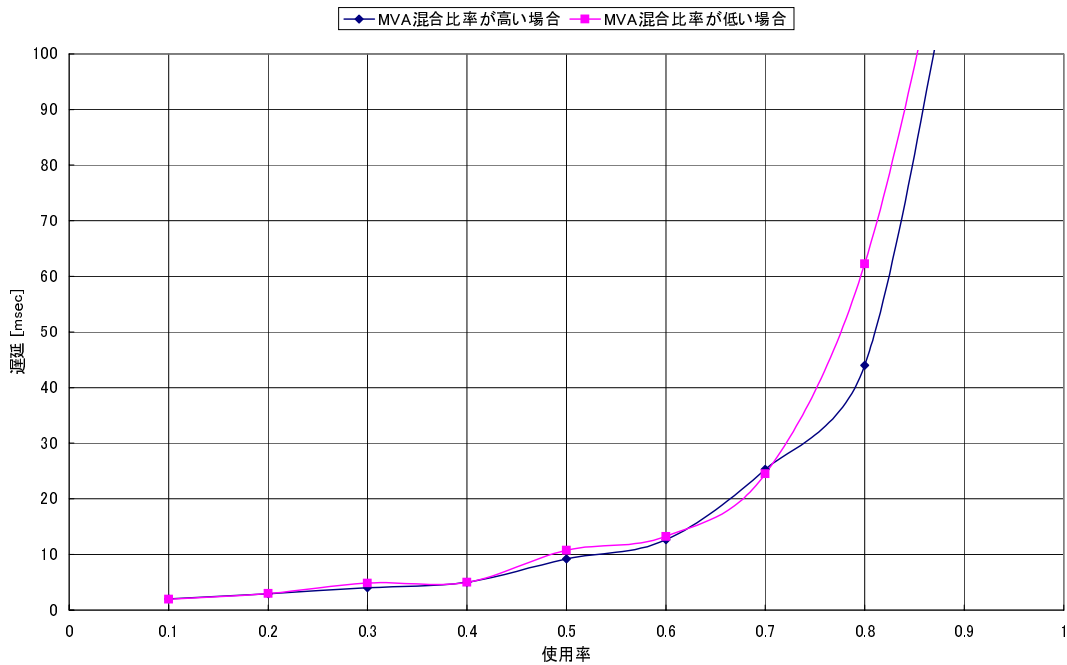


図 5.20: MVA トラフィックの混合比率の違いによるパケットの遅延の最大値

の方が、BEA トラフィックは入り込みやすいということを示している。

5.4 優先制御のみを行った場合との比較

第3章で述べたタイムスロットのスケジューリングポリシーでは、転送フレーム送出前であれば、既にタイムスロットに書き込まれた AVA パケットをキューに戻し、MVA パケットを書き込むことをしているが、このことについて検討を行う。この機能を止めて、優先スケジューリングのみを行い、いったん書き込まれたタイムスロットはそのままにしておいた場合の、パケットと画像フレームの遅延の平均を図 5.4、図 5.4 に示す。なお、MVA と AVA の混合比は 1:1 としてある。これを図 5.2.1、図 5.2.2 と比較してみる。図 5.4 に、プリエンブションを行った場合と、行わなかった場合の MVA・AVA パケットの遅延のを示す。また、併せて示してある数字は、どれくらいプリエンブションが行われたかのを示す。使用率が高くなるにつれて、プリエンブションが行われる確率が高くなっているが、特に使用率 0.9 においては、半数以上のパケットがプリエンブションの対象となっている。プリエンブションを行うと、MVA パケットの遅延は減少し、AVA パケットの遅延が増加している。AVA パケットをある程度犠牲にすることにはなるが、その結果、MVA パケットの遅延はほぼ一定となり、遅延の揺らぎが少ない、高い品質を保證できるようになる。

