

Diffserv における平均滞在時間を用いた バッファ管理方式の検討

前田 義人[†] 保岡 裕[†] 甲藤 二郎[†] 大久保 榮^{††}

[†]早稲田大学大学院理工学研究科 〒169-8555 東京都新宿区大久保 3-4-1

E-mail: [†] {yoshi,yasuoka,katto}@katto.comm.waseda.ac.jp

^{††}〒169-0051 東京都新宿区西早稲田 1-3-10

E-mail: ^{††} sokubo@waseda.jp

あらまし Diffserv ルータでは、IP ヘッダのクラス情報に基づくスケジューリングを行い、サービスの階層化を実現している。しかし、既存のバッファ管理アルゴリズムでは、VoIP やインターネット放送のような遅延特性を重視するアプリケーションの要求を十分に満足しているとは言い難く、適切なサービスを提供できていない。そこで本稿では、実時間性の要求されるクラスのキュー内平均滞在時間を減少させるため、キュー内の平均滞在時間を閾値として用い、遅延ベースでコントロール可能なバッファ管理方式を提案し、シミュレーションによって提案方式の有効性を明らかにする。

キーワード Diffserv, バッファ管理, 遅延要求

Buffer Management with Delay Constraints for Differentiated Services

Yoshihito MAEDA[†] Yutaka YASUOKA[†] Jiro KATTO[†] Sakae OKUBO^{††}

[†] Graduate School of Science and Engineering, Waseda University

3-4-1 Okubo, Shinjuku-ku, Tokyo, 169-8555 Japan.

E-mail: [†] {yoshi,yasuoka,katto}@katto.comm.waseda.ac.jp

^{††} 1-3-10 Nishi-Waseda, Shinjuku-ku, Tokyo, 169-0051 Japan.

E-mail: ^{††} sokubo@waseda.jp

Abstract Diffserv routers carry out packet classification and scheduling according to IP header information and provide service differentiation. However, current algorithms do not sufficiently satisfy delay requirements for realtime oriented applications such as VoIP and Internet broadcasting. In this paper, we therefore propose a novel buffer management algorithm that is explicitly controllable with delay parameters to decrease packet queuing time of specific Diffserv classes. Computer simulations are carried out and validity of our proposal is quantitatively clarified.

Keyword Diffserv, Buffer Management, Delay Requirements

1. はじめに

近年、VoIP や映像配信などリアルタイム、準リアルタイム IP アプリケーションが企業網や地域網で数多く使用されるようになってきた。しかし、現行の様々なトラヒックが混在するベストエフォート網においては、遅延やジッタ、パケット廃棄などによる音声や映像の品質劣化が生じる可能性がある。そのような背景の下、インターネット上を流れる様々なデータの種類に応じたサービスを提供するために、ネットワーク QoS(Quality of Service)を提供するための研究開発が盛んに進められている。その一例として、IETF(Internet Engineering Task Force)において、IP 層で QoS を保証するための技術として Diffserv(Differentiated Services)[1] 技術が標準化されている。

Diffserv は、フローが流入するエッジルータにおいて、EF (Expedited Forwarding)[2]、AF (Assured Forwarding)[3]、BE (Best Effort) と呼ばれるサービスクラスのいずれかにフローを集約し、クラス毎の優先制御を行なうことでスケーラビリティを確保しつつ、QoS をクラス単位で提供するサービスである。これにより、バックボーンなどのフローが集中するところでも、あらかじめ決められた QoS を提供することが可能になる。また、AF クラスには 4 つのクラス(AF1 ~ AF4) が定義されており、さらにはクラス毎に 3 段階の廃棄優先度 (Green、Yellow、Red) を設定することができる (trTCM: two rate Three Color Marker)[4]。これにより、主に Class-based の相対的な優先制御と最低帯域保証を実現することが可能となる。

しかし、Diffserv で用いられる従来のバッファ管理アルゴリズムは、バッファリングを許容したクラス毎の最低帯域の保証とパケット廃棄率の低減に主眼が置かれ、とりわけ輻輳の発生時に、必ずしも各アプリケーションが必要とする遅延要求を満足する設計とはなっていない。

