統計数理(2007)
第 55 巻 第 1 号 177–195

［研究詳解］

トラヒックの同定・推定に基づく シェーピングアルゴリズムの検討

本多 泰理†

(受付 2005 年 12 月 26 日；改訂 2006 年 11 月 15 日)

要　　旨

本稿では，ネットワーク上のデータの流れ，即ちデータトラヒック(以下，単にトラヒック)の同定・推定を，パケット毎の到着間隔とパケット長に着目して実現すると共に，その適用領域として，トラヒックシェーパー(帯域制御装置)におけるパケットロスの低減と出力スループット維持の効果を実現するアルゴリズムを提案する．手法として，パケット毎の到着間隔およびパケット長の時系列データを多変数時変係数 AR モデルとして記述・同定する．更に得られたモデルにカルマンフィルタを適用し，一期先のパケット到着時間間隔およびパケット長を推定する．上記推定手法に基づくトラヒックシェーピングアルゴリズムは，従来の Token Bucket アルゴリズムに比較して，トラヒックの一期先予測に基づきトークン生成率を動的に可変とすることにより，トラヒックシェーパーの入力ポートにおけるパケットロス率の低減と出力トラヒックのスループット劣化の抑制を実現する．我々は更に，提案手法の有効性を数値シミュレーションにより確認する．ただし本稿では，入力ポートにおけるキューイングは行わず，シェーパーの転送レートはシェーピングレートに比して充分高いものと仮定している．

キーワード： 時変係数 AR モデル，カルマンフィルタ，トラヒックシェーピング．

1. 背景

トラヒックシェーピングは入力トラヒックのバースト性を出力トラヒックにおいて平滑化する技術であり，従来研究も数多く行われ，その中でも Token Bucket 等のアルゴリズムが比較的高い効果を表すことが示されてきた(Chang et al., 2004; Elwalid and Mitra, 1997; Francini and Chiussi, 2000)．

しかしながら，Token Bucket においても，入力トラヒックが高いバースト性を示す場合，シェーパー内部におけるパケットロス率が上昇することが指摘されている(Ji, 2001a)．この場合，トラヒックシェーパーの入力ポートにおいてパケットが廃棄されるため，出力ポートにおけるパケット送信レートは低下し，スループットの低減を誘発する．こうした場合においてスループットを維持するためには，Token Bucket の様な固定的なトークン生成によるパケット通過可否判定ではなく，何らかの動的な判定アルゴリズムによるパケット廃棄の抑制が必要である．上記の事情に鑑み筆者は，入力トラヒックパターンに対しトークン生成率を動的に調整し，シェーパー内部におけるパケットロス率を低減する手法を提案してきた．筆者は，H^∞ フィルタの適用によるパケット到着間隔に対する一期先予測に基づきトークン生成率を動的に

† 慶應義塾大学大学院 理工学研究科：〒223–0061 神奈川県横浜市港北区日吉 3–14–1

調整し，上記の効果が得られることを示した(Honda, 2006)．該文献においては状態空間モデルにおいて，事前に用意した複数のモデルによる予測値と実データの残差を比較し，オンラインで最適なモデルを選択するアプローチを採用した．しかしながら，フィルタリングに用いる初期値や係数を決定するための同定手法の精緻化，一期前以前のパケットによる影響の考慮，時系列データに周期的変動要因が存在するケースや，複数フロー入力時への対応等の課題が残存した．ここで我々はトラヒックフローを，宛先 IP アドレス，送信元 IP アドレス，宛先ポート番号，送信元ポート番号，プロトコルタイプの組合せにより定義する．例えば，同一の IP アドレスを有するホスト間の通信であっても，異なるアプリケーションが異なるポート番号を使用して通信を行う場合，各アプリケーションのデータは異なるトラヒックフローとして区別される．また Honda (2006) の残存課題の例として，ネットワーク上のフラグメントパケットのトラヒックフローが考えられる．経路上でフラグメントされた複数のフラグメントパケットが連続してネットワーク上を流れる場合，単一のパケットが複数のパケットにフラグメントされる．更に，その到着順序や到着間隔はフラグメントを実施する中継ノードや配信ノードの処理に依存する．従って一連のパケットの到着間隔・パケット長は互いに相関をもつため，単に一期前のパケットのみならず，一期前以前のパケットによる影響をも考慮してその特性を判断する必要がある．

本稿ではこれらの課題を解決するため，多変数時変係数 AR モデルを採用する．既往研究においても，トラヒック分析においてパケット到着間隔に着目した ARX モデルの採用例を見ることができる(森田 他, 2001)．しかしながら，実際のトラヒックにおいては相異なる属性を有する複数のデータフローが存在すること，また Token Bucket はパケット長に基づき個々のパケットの通過可否を決定するアルゴリズムであることから，パケット到着間隔とともにパケット長を考慮する必要がある．例えば VoIP (Voice over IP：IP 電話等に代表される，IP 網上での音声データ通信)のように，主にショートパケットから構成されるトラヒックフローもあれば，映像配信等のようにロングパケットから構成されるトラヒックフローも存在する．

さらに本稿では，対象トラヒックにおける周期的変動要因の考慮を可能とするため，時系列データの定常性の仮定は採用しない．本稿では，上記のトラヒックモデルに基づくトラヒック同定・推定手法を提案するとともに，トラヒックの一期先予測に基づくシェーピングアルゴリズムを提案する．

2. トラヒックシェーピングについて

トラヒックシェーパーとは，①ネットワーク上のデータトラヒックを所定の転送レートに調整し転送すること，②同時に各パケットの送出間隔を平滑化すること，の 2 点を機能として具備する装置であり，ネットワークの品質保証や帯域制御に使用される．例えば，通信事業者網において，配信サーバ下流に設置することにより映像等配信サービスの品質保証に使用される等の利用シーンが想定される．通常トラヒックシェーパーは主に図 1 のように構成され，入力ポートと出力ポートで異なる機能分担が実装されている(実装依存ではあるものの，概要としてはここで示す構成が一般的であり，本稿では図 1 の構成を前提として議論を行う)．すなわち，入力ポートでは設定されたトラヒックレートに基づき，出力ポート側でのバッファ溢れを防止するため，入力トラヒックに対してパケットの通過可否を決定する．次に，通過を許可されたパケットは出力ポート側のキューへ転送され，出力スループットの設定に応じ，所定のアルゴリズムにより送出間隔の平滑化が実施される．各パケットは一度バッファに格納され送出間隔の調整が行われるが，バッファ長を上回るパケットが到着した場合には破棄される．本稿では，入力ポートにおけるパケットの通過可否判定のアルゴリズムの検討を行うこととし，以

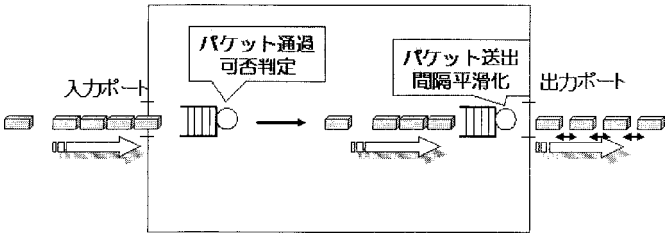

図 1. トラヒックシェーパーの機能構成.

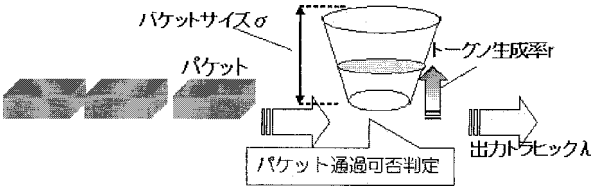

図 2. Token Bucket の概要.

下にその概要を述べる.

