

# 優先度付きキューを用いた確率的 QoS 保証におけるポリシング法

岡 敏生\* 藤川 賢治† 岡部 寿男† 池田 克夫‡

\* 東京大学大学院新領域創成科学研究科

† 京都大学大学院情報学研究科

‡ 大阪工業大学情報科学部

本論文ではストリーム型トラフィックを対象としたポリシング手法を提案する。インターネットの普及により近い将来、電話・遠隔会議・放送などのシステムを現在よりも廉価で実現できるようになると思われる。これらのストリーム型トラフィックは遅延についての要求が一般に厳しい。したがって、バックボーンにおいて高速に QoS(Quality of Service) を保証するルータが望まれる。そのような背景をもとに、Policed Priority Queuing(PPQ) と呼ばれるスケジューリング方式が提案されている。PPQ では各出力リンク毎に複数のフローで一つのキューを共有している。そのため送信規則を違反するフローのパケットを廃棄しないと、他のフローのパケット損失率が上昇してしまう。そこで本研究では各フローが PPQ の送信規則を遵守しているか監視する手法の提案を行なった。提案手法では良く知られているトークンバケット法を効果的に利用する方法について言及している。本手法を用いることで適切な利用を妨げないで、不正フローのポリシングを行うことができる。それによって、一般的な利用の範囲でのフローの孤立化を行うことができる。

## A Policing Method for Statistical QoS Guarantee using Priority Queuing

Toshio Oka\* Kenji Fujikawa† Yasuo Okabe† Katsuo Ikeda‡

\*Graduate School of Frontier Sciences,  
the University of Tokyo

†Graduate School of Informatics,  
Kyoto University

‡Faculty of Information Science, Osaka Institute of Technology

In this paper, we propose a policing method for stream traffic. Inexpensive systems of remote meeting, telephone, and broadcasting are going to be realized by the development of the Internet. The requirement for delay of these stream traffics is generally demanding. Therefore, it is desirable that high-speed QoS routers are introduced to the backbone network. In order to solve the situation, a scheduling algorithm called Policed Priority Queuing(PPQ) has been proposed. In PPQ router, many flows of the same output link share a QoS queue. Unless the packets of illegal flows are not discarded, the packet loss rates of other flows might rise. In this paper, we propose a policing method that detects illegal flows and discards the packets of them. We refer to an effective use of a well-known policing method, or token bucket. By using the proposed method, you can police the illegal flows without obstructing proper use. We aims at the realization of isolation of flows within a general use.

### 1 はじめに

インターネットを利用して実時間で音声や映像を伝送しようという試みが広くなされている。本研究では、それらの通信品質 (Quality of Service, QoS) を保証するためのスケジューリングアルゴリズム Policed Priority Queuing(PPQ) [1, 2, 3] に用いるトラフィック監視 (policing, ポリシング) の手法を提案する。PPQ では保証される最大遅延が各フローの予約帯域ではなく、出力リンク速度の逆数に比例するため、保証される最大遅延が小さい。そのため、遅延に敏感なストリーム型のトラフィックに適している。また PPQ ではキューイングにおいて FIFO(First-in First-out) を用いている上、キューイングに必要なバッファが小さくて済むため、実装が比較的容易になっている。こ

れらの性質から PPQ を実装したルータはストリームの QoS を保証するバックボーンルータに最適である。PPQ では複数のフローで一つのキューを共有しているため、送信規則を違反するフローのパケットを廃棄しないと、他のフローのパケット損失率が上昇してしまう。そこで本研究では各フローが PPQ の送信規則を遵守しているか監視する手法の提案を行なった。PPQ では WFQ に代表される決定的スケジューリングアルゴリズム [4, 5, 6] と異なり、あらゆる場合において理論的にパケット損失を保証することは行わない。しかしながら本研究で提案するポリシング手法を用いることで、一般的利用におけるパケット損失を確率的に保証することができる。

## 2 準備

### 2.1 決定的スケジューリングアルゴリズムを用いてストリームを扱う問題点

ストリーム型トラフィックを扱う上で WFQ などの決定的アルゴリズムを用いることは、以下の二点において望ましくない。

- フロー数が多くなると実装が容易でない
- 保証される最大遅延がストリーム型アプリケーションとしては大きい

決定的保証を行うアルゴリズムでは、他のフローの振舞いによらず通信品質を保証するために、バッファサイズが大きくなりがちである。又、パケットの転送順序を決定する処理が複雑であり、実装上不利である。