そこで、本稿では、AFクラスの利用を前提に、キュー内平均滞在時間を考慮したバッファ管理アルゴリズムを提案し、クラス（アプリケーション）ごとの遅延要求を満足する Diffserv ルータの構成方式について検討を行う。さらに、コンピュータシミュレーションを実行し、その結果についての考察を行う。

2. Diffserv における優先制御

Diffserv での優先制御(図 1)において、クラス間の転送順位に関するスケジューリング方式として WRR(Weighted Round Robin)[5]、WFQ(Weighted Fair Queuing)[6]等が一般的に用いられている。また、キュー内の滞在時間を用いたスケジューリング方式として WTP(Waiting Time Priority)[7]という方式も提案されている。一方、廃棄制御を行うバッファ管理方式としては、Drop Tail、RED(Random Early Detection)[8]等が知られている。またクラシファイヤの詳細を図 2 に示す。

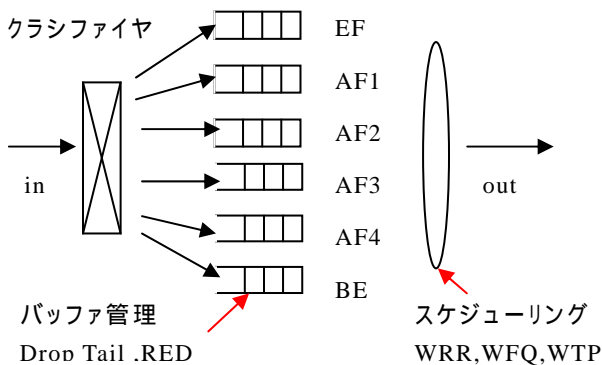
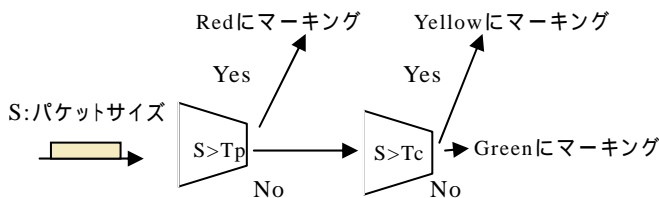


図 1 Diffserv における優先制御



- PIR: ピークデータレート[bps]
- PBS: PIRに関するバーストサイズ[byte]
- Tp: トークンパケットのサイズ[byte]
- CIR: 保証レート[bps]
- CBS: CIRに関するバーストサイズ[byte]
- Tc: トークンパケットのサイズ[byte]

図 2 クラシファイヤ部の詳細(trTCM)

従来方式の問題点として、バッファ管理方式として Drop Tail や RED を用いたとき、主に輻輳の発生時にキュー内に溜まるパケット数が急激に増加し、サービスを受けるまでの時間が長くなり、特定のクラスの遅延品質が大幅に劣化してしまうことが挙げられる。また、この結果は、後続するスケジューリング方式と関連して、他のクラスの遅延品質の劣化を引き起こすこともある。

そこで本稿では、遅延要求の厳しい特定のクラスのキュー内平均滞在時間を減少させるため、キュー内の平均滞在時間を閾値として用い、遅延ベースで明確なコントロールが可能なバッファ管理方式を提案する。以下に、提案するバッファ管理アルゴリズムについての詳細を述べる。

3. 提案方式

3.1 バッファ管理アルゴリズム

Diffserv の実現例として、Green、Yellow、Red にマーキングされたパケット(図 2)に対して、それぞれ異なる RED パラメータを設定する方式が提案されている。一方、本稿では、RED における平均キュー長の最小閾値、最大閾値の代わりに、Green、Yellow、Red にマーキングされたパケットに対してキュー内の平均滞在時間の最小閾値、最大閾値、廃棄確率を設定し、廃棄を行なうバッファ管理方式を提案する。このように、キュー長の代わりに時間を閾値とすることで、キュー内の滞在時間を制御することが可能になると見込まれる。

