

FIFO キューの packets 廃棄確率に対する IS 法を用いたオンライン推定

小林 信裕[†] 中川 健治^{††a)}

On-line Estimation by Importance Sampling for the Packet Loss Probability of FIFO Queue

Nobuhiro KOBAYASHI[†] and Kenji NAKAGAWA^{††a)}

あらまし 本論文では、FIFO キューの packets 廃棄確率を NS-2 を用いた IS シミュレーションによって推定する新たな方法を提案する。従来の IS 法のように packets の到着レートを上げるのではなく、処理レートを下げてキュー長を伸ばし廃棄事象を多く発生させる。それによって、各ノード内で IS 法を実行できるため、オンラインでの廃棄確率の推定が可能になる。MC 法と従来 IS 法及び提案 IS 法によるシミュレーション結果を比較し考察する。

キーワード FIFO キュー, packets 廃棄確率, IS シミュレーション, オンライン推定

1. ま え が き

インターネットにおいて最も多く利用されている FIFO (First In First Out) キューにおける packets 廃棄確率をネットワークシミュレータ NS-2 を用いて推定する。

NS-2 を用いて packets 廃棄確率シミュレーションを行う場合、モンテカルロ (Monte Carlo, MC) シミュレーション法によるのが一般的である。NS-2 に MC 法を適用すると、シミュレーションモデルの確率的構造を直接利用することができ、簡単にプログラムを作成することができる。しかし、ITU-T Rec. Y.1541 [3] で規定されている最も厳しいクラスの packets 廃棄確率である 10^{-5} 以下の小さい廃棄確率の場合、注目する packets 廃棄事象がなかなか発生せず、シミュレーションに時間がかかり、また推定値の信頼性も低い。

そこで、NS-2 によるシミュレーションの高速化と推定値の推定精度の向上のためにインポートランスサンプリング (Importance Sampling, IS) シミュレーション

法を適用する。IS 法は実際の確率分布と異なる確率分布を使用して小さい確率の事象をより多く発生させ、得られた値を補正して目標とする確率を得る方法である [1], [5]。筆者らはこれまでに、NS-2 による DRR (Deficit Round Robin) スケジューラの packets 廃棄確率推定に対して IS 法を適用し、その特性を評価した [4]。

通常、IS 法では、packets 廃棄事象が MC 法よりも多く発生するように到着レートを上げてシミュレーションを行い、得られた値を補正して真の廃棄確率に対する不偏推定値を得る。到着レートを適切な値に設定することによって、推定値の分散がもとの MC 法による推定値の分散よりも小さくなる。分散は発生する標本数に反比例するので、分散を同程度にすれば IS 法は MC 法よりも標本数が少なく済む。すなわち、シミュレーションが高速化される [1], [4]。

以上が IS 法の一般的な説明であるが、その IS 法の問題点について考えてみよう。上記のような IS 法では、到着レートを上げて廃棄事象が頻繁に発生するようにしているのであるが、より現実的なシミュレーション、あるいは実際のネットワークにおける packets 廃棄確率の評価において、到着レートを上げることが可能だろうか。ネットワーク全体をモデル化してシミュレーションをオフラインで考えれば可能である。

[†] (株) イートラスト, 長岡市
eTRUST, Nagaoka-shi, 940-0871 Japan

^{††} 長岡技術科学大学, 長岡市
Nagaoka University of Technology, Nagaoka-shi, 940-2188 Japan

a) E-mail: nakagawa@nagaokaut.ac.jp

しかし、ユーザが送信するトラフィックをそのまま利用してオンラインで推定を行う場合には到着レートを上げてユーザトラフィックを実際と異なるものにするにはできない。

ユーザトラフィックを変形しないでそのまま観測して高速なシミュレーションはできないだろうか。もし、それができればパケット廃棄確率を高速に推定して、得られた値を用いてリアルタイムに、例えば、パケット送信のレート制御に使用することができる。

IS法の重要な点はパケット廃棄事象を多く発生させることである。パケットの到着レートを増加させるのではなく、処理レートを低下させることによってキュー長を伸ばして廃棄事象を多くすることができる。この方法の特徴はユーザトラフィックを変化させず、ネットワーク側だけの処理によってIS法を実現できることである。具体的には、ネットワーク側では実際のパケット処理と並行して仮想的に遅いパケット処理を行ってキュー長を伸ばす。IS法では、統計量としてMC法よりも長くなったキュー長が得られればよいのだから、これをNS-2への付加的なカウンタとして実現する。

遅いパケット処理をするのに新たに別のタイマーを設けてパケット処理の時間を長くする方法はとらない。NS-2がevent drivenなシミュレータであることの特徴を生かして、FIFOキューへのパケット到着(enqueue)イベントとキューからの送出(dequeue)イベントのみを観測時点としてキュー長を変化させ、最適なISシミュレーションを行う。