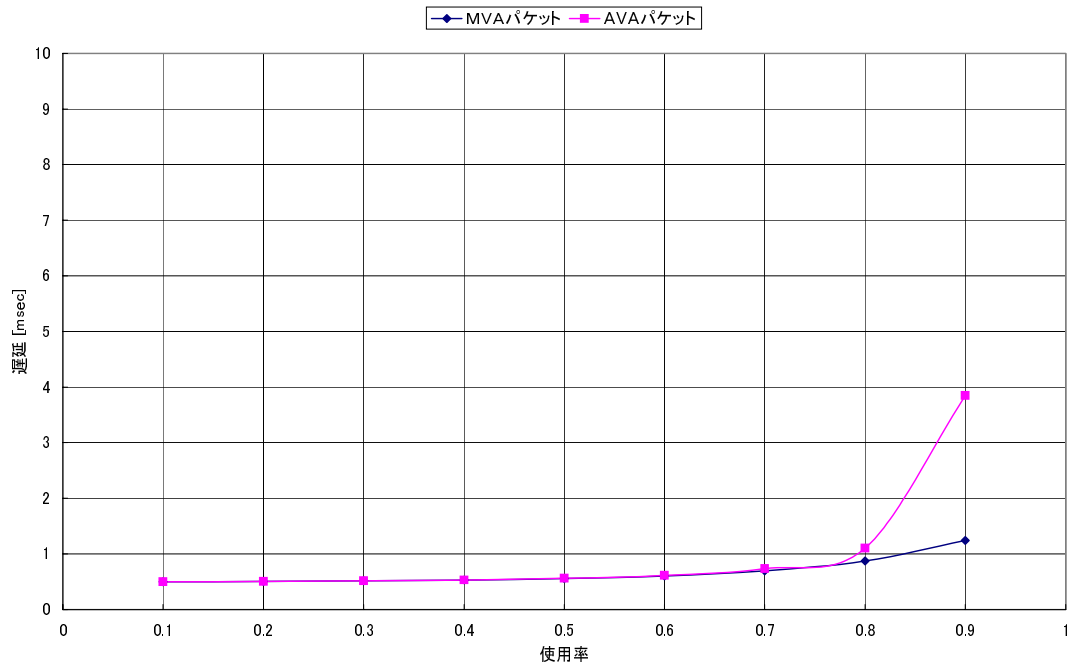


図 5.21: 優先制御のみを行った場合のパケットの遅延の平均 (混合比 MVA:AVA = 1:1)