また提案方式では、[9]における閾値設定の問題を解決するため、式 3.1 を基に RED における平均キュー長の閾値を平均滞在時間にマッピングしたものを閾値として扱う。ここで RED のパラメータの推奨値として [10]の値を参考にした。

$$\text{閾値} = \frac{\text{パケットサイズ}}{\text{リンク速度}} \times \text{REDパラメータ(推奨値)} \quad (\text{式 3.1})$$

提案方式では、パケットが到着するたびに、以下のアルゴリズムに従って、廃棄制御を実行する。

STEP1 各キューの平均滞在時間 avg_time を求める。

この avg_time はキューに入った時間からキューの先頭に来るまでの待ち時間を滞在時間とし、ローパスフィルタにより平均時間を見積もる。このフィルタの時定数はパラメータ w によって制御し、以下の式で表される。なお、current_time は最新のパケットが到着した時点で計測された滞在時間である。

$$\text{avg_time} = (1-w) * \text{avg_time} + w * \text{current_time}$$

STEP2 各クラスに設定した閾値 max_th 、 min_th を avg_time と比較する。このときあるクラスにおいて、以下の条件分岐に従ってパケットのエンキュー、廃棄を決定する。

```

if (avg_time < min_th)
    パケットをエンキューする。
else if (min_th < avg_time < max_th)
    クラスに設定された廃棄確率(p)に従いパケットを廃棄する。
else (avg_time > max_th)
    パケットを廃棄する。

```

上記のようにして、各キュー内のパケット平均滞在時間を計測し、ある閾値を越えたら、遅延とジッタを抑えるため、キュー内の平均滞在時間に基づいた廃棄を行う。この方式によりフローが集中したクラスに対してだけでなく、他のクラスに対しても遅延とジッタを抑えることが可能となる。

3.2 提案方式と RED の違い

式 3.1 からわかるように、キュー長における閾値と、滞在時間における閾値には明確な関係があり、キューが1つしかなく、さらにスケジューラが存在しない場合は、両者は等価であり、結果は変わらない。しかし、Diffserv では、図 3 のようにクラス毎に複数のキューを設け、さらにスケジューラと1つの出力キューを加えた制御を行うので、あるキューからパケットが出力キューにスケジューリングされるまで、ほかのキューのパケットはサービスを待たなければならないことになる。このことから、Diffserv の仕組みでは、キュー長と滞在時間の閾値は等価にならず、結果が変わると考えられる。

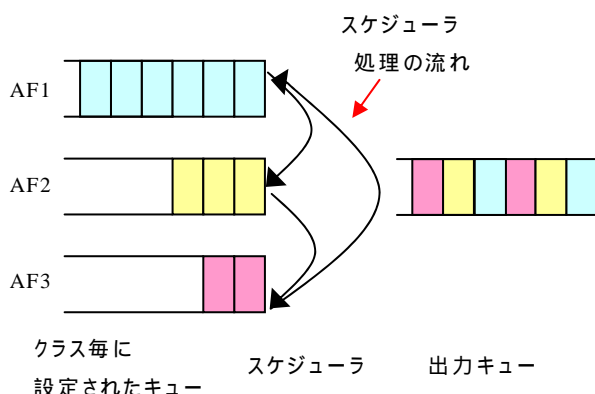


図 3 優先制御部分の詳細

上図のように、AF1、AF2、AF3 の各キューへと処理が移行しスケジューラが設定に従い出力キューへとパ

ケットを移動させていく。キューが多段に接続されているとき、AF2、AF3 クラスのパケットは、AF1 クラスのキューが処理されている時間、サービスを待つことになる。このため、キューにパケットが溜まっていなくても、滞在時間が増加するので、滞在時間をパラメータとして扱っている本提案方式では、廃棄制御が早く始まり、キューにパケットが溜まりにくくなる。この結果としてキュー内の滞在時間が減少し、遅延特性が改善される。