そこで、本研究の目的を下記の3点とする。

(I) FIFOキューのパケット廃棄確率をNS-2を用いたISシミュレーションによって推定する。

(II) 到着レートを上げるのではなく処理レートを下げてキュー長をMC法よりも伸ばし廃棄事象を多く発生させる。それによってオンライン推定を可能にする。

(III) MC法とIS法によるシミュレーション結果を比較し、考察する。

2. 待ち行列に対するISシミュレーション

IS法を待ち行列に適用する一般的な方法について考える。定常状態におけるキュー長を Q とし、 Q がある値 q を超える確率 $P(Q > q)$ に対するIS法を考える。 $P(Q > q)$ をキュー長の裾確率という。 q が大きいと事象 $\{Q > q\}$ はなかなか発生しないので、元の分布 P と異なる別の分布 P' を用いて事象 $\{Q > q\}$ を多く

発生させて、重み関数によって補正して $P(Q > q)$ の推定値を得る[1]。 P' を用いてシミュレーションを行うので、 P' をシミュレーション分布という。

待ち行列におけるIS法では裾確率 $P(Q > q)$ の推定をサイクル単位で考える[2]。サイクルとはキュー長が正である一つの時間区間のことである。

2.1 キュー長の裾確率に対するIS推定値

元の分布 P によって生成されたパケットの処理終了時点を $t(1), t(2), \dots, t(j), \dots$ とする。シミュレーション分布 P' によって生成されたパケットによる時刻 $t(j)$ でのキュー長を $Q'(t(j))$ とすると、 $P(Q > q)$ に対するIS推定値 \hat{P}_{IS} は以下のように与えられる[2]。

$$\hat{P}_{IS} = \frac{1}{D} \cdot \frac{1}{M} \sum_{m=1}^M \sum_{j=1}^{\tilde{M}} 1_q(Q'(t(j))) W(j) \quad (1)$$

$$D = \frac{1}{\tilde{M}} \sum_{m=1}^{\tilde{M}} \sum_{j=1}^{\tilde{M}} W(j) \quad (2)$$

$$W(j) = \frac{P(Q'(t(1)), \dots, Q'(t(j)))}{P'(Q'(t(1)), \dots, Q'(t(j)))} \quad (3)$$

ここで、 M 及び \tilde{M} はサイクル数を表す。また、 $1_q(Q'(t(j)))$ は条件 $Q'(t(j)) > q$ が成り立つとき1、そうでないとき0となる関数、すなわち、事象 $\{Q' > q\}$ の指示関数を表す。 $W(j)$ は重み関数であり、 D は平均サイクル長のIS推定値である。(1)において \hat{P}_{IS} の $1/D$ を除く部分は1サイクル中でキュー長が q を超えている時間の長さのIS推定値である[2]。また、(1)の j に関する和は以下のようにする。待ち行列に対するIS法において P' によるトラフィック密度が1より大きい場合はISのキュー長 Q' が時間とともに発散してしまう。そこで、(1)において $j = 1, 2, \dots$ としていき、 $1_q(Q'(t(j))) = 1$ 、すなわち $Q'(t(j)) > q$ となる番号 j を適当な回数観測したら観測を打ち切ってサイクルを終了させる。ここで、適当な回数とは、統計量 $\sum_{j=1}^{\tilde{M}} 1_q(Q'(t(j))) W(j)$ が十分に収束したと考えられる回数である。経験的にこの回数は約20程度である。このように途中でサイクルを打ち切る方法をダイナミックISという[2],[4]。

時点 $t(j)$ は元の分布 P によって生成したパケットの処理終了時点であるが、 P' の選び方によっては $t(j)$ は Q' にとって特に意味のない時点となるかもしれない。しかし、 $\{t(j)\}_{j=1,2,\dots}$ が十分に密な時点列であれば、式(3)の重み関数 $W(j)$ 、すなわちゆう度比の計算は十分に精度が高いといえる。

2.2 最適シミュレーション分布

離散時刻 $t(j)$ におけるもとの分布 P によるキュー長を $Q(t(j))$, $j = 1, 2, \dots$ とする. $Q(t(j))$ が Markov 連鎖をなすとし, $Q(t(j-1)) > 0$ のとき i.i.d. (独立同分布) 確率変数 X によって

$$Q(t(j)) = Q(t(j-1)) + X, \quad j = 1, 2, \dots \quad (4)$$

と表されているとする. つまり, キュー長の確率は X によって決まる. このとき, \hat{P}_{IS} の分散 $V[\hat{P}_{\text{IS}}]$ を最小にする X^* は次のように与えられる [1].

X のモーメント母関数を $\varphi_X(\theta) \equiv E(e^{\theta X}) = \sum_x e^{\theta x} P(X = x)$ とし, $P(X = x)$ に対する指数測度変換を

$$P(X' = x) = \frac{e^{\theta x} P(X = x)}{\varphi_X(\theta)}, \quad \theta \in \mathbb{R} \quad (5)$$

と定義する. 次の定理が成り立つ [1].

[定理 A] $V[\hat{P}_{\text{IS}}]$ を最小にする確率変数 $X' = X^*$ は指数測度変換 $P(X^* = x) = e^{\theta^* x} P(X = x)$ によって与えられる. ここで, θ^* は方程式 $\varphi_X(\theta^*) = 1$, $\theta^* > 0$ の解である.

以下では, 分散 $V[\hat{P}_{\text{IS}}]$ を最小にするシミュレーション分布を「最適」シミュレーション分布という [6], [7].