また最大遅延については、すべてのフローのバッファが空でない場合を想定して、最大遅延を算出している。そのため最大遅延の項にフローの重みが含まれてしまい、リンクが高速であっても改善の効果が少ない。

### 2.2 Policed Priority Queuing(PPQ) の概要

2.1 節で述べたような決定的アルゴリズムの問題点を解決するために、Policed Priority Queuing(PPQ) [1, 2, 3] と呼ばれる手法が提案されている。

PPQ の特長を列挙する。

- 通信品質保証されるパケットはベストエフォート方式のフローの影響を受けない
- 最大遅延は QoS 保証用のキューの最大キュー長とパケットサイズ、出力リンク速度によって決定される。リンクが高速であるほど遅延は小さくなり、保証される最大遅延はフロー数が多い場合、WFQ よりも十分小さい
- 各フローはキューを共有するのでバッファサイズは小さくてよい
- 各パケットの処理は定数時間でできる
- パケット損失率は、QoS 保証されるフローの帯域利用率と最大キュー長によって設定できる
- 実装が容易である

本稿ではベストエフォート方式については触れず、QoS 保証に関する部分について扱う。

### 2.3 PPQ における送信規則

バースト性の高いトラフィックは合流しても高いバースト性を有するため、キューイングに必要なキュー長が大きくなってしまう。そのため、PPQ ではパケットの到着に関して

- 予約レートをパラメータとするポアソン到着であるか、またはレート・バースト性が共にそれよりも低い到着でなくてはならない
- 他のフローと同期してはならない

という制限が設けられている。

互いに独立なポアソン到着が合流してもポアソン到着になることが知られているので、各フローについて PPQ でのパケット到着の制限が守られた場合、ルータに到着するパケットの到着パターンはポアソン到着であるか、またはそれよりもバースト性が低い到着になると考えられる。

### 2.4 PPQ におけるポリシングの要件

PPQ におけるポリシングは以下の要件を満たす必要がある。

- パケットを受信するレートを予約帯域程度に抑制すること
- パケットの到着パターンのバースト性を抑制すること
- 第一種の誤り率が低いこと
- 容易に実装できること

PPQ では所定のパケット損失率を達成できるように予約を行っている。そのため予約帯域を大幅に上回るパケットが到着すると、他のフローのパケット損失率に影響をおよぼしてしまう。従ってその場合、ポリシングによって一部のパケットを廃棄する必要がある。

ポリシングのもう一つの役割としてバースト性の抑制がある。バースト性の高い送信の場合、長期的に観測すれば予約レートを守っていたとしても、一時的にパケットの到着レートが高くなってしまう。それによって他のフローのパケットが廃棄されてしまう可能性が生じる。そのため PPQ におけるポリシングでは、ポアソン到着よりもバースト性が高い場合パケットを廃棄しなくてはならない。

以上二点が PPQ におけるポリシングの役割であるが、それらを実現する上で第一種の過誤に注意する必要がある。PPQ におけるポリシングは、送信規則を違反しているかどうかを調べる検定とみなすことができる。その際、適切な送信を行なっているにもかかわらず違反とみなされる確率(第一種の誤り率)は、十分低く設定されなくてはならない。本研究では誤検出による被害を軽微にするため、「第一種の誤り率がキュー溢れによるパケット損失率を十分下回るようにする」という方針をとる。

### 2.5 複数ノードを経由するトラフィックの到着パターンに関する考察

複数ノードを経由する場合、しばしばトラフィックパターンが変化することが知られている [7]。PPQ においては送信時のトラフィックパターンが、ポアソン過程よりもバースト性が高くないように定められているが、複数ノードを経由する過程でバースト性が高くなってしまいかも知れない。その場合、ポリサーにおいてポアソン到着を基準にバースト性の検定を行うと、適正な送信を行なっているにもかかわらずパケットが廃棄されてしまう。そのため、複数ノードを経由したパケットの到着パターンが、ポアソン到着よりもバースト性が高くないことが望ましい。