2.1 Token Bucket アルゴリズムについて

トラヒックシェーピングのアルゴリズムとして, 最も広範に知られているものの一つに Token Bucket (または Leaky Bucket)がある. 該アルゴリズムにおいては, トラヒックシェーパーは一定量のバッファ(Bucket)と, 入力パケットの転送可否の指標として用いられるトークンとから構成される(図 2). トークンは時間に対して比例的に Bucket に蓄積されるが, パケット受信時に所定量だけ減算される. 同時に, 減算後のバッファ内に残存するトークン量と所定のアルゴリズムに基づき該パケットの転送/廃棄を決定する. 以上のアルゴリズムにより, 出力ポートへ転送されるトラヒックは下式の意味において所定の上限をもち, 平滑化される.

$$\int_{\tau}^{t} \lambda(t)dt \leq r*(t-\tau)+\sigma \tag{2.1}$$

ここで $\lambda(t)$ は時刻 t における出力ポートへ転送されるトラヒックレート, r は時刻 t におけるトークン生成率, σ はトークンが蓄積されるバッファ(Bucket)のバッファ長を表す. 該アルゴリズムはまた, 多くのトラヒックシェーピングアルゴリズムの中でも比較的良好なスループットを示すことが知られている(Ji, 2001b). 以下に Token Bucket のアルゴリズムを示す:

$$\begin{aligned}
&\text{if} \quad buffer_k + r*(t_k - t_{k-1}) - c(l_k) < 0,\\
&/*\,\text{discard the packet}\,*/ \quad buffer_k = buffer_{k-1}\\
&\text{else} \quad buffer_k = \max\{bucket, buffer_{k-1} + r*(t_k - t_{k-1}) - c(l_k)\}.
\end{aligned}$$

ここで t_k, $bucket$, $buffer_k$, l_k, $c(\cdot)$ はそれぞれ, k 番目のパケット到着時刻, バケット長, 時刻 t_k における残存トークン量, パケット長, およびパケット長を引数にとる正値単調増加関数を示す.

2.2 Token Bucket の課題

前節では Token Bucket の概要を説明した．一方既知の事実として，入力トラヒックが高いバースト性を示す場合，Token Bucket はシェーパーの入力ポートにおけるパケットロス率の増大を招き，スループットも限定的となる．ネットワーク上のトラヒックは一般に高い分散を示す傾向があるため，実環境における Token Bucket の適用においては上記の点が課題となる．特に映像配信等の高レベルの QoS (Quality of Service) 保証が要求されるネットワークサービスにおいて，上記の課題が顕在化する．本稿で筆者は，トラヒック予測に基づきパケットのバースト到着時のパケットロス率の低減を実現するシェーピング方式を提案する．

2.3 出力ポート側の動作

前節では入力ポートにおけるパケット通過可否の判定アルゴリズムについて概説した．一方出力ポートにおいても，類似のアルゴリズムによりパケットの送出間隔の平滑化が図られる．即ち，所与の設定レートに対して所定の率で時間に比例してバッファにトークンが蓄積される．パケット到着時に，バッファ内に蓄積されたトークンはパケット長に応じ減算され，減算可の場合パケットは送出され，不可の場合パケットは待ち行列にキューイングされる．ただし，キュー長が所定量を上回った場合，該パケットは破棄される．このように出力側ではパケットをバッファリングすることにより，送出間隔の平滑化を実現する．

従って入力ポートにおいてはパケット破棄によるパケットロスが発生するのに対し，出力ポートではパケットのキューイングに起因する遅延が発生する傾向がある．パケット到着過程が高いバースト性を示す場合，出力ポートにおける遅延の増大とバッファ溢れの防止，またシェーパーの処理負荷抑制のため，入力ポートにおけるパケット通過可否の判定が行われるのである．

3. 提案方式について

前節で提示した課題に対して，我々は多変量時変係数 AR モデルの導入により，パケット毎の時系列データに対する同定とカルマンフィルタによる一期先予測を組合せ，トラヒック推定を行う．これに基づきトークン生成率を動的に調整した場合の，従来の Token Bucket に対する有効性を示す．

3.1 トラヒックのモデル化

時変係数 AR モデルは，通常の AR モデルに対して時間的に変化する係数を採用するとともに，係数の変化に時間的変動要因が含まれるケースを想定したモデルであり，時間的変動要因を要素として含む時系列の解析が可能となる．筆者は，既にトラヒックのパケット到着間隔の時系列に着目し，これを状態量として H^∞ フィルタによる推定法と，それに基づくシェーピングアルゴリズムを提案している．しかしながら，該検討においてはパケット長は固定長でモデル化していた．また，一期前のパケット到着間隔による影響は考慮可能であったが，それ以前のパケットによる影響を考慮することができなかった．そこで本稿では，パケット到着間隔とパケット長を要素とする 2 次元時変係数 AR モデルを採用する．AR モデルの次数の推定決定には AIC (Akaike Information Criterion) を用い，最小二乗法により必要なパラメータを推定，これを初期値としてカルマンフィルタを適用し一期先予測を行う．次節以降，概略を説明する．

3.2 多変量時変係数 AR モデルの推定

本節では，与えられたパケットの時系列から多変量時変係数 AR モデルの推定を行う手法に

ついて説明する．本手法は，北川(2005)の手法に基づいている．

$$\boldsymbol{y}_n = \sum_{j=1}^{m} \boldsymbol{A}_{nj}\boldsymbol{y}_{n-j} + \boldsymbol{w}_n. \tag{3.1}$$

ここで n はトラヒックシェーパーへの累積パケット到着数を表し，$\boldsymbol{A}_{nj} \in M(\boldsymbol{R})_{2\times 2}$ は係数行列，$\boldsymbol{w}_n$ は観測量のノイズを表す．また $\boldsymbol{y_n}$ はそれぞれ n 番目のパケット到着時刻におけるパケット到着間隔，パケット長を要素とするベクトルである．ここでパケット到着間隔の定義を述べる．例えば，午前 0 時から計測を開始し，パケットが①0 時 0 分 3 秒，②0 時 0 分 4 秒，③0 時 0 分 7 秒，④0 時 0 分 16 秒に到着したとする．この場合，時刻①のパケットについては到着間隔は定義されない．時刻②，③，④におけるパケット到着間隔はそれぞれ，1 秒，3 秒，9 秒と定義される．

係数行列 $\boldsymbol{A}_{nj}$ は以下の関係を満たすものとする：

$$\triangle^2 \boldsymbol{A}_{nj} = \boldsymbol{V}_{nj}, \quad j = 1,\ldots,m \tag{3.2}$$

上式は係数の時間的推移を示すものである．ただし，$\triangle$ は差分オペレータで $\triangle \boldsymbol{A}_{nj} = \boldsymbol{A}_{n,j} - \boldsymbol{A}_{n-1,j}$，2 次元ベクトル $\boldsymbol{w}_n$ は平均 $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$，分散共分散行列 W のガウス分布に従い，各 $\boldsymbol{V}_{nj} \in M(\boldsymbol{R})_{2\times 2}$ もガウス分布に従い，その第 i 列ベクトルはそれぞれ平均は共に $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$，分散共分散行列は $\begin{pmatrix} \tau_1^{(i)2} & 0 \\ 0 & \tau_2^{(i)2} \end{pmatrix} (i = 1,2)$ に従うものとする．$\tau_1^{(i)}$，$\tau_2^{(i)}$ はともにカルマンフィルタの中で推定値が求められるものであるが，その初期値はここでは与えられたものとする．我々は最小二乗法を用いて所与のパケット時系列データから上式における各種パラメータを同定する．対象となる AR モデルの最大次数 m を $m = 5$ と定め，$N > m$ なる N をとり，N 個のパケットデータ $\{\boldsymbol{y}_1,\ldots,\boldsymbol{y}_N\}$ がキャプチャされているものとする．最初に我々は(3.1)を以下の形に変形する：

$$\boldsymbol{y}_n = B_0\boldsymbol{y}_n + \sum_{i=1}^{m} B_i\boldsymbol{y}_{n-i} + \tilde{\boldsymbol{w}}_n \tag{3.3}$$

但し B_0 は下三角行列：

$$B_0 = \begin{pmatrix} 0 & 0 \\ b_0(2,1) & 0 \end{pmatrix} \tag{3.4}$$

各係数行列 $B_i (i = 1,\ldots,m)$ の (p,q) 成分は $b_i(p,q)$ と表されるものとする．またこの時，$\tilde{\boldsymbol{w}}_n$ の分散共分散行列 $\tilde{W}$ は対角行列 $\begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix}$ となるように変換を行うものとする．(3.3)は

$$\boldsymbol{y}_n = (I - B_0)^{-1} \sum_{i=1}^{m} B_i\boldsymbol{y}_{n-i} + (I - B_0)^{-1}\tilde{\boldsymbol{w}}_n \tag{3.5}$$