2.2.1 M/M/1 の最適シミュレーション分布

定理 A を, 例えば, M/M/1 待ち行列に応用して最適シミュレーション分布を計算すると次のようになる [1].

到着率を λ とし, 処理率を μ とする. M/M/1 のキュー長変化を連続時間マルコフ連鎖として表現したものを一様化 [5] によって離散時間マルコフ連鎖とすると, 式 (4) の X は次のような確率変数となる.

$$X = \begin{cases} +1, & \text{at prob. } \bar{\lambda} \\ -1, & \text{at prob. } \bar{\mu} \end{cases} \quad (6)$$

ここで, $\bar{\lambda} = \lambda/(\lambda + \mu)$, $\bar{\mu} = \mu/(\lambda + \mu)$ である. X のモーメント母関数は $\varphi_X(\theta) = \bar{\lambda}e^{\theta} + \bar{\mu}e^{-\theta}$ となるので, $\varphi_X(\theta^*) = 1$, $\theta^* > 0$ を解いて $\theta^* = \log(\bar{\mu}/\bar{\lambda})$ となる. したがって, 最適な X^* は定理 A の $P(X^* = x) = e^{\theta^* x} P(X = x)$ より

$$X^* = \begin{cases} +1, & \text{at prob. } \bar{\mu} \\ -1, & \text{at prob. } \bar{\lambda} \end{cases} \quad (7)$$

となる. つまり, 元の到着率 λ と処理率 μ に対して, 最適シミュレーション分布の到着率は $\lambda^* = \mu$, 処理率

は $\mu^* = \lambda$ となり, 到着率と処理率を入れ替えたものとなる. 元のトラフィック密度は $\rho = \lambda/\mu < 1$ なので, 最適シミュレーション分布では $\rho^* = \mu/\lambda > 1$ となる. つまり, 不安定なキューになりキュー長 Q^* は時間とともに発散してしまう. そこで, 事象 $\{Q^* > q\}$ を十分に多く観測したら 2.1 で述べたダイナミック IS によってサイクルを強制的に打ち切る.

3. FIFO キューのモデル化

本章では, FIFO キューへのケット到着と Byte 単位でのケット処理をモデル化する.

3.1 ペケット到着のモデル化

キューに到着するトラフィックには K 種類のケット長 l_k [Byte], $k = 1, \dots, K$ のケットが含まれているとする. ペケット長 l_k のケットの到着レートを Λ_k [Byte/s] とする. フロー数は特に考慮せず, すべてのフローを多重化した結果のトラフィックにおけるケット長 l_k のケットの到着レートを Λ_k と考えればよい. ペケット長 l_k のケットは到着率 Λ_k/l_k [packet/s] のポアソン過程に従うとする.

3.2 ペケット処理のモデル化

到着したケットはサイズ ∞ のバッファに格納され, 到着順に処理される. 処理レートを μ [Byte/s] とする. 本論文では, 有限のバッファサイズ q をもつ FIFO キューにおけるケット廃棄率を, バッファサイズ無限大の FIFO キューにおけるキュー長の裾率 $P(Q > q)$ で近似する. そこで, 本論文では便宜上, $P(Q > q)$ をケット廃棄率と呼ぶことにする.

以上の定義に基づいて, FIFO キューはパラメータ

$$S = (\Lambda_k, \mu)_{k=1, \dots, K} \quad (8)$$

によって規定されると考える.

3.3 FIFO キューにおけるキュー長変化の確率的構造

時刻 t においてキュー内にあるケット長 l_k の全ケットの合計バイト数を $Q_k(t)$ [Byte], $k = 1, \dots, K$ とし, $Q(t) = \sum_{k=1}^K Q_k(t)$ [Byte] とおく. ペケットの処理終了時刻を $t_d(1), t_d(2), \dots, t_d(j), \dots$ とする. ここで, 添字の d はキューからの送出 (dequeue) を表す. 時刻 $t_d(1)$ に処理が終了するケットの処理開始時刻を $t_d(0)$ とおく.

時間区間 $t_d(j-1) < t \leq t_d(j)$ に処理されるケットのケット長を $L(j)$ [Byte] とする. そして, その時間区間 $t_d(j-1) < t \leq t_d(j)$ に到着するケット長 l_k の

パケットの個数を $n_k(j)$ 個とし、 $A(j) = \sum_{k=1}^K n_k(j)l_k$ とおく。 $A(j)$ は時間区間 $t_d(j-1) < t \leq t_d(j)$ に到着する全バイト数を表す。このとき、次の漸化式が成り立つ；

$$Q(t_d(j)) = Q(t_d(j-1)) + A(j) - L(j), \quad j = 1, 2, \dots \quad (9)$$

ここで、キュー長差 $Q(t_d(j)) - Q(t_d(j-1)) = A(j) - L(j)$ について考える。変数 j を省略して、 $A = A(j)$ 、 $L = L(j)$ として、 $X = A - L$ とおく。また、最大パケット長を $l_{\max} = \max(l_1, \dots, l_K)$ とする。 X の確率 $P(X = x)$ 、 $x \geq -l_{\max}$ 、及び X のモーメント母関数 $\varphi(\theta)$ を求める。