4. シミュレーション評価

本稿では ns-2(network simulator ver.2)[11]を用いたシミュレーションによる定量評価を行った。トラヒックのクラスとして Diffserv の DSCP(Diffserv Code Point)を利用し、最優先クラス(AF1)、中優先クラス(AF2)、低優先クラス(AF3)の3つのクラスを定義する。シミュレーショントポロジを図 4 に示す。

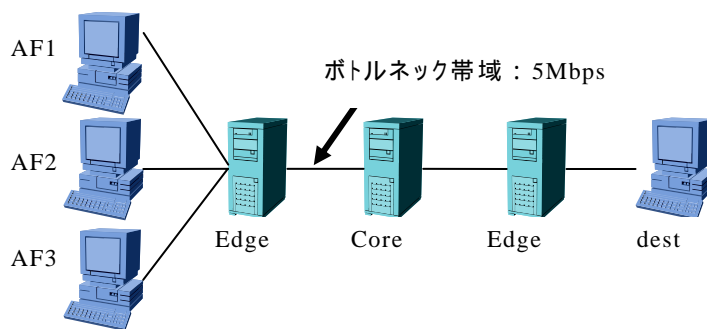


図 4 シミュレーショントポロジ

具体的なアプリケーション例として、AF1 では最も遅延を短くしなければならない VoIP、AF2 ではある程度の短い遅延を必要とするストリーミング (片方向インターネット放送)、AF3 では遅延を必要以上に考慮しなくてもよい FTP や HTTP を想定した。また、VoIP は UDP、ストリーミングは TCP フレンドリ、FTP、HTTP は TCP で転送するものとした。条件を表 1 に示す。ここでは、AF1 クラスに有効帯域以上の VoIP フローが流入している。このシミュレーション条件の下、バッファ管理の方式を全てのクラスにおいて RED を使ったものと、AF1 と AF2 クラスにおいては提案方式を、AF3 クラスにおいては RED を用いたシミュレーションを行った。ここで AF3 クラスにおいて提案方式を用いない理由であるが、AF3 クラスにおいて想定しているアプリケーションは遅延要求を考慮すべきものではないので、遅延を制御することを目的とする提案方式は使用せず、RED を用いた。また、予備実験の結果として、TCP フローに対して提案方式を使用すると、提案方式の特徴である早いタイミングのパケット廃棄により、

TCP の輻輳制御が敏感に働くことになり、結果としてスループットが劣化することを確認している。そこで、以上の条件で、各クラスのキュー内平均滞在時間とパケットロス率を測定した。

さらに、各クラスの End-to-End 遅延も測定した。

表 1 シミュレーション条件

	プロトコル	Rate [Mbps]	Size [Kbytes]	WRR [%]	有効帯域 [Mbps]
AF1	UDP	0.6	1	10	0.5
AF2	TCP Friendly	可変	1	40	2
AF3	TCP	可変	1	50	2.5

また、今回の提案において遅延を制御できるかを検討するために、提案方式におけるキュー内滞在時間の閾値、パケット廃棄率を変化させたシミュレーションを実行し、その際の 2 種類の設定値を表 2、3 に示す。

表 2 提案方式のパラメータの設定 1

	CP	min_th [msec]	max_th [msec]	廃棄率(p)
AF1	10(G)	16	32	0.01
	11(Y)	8	16	0.10
	12(R)	4	8	0.20
AF2	20(G)	16	32	0.01
	21(Y)	8	16	0.10
	22(R)	4	8	0.20

表 3 提案方式のパラメータの設定 2

	CP	min_th [msec]	max_th [msec]	廃棄率(p)
AF1	10(G)	16	32	0.01
	11(Y)	4	8	0.20
	12(R)	2	4	0.20
AF2	20(G)	16	32	0.01
	21(Y)	16	32	0.01
	22(R)	16	32	0.01

設定 1 は設定 2 を比較検討する際のデフォルトのパラメータであり、設定 2 は遅延要求の厳しい AF1 クラスにおいて、3 段階の廃棄確率をあらわす Green、Yellow、Red の Yellow、Red にマーキングされたパケットに対し、さらに厳しいパラメータを設定した。また、AF2 クラスにおいては、ある程度の遅延要求を満たせばよいということで、Yellow、Red に相当するパラメータを Green と同様のパラメータとし、キューに