の形に変換することができる．分散共分散行列 $\tilde{W}$ が対角行列であることから，我々は(3.5)の各係数行列の行成分および分散を独立に推定することができる．そこで，まず

$$X = \begin{pmatrix} \boldsymbol{y}_m^T & \cdots & \boldsymbol{y}_1^T & \boldsymbol{y}_{m+1}^T \\ \boldsymbol{y}_{m+1}^T & \cdots & \boldsymbol{y}_2^T & \boldsymbol{y}_{m+2}^T \\ \vdots & \cdots & \cdots & \cdots \\ \boldsymbol{y}_{N-1}^T & \cdots & \boldsymbol{y}_{N-m}^T & \boldsymbol{y}_N^T \end{pmatrix}$$

に対しハウスホルダー変換を施し，

$$S1=\begin{pmatrix} s_{11} & s_{12} & \cdots & s_{1,2(m+1)} \\ 0 & s_{22} & \cdots & s_{1,2(m+1)} \\ 0 & \vdots & \cdots & s_{2m,2(m+1)} \\ 0 & \cdots & \cdots & \cdots \end{pmatrix}$$

の形にする．この時，$S1$ の左上の $(2m+1,2m+1)$ 小行列に基づき j 次モデルの $\boldsymbol{y}_n$ の第一成分

$$y_n(1)=\sum_{i=1}^{j} b_i(1,1)\boldsymbol{y}_{n-i}(1)+\cdots+\sum_{i=1}^{j} b_i(1,k)\boldsymbol{y}_{n-i}(k)+((I-B_0)^{-1}\tilde{\boldsymbol{w}}_n)(1)$$

の各係数および分散，AIC の推定を下式により算出できる：

$$\hat{\sigma}_j^2(1)=\frac{1}{N-m}\sum_{i=2j+1}^{2m+1} s_{i,2m+1}^2 \tag{3.6}$$

$$AIC_j(1)=(N-m)(\log 2\pi\hat{\sigma}_j^2(1)+1)+2(kj+1) \tag{3.7}$$

同様にして，第二成分のモデルの推定を行うため，X に反復的にハウスホルダー変換を施した行列

$$S2=\begin{pmatrix} s_{11} & \cdots & s_{1,10} & s_{1,11} & s_{1,12} \\ s_{21} & \cdots & s_{2,10} & 0 & s_{2,12} \\ 0 & \cdots & \vdots & 0 & \vdots \\ \vdots & \cdots & s_{11,10} & 0 & \vdots \\ \vdots & \cdots & 0 & 0 & s_{12,12} \\ 0 & \cdots & \cdots & \cdots & \cdots \end{pmatrix}$$

により

$$\hat{\sigma}_j^2(2)=\frac{1}{N-m}\sum_{i=2j+2}^{2m+2} s_{i,2m+2}^2 \tag{3.8}$$

$$AIC_j(2)=(N-m)(\log 2\pi\hat{\sigma}_j^2(2)+1)+2(kj+2) \tag{3.9}$$

から算出できる．以上の過程で得られた各次数の中で，最小の AIC を提示する m_0 を最適なモデル次数として選択する．なお，ここではパケット長に対する推定精度をより重視するものとし，m_0 の決定には第二成分の AIC を使用した．最終的に，(3.1) の A_{nj} は $A_{nj}=(I-B_0)^{-1}B_j$，W は $W=(I-B_0)^{-1}\tilde{W}((I-B_0)^{-1})^T$ により得られる．こうして後述するカルマンフィルタの初期値を算出することができる．

3.3 カルマンフィルタによる一期先予測

前節で得られた推定値を初期値として，我々はカルマンフィルタを構成し，一期先予測を行う．我々は (3.2) を以下の様に書き換える：

$$\begin{pmatrix} \boldsymbol{A}_{n,j} \\ \boldsymbol{A}_{n-1,j} \end{pmatrix}=\begin{pmatrix} 2\boldsymbol{I} & -\boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{0}_{2\times 2} \end{pmatrix}\begin{pmatrix} \boldsymbol{A}_{n-1,j} \\ \boldsymbol{A}_{n-2,j} \end{pmatrix}+\begin{pmatrix} \boldsymbol{V}_{nj} \\ \boldsymbol{0}_{2\times 2} \end{pmatrix} \qquad j=1,\ldots,m_0$$

($\mathbf{0}_{2\times2}$ は 2×2 次零行列)．更に

$$\boldsymbol{X}_n = \begin{pmatrix} \boldsymbol{A}_{n,1}^T & \boldsymbol{A}_{n-1,1}^T & \cdots & \boldsymbol{A}_{n,m_0}^T & \boldsymbol{A}_{n-1,m_0}^T \end{pmatrix}^T,$$

$$\boldsymbol{C}_n = \begin{pmatrix} \boldsymbol{y}_{n-1}^T & \mathbf{0}^T & \boldsymbol{y}_{n-2}^T & \mathbf{0}^T & \cdots & \boldsymbol{y}_{n-m_0}^T & \mathbf{0}^T \end{pmatrix}$$

($\mathbf{0}$ は 2 次零ベクトル) と置くことにより，当初の式は

$$\boldsymbol{X}_n = \boldsymbol{F}\boldsymbol{X}_{n-1} + \boldsymbol{B}\boldsymbol{V}_n \tag{3.10}$$

$$\boldsymbol{y}_n^T = \boldsymbol{C}_n\boldsymbol{X}_n + \boldsymbol{w}_n^T \tag{3.11}$$

と書き換えられる．但し

$$\boldsymbol{F} = \begin{pmatrix} \boldsymbol{G} & \cdots & \mathbf{0}_{4\times4} \\ \vdots & \vdots & \vdots \\ \mathbf{0}_{4\times4} & \cdots & \boldsymbol{G} \end{pmatrix}, \quad \boldsymbol{G} = \begin{pmatrix} 2\boldsymbol{I} & -\boldsymbol{I} \\ \boldsymbol{I} & 0 \end{pmatrix},$$

また

$$\boldsymbol{B} = \begin{pmatrix} \boldsymbol{I} & \mathbf{0}_{2\times2} & \cdots & \mathbf{0}_{2\times2} \\ \mathbf{0}_{2\times2} & \cdots & \cdots & \mathbf{0}_{2\times2} \\ \mathbf{0}_{2\times2} & \boldsymbol{I} & \cdots & \mathbf{0}_{2\times2} \\ \mathbf{0}_{2\times2} & \cdots & \mathbf{0}_{2\times2} & \mathbf{0}_{2\times2} \\ \vdots & \cdots & \cdots & \vdots \\ \mathbf{0}_{2\times2} & \cdots & \mathbf{0}_{2\times2} & \boldsymbol{I} \\ \mathbf{0}_{2\times2} & \cdots & \cdots & \mathbf{0}_{2\times2} \end{pmatrix}, \quad \boldsymbol{V}_n = \begin{pmatrix} \boldsymbol{V}_{n1} \\ \boldsymbol{V}_{n2} \\ \vdots \\ \boldsymbol{V}_{n,m_0} \end{pmatrix}$$

で与えられる．

次にカルマンフィルタを以下のように構成する：

$$\boldsymbol{K}_n^{(i)} = \boldsymbol{P}_{n/n-1}^{(i)}\boldsymbol{C}_n^T[\boldsymbol{C}_n\boldsymbol{P}_{n/n-1}^{(i)}\boldsymbol{C}_n^T + \boldsymbol{R}_n^{(i)}]^{-1} \tag{3.12}$$

$$\boldsymbol{X}_{n/n}^{(i)} = \boldsymbol{X}_{n/n-1}^{(i)} + \boldsymbol{K}_n^{(i)}[y_n(i) - \boldsymbol{C}_n\boldsymbol{X}_{n/n-1}^{(i)}] \tag{3.13}$$

$$\boldsymbol{X}_{n+1/n}^{(i)} = \boldsymbol{F}\boldsymbol{X}_{n/n}^{(i)} \tag{3.14}$$

$$\boldsymbol{P}_{n/n}^{(i)} = \boldsymbol{P}_{n/n-1}^{(i)} - \boldsymbol{K}_n^{(i)}\boldsymbol{C}_n\boldsymbol{P}_{n/n-1}^{(i)} \tag{3.15}$$

$$\boldsymbol{P}_{n+1/n}^{(i)} = \boldsymbol{F}\boldsymbol{P}_{n/n}^{(i)}\boldsymbol{F}^T + \boldsymbol{B}\boldsymbol{Q}_n^{(i)}\boldsymbol{B}^T, \quad (i=1,2). \tag{3.16}$$