バッファサイズ $= \infty$ としているので、到着したパケットは廃棄されず必ずサーバで処理される。したがって、処理されるパケットの割合はパケット到着率 Λ_k/l_k に比例する。そこで、 $\pi_k = P(L = l_k)$ とおくと、

$$\pi_k = \frac{\Lambda_k/l_k}{\sum_{k'=1}^K \Lambda_{k'}/l_{k'}}, \quad k = 1, \dots, K \quad (10)$$

を得る。長さ l_k のパケットに対する処理時間を $\tau_k = l_k/\mu$ [s] とおく。長さ τ_k の時間区間におけるパケット長 $l_{k'}$ のパケットの到着率 $\lambda_{kk'}$ は

$$\lambda_{kk'} = \tau_k \frac{\Lambda_{k'}}{l_{k'}} = \frac{l_k}{\mu} \frac{\Lambda_{k'}}{l_{k'}} \quad (11)$$

となる。更に、

$$f(\theta) = -\theta + \frac{1}{\mu} \sum_{k=1}^K \frac{\Lambda_k}{l_k} (e^{l_k \theta} - 1) \quad (12)$$

とおく。また、記号の簡単化のために、ポアソン確率を $R(n; \lambda) = \lambda^n e^{-\lambda}/n!$ と略記し、 $n \cdot l = \sum_{k=1}^K n_k l_k$ と書く。

以上の記号の下で、文献 [4] と同様にして、

$$P(X=x) = \sum_{k=1}^K \pi_k \sum_{n \cdot l = x + l_k} \prod_{k'=1}^K R(n_{k'}; \lambda_{kk'}), \quad (13)$$

$$x \geq -l_{\max},$$

$$\varphi_X(\theta) = \sum_{k=1}^K \pi_k \exp(f(\theta)l_k) \quad (14)$$

を得る。そして、

$$\varphi_X(\theta^*) = 1, \quad \theta^* > 0 \quad (15)$$

を満たす $\theta = \theta^*$ が唯一つ存在することが分かる [4]。

3.4 FIFO キューの最適シミュレーション分布

FIFO キューのパケット廃棄確率に対する IS 推定値 \hat{P}_{IS} の分散を最小にするシミュレーション分布、すなわち最適シミュレーション分布を決定する。それは文献 [4] と同様にして

$$P(X^*=x) = \sum_{k=1}^K \pi_k \sum_{n \cdot l = x + l_k} \prod_{k'=1}^K R(n_{k'}; \lambda_{kk'}^*), \quad (16)$$

$$x \geq -l_{\max},$$

$$\lambda_{kk'}^* = e^{l_{k'} \theta^*} \lambda_{kk'}, \quad k, k' = 1, \dots, K \quad (17)$$

となることが分かる。

4. オンライン IS 推定

さて、ここで、(17) について考える。元の FIFO キューのキュー長変化規則 X に関するシステムパラメータを $S = (\Lambda_k, \mu)_{k=1, \dots, K}$ とし、IS 法の最適なキュー長変化規則 X^* に関するシステムパラメータを

$$S^* = (\Lambda_k^*, \mu_k^*)_{k=1, \dots, K} \quad (18)$$

とする。(11) と同様に

$$\lambda_{kk'}^* = \frac{l_k}{\mu_{k'}} \frac{\Lambda_{k'}}{l_{k'}} \quad (19)$$

となるので、(11)、(17)、(19) より、

$$\frac{\Lambda_k^*}{\mu_k^*} = e^{l_k \theta^*} \frac{\Lambda_k}{\mu}, \quad k = 1, \dots, K \quad (20)$$

を得る。IS 法において、到着レート Λ_k^* [Byte/s] と処理レート μ_k^* [Byte/s] が (20) の関係を満たしていれば、(18) の S^* は最適なパラメータである。(20) によるパラメータ Λ_k^* 、 μ_k^* の決定には任意性がある。特に、 $\Lambda_k^* = e^{l_k \theta^*} \Lambda_k$ 、 $\mu_k^* = \mu$ としたのが、文献 [4] である。[4] は、多くの IS 法と同様に到着レートを増加させてキュー長を伸ばす方法である。本論文では、

$$\Lambda_k^* = \Lambda_k, \quad \mu_k^* = e^{-l_k \theta^*} \mu, \quad k = 1, \dots, K \quad (21)$$

として IS 法を実行する。(21) によるパラメータ設定の特徴は、 $\Lambda_k^* = \Lambda_k$ 、すなわちユーザトラフィックを変化させないということである。そして、 $\mu_k^* = e^{-l_k \theta^*} \mu$ として処理レートを元の μ より小さくしてキュー長を伸ばすのである。

IS 法におけるキュー長を IS キュー長 (IS queue length) という。特に最適なパラメータ (21) による IS キュー長を最適な IS キュー長といい、 Q^* と表す。 Q^*

を計算するためのカウンタを以下のように提案する。

4.1 オンラインでの IS キュー長の計算

前節でも述べたように、本論文での IS シミュレーションの特徴は、到着レートを変えずに処理レートをお小さくしてキュー長を伸ばし廃棄事象を多く発生させることである。ユーザトラヒックを変えずに処理を変えるということは、IS 法を全てネットワーク側の処理の変更だけで実現できるということである。したがって、従来の IS 法と異なり、IS 法をオンラインで実行できる。