本節ではポアソン過程に従って送信したトラフィックの性質が、複数ノードを経由してもほとんど変化しないことを示す。

そのため先ず特定の仮定を設け、その仮定の下ではノードを経由してもトラフィックパターンがポアソン到着であることを示す。その後仮定がない場合でも、バースト性がほとんど変化しないことを述べる。  
仮定

1. ルータにおけるキュー長は無限大
2. パケット到着時にはキューは定常状態にある。定常状態においてキュー長が  $j$  である確率を  $q^\infty(j)$  とする
3. パケットが到着してから出力されるまでの時間を  $j \times \Delta_o[\text{sec}]$  に数値上補正するとする ( $j$  は到着直後のキュー長,  $\Delta_o[\text{sec}]$  は出力間隔とし, 十分小さいこととする)

フロー  $i$  の入力側の到着をパラメータ  $\lambda_i$  (定数) のポアソン到着とする。

出力インターフェイス側で、時間  $(t_0, t_0 + \Delta t]$  にパケットが出力される確率  $P_0$  を求める。( $t_0$  は任意)

$(t_0 - j \times \Delta_o, t_0 - j \times \Delta_o + \Delta t]$  に入力側にパケットが到着する確率を  $P^{0j}$ , 入力側に到着したパケットが入った直後のキュー長が  $j$  である確率を  $q_0(j)$  とすると  $q_0(j) = q^\infty(j)$  なので

$$P_0 = \sum_{j=0}^{\infty} P^{0j} \times q_0(j) \quad (1)$$

$$= \sum_{j=0}^{\infty} (\lambda_i \Delta t + o(\Delta t)) \times q^\infty(j) \quad (2)$$

$$= (\lambda_i \Delta t + o(\Delta t)) \times \sum_{j=0}^{\infty} q^\infty(j) \quad (3)$$

$$= \lambda_i \Delta t + o(\Delta t) \quad (4)$$

同様に時間  $(t_1, t_1 + \Delta t]$  にパケットが出力する確率  $P_1$  ( $t_1$  は任意) は

$$P_1 = \sum_{j=0}^{\infty} P^{1j} \times q_1(j) \quad (5)$$

$$= \lambda_i \Delta t + o(\Delta t) \quad (6)$$

$P_0, P_1$  はともに  $\Delta t$  の一次の項が定数なので互いに独立。よって出力されたパケットもポアソン到着になる。従って、上記の仮定を設けると複数ノードを経由しても、トラフィックパターンはポアソン到着となる。

次に仮定 3 がない条件を考える。するとパケットの到着間隔は、一つのノードを経由する過程で最大で  $\Delta_o$  ずれる。しかし出力リンク速度に対して予約帯域が十分小さい場合、そのずれは到着間隔に対して十分小さいので無視できる。

仮定 1 がない条件を考えるとパケットが損失することがあるので、ノードを経由した場合同じパラメータのポアソン到着になるとは限らない。しかしその場合バースト性はむしろ小さくなる。

以上からポアソン過程に従って送信した場合、複数のノードを経由してもトラフィックのバースト性はほとんど高くないことがわかる。ただし、仮定 2 に関しては議論を尽くす必要がある。

一方到着がポアソン到着よりもバースト性が低い場合を考えると、複数ノードを経由することによって、ポアソン到着よりもバースト性が高くなることはないと考えられる。

### 3 PPQのためのポーリング手法の提案

#### 3.1 送信パターン

同期的な送信はパケット損失率に悪影響を与えてしまうが、本研究では高速バックボーンに被害を与える程正確に同期させることは困難と考え、各フローは同期的に送信しないとする。

そうすると、違反トラフィックパターンは帯域を大きく違反するものと、そうでないものに分類できる。前者のパターンに対しては、多段トークンバケット法を提案する。一方、後者は多段トークンバケット法では検出できないため、マスターバケット法を提案する。

#### 3.2 多段トークンバケット法の提案

ポーリングにはしばしばトークンバケットが用いられる。ただし PPQ において一つだけトークンバケットを用いることは以下の理由で許容できない。

1. トークン追加レートが大きいバケットを用いると予約帯域を違反しているフローを検出できない
2. バケットサイズが大きいバケットを用いるとバースト性の高いトラフィックが通過してしまう
3. レート、バケットサイズ共に小さいバケットを用いると、適正なトラフィックのパケットを大量に破棄してしまう