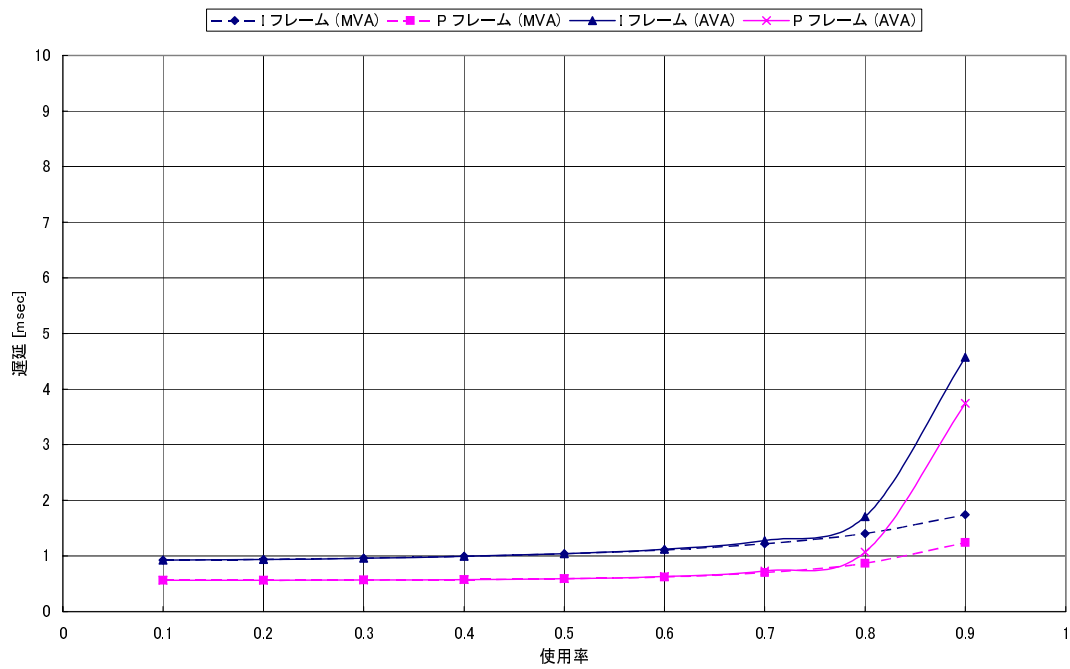


図 5.22: 優先制御のみを行った場合のフレームの遅延の平均 (混合比 MVA:AVA = 1:1)

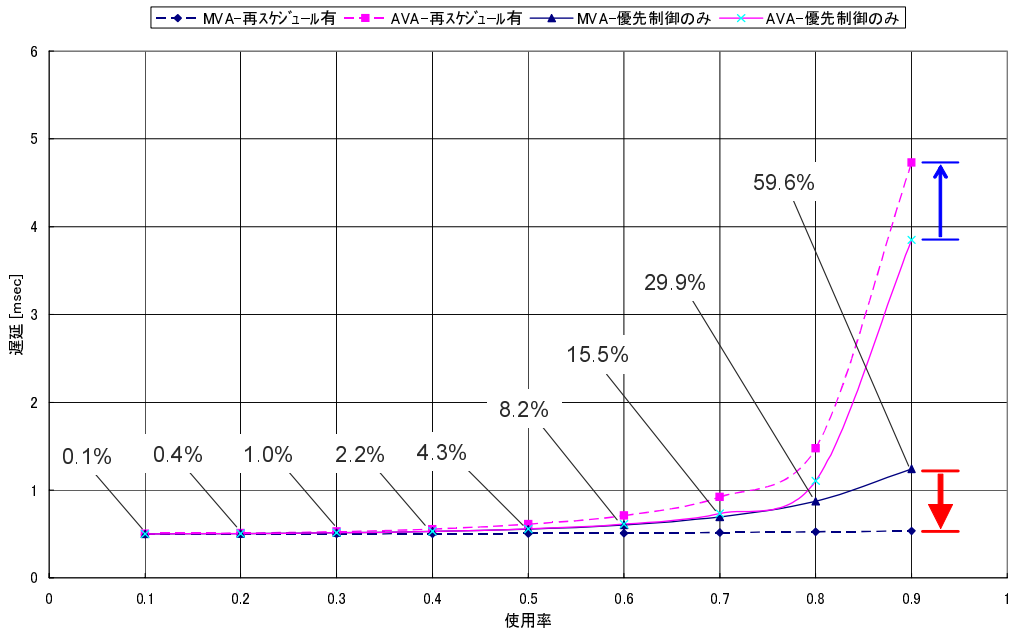


図 5.23: プリエンプションを行った場合と行わなかった場合の比較

第6章 結論

6.1 要旨

本論文では、同期通信に基づき、ラベルスイッチングの技術を併用する、ラベル付き同期転送方式を提案した。そしてサービスクラスを MVA・AVA・BEA 3 つに分けて優先制御制御を行い、さらにフレーム転送を行っている点を利用した再スケジューリングを行うことにより QoS を保証する機構を示した。本提案方式による通信網に流すトラヒックとして、リアルタイム型トラヒック、ベストエフォート型トラヒックのモデルを示し、シミュレーション実験を行い、性能の評価を行い、以下のような結果を得た。