ここで $\boldsymbol{X}_{n/n}^{(i)}$ $(i=1,2)$ は状態量 X_n の時刻 n における推定量の第 $i\,(i=1,2)$ 列，$\boldsymbol{X}_{n+1/n}^{(i)}$ $(i=1,2)$ は状態量 X_{n+1} の時刻 n における推定量の第 $i\,(i=1,2)$ 列，$\boldsymbol{K}_n^{(i)}$ $(i=1,2)$ は各列の推定量算出のカルマンゲイン，$\boldsymbol{P}_{n/n}^{(i)}$ $(i=1,2)$ および $\boldsymbol{P}_{n+1/n}^{(i)}$ $(i=1,2)$ は $\boldsymbol{X}_{n+1/n}^{(i)}$，$\boldsymbol{X}_{n/n}^{(i)}$ の推定誤差共分散行列，また $\boldsymbol{Q}_n^{(i)}$，$\boldsymbol{R}_n^{(i)}$ はそれぞれ $\boldsymbol{V}_n$ の第 i 列の分散共分散行列および $\boldsymbol{w}_n$ の第 i 成分の分散である．上のアルゴリズムにより，状態量の一期先予測値 $\boldsymbol{X}_{n+1/n} = (\boldsymbol{X}_{n+1/n}^{(1)}, \boldsymbol{X}_{n+1/n}^{(2)})$ に基づき，$\boldsymbol{y}_{n+1/n}^T = \boldsymbol{C}_{n+1}\boldsymbol{X}_{n+1/n}$ により，観測量の一期先予測値を算出する．

3.4 提案方式によるシェーピングアルゴリズム(単一フロー)

前節まで，本検討におけるトラヒック推定アルゴリズムの概要を説明した．本節では，前節のアルゴリズムに基づくトラヒックシェーピング方式の提案を行う．提案方式のシェーピングアルゴリズムは，トークン生成率をトラヒック予測に基づき動的に調整することを特徴とする．本方式は筆者の従来よりの提案方式であるが，本稿では更に複数フローを対象とするシェーピングについても考察を進める．Token Bucket におけるトークン生成率は，(3.17)(再掲)の意味においてシェーパーの出力側ポートへ転送されるトラヒックの上限値を規定するものである：

$$\int_{\tau}^{t} \lambda(t)dt \leq r * (t-\tau) + \sigma \tag{3.17}$$

ここで筆者は，シェーパーにおけるクレジットの概念を導入する．即ち，現在までの受信トラヒック量の総和と，上式(3.17)の右辺で規定される量との差分をクレジットとし，その値が正である場合にはまだ受信トラヒック量に余裕があるものと判断し，非負である場合にはこれ以上の受信はできないものと判断する：

$$credit(t) = r(t-\tau) + \sigma - \int_{\tau}^{t} \lambda(t)dt \tag{3.18}$$

但し右辺第 3 項は時刻 t までの受信バイト量を表す．これは，通常の Token Bucket による受信許容量と比較して，現在の受信トラヒック量の時間的総和での多寡を判断し，受信トラヒック量が少ない場合には通常の Token Bucket の受信トラヒック量の限界値よりも少ないため，まだ受信可能であると判断するためである．クレジットが正である場合に限り，トラヒックの一期先予測に応じてトークン生成率を決定する：

$$\text{if} \quad credit_k > 0 \quad \text{then}, \tag{3.19}$$

$$\text{if} \quad buffer_k > c(\hat{l}_{k+1}), \ \text{then} \quad token_{t_{k+1}} = r \tag{3.20}$$

$$\text{else} \quad token_{t_k} = \max\{r, \gamma * (c(\hat{l}_{k+1}) - buffer_k)/(\hat{t}_{k+1} - t_k)\} \tag{3.21}$$

$$\text{else} \quad token_{t_k} = r \tag{3.22}$$

ここで t_k，$credit_k$，$token_{t_k}$，$buffer_k$，$\hat{t}_k$，$\hat{l}_k$，$c(\cdot)$ はそれぞれ，k 番目のパケット到着時刻，時刻 t_k におけるクレジットの値，時刻 t_k から次のパケット到着時までにおけるトークン生成率，時刻 t_k における残存トークン量，一期先予測による推定パケット到着時刻，推定到着パケット長，パケット長を引数にとる正値単調増加関数を示す．(3.19)–(3.21)はクレジットに余裕がある場合，現時点でトークン残存量に余裕がある場合にはトークン生成率の初期値を採用すること，推定パケット到着時刻においてトークン残存量に余裕が無い場合にはトークン生成率を向上させることを示す．その際，正数 γ 倍の余裕を持たせている．また(3.22)はクレジットに余裕が無い場合にトークン生成率は初期値を採用することを意味する．なお本検討では $\gamma = 1.5$，$c(x) = x$ としている．上記の処理により，入力トラヒックレートが時間大域的には式(3.17)の意味での上限を上回ることなく，かつ上記の推定値に γ 倍の余裕を持たせて設定されたトークン生成率以下であれば，入力ポートにおいてはパケットの通過を許容することが可能となる．ただし，出力ポートの入力キューにおいて蓄積されたパケットがバッファ量を超過した場合にはこの限りではない．式(3.17)の右辺と，現在までの受信トラヒック量との差分を許容量として蓄積し，バースト的トラヒック発生時のトークン生成に充てることにより，固定的トークン生成によりパケット廃棄の判断を行う通常の Token Bucket に比して柔軟なシェーピング効果を得ることが可能となる．

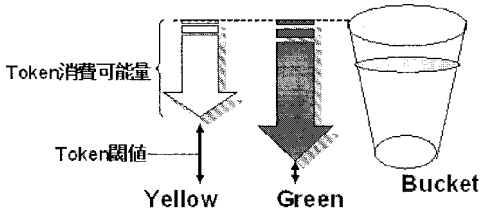

図 3. Color Awared Token Bucket の概要.

3.5 提案方式によるシェーピングアルゴリズム(複数フロー)

前節では，本検討におけるトラヒックの一期先予測を応用したシェーピングアルゴリズムを，単一のトラヒックフローに対し提案した．本節では複数フローをシェーピング対象とする場合におけるシェーピングアルゴリズムについて概要を述べる．複数フローに対する既存のシェーピングアルゴリズムとしては Color Awared Token Bucket および Group Policing が代表的である．

Color Awared Token Bucket とは，シェーピング対象となる各トラヒックフローに優先度を示す属性値として色の名前を定義し，色毎に定められた所定の閾値を Token Bucket の残存トークン量が下回った場合に該トラヒックに属するパケットを廃棄する手法である(概要を図 3 に示す)．残存トークン量の閾値は優先度の高い色のフローほど低く，優先度が低い色のフローほど高く設定される．図 3 の場合，閾値が低い Green が割り当てられたトラヒックは高優先であり，低優先の Yellow は閾値が高く設定されているためよりパケット廃棄率が高い．一方 Group Policing は，各トラヒックフローに所定の重み付けを設定し，重みに従いトークン生成率を割り当てる手法である．

本稿では前者の Color Awared Token Bucket を採用し，複数フローをシェーピング対象とする場合において，フローの優先制御と同時にパケットロスの低減を実現するシェーピングアルゴリズムを新規に提案する．提案手法では，前記のトラヒック予測のアルゴリズムは各フロー毎に実施し，一期先予測値を適用する．即ち，(3.10), (3.11)の形の状態方程式をフロー毎に構成し，各フローにつき到着間隔およびパケット長に対する一期先予測を算出する:

$$\boldsymbol{X}_{n_i} = \boldsymbol{F}\boldsymbol{X}_{n_i-1} + \boldsymbol{B}\boldsymbol{V}_{n_i} \tag{3.23}$$

$$\boldsymbol{y}^T_{n_i} = \boldsymbol{C}_{n_i}\boldsymbol{X}_{n_i} + \boldsymbol{w}^T_{n_i} \quad (i = 1, 2, \ldots, N) \tag{3.24}$$