パケットが溜まりやすくなり、ある程度の遅延を許容できるような設定を行った。

5. シミュレーション結果

5.1 平均滞在時間とパケット廃棄率

バッファ管理方式として全てのクラスにおいて RED を使ったものと、AF1 と AF2 クラスにおいては提案方式を使ったもののそれぞれについて、パケットのキュー内平均滞在時間を図 5、パケットロス率を図 6 に示す。

まず、キュー内の平均滞在時間に関しては、RED を用いると、キュー長閾値までパケットが溜まる分、滞在時間が増加していることがわかる。今回スケジューリング方式として WRR を用いているがこの方式では重みの設定により 1 ラウンドに約 16msec の時間がかかる(図 7 参照)。AF1 クラスにおいてキューにパケットが一つ溜まるとこの時間だけ滞在時間が増加することとなり、キューにパケットが溜まる個数×16msec 分だけ滞在時間が増加していく。この結果、今回の RED パラメータ条件下では、平均として約 200msec 程度まで滞在時間が延びてしまうのである。一方、提案方式を用いると、本方式の特徴である、早いタイミングのパケット廃棄により、キューにパケットがほとんど溜まらないので、滞在時間が抑えられていることがわかる。

また、全体のパケットロス率に関しては、AF1 クラスにおいては、提案方式の方がロスが増加しているが、これは有効帯域を越えた分のロスであり、問題は無いと考えられる。AF2 クラスにおいても若干提案方式においてロスが見られるが、RED との差は 1% 以下であり、滞在時間が大きく減少していることを考慮すると、提案方式において有効性があると考えられる。

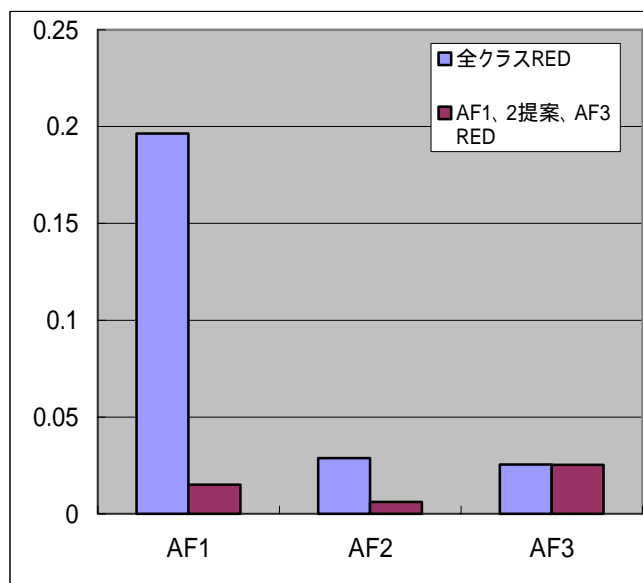


図 5 各クラスのキュー内平均滞在時間 [sec]

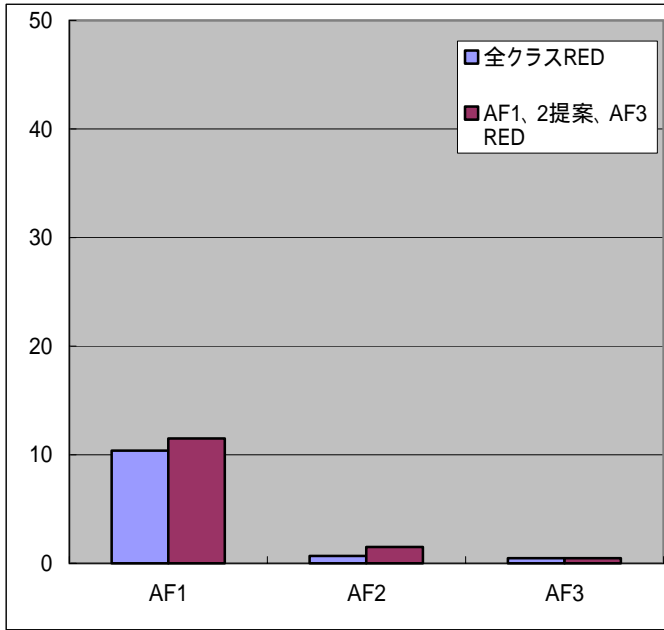


図 6 各クラスのペケットロス率[%]