- MVA 又は AVA トラヒックを単体で流した場合、使用率 0.9 においても平均 3 転送フレーム程度の遅れで転送ができる。
- BEA トラヒックを単体で流した場合の遅延は、FIFO キューで非同期転送を行う場合に比べ、フレーム転送を行っているために使用率が低い場合でも一定の遅延が発生してしまうが、遅延時間はほぼ一定で、遅延のゆらぎを少なくすることができる。
- MVA・AVA トラヒックを等しい割合で混合した場合、平均で見れば使用率 0.9 においても MVA トラヒックの遅延は 1 転送フレーム以内であり、AVA トラヒックの遅延も 4~5 転送フレーム程度である。しかし、MVA トラヒックの画像フレームは、I フレームが約 45 秒に 1 フレーム、P フレームが約 66 秒に 1 フレームの割合で遅延の許容範囲を超えてしまうものが出る。
- MVA・AVA トラヒックを異なった割合で混合した場合、MVA トラヒックは AVA トラヒックにあまり影響されることなくの遅延はほぼ一定で高い品質が保たれているが、AVA トラヒックの遅延は MVA トラヒックの割合によって大きく左右される。MVA トラヒックが極端に多い場合、MVA トラヒックと AVA トラヒックの比が 8 : 2 であった場合、AVA トラヒックの画像フレームは、使用率 0.9 においては、全体の約 5% のフレームの遅延が許容範囲を超えてしまう。使用率が 0.8 の時も、約 28 分に 1 フレームの割合で遅延が今日範囲を超えてしまう。しかし、サービスクラスが AVA であることを考えると、この程度の遅延であれば問題のない範囲である。逆に、MVA トラヒックが極端に少ない場合、MVA・AVA トラヒックの比が 2:8 であった場合は、遅延の最大値でも、使用率 0.9 において AVA トラヒックの品質が保たれている。上記の両比率を保ったまま、そこにベストエフォート型トラヒックを混ぜ

た場合、AVAトラヒックの品質への悪影響はなかった。AVAトラヒックは、MVAトラヒックの影響を受けるが、BEAトラヒックの影響は受けない。また、BEAトラヒックは、MVAトラヒックの割合が高い場合と低い場合において、平均の遅延はあまり差がみられなかったが、遅延の最大値でみると、使用率が高いところでは、MVAトラヒックの割合が高い時のほうが、遅延の最大値が小さくなっていた。MVAトラヒックの割合が少なく、残りをAVAトラヒックで埋めてしまうよりも、MVAトラヒックの割合が多く、残りをAVAトラヒックで埋めきれないところにBEAトラヒックは入り込みやすいことを示している。そして、このBEAトラヒックの帯域を犠牲にすることによって、MVA・AVAトラヒックの品質を改善することができた。

- MVAトラヒックとAVAトラヒックを等しい割合で混合した状態において、タイムスロットの再スケジューリング機構を止め、優先制御のみを行った場合、再スケジューリングを行った場合に比べ、MVAトラヒックの遅延は増加し、AVAトラヒックの遅延は減少した。AVAの品質を少々犠牲にすることにはなるが、MVAトラヒックの品質をかなり良い状態に保つことが可能となる。

6.2 結び

本提案方式は、キューイングを行い、優先制御及びプリエンプションを行う方式であるため、完全なQoS保証を行うことはできない。しかし、シミュレーションにおいて、MVAトラヒックは全て十分な品質の保証ができておりといえる。これは、フレーム転送を行い、プリエンプションを行った結果であると思われる。また、AVAトラヒックについても、MVAトラヒックの全体に占める割合にもよるが、接続制御の段階で、AVAトラヒックを受け付けすぎなければ、品質を保証することができる。

そして、BEAトラヒックを混ぜることによって、AVAトラヒックのピークが重なったときなど、AVAトラヒックが予約している帯域をオーバーするときは、このBEA帯域を使用して転送を行うことにより、高い使用率においても、品質を保証できる確率が高くなる。

謝辞

本研究を行うにあたり、始終熱心な御指導を賜りました日比野靖教授に感謝致します。また、適切な御助言を賜りました田中清史助教授、丹康雄助教授、菅原英子助手に感謝致します。そして、その他にも、貴重なご意見、御討論を頂きました、日比野研究室・田中研究室の皆様感謝致します。