

[奨励論文] 高速ネットワークにおける TCP パケットロス監視技術

山崎 康広 下西 英之 村瀬 勉

NEC システムプラットフォーム研究所

〒211-8666 神奈川県川崎市中原区下沼部 1753

E-mail: {y-yamasaki@ay, h-shimonishi@cd, t-murase@ap}.jp.nec.com

あらまし 近年、回線速度の高速化に対応できるネットワーク監視技術として、パケットサンプリング技術が注目されてきている。これまでのサンプリング技術の研究では、ネットワーク品質の重要な指標であるパケットロス率の計測に関する議論はなかった。この問題に対して筆者らは TCP の振る舞いに着目することで、サンプリングで観測できたパケットロスの一部から、全パケットロス量を推定する手法を提案してきた。提案手法では、推定パケットロス率が推定式の反復計算により真のパケットロス率に収束していくことを、シミュレーション評価・実機評価により経験的に示し、その有効性を確認している。本稿では、提案アルゴリズムの数学的意味を明確にすることにより、推定アルゴリズムが真のパケットロスに収束することを解析的に明らかにする。さらにシミュレーションにより、反復過程を明らかにする。これにより、ほとんどの場合数回程度の反復計算で真のパケットロス率を推定できることが分かった。

キーワード ネットワーク性能 サンプリング計測、パケットロス率、TCP 性能

TCP Packet Loss Measurement Technique in the High-Speed Network Environment

Yasuhiro YAMASAKI Hideyuki SHIMONISHI and Tutomu MURASE

System Platforms Research Laboratories, NEC Corporation

1753 Shimonumabe Nakahara, Kawasaki, Kanagawa, 105-0123 Japan

E-mail: {y-yamasaki@ay, h-shimonishi@cd, t-murase@ap}.jp.nec.com

Abstract Recently, as a monitoring technology flexibly applicable to the high-speed network, packet sampling technology has attracted attention. However, in the previous studies, there is little argument on the packet loss rate that is an important index of the network. Taking TCP flow level into consideration, we have proposed new method that can estimate all packet losses through a part of observed packet losses from sampled packets. At the same time, we have confirmed the validity of this method through simulation and proto-typing. However, previous estimation algorithm shows experimentally that estimation packet loss is equal to true packet loss. In this study, we analytically show that estimation algorithm can converge to true packet loss. And we show the limitations of sampling performance by the converging conditions.

Keyword Network Performance, Sampling Measurement, Packet loss rate, TCP performance

1. はじめに

近年、ネットワークの監視技術では、回線速度の向上に伴い、ネットワーク監視装置の高速化への対応が問題になってきている[1]。従来のネットワーク監視技術は、全通過パケットを監視する必要があり、高速ネットワークの監視が非常に難しい高コストになるという問題がある。

この問題を解決する技術として、サンプリング計測技術が注目を浴びている。サンプリング計測技術とは、一部のパケットをサンプリング計測し、精度良い推定を行うことにより、計測コストの軽減並びに高速化を実現する技術である。この技術では、サンプリング計測時に通過パケット量を推定する手法[2]、サンプル数と統計結果の信頼度の評価[3][4]、リンク帯域の占有率の大きいフローを特定する手法[5][6]、やフロー発生数やフローサイズに関する統計情報を推定する方法[7][8]、TCP フローレベルの性能劣化を検出する方法[9][10]等が提案されている。しかしながら、ネットワーク管理の最も重要な指標の一つである、パケットロス率については議論されていなかった。

そこで著者らは、TCP の振る舞いに着目することにより、サンプリング計測時にもパケットロス率を監視可能な方式を提案している[11][12]。これにより今まで実現できなかった高速ネットワーク環境でのパケットロス率の把握が可能となる。

これまで提案手法では、推定パケットロス率が推定式の反復計算により、真のパケットロス率に収束していくことを、シミュレーション評価や実装評価等で経験的に示してきた。本稿では、提案アルゴリズムの数学的意味を明確にすることにより、この反