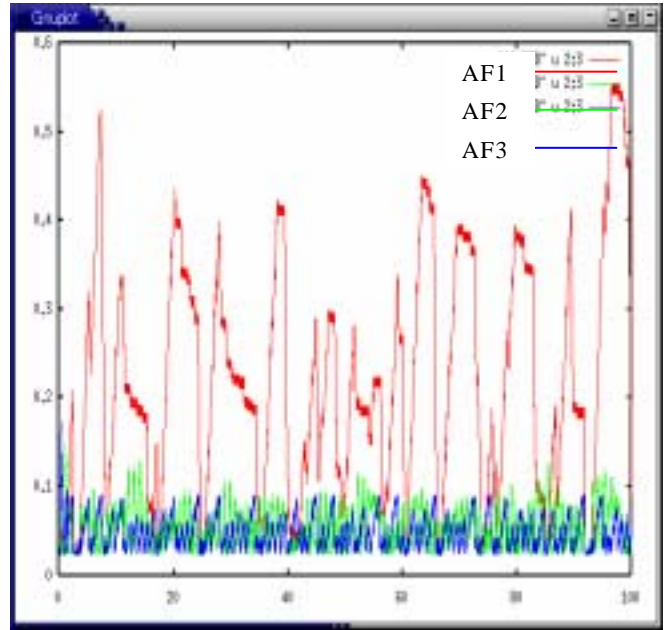


図 8 各クラスの End-to-End 遅延[sec] (ALL RED)

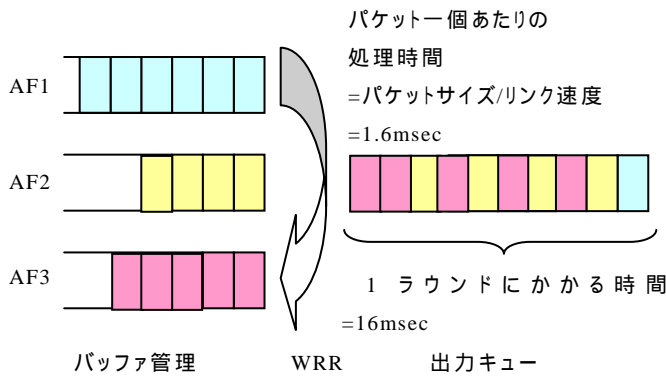


図 7 WRR スケジューリングによる遅延の増加

5.2 End-to-End 特性

バッファ管理方式を全てのクラスにおいて RED を使ったものと、AF1 と AF2 クラスにおいて提案方式を使ったものの End-to-End 遅延を図 8、図 9 に示す。

バッファ管理方式で、全てのクラスにおいて従来方式の RED を使った場合、エッジルータで輻輳が起きているため、キューイング遅延が大きく増加し、その影響により End-to-End 遅延も大きく増加していることがわかる。

一方、バッファ管理方式で AF1 と AF2 クラスでは提案方式、AF3 クラスでは RED を使った場合、End-to-End 遅延が小さくなり、またジッターも小さくなっていることがわかる。これは、エッジルータの制御によりキューイング遅延が減少しているためである。