処理レートを小さくするためにすぐに考えられる方法は、処理時間を管理する新たなタイマーを設けて packets をゆっくり処理することだが、新たなタイマーを NS-2 に実装するとシステムが複雑になる。そこで、本研究では、NS-2 が event driven で動作するシミュレータであるという特徴を利用して、最適な IS キュー長 Q^* を簡単に計算するカウンタを提案する。

4.2 ISQL カウンタとそのアルゴリズム

最適な IS キュー長を計算するために以下のカウンタを用意する。

Q^* [Byte]	最適な IS キュー長を表すカウンタ
$C_k, k = 1 \dots, K$	パケット処理を遅くするためのカウンタ

これらのカウンタの組 $\{Q^*, C_k, k = 1, \dots, K\}$ を総称して ISQL カウンタ (IS Queue Length counter) という。

ISQL カウンタについて説明する。FIFO キューへ packets が到着する時刻を $t_e(1), t_e(2), \dots, t_e(i), \dots$ とし、packets が処理されて FIFO キューから送出される時刻を $t_d(1), t_d(2), \dots, t_d(j), \dots$ とする。添字の e, d は、それぞれ enqueue, dequeue の頭文字である。また、時刻 t に到着する、あるいは送出される packets の index を返す関数を $\text{PktInd}(t)$ とする。ここで、packets の index とは、packets の長さが l_k のときの添字 k のことである。

ISQL カウンタの動作アルゴリズムの擬似コードと図を下記に示す。

ISQL カウンタの動作アルゴリズム

```

1  at time 0
2   $Q^* = 0$ ;
3   $C_k = 0, k = 1, \dots, K$ 
4  at enqueue time  $t_e(i)$ 

```

```

5  let  $k = \text{PktInd}(t_e(i))$ ;
6   $Q^* = Q^* + l_k$ 
7  at dequeue time  $t_d(j)$ 
8  let  $k = \text{PktInd}(t_d(j))$ ;
9   $C_k = C_k + 1$ 
10 If  $C_k \geq \exp(l_k \theta^*)$ 
11   then  $Q^* = Q^* - l_k$ ;
12    $C_k = C_k - \exp(l_k \theta^*)$ 

```

上記の動作アルゴリズムについて説明する。まず、基本的なこととして、この動作は IS キュー長 Q^* に関するサイクル単位で実行されている。つまり、 $Q^* = 0$ のときを時間原点として、 $Q^* > 0$ である期間、すなわちサイクルを単位として推定値が得られる。

第1~3行において、時刻0でISキュー長 Q^* とカウンタ C_k が初期化される。第4~6行で、FIFOキューにpacketsが到着する時刻(enqueue time) $t_e(i)$ において、到着したpacketsのpackets長 l_k がISキュー長 Q^* に加算される。このpackets到着時の動作は、FIFOキューのキュー長に対する動作と同じである。次に、FIFOキューからpacketsが送出される時刻(dequeue time) $t_d(j)$ において、送出されたpacketsに対応するカウンタ C_k が1加算される(第9行)。そして、カウンタ値 C_k が $\exp(l_k \theta^*)$ 以上となったときに(第10行)にISキュー長 Q^* から対応する1packets分のByte数 l_k が減算され(第11行)、 C_k から $\exp(l_k \theta^*)$ が減算される(第12行)。第9~12行の動作は、カウンタ C_k が1ずつ増加してゆき、 $\exp(l_k \theta^*)$ 以上になったときに Q^* からpackets長 l_k が送出されるので、長さ l_k のpacketsに対する処理時間が約 $\exp(l_k \theta^*)$ 倍になること、言い換えると処理レートが約 $\exp(-l_k \theta^*)$ 倍になることを意味する。この効果によってISキュー長 Q^* が伸びるのである。 $\exp(l_k \theta^*)$ が整数でないときには第12行の減算で端数が生じるが、その端数は次回に繰り越されるので、長時間で考えれば長さ l_k のpacketsはほぼ $\mu^* = \mu \exp(-l_k \theta^*)$ のレートで処理されることになる。ここで注意すべきことは、ISキュー長 Q^* からのpackets長 l_k の送出は必ずしも到着順には行われていないことである。すなわち、 Q^* の中ではpacketsの個性は失われ、全packetsの合計Byte数のみが Q^* によって管理されている。このことを図1の Q^* においてpackets間の点線で示している。我々の目的は事象 $\{Q > q\}$ の確率を得ることなので、packetsの処理順は結果に影響を

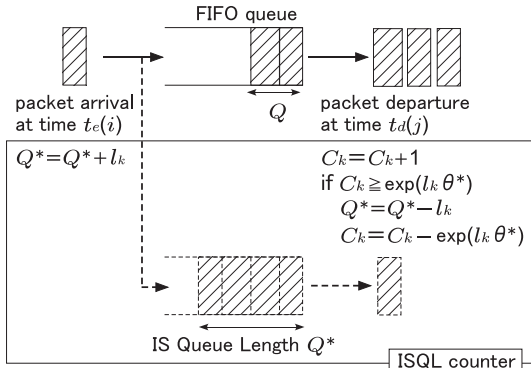


図1 ISQL カウンタによる処理レートの低減と IS キュー長

Fig. 1 Reduction of service rate and IS queue length by ISQL counter.