ここで N はフロー数を，n_i はフロー i に属するパケットの現時点までの累積到着数を示す．各フローのトラヒックの一期先予測値が算出されると，シェーパーはこれに基づきバッファ残量を推定し，動的にトークン生成率を算出する．以下に 2 フローをシェーピング対象とした場合の例を示す．この場合，現在時刻 t におけるフロー i の到着間隔の一期先予測値 $\hat{t}_{n_i+1}$ は上記で算出される $\boldsymbol{y}^T_{n_i}$ の第一成分と現在時刻 t の和として算出される：$\hat{t}_{n_i+1} = t + (\boldsymbol{C}_{n_i+1}\boldsymbol{X}_{n_i+1/n_i})^{(1)}$. ここで $\boldsymbol{X}_{n_i+1/n_i}$ は n_i 番目のパケット情報に基づく n_i+1 番目の予測値を，また右辺の右上の添字は第一成分を意味する．フロー 1 の n_1 番目またはフロー 2 の n_2 番目のパケットが到着した時刻を現在時刻 t として，これらの値との大小関係の組合せに基づき，場合分けされた判定フローに従いトークン生成率等が決定される．

(フロー 1，2 ともに一期先予測到着時刻が現在時刻以前と算出される場合)

$$\text{if } \hat{t}_{n_1+1} < t \text{ and } \hat{t}_{n_2+1} < t \text{ then } token_t = r \tag{3.25}$$

(上記以外で，次の到着フローがフロー 1 であると予測される場合)

(3.26) if $(\hat{t}_{n_1+1} > t$ and $\hat{t}_{n_1+1} < \hat{t}_{n_2+1})$ or $(\hat{t}_{n_1+1} > t$ and $t > \hat{t}_{n_2+1})$ then,

(3.27) if $credit > 0$

(3.28) if $buffer_t > c(\hat{l}_{n_1+1})$, then $token_{n_1+1} = r$

(3.29) else $token_t = \max\{r, \gamma * (c(\hat{l}_{n_1+1}) - buffer_t)/(\hat{t}_{n_1+1} - t)\}$

(3.30) else $token_t = r$

(上記以外で，次の到着フローがフロー 2 であると予測される場合)

(3.31) if $(\hat{t}_{n_2+1} > t$ and $\hat{t}_{n_2+1} < \hat{t}_{n_1+1})$ or $(\hat{t}_{n_2+1} > t$ and $t > \hat{t}_{n_1+1})$ then,

(3.32) if $credit > 0$

(3.33) if $buffer_t > c(\hat{l}_{n_2+1})$, then $token_{n_2+1} = r$

(3.34) else $token_t = \max\{r, \gamma * (c(\hat{l}_{n_2+1}) - buffer_t)/(\hat{t}_{n_2+1} - t)\}$

(3.35) else $token_t = r$

ここで t，$token_t$，$buffer_t$，$\hat{t}_{n_i+1}$，$\hat{l}_{n_i+1}$ $(i = 1, 2)$，$c(\cdot)$ はそれぞれ，現在時刻(あるフローに属するパケットの到着時点)，時刻 t から次のパケット到着時までにおけるトークン生成率，時刻 t における残存トークン量，フロー i の次のパケット($n_i + 1$ 番目の)到着時刻およびパケット長に対する一期先予測値，パケット長を引数にとる正値単調増加関数を示す．(3.25) は全てのフローについて推定される一期先予測到着時刻が現在時刻よりも小さい場合には，パケットの推定到着時刻の推定は行わず，トークン生成率として初期値を採用することを表す．(3.26) はフロー 1 の一期先予測到着時刻が現在時刻よりも大きく，フロー 2 の一期先予測到着時刻未満であるか，またはフロー 2 の一期先予測到着時刻のみが現在時刻以前である状況においては，次に到着するパケットはフロー 1 であると判断することを示す．フロー 2 の一期先予測到着時刻が現在時刻未満の場合はフロー 2 のパケットが先着する場合もあり得るが，フロー 2 のパケットが欠落している場合等も考えられるため，ここでは次の到着パケットの予測をこのように定義する．(3.27) − (3.29) は上記の条件の下でクレジットに余裕がある場合，推定パケット長に比して現時点で残存トークン量に余裕があればトークン生成率として初期値を採用し，無ければトークン生成率を予測値に基づき上昇させることを意味する．その際，正数 γ 倍の余裕を持たせている．ここでも $\gamma = 1.5$，$c(x) = x$ としている．(3.30) はクレジットに余裕が無ければトークン生成率として初期値を採用することを示す．(3.31) − (3.35) は次の到着パケットがフロー 2 と推定される場合についての同様の設定を表す．

3.6 提案方式における出力ポート側のフロー

本節では，入力ポート側で提案方式のアルゴリズムを導入した場合に予想される出力ポート側のトラヒック挙動に関する検討を行う．前述の通り，出力ポートにおいては Token Bucket に基づき出力パケットの送出間隔の平滑化が行われる．しかしながら，到着パケットが高いバースト性を提示する場合，通常の Token Bucket によるパケット通過可否の判定が入力ポートで行われた場合，入力ポートにおけるパケットロス率が増大する可能性が高い．その場合，出力ポートへ転送されるパケットの間隔は拡大し，出力ポートにおける送出間隔が必要以上に増大する可能性がある．このため，出力ポートから送出されるトラヒックのスループットは設定レートより劣化する可能性が考えられる．一方提案方式の場合，入力ポートにおけるパケッ

トロス率の低減効果が有効に機能すれば，出力ポートへは設定レートに近い流量のトラヒックが流入し，出力スループットの劣化を一定レベルで抑制可能であると考えられる．

4. 数値シミュレーションによる提案方式の検証

本節では，該方式のケーススタディを作成し，数値シミュレーションによりその有効性を検証する．

4.1 ケーススタディ

ケーススタディ(1)．単一の入力トラヒックに対する提案手法によるトラヒック予測と，これに基づく提案手法によるシェーピングアルゴリズムを Token Bucket と比較する．対象の入力トラヒックのパケット到着間隔は区分的にガウス分布に従い，一部の時間区間(360–420 ミリ秒)において突発的にバースト性を示す特性を有するものとする．またパケット長はシミュレーション時間を通じ平均 1450 bytes のガウス分布に従うものとする．ケーススタディ(1)のパラメータの詳細は表 1 に示す．また提案方式のシェーピングの初期パラメータを表 2 に示す．またこの時，出力ポートでは 50 Mbps シェーピングを実施するものとする．ケース(1)，ケース(2)ともに $\tau_1^2 = 0.1, \tau_2^2 = 2.0$ とした．また出力ポートにおけるキューサイズは 5 Mbytes と設

表 1. サンプル時系列の設定(1)．

時間 (ミリ秒)	0-360	360-420	420-1000
平均間隔（μ 秒）	240	120	240
間隔分散	0.1	0.1	0.1
平均パケット長 (bytes)	1450	1450	1450
パケット長分散	2.0	2.0	2.0

表 2. シェーピングパラメータ設定(1)．

入力ポート		出力ポート	
バッファ長	トークン生成率	バッファ長	トークン生成率
100kbytes	50Mbps	100kbytes	50Mbps(50bytes/80μsec)

表 3. サンプル時系列の設定(2)．

時間 (ミリ秒)	0-200	200-242	242-300	300-349	349-400
フロー 1・平均間隔（μ 秒）	200	140	200	200	200
フロー 1・間隔分散	0.05	0.05	0.05	0.05	0.05
フロー 1・平均パケット長 (bytes)	1450	1450	1450	1450	1450
フロー 1・パケット長分散	2.0	2.0	2.0	2.0	2.0
フロー 2・平均間隔（μ 秒）	300	300	300	196	300
フロー 2・間隔分散	0.05	0.05	0.05	0.05	0.05
フロー 2・平均パケット長 (bytes)	1450	1450	1450	1450	1450
フロー 2・パケット長分散	2.0	2.0	2.0	2.0	2.0

表 4. シェーピングパラメータ設定(2)．

入力ポート				出力ポート	
バッファ長	トークン生成率	閾値 (フロー 1)	閾値 (フロー 2)	バッファ長	トークン生成率
100kbytes	100Mbps	0	1kbytes	100kbytes	100Mbps(100bytes/80μ sec)

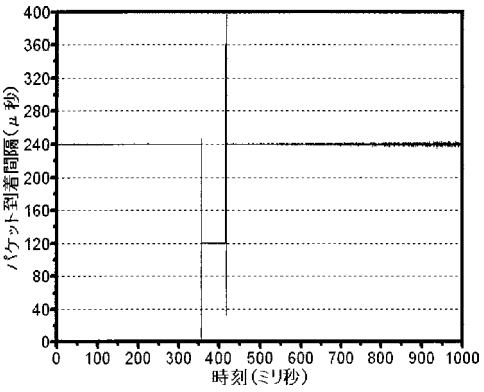

図 4. 推定パケット到着間隔(1).