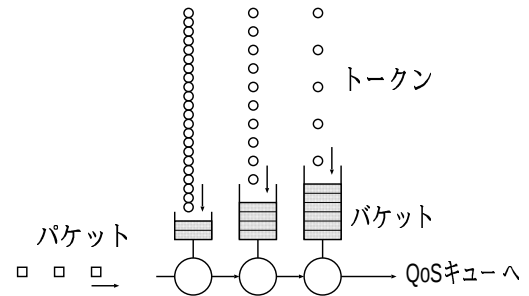


図 1: 多段のトークンバケット

そこで本節では、それらの問題を解決する手法として多段トークンバケット法 (図 1) を提案する。

多段トークンバケット法ではフロー毎にサイズの異なるバケットを複数用意し (各々のサイズを  $B_1 \sim B_k$  [個] とする), その中にトークンを蓄えることとする。パケットが到着するとそれぞれのバケットの中のトークン量を調べ、一つでもトークンのないバケットがあればそのバケットを廃棄する。すべてのバケットにトークンがある場合、到着したパケットを伝送する為 QoS 保証用のキューに入れ、すべてのバケットが

ら一つずつトークンを除去する。トークンは各バケット毎に決めるレート ( $R_1 \sim R_k$ : 予約レート  $\lambda$ [pps] よりも大きい) で補充する。トークンの個数以上にパケットの送信はできないので、任意の時間区間  $(t_0, t_0 + \delta t]$  において QoS 保証用のキューに入れられるあるフローのバケット数  $n$  は

$$n \leq \min_{i=1 \dots k} (B_i + R_i \times \delta t) \quad (7)$$

を満たし、予約レートを違反した送信やバースト性が極端に高い送信を抑えることができる。

この手法では PPQ で定められた適正な送信を行っても、ポリシングによってパケットが廃棄されてしまうことがある。しかし、その確率がキュー溢れによるパケット損失率よりも十分小さくなるように、バケットサイズ  $B$  およびトークンの追加レート  $R$  を設定することで、その影響を無視できる程小さくすることができる。

そこでパケットの到着がポアソン到着である場合について、パケットが到着した時点でトークンが存在しない確率を求める。ここではバケットの空き容量を、待ち行列理論における待ち行列長に見立てることによって、その確率を割り出す。すなわちトークンがいっぱいである時は待ち行列長が 0、トークンが空である時は待ち行列長が最大であるとみなす。バケットが一つ到着すると、トークンは除去されるので待ち行列長は 1 増える。逆にトークンが追加されると待ち行列長は 1 減る。

そうするとパケットの到着はポアソン到着であり、トークンが追加されるまでの時間は一定なので、バケットの空き容量は  $M/D/1/K$  待ち行列モデルに従う。 $A_j$  は時間  $\frac{1}{R}$ [sec] に  $j$ [個] のパケットが到着する確率)

よって  $M/D/1/K$  待ち行列モデルからパケット到着時にトークンがない確率 (即ちポアソン到着であるにも拘らずパケットが廃棄される確率) が求まるので、その確率がキュー溢れによるパケット損失より十分小さくなるようにパラメータを設定すればよい。

表 1 に到着がパラメータ  $\lambda_i$  のポアソン到着である場合に、各バケットで廃棄される確率が  $10^{-6}$  以下になるような、トークン追加レート  $R$  及びバケットサイズ  $B$  の関係を示した。例えば、予約レートが 50[pps]、トークン追加レートが  $50 \times 1.25$ [個/sec] で、バケットサイズが 30 のとき、パケット到着時にトークンがない確率は  $10^{-6}$  以下である。

到着がポアソン到着よりもバースト性が低い場合は、パケットが廃棄される確率はさらに低くなる。従って適切な送信に対してはポリシングによる第一種の誤り率を十分低くすることが可能である。