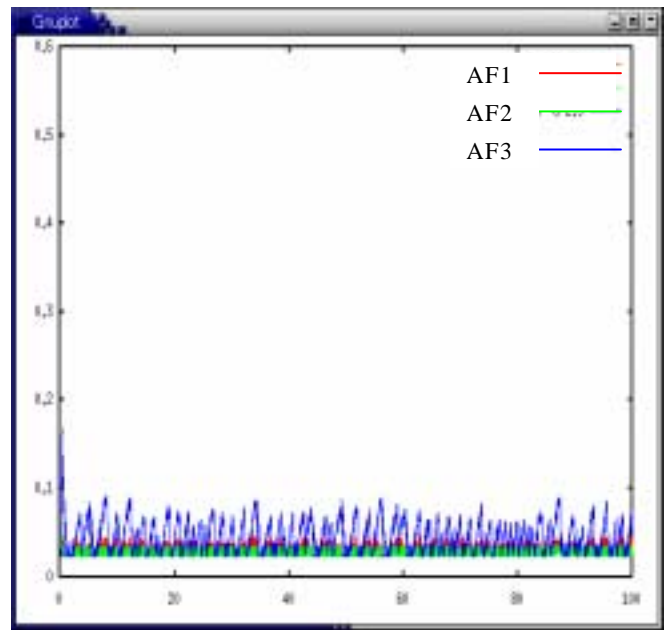


図 9 各クラスの End-to-End 遅延[sec] (提案 + RED)

5.3 提案方式の閾値の影響

AF1 と AF2 クラスに提案方式を用いた場合において、2 種類の異なるキュー内滞在時間の閾値ならびにパケット廃棄率を設定したシミュレーションについて、トラフィックが集中する入力側のエッジルータにおける各クラスでのキュー内滞在時間を測定した。図 10 に設定 1 における滞在時間のヒストグラムを、図 11 に設定 2 における滞在時間のヒストグラムを示す。横軸が滞在時間、縦軸が頻度である。