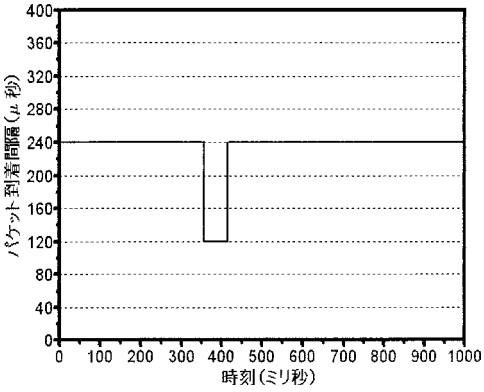

図 5. パケット到着間隔実値(1).

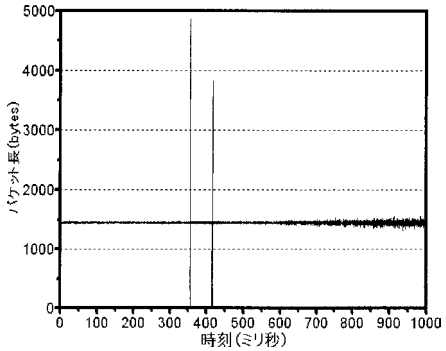

図 6. 推定パケット長(1).

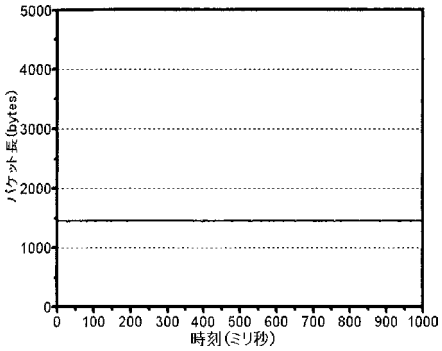

図 7. パケット長実値(1).

定した.

ケーススタディ(2). ケーススタディ(2)では, 複数トラヒックに対する予測と, 提案方式によるシェーピングアルゴリズムを Color Awared Token Bucket に適用した場合の有効性を確認する. ここでは優先度の異なる 2 本のトラヒックを対象とする. 各フローに対する設定は表 3 に提示する. ここでもケーススタディ(1)同様, 各トラヒックのパケット到着間隔は特定の時間区間(フロー 1:200–242 ミリ秒, フロー 2:300–349 ミリ秒)においてバースト性を示し, 各時間区間においてはそれぞれガウス分布に従うものとする. パケット長はシミュレーション時間を通じ平均 1450 bytes のガウス分布に従うものとする. また今回は, 両フローの出力ポートは同一であり, 出力ポートにおいては両フローの区別および優先制御は実施せず, 両フローを併せたトラヒックに対して 100Mbps シェーピングを実施するものとする. 出力ポートにおけるキューサイズは 10 Mbytes と設定した(表 3, 表 4).

4.2 結果

本節ではシミュレーション結果を示すとともに, 結果に対する考察と提案方式の有効性の確認を行う.

(1)図 4–図 7 はそれぞれ単一の入力トラヒックのパケット到着間隔およびパケット長に対する推定値と(シミュレーション上の)実測値を示すものである. これより, 各時間区間における

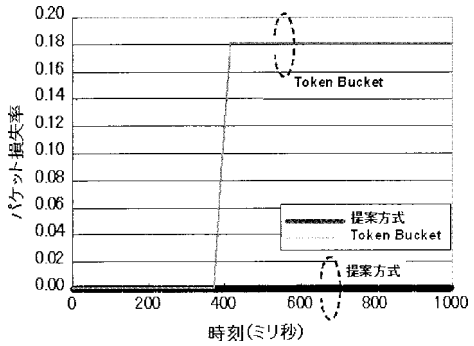

図 8. パケットロス率(1).

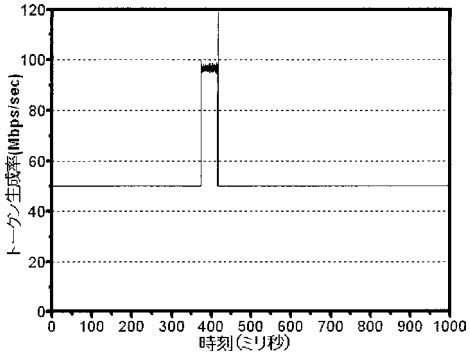

図 9. 提案方式によるトークン生成率(1).

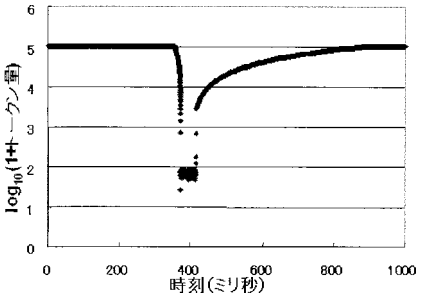

図 10. 提案方式におけるトークン残存量(1).

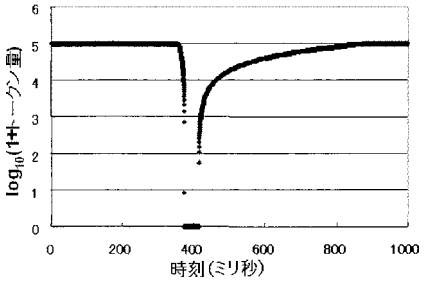

図 11. Token Bucket におけるトークン残存量(1).

パケット到着間隔の推定精度を確認することができるとともに，時間区間の境界においては一時的に推定値に変動が発生することが確認できる．次に該推定結果に基づき提案方式によるシェーピングを実施した場合の結果を示す．図 8 は同一の入力トラヒックに対する提案方式のシェーピングアルゴリズムと通常の Token Bucket 適用時におけるシェーパー内パケットロス率の比較を示している．図 8 から，入力トラヒックがバースト性を示す時間帯において通常の Token Bucket においてはパケットロスが発生しているのに対し，提案方式においてはパケットロスが発生していない事が確認できる．図 9 は入力ポートにおける，提案方式によるトークン生成率の推移を，また図 10，図 11 はそれぞれ提案方式および Token Bucket におけるトークン残存量の推移を示す．図 10，図 11 においては縦軸に対数プロットを適用している．これより，提案方式においては入力トラヒックがバースト性を示す時間帯においてトークン生成率を増大させている事，それによりトークン残存量の減少の抑制を実現しパケットロスを抑制している事が確認できる．図 12 および図 13 はそれぞれ，提案方式および Token Bucket における，出力ポートからのパケット送出間隔である．両図を比較する．まず図 12 が示すように，提案方式ではほぼ一定のパケット送出率であり，かつ送信レートは設定値の 50Mbps を示している．ただし，バースト到着時まではパケット到着間隔が 240 ミリ秒前後であり，これは 50Mbps シェーピングの最大出力速度に満たないため，到着パケットが即時送出されているのに対し，360–420 ミリ秒におけるバースト発生後は出力ポート側に蓄積されたパケットが残存するため最大出力速度を実現する 232 ミリ秒程度の間隔で送出が行われている．蓄積パケットはバースト発生時に最大となり，その後減少していくため，2100 ミリ秒付近で送出間隔は 240 ミリ秒に戻る．これに対し図 13 に示すように，Token Bucket ではパケット到着がバースト性を示す

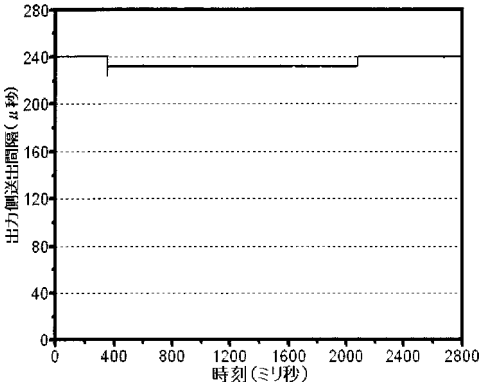

図 12. 提案方式の出力パケット送出間隔(1).