複数のトークンバケットを用いてポリシングする際には、表 1 のように第一種の誤り率が同程度のバケットの組合せを用いることが、効率的なポリシングと考えられる。

### 3.3 マスターバケット法の提案

#### 3.3.1 高いバースト性を持つ送信の防止

バースト性の高い送信でありながら、多段トークンバケット法で監視できない典型的なパターンを以下に挙げる。

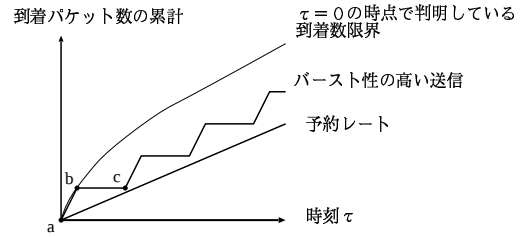


図 2: バースト性の高い送信

1. サイズの小さいバケット (例えば図 1 の 1 番左のバケット) のトークンが空になるまで速いレートで送信する (図 2 の線分 ab)
2. そのバケットのトークンが空になった時点で送信を中断する (図 2 の線分 bc)
3. そのバケットのトークンが溜った時点で、再び送信を再開する

このパターンを繰り返した場合、多段のトークンバケットではポリシングによってパケットを廃棄することができないので、高いバースト性を持った到着パターンが通過してしまう。(図 2)

図 2 のような送信は、一時的に送信を中断してトークンバケットのトークンが溜まるまで待つため、多段トークンバケットの枠組みでは検出できなかった。

しかしこのような送信では、送信しない区間を設けることでバースト性を生み出だしている。したがって送信しない区間を見つけることでそのような送信を防ぐことができる。

#### 3.3.2 マスターバケット法

バースト性が高い送信に対する手法として、マスターバケット法を提案する。この手法ではパケットの送信が行われない区間の検出に状態遷移を導入する。

具体的には、あるバケットにトークン (マスターバケットトークン) を一定レートで追加してゆき、パケットが到着する毎にマスターバケットトークンを除去する。到着間隔が広い程、除去するマスターバケットトークンの数を多くしてやれば、送信しない区間の長いトラフィックパターンではマスターバケットトークンがなくなってしまう。マスターバケットトークンがなくなれば送信がなかったとして、その区間の多段トークンバケットへのトークンの追加を行わない。

こうすることで、パケット到着がない区間の多段トークンバケットへのトークン追加がなされないため、バースト的な送信を防ぐことができる。

この手法においても、適正な送信を誤って検出してしまふ可能性があるため、誤検出の確率を保証しなくてはならない。これは到着間隔を離散化して離散マルコフ連鎖を用いることで誤検出の確率を求めることができる。

#### 3.3.3 バースト検出の速度

このマスターバケットを用いた場合のバースト検出の速度を検証する。パケットを連続的に  $n$  個送信した

表 1: 第一種誤り率が  $10^{-6}$  のパケットのレートとサイズの関係

|                       |      |      |      |      |      |      |      |      |      |      |      |      |
|-----------------------|------|------|------|------|------|------|------|------|------|------|------|------|
| $\frac{R}{\lambda_i}$ | 1.01 | 1.05 | 1.10 | 1.25 | 1.43 | 1.66 | 2.00 | 2.50 | 3.33 | 5.00 | 10.0 | 50.0 |
| B                     | 490  | 112  | 60   | 30   | 23   | 14   | 12   | 9    | 8    | 6    | 5    | 4    |

表 2: バーストのパラメータ  $n$  と検出されるまでの個数

|    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|
| n  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 |
| 個数 | 21 | 22 | 21 | 21 | 19 | 22 | 25 | 19 | 21 |

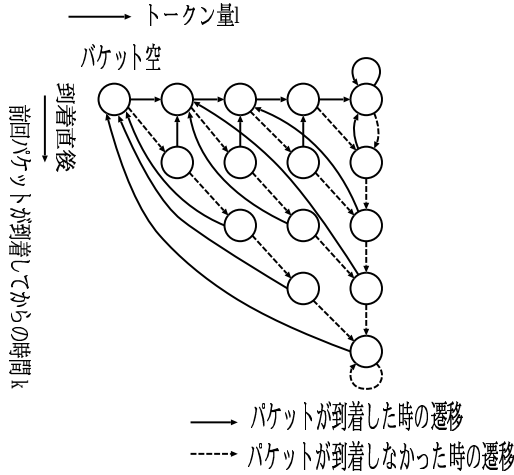


図 3: 有限マルコフ連鎖の状態遷移図 (パケットサイズ 4)