与えない。

4.3 ISQL カウンタによる IS 推定値

式 (1), (2), (3) のキュー長 Q' として ISQL カウンタで得られた最適な IS キュー長 Q^* を用いて、パケット廃棄確率 $P(Q > q)$ に対する IS 推定値 \hat{P}_{IS} は [4] と同様にして以下のように与えられる。

$$\hat{P}_{IS} = \frac{1}{D} \cdot \frac{1}{M} \sum_{m=1}^M \sum_{j=1}^M 1_q(Q^*(t_d(j))) W(j) \quad (22)$$

$$D = \frac{1}{M} \sum_{m=1}^M \sum_{j=1}^M W(j) \quad (23)$$

$$W(j) = \exp(-\theta^* Q^*(t_d(j))) \quad (24)$$

(22), (23) の j に関する和は 2.1 の最後の部分に述べたようにシミュレーションの途中でサイクルを打ち切るダイナミック IS 法によって行う。

5. シミュレーションによる比較評価

幾つかの条件のもとで、FIFO キューのパケット廃棄確率に対する推定値の精度、得られる推定値の限界、シミュレーション時間について考察する。比較するシミュレーション法は、MC 法、到着レートを大きくする従来 IS 法 (conventional IS) 及び処理レートを小さくする本論文での提案 IS 法 (proposed IS) である。ここで、従来 IS 法とは、文献 [4] で作成した DRR に対する IS 法を FIFO キューに対応するように修正したものである。

特性の比較は、提案 IS 法がもつ IS 法としての基本的特性が従来 IS 法と同等であることを確認するもの

表 1 パケット長に関するパラメータ
Table 1 Parameters on packet length.

異なるパケット長の種類	$K = 1$
パケット長 [Byte]	$l = 128$

表 2 FIFO キュー及び従来 IS 法のパラメータ
Table 2 Parameters on FIFO queue and conventional IS.

	case 1	case 2	case 3	case 4
到着レート Λ [kByte/s]	12.5	50.0	75.0	100
処理レート μ [kByte/s]	125	125	125	125
トラフィック密度 Λ/μ	0.1	0.4	0.6	0.8
最適な到着率 Λ^* [kByte/s]	464.4	252.5	193.5	154.0

表 3 ISQL カウンタのパラメータ
Table 3 Parameters on ISQL counter.

	case 1	case 2	case 3	case 4
θ^*	2.82	1.26	7.40	3.37
	$\times 10^{-2}$	$\times 10^{-2}$	$\times 10^{-2}$	$\times 10^{-2}$
最適処理レート $\mu^* = \mu e^{-l\theta^*}$ [kByte/s]	3.36	24.76	48.47	81.24
処理レートの比 μ/μ^*	37.15	5.05	2.58	1.54
トラフィック密度 Λ/μ^*	3.72	2.02	1.55	1.23

であって、従来 IS 法よりもこれらの特性において優れているということを主張するものではない。本論文で提案するオンライン IS 法の特徴は、実際に稼働しているネットワークのトラフィックをそのまま利用してパケット廃棄確率の高速な推定ができることであるが、特徴である「オンライン」ということを数値的に主張することは難しい。本章では、したがって、提案 IS 法の IS 法としての基本的性能を評価するものである。

5.1 シミュレーション 1

表 1, 表 2 に示す条件でシミュレーションを行う。すべてのパケット長が等しく 128 [Byte] である。到着レートを 4 種類変えて調べる。表 3 に ISQL カウンタのパラメータを示す。表 3 において処理レートの比 μ/μ^* が、例えば case1 では 37.15 となっているが、これは図 1 において、元の処理速度の $1/37.15$ のゆっくりした速度でパケットを処理するのが最適であることを表している。また、case1 では元のトラフィック密度 Λ/μ が表 2 より 0.1 であり、最適なトラフィック密度 Λ/μ^* が表 3 より 3.72 である。つまり $\Lambda/\mu^* > 1$ なので Q^* は不安定である。

図 2 にパケット廃棄確率の推定値を示す。MC 法ではサイクル数を 10^7 とし、従来 IS 法及び提案 IS 法で

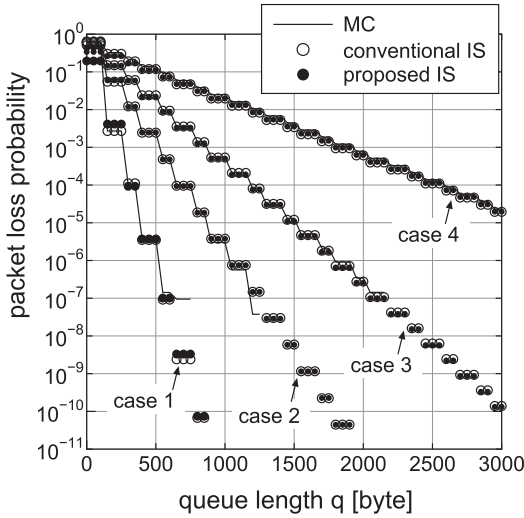


図2 MC法, 従来IS法, 提案IS法による推定値の比較
Fig. 2 Comparison of estimates by MC, conventional IS and proposed IS.