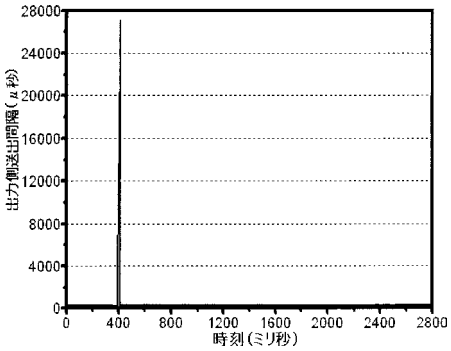

図 13. Token Bucket の出力パケット送出間隔(1).

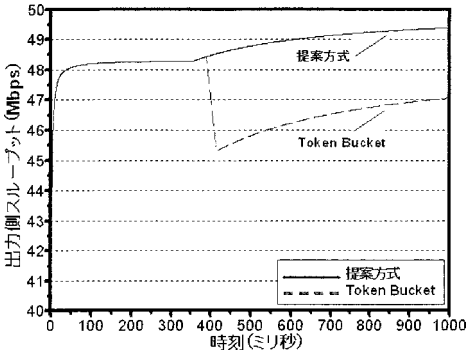

図 14. 出力側平均スループット(1).

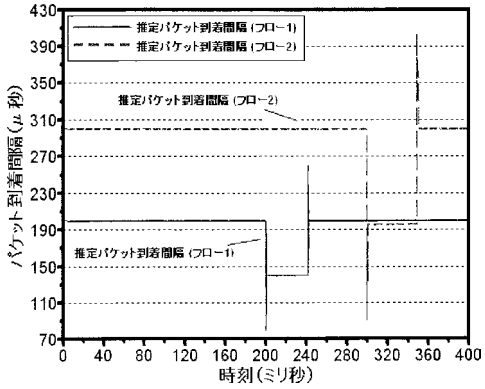

図 15. 推定パケット到着間隔(2).

360–420 ミリ秒付近において急激にパケット送出間隔が大きくなっていることがわかる．これは，入力ポートでのパケットロスにより，出力ポートへの到着パケット間隔が大きくなっているためである．結果として図 14 に示すとおり，シェーパーの出力ポートから送出されるトラヒックの平均スループットは，Token Bucket に比して提案方式ではスループットの劣化を抑制可能であることが確認された．

(2) 次に図 15 に，2 本の入力トラヒックに対するパケット到着間隔の推定値の結果を示す．ケーススタディ (1) と同様，バースト開始および終了付近(200 ミリ秒，242 ミリ秒，300 ミリ秒，349 ミリ秒) においては一時的に推定値に変動が発生することが確認できるが，それ以外の部分においては実測値と推定値はほぼ一致した．パケット長についても同様の結果を得たが図は割愛する．図 16 および図 17 はそれぞれフロー 1 (高優先) とフロー 2 (低優先) に対する提案方式および Token Bucket 適用時の入力ポートにおけるパケットロス率を示す．提案方式によるパケットロス率の低減効果が両フローについて確認できる．またフロー 1 におけるパケットロス率の低減の方が顕著であることが確認できる．この理由について，フロー 1 は高優先，即ちトークン残存量が低い場合においても高い確率でパケットの通過が許可されるため，パケットロス率は低減されるためと考えられる．

図 18 は入力ポートにおける提案方式によるトークン生成率の推移，図 19，図 20 はそれぞれ提案方式および Token Bucket におけるトークン残存量の推移を示す．これらの結果について

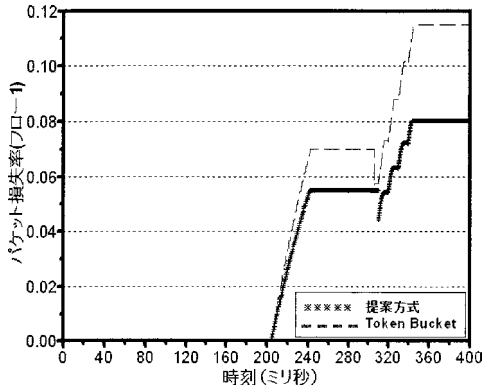

図 16. パケットロス率(フロー 1) (2).

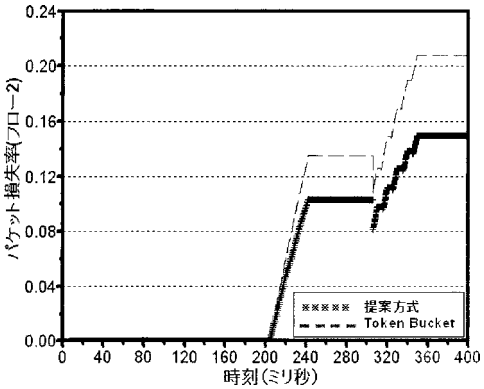

図 17. パケットロス率(フロー 2) (2).

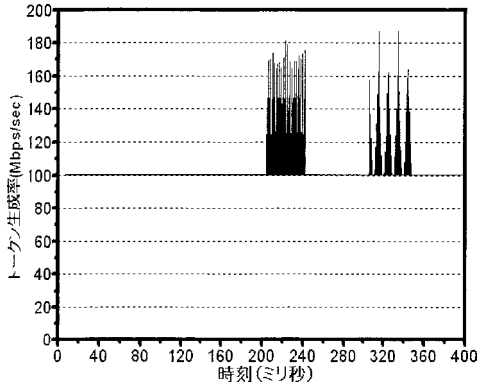

図 18. 提案方式によるトークン生成率(2).

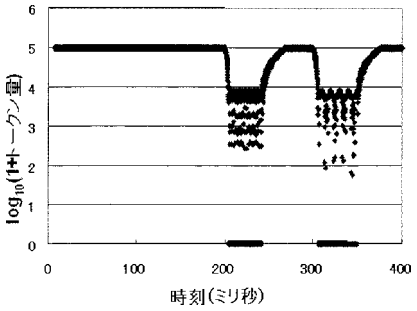

図 19. 提案方式におけるトークン残存量(2).

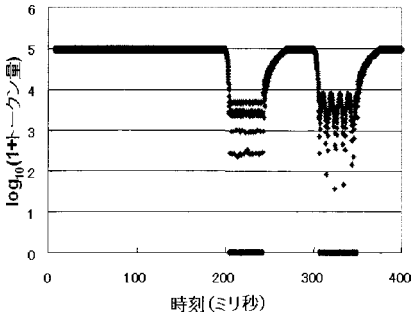

図 20. Token Bucket におけるトークン残存量(2).

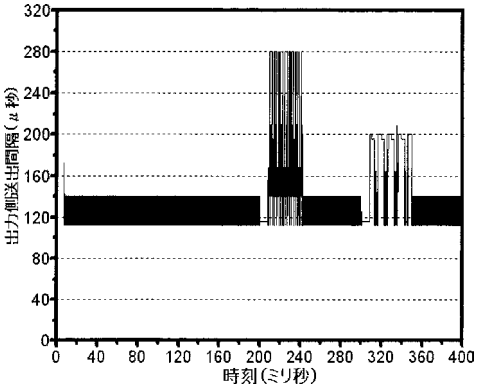

図 21. 提案方式の出力パケット送出間隔(2).

も，ケーススタディ(1)と同様，トークン残存量における提案方式の優位性とトークン生成率の動的な変化を確認する事ができる．

図 21 および図 22 はそれぞれ，提案方式および Token Bucket における，出力ポートでのパケット送出間隔である．ケーススタディ(1)と比較しバースト発生時におけるパケット送出間

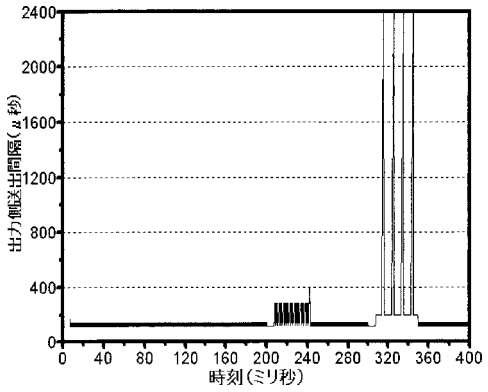

図 22. Token Bucket の出力パケット送出間隔(2).