のち、通常の  $n$  倍の間送信を中断するといったバースト性の高い送信について、いくつパケットを送信した時点で異常が検出できるかを表 2 に示した。ここでは有限マルコフ連鎖が 定常状態に収束するように、パケットの到着に応じて除去するトークンの個数を次のように設定した。なおこれらの式及びパラメータは恣意的に決定したものである。ここでは、トークン追加レートを  $\lambda_i \times 1.25$  [個/sec]、到着間隔は  $k \times \Delta_s$  に離散化した。 $(\Delta_s = \frac{1}{\lambda_i} \text{[sec]})$  パケットサイズは 33 とし、パケットが到着すると  $f(k) = k + 0.2 \times \sqrt{33 \times k}$  [個] のトークンを除去する。この結果から比較的早期に、多段トークンバケット法で検出できないバーストを抑止できることが示された。

## 4 評価

本章では 3 章で提案した多段トークンバケット法およびマスターバケット法の評価を行う。

### 4.1 多段トークンバケット法の評価

多段トークンバケットの性能を評価する為、帯域違反がなされた場合についてシミュレーションを行なった。これによって不正なフローのパケットをどれだけ廃棄できるかを検証する。トークンバケットには表 1 に挙げる 12 種類のパケットの中から 5 つ選んで用いた。

### 4.2 シミュレーション

シミュレーションでは、すべてのフローが予約レートの 20 倍のレートで送信したとして検証する。(図 4)

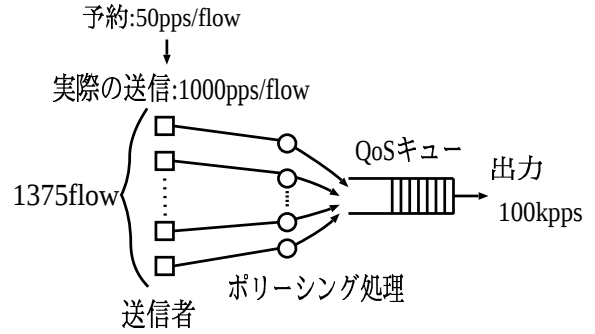


図 4: シミュレーション

シミュレーションの設定条件を以下に示す。

- ポリサーを通過してキューにパケットが入るレートが出力リンク速度の 0.8 倍になるように、出力リンク速度の 0.6875 倍まで帯域予約を認めた。
- 出力リンク速度は 100k[pps] とすると、全体で 68.75k[pps] まで予約を認めてよいので、フロー 1 つの予約帯域を 50[pps] として 1375 フローあるとした。(  $100000[\text{pps}] \times 0.6875 = 50[\text{pps}] \times 1375$  )
- キュー溢れによるパケット損失率は  $10^{-5}$  になるように最大キュー長を 26 とした。
- フローの開始時刻にはランダム性を持たせる。
- フローの持続時間  $T = T_{\min} + T_{\text{addition}}$  とした。 $T_{\min} = 10[\text{sec}]$  とし、 $T_{\text{addition}}$  は平均 10[sec] の指数分布に従うようにした
- フローは持続時間が経過すると終了し、新たなフローが開始するとする。フロー開始時にはパケットがいっぱいであるとする。
- 各フローは同期して送信を行わない。
- 300[sec] 間の転送処理をシミュレートした。

シミュレーション結果は表 3 の通り。シミュレーションでは帯域を違反している為、ポリシングで廃棄されたパケットを損失率に含めるのは不適當である。従って、パケット損失率の計算ではポリシングによる廃棄数を含めなかった。シミュレーション結果よりすべてのフローが帯域違反をしても、パケット損失率は  $10^{-5}$  を達成できることが示された。

表 3: 多段トークンバケット法によるポリシング結果

| 受信パケット数   | ポリサーでの廃棄  | キュー溢れ | 転送数      | パケット損失率               |
|-----------|-----------|-------|----------|-----------------------|
| 415392571 | 391656321 | 45    | 23736205 | $1.90 \times 10^{-6}$ |