表4 パケット長に関するパラメータ
Table 4 Parameters on packet length.

異なるパケット長の種類	$K = 2$
パケット長 [Byte]	$l_1 = 128, l_2 = 256$

表5 FIFO キュー及び従来IS法のパラメータ
Table 5 Parameters on FIFO queue and conventional IS.

	case 1	case 2	case 3	case 4
到着レート Λ_1 [kByte/s]	0.714	1.43	2.14	2.86
到着レート Λ_2 [kByte/s]	24.3	48.6	72.9	97.1
処理レート μ [kByte/s]	125	125	125	125
トラフィック密度 $(\Lambda_1 + \Lambda_2)/\mu$	0.2	0.4	0.6	0.8
最適な到着レート Λ_1^* [kByte/s]	2.73	3.25	3.45	3.55
最適な到着レート Λ_2^* [kByte/s]	355.8	249.3	190.3	150.5

は 10^4 とした。この条件で MC 法と両方の IS 法の標本分散が同程度となる。図 2 より MC 法では約 10^{-6} が推定値の限界であるが、従来 IS 法と提案 IS 法とも限界なく推定値が得られている。

5.2 シミュレーション 2

表 4, 表 5 にシミュレーション条件を示す。このシミュレーションでは、 $K = 2$, すなわち 2 種類の異なる長さのパケットが存在する場合である。ISQL カウンタのパラメータを表 6 に示す。表 6 において、例えば、

表 6 ISQL カウンタのパラメータ
Table 6 Parameters on ISQL counter.

	case 1	case 2	case 3	case 4
θ^*	1.04 $\times 10^{-2}$	6.39 $\times 10^{-3}$	3.74 $\times 10^{-3}$	1.71 $\times 10^{-3}$
最適処理レート $\mu_1^* = \mu e^{-l_1 \theta^*}$ [kByte/s]	32.67	55.18	77.42	100.44
最適処理レート $\mu_2^* = \mu e^{-l_2 \theta^*}$ [kByte/s]	8.54	24.35	47.95	80.71
処理レートの比 μ/μ_1^*	3.83	2.27	1.61	1.24
処理レートの比 μ/μ_2^*	14.64	5.13	2.61	1.55
トラフィック密度	2.53	1.92	1.51	1.22

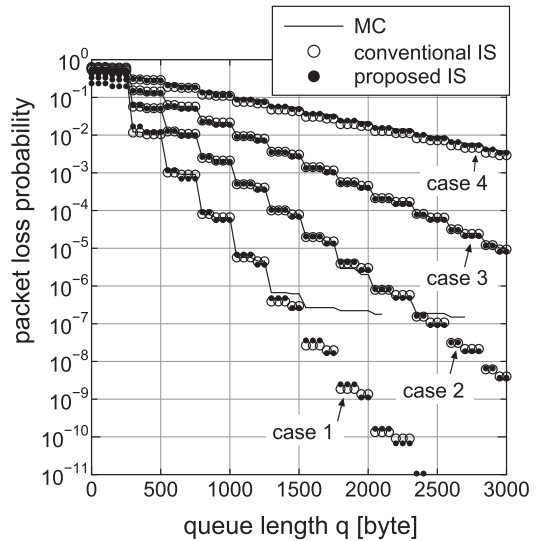


図3 MC法, 従来IS法, 提案IS法による推定値の比較
Fig. 3 Comparison of estimates by MC, conventional IS and proposed IS.

case1 では最適な処理レートが $\mu_1^* = 32.67$ [kByte/s], $\mu_2^* = 8.54$ [kByte/s] となり、短いパケット長と長いパケット長とで最適な処理レートが異なることに注意する。パケット廃棄確率に対する推定値を図 3 に示す。図 3 から提案 IS 法は従来 IS 法とほぼ同程度の特性をもつことが分かる。従来 IS 法と提案 IS 法の推定値にわずかな誤差があるが、この誤差の理由について考える。提案 IS 法では処理レートを下げる方法としてタイマーを設けるのではなく FIFO キューにおけるパケット到着、送出手のタイミングで IS キュー長 Q^* を計算している。タイマーなら完全に最適な処理レートを実現できるが、提案 IS 法では簡易的な方法なので、それによる誤差と考えられる。

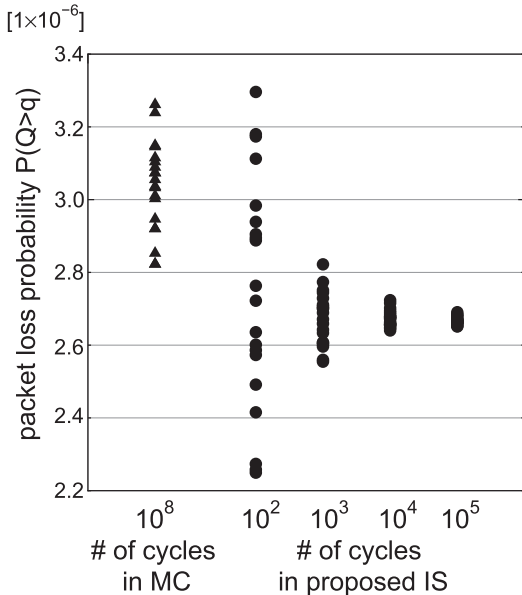


図4 MC法と提案IS法による推定値のばらつき比較
Fig. 4 Comparison of dispersion of estimates by MC and proposed IS.