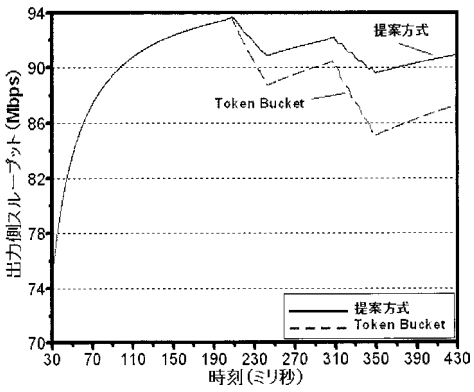

図 23. 出力側平均スループット(2).

隔は提案方式においても大きくなるが，Token Bucket に比してその増大は抑制されていることが確認される．出力ポートから送出されるトラヒックの平均スループットの比較を図 23 に示す．ここでも Token Bucket に比して提案方式ではスループットの劣化を抑制可能であることが確認された．ただし，ケーススタディ (1), (2) ともに出力ポートのキューにおけるパケットロスは発生していない．

4.3 考察

前節で得られた結果に対する考察を行う．まず，トラヒックシェーピングの意義と提案方式の位置付けについて整理する：

1. トラヒックシェーピングは映像配信，IP 電話等の高度のリアルタイム性が要求されるネットワークサービスにおいて，パケット送出時やネットワーク上の中継機器通過時に発生するパケット送出間隔のばらつきを平滑化し，サービス品質劣化を抑制するための技術である．そのため各フローにより使用される上限帯域の設定が必要であり，単一フロー・複数フローに関わらず，Token Bucket 等に見られる送受信レートの制御が行われる．即ち，所定の値を超過する入力パケットの廃棄による受信レートの抑制と，バッファリングされたパケットの送出間隔の調整による送信レート制御である．
2. 上記機能の実装形態として，通常入力ポートにおける受信レート制御と出力ポートにおける送信レート制御が行われる．前者は受信レートの上限設定および出力ポート側のバッファリングに起因するシェーパー内部での遅延の抑制効果を有し，後者は送信レートの制限とパケット送出間隔の平滑化効果を有する．
3. 一方 Token Bucket による入力ポートでの受信レート制御は，トラヒックのバースト到着時にパケットロスの増大を招く傾向があった．そこで本稿では，各時点におけるトラヒック予測と受信量に基づき柔軟にトークン生成率を調整し，パケットロス抑制とシェーピング機能を同時に実現する方式を提案した．また上記方式の提案に先立ち，トラヒックの同定手法に対する検討を行った．

提案方式によれば，単一トラヒック，複数トラヒックのいずれの場合においても，パケット到着間隔の予測に基づきトークン生成率を動的に調整するため，高いバースト性を示す入力トラヒックに対するパケットロス率の抑制とパケット送出間隔の平滑化の双方を実現可能である．但し，本稿で実施したケーススタディにおいては，入力ポートにおけるキューイングは行わないものとの前提に立つ．また本稿ではシェーパーの転送レートはシェーピングの帯域上限値に

比して充分に高いものとする．

ケーススタディにおいて，パケット到着間隔およびパケット長がガウス分布に従う各区間においては，シェーピング対象トラヒックフローが単一および複数の場合とも，提案方式による推定は比較的高い精度を実現した．また，この推定に基づき動的にトークン生成を行う提案方式の，Token Bucket に対する有効性を，入力ポートにおけるパケットロス率低減効果と，出力トラヒックのスループット維持の観点から確認した．また異なる優先度を有する複数の入力トラヒック存在時においても，フロー毎の推定に基づきトークン生成率を調整する提案方式の有効性を確認した．一方で，パケット到着間隔の分布が急激な変化を示す時間区間の境界においては局所的に推定値と実測値の間に誤差が生じることも確認された．これは状態方程式における残差分布が非ガウス性を有する事に起因し，解決策として非ガウス分布に対する有効性を有する他のフィルタリング方式，例えばモンテカルロフィルタ等の適用が必要であると考えている．採用するフィルタ方式についてはこれらの他にも複数考えられるが，ロバスト性の観点からは以前に筆者が採用した H^∞ フィルタとの比較が興味深い．一方で，H^∞ フィルタ適用時にはシステムが過度に冗長性を示す事も懸念されるため，H_2，H^∞ 混合ノルムによるフィルタの適用も考えられる．その他，Moving Horizon Estimator 等も考えられるが，実運用環境においては計算処理負荷の観点も考慮したフィルタリング方式の採用が望まれ，より一般的な入力トラヒックパターンに対する推定精度の向上と計算処理負荷軽減の両観点から提案方式の更なる向上を果たしていく必要があると考えている．

4.4 結論

本稿では，所与のトラヒックデータから，特にパケット到着間隔とパケット長の時系列データに着目して一期先トラヒック推定を行う手法と，これを利用したシェーピングアルゴリズムの提案を行った．該推定手法の適用により，トラヒックがガウス分布を示す条件の下では，パケット到着間隔およびパケット長の両方に対し，高い精度で推定を行うことができた．一方で，トラヒックが非ガウス分布を示す場合や，パケット長の変動が大きい場合における推定精度に対しては課題が残った．提案方式によるシェーピングアルゴリズムは，単一および複数のトラヒックフローの各場合において Token Bucket と比較しパケットロス率の低減効果が確認された．本検討では①多変数モデルの採用によりパケット到着間隔，パケット長の両方を推定対象としていること，②AR モデルの採用により一期以上前のパケットによる影響も考慮可能である点，③時変係数 AR モデルの採用により，長期的変動要因の推移のモデル化も可能である点，に着目しトラヒックのモデル化を行った．今後の課題として，①トークン生成率の設定とパケット到着レートがこれを超過する確率に対する考察，②より一般的な入力トラヒックパターンに対する有効なフィルタリング方式の比較，③バッファ長の考慮，④AR モデルにおける残差分布が非ガウス分布を示す場合における推定精度の向上，⑤計算処理負荷の軽減に対する検討，の必要性を認識している．

参 考 文 献

Chang, Chung-Ju, Yu, Chung-Hsun and Lin, Li-Fong (2004). Inteligent leaky bucket algorithm for sustainable-cell-rate usage parameter control in ATM networks, *IEEE Transactions on Multimedia*, **6**(5), 749–759.

Elwalid, Anwar and Mitra, Debasis (1977). Traffic shaping at a network node: Theory, optimum design, admission control, *IEEE INFOCOM'97*, **2**(7–11), 444–454.

Francini, Andrea and Chiussi, Fabio M. (2000). Minimum-latency dual-leaky-bucket shaping for

packet multiplexers: Theory and implementation, *IEEE Quality of Service, 2000. Eighth International Workshop*, **2**(7–11), 19–28.

Honda, Hirotada (2006). A method of traffic shaping using H^{∞} filtering, *SAINT 2006*, 10–13.

Ji, Yusheng (2001a). A study of traffic control and effect on QoS, *NII Journal*, **2**, 1–8.

Ji, Yusheng (2001b). Traffic control for quality of service in the information platforms, *NII Journal*, **3**, 23–33.

北川源四郎(2005).『時系列解析入門』, 岩波書店, 東京.

森田光茂, 大崎博之, 村田正幸(2001). インターネットにおけるパケット伝送遅延時間の測定およびシステム同定によるモデル化に関する検討, 電子情報通信学会技術研究報告, **SSE2001–243**, 55–62.

Proceedings of the Institute of Statistical Mathematics Vol. 55, No. 1, 177–195 (2007) 195

A Method of Traffic Shaping Based on a Traffic Identification and Estimation

Hirotada Honda

Faculty of Science and Technology, Keio University

This paper proposes a method of assimilation and estimation of packet arrival intervals and packet length. Based on the estimation method, we also propose a new traffic shaping algorithm, which enables reduction of packet loss at the traffic shaper, and maintains throughput. We adopt a multidimensional AR model with time varying coefficients to model the network traffic. In our model, the unknown variable consists of packet arrival intervals and length of arrival packets. A Kalman filter is applied to estimate the next packet's arrival time and length. Compared to the token bucket algorithm, the proposed method of traffic shaping reduces packet loss and maintains throughput by flexibly changing the token generation rate based on the estimated values. Using numerical simulations, we verify the effectiveness of the proposed method.

Key words: Time variant AR model, Kalman filter, traffic shaping.