### 4.3 考察

4.2 節で行なったシミュレーションではすべてのフローが帯域違反をするという非常に厳しい条件下にもかかわらず、多段トークンバケット法を用いることで QoS を保証できた。このことから送信者が欲張りな送信を行なった場合には多段トークンバケット法が非常に有効であると考えられる。

一方適切にシェーピングされていないフローについても、マスターバケット法によって早期にバーストを検出できることが示された。

さらにこれらの手法は

- 不正なトラフィックに対する強力な抑止として作用する
- ポリシングする上で保持すべき情報が少なく処理も単純である (パケット当たり  $O(1)$  の計算量)
- 第一種の誤り率を十分小さくできる

ということを共通の特長として備えていることも示された。

本稿のシミュレーションではパケット損失率を  $10^{-5}$  としてパラメータの設定を行なったが、用途に応じてパケット損失率を変更することは可能である。

ただしマスターバケット法は、多段トークンバケット法よりも計算量が数倍多いのに対し、それに見合う働きをするかについては検討が必要である。

## 5 おわりに

本稿では、優先度付き待ち行列を用いて QoS を保証するスケジューリングアルゴリズム PPQ におけるポリシング法を提案、評価した。

PPQ では実用を重視しているため特殊なケースを除外しており、通常の利用範囲では複数のフローでキューを共有しながら QoS を保証できる。PPQ は保証される最大遅延が極めて小さい上、単純な構成になっているため実装も簡易である。多くのスケジューリングアルゴリズムでは、他のフローから悪影響を防ぐためにそれぞれのフローを独立に扱っているが、PPQ では本研究で提案したポリシングを用いて孤立化を実現している。

提案した多段トークンバケット法はボアソン到着という理論的に単純な到着パターンからパケットサイズを導き出したが、一般的なトラフィックを扱う上でそれ以上バースト性は許容すべきでないため、今回求めたパケットサイズが上限になる考える。

多段トークンバケット法において重要なことは、到着がボアソン到着であれば同時に用いるバケットの第一種の誤り率が同程度にできるということである。ストリームの一般的な到着について第一種の誤り率が同程度になるようなパラメータを求めることが理想的で

あるが、本研究で求めたパラメータは十分実用的なものであり、解析的にも容易であるという長所を備えている。

多段トークンバケット法自体は一部のトラフィックに対応できないが、マスターバケット法と組み合わせることによって、より広いトラフィックに対応できるようになっている。

今回提案したポリシング法は確率的に QoS を保証しているためネットワークが高速である程正確に帯域違反を検出できる。以上の観点からトラフィック監視を用いた通信品質保証が、未来のネットワークにおいて重要であると考ええる。

今後の課題としては、ハードウェアルータを用いた実験を行うと共に、実用・理論の両面から PPQ とそのポリシング手法をより深く検証する必要がある。

## 参考文献

- [1] Fujikawa, K., Fujimoto, Y., Ikeda, K., Matsu-furu, N. and Ohta, M.: Comfortable Service: A New Type of Integrated Services Based on Policed Priority Queuing, 情報処理学会研究報告 (2000).
- [2] 藤川賢治, 藤本義人, 太田昌孝, 岡部寿男, 池田克夫: Policed Priority Queuing (PPQ) を用いたフローごとの QoS 保証のための Comfortable Service (CS) の提案, *DICOMO2001* (2001).
- [3] 藤本義人: M/D/1/K 待ち行列モデルを用いた 通信品質保証の要件に関する考察, 京都大学工学部 情報学科特別研究報告書 (September, 1999).
- [4] K.Parekh, A. and G.Gallager, R.: A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks-The Single Node Case, *INFOCOM '92*, pp. 0915-0924 (1992).
- [5] Demers, A., Keshav, S. and Shenker, S.: Analysis and Simulation of a Fair Queueing Algorithm, *Proc. of ACM SIGCOMM 1989*, pp. 1-12 (1989).
- [6] D.Stilidiadis, A.: Latency-rate servers: A general model for analysis of traffic scheduling algorithms, *INFOCOM '96*, pp. 0111-0119 (1996).
- [7] 西田 武志 監訳 Craig Patridge 著: ギガビット ネットワーク, ソフトバンク株式会社 (1995).