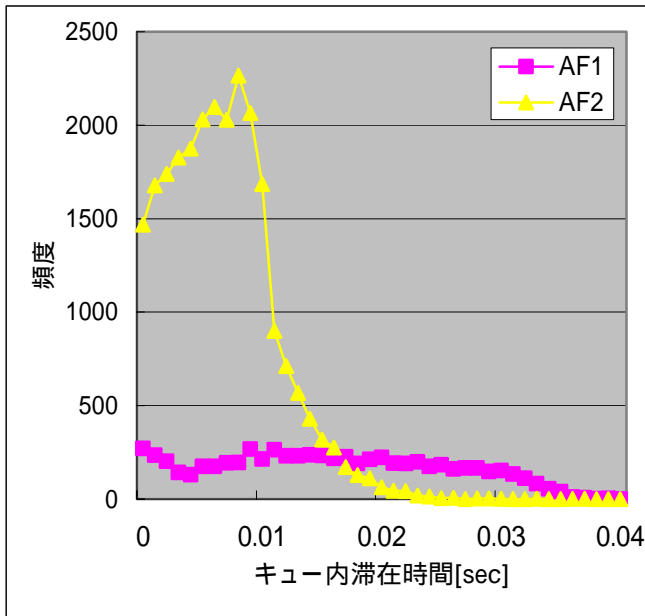


図 10 設定 1 における滞在時間のヒストグラム

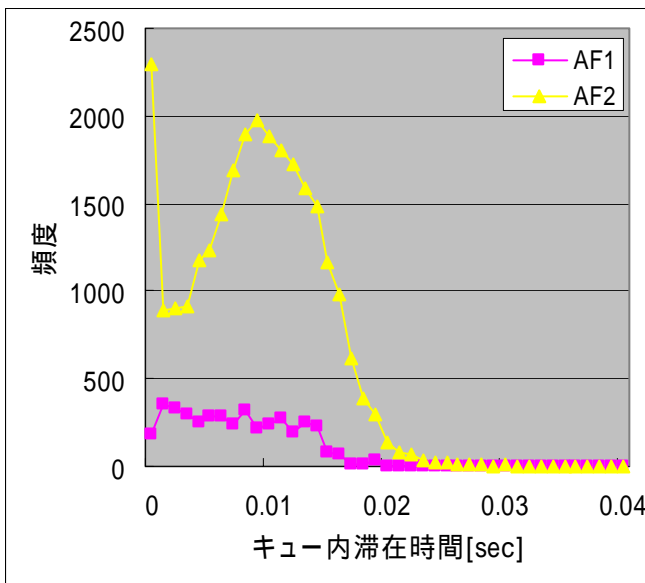


図 11 設定 2 における滞在時間のヒストグラム

AF1 クラスの分布をみると滞在時間はほぼ均等になっている。これは UDP を用いた通信であり、輻輳制御を行わないからである。しかし、契約を超え、Red にマーキングされたパケットを積極的に廃棄しているので、その影響が少なく、滞在時間が抑えられている(図 12 参照)。閾値設定を厳しくするほど、このパケット廃棄が早くなり、他のパケットへの影響が小さくなり滞在時間が減少する。

AF2 クラスに関しては、分布が大きい部分があるが、これは閾値の設定に影響される。TCP フレンドリはロス等のネットワーク状況によりレート制御を行うので、時間を閾値として廃棄を行う提案方式により影響を受

ける。よって、最も厳しく設定した閾値付近に滞在時間の分布が増加するのである。

本来廃棄されるべきパケットであるが、このパケットがキューに滞在すると、他の契約内パケットの滞在時間にも影響を与えてしまう。

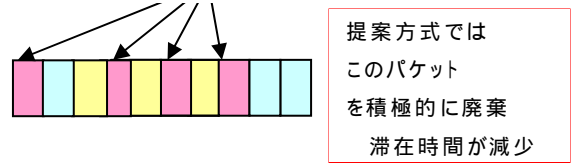


図 12 輻輳状態のキューに関する考察

6. まとめ

本稿において、平均滞在時間を用いたバッファ管理方式の比較検討を行った。シミュレーションの結果から、エッジルータにおいて、リアルタイム性が要求されるアプリケーションに対してキュー内の滞在時間を減少させ、結果として End-to-End での遅延においても特性を改善することができた。また、アプリケーションの遅延要求に応じてパラメータを変えることにより各クラスの遅延特性を制御可能であるということがわかり、提案方式の有効性を示すことができた。

謝辞

この研究は TAO 委託研究課題"通信ネットワーク利用放送技術"の研究開発のサポートによる。

文献

- [1] S.Blake, D.Black, M.Carlson, E.Davies, Z.Wang, W.Weiss, An Architecture for Differentiated Services, RFC2475, Dec.1998.
- [2] V.Jacobson, K.Nichols, K.Poduri, An Expedited Forwarding PHB, RFC2598, June. 1999.
- [3] J.Heinanan, F.Baker, W.Weiss, J.Wroclawski, Assured Forwarding PHB Group, RFC2597, June.1999.
- [4] J.Heinanan,R.Guerin, A Two Rate Three Color Marker, RFC2698, Sept.1999.
- [5] M. Katevenis, S. Sidiropoulos, C.Courcoubetis, Weighted round-robin cell multiplexing in a general-purpose ATM switch chip, IEEE Journal on Selected Areas in Communications, October 1991.
- [6] S. J. Golestani, A Self-Clocked Fair Queueing Scheme for Broadband Applications, Proc. IEEE INFOCOM, pp. 636--646, 1994.
- [7] C.Dovrolis, P.Bhagwat, A Case for Relative Differentiated Services and the Propotional Differentiation Model, IEEE Network, Sep.1999.
- [8] S.Floyd, V.Jacobson, Random Early Detection gateways for Congestion Avoidance, IEEE/ACM Transactions on Networking, August 1993.
- [9] 後藤 渉,保岡 裕,甲藤 二郎,遅延を考慮した適応 Diffserv スケジューラとバッファ管理アルゴリズムの提案, 信学技報 NS2002-173
- [10] N.Seddigh,B. Nandy, and P. Piedad, Study of TCP and UDP Interaction for AF PHB, IEEE HPSR'2001, May 2001
- [11] Network Simulator-ns (ver.2), <http://www.isi.edu/nanam/ns>, UCB/LBN/VINT.