5.3 シミュレーション3 [高速性能評価]

次に、MC法と提案IS法のシミュレーション時間を比較する。前述のシミュレーション2におけるcase2で $q = 2000$ [Byte]のときのパケット廃棄確率を推定するのにかかる時間を比較する。公平な比較を行うために次の手順による。まず、MC法において1回のシミュレーションでのサイクル数を 10^8 として $P(Q > q)$ に対する1個の推定値を得る。同じことを繰り返して、20個の推定値を得て、それらの標本分散を計算する。次に提案IS法でサイクル数を $10^2, 10^3, 10^4, 10^5$ としてそれぞれ20個の推定値を得て標本分散を計算する。最初にMC法で得られた分散に最も近い分散を与えるサイクル数を選ぶ。この過程を示したのが図4である。図4よりMC法の 10^8 サイクルに対して提案IS法の 10^3 サイクルのときがほぼ同等の分散となることが分かる。このようにしてそれぞれのシミュレーションのサイクル数を決定して実行した結果を表7に示す。推定値の平均と分散がほぼ等しいので同程度の推定精度といえる。このときのシミュレーション時間は、表7よりMC法では379.8秒であり、提案IS法では0.093秒となった。したがって、この場合、約4000倍の高速化が得られている。

表7 MC法と提案IS法のシミュレーション時間の比較
Table 7 Comparison of simulation time by MC and proposed IS.

	MC	proposed IS
サイクル数	10^8	10^3
平均推定値	3.03×10^{-6}	2.67×10^{-6}
標本分散	1.47×10^{-14}	5.01×10^{-15}
シミュレーション時間	379.8 [s]	0.093 [s]

6. むすび

本論文では、FIFOキューのパケット廃棄確率をオンラインで高速に推定するISシミュレーション法を提案した。

従来のIS法ではキューへの入力トラフィックをモデル化し、モデルの到着レートを上げてパケット廃棄事象を多く発生させる。しかし、その方法では実際のキューとは別なシミュレーションになってしまう。

本研究では、ユーザからのパケットトラフィックをそのまま利用して、処理レートを下げることによって廃棄事象を増やす新たなIS法を提案した。それによってMC法と同様にオンラインでの推定ができ、更にMC法よりも高速に推定を実行できる。実際に、提案するIS法のシミュレーションプログラムをNS-2で作成した。オンライン推定のために「ISQLカウンタ」をFIFOキューに置き、NS-2のevent drivenなシミュレータとしての特徴を生かした方法でパケット処理レートを小さくしてキュー長を伸ばすアルゴリズムを作成した。幾つかのシミュレーションを行い、得られた推定値の精度、推定限界、高速性について評価した。

今後は、提案したIS法を用いて高速にリアルタイムのパケット廃棄確率を推定し、その値をすぐにレート制御のために使うことを検討していく。

文 献

- [1] J.A. Bucklew, Large Deviation Techniques in Decision, Simulation, and Estimation, Wiley, New York, 1990.
- [2] M. Devetsikiotis and J.K. Townsend, "Statistical optimization of dynamic importance sampling parameters for efficient simulation of communication networks," IEEE/ACM Trans. Netw., vol.1, no.3, pp.293-305, June 1993.
- [3] ITU-T Recommendation Y.1541, <http://www.itu.int/rec/T-REC-Y.1541-200602-I/en>
- [4] 和泉光紀, 中川健治, 横谷哲也, "DRRスケジューラのパケット廃棄確率算出におけるNS-2へのインポートランスサンプリングシミュレーションの適用について," 信学論

(B), vol.J94-B, no.5, pp.698-707, May 2011.

- [5] 中川健治, “モンテカルロシミュレーション基礎—推定精度評価の問題点とその克服,” 信学通誌, no.6, 秋号, pp.11-20, 2008.
- [6] 小川耕司, 中川健治, “MMPP/D/1 キューイングにおける最適 IS シミュレーション分布,” 信学論 (B-I), vol.J80-B-I, no.2, pp.64-73, Feb. 1997.
- [7] J.S. Sadowsky, “Large deviations theory and efficient simulation of excessive backlogs in a GI/GI/m queue,” IEEE Trans. Autom. Control., vol.36, no.12, pp.1383-1394, 1991.

(平成 23 年 9 月 8 日受付, 12 月 27 日再受付)



小林 信裕

平 21 長岡技科大・工・電気電子情報卒.
平 23 同大学院修士課程了. 同年(株)
イトラスト入社.



中川 健治 (正員)

昭 55 東工大・理・数学卒. 昭 60 同大大
学院博士課程満期退学. 昭 60 NTT 研究
所入社. 情報理論, ATM 容量設計等の研
究に従事. 平 4 長岡技科大工学部助教授,
平 19 同准教授, 平 24 同教授. ネットワー
ク特性評価, 待ち行列理論, 大偏差理論等
の研究に従事. 理博. IEEE, 日本 OR 学会, 日本数学会, 日本
工業教育協会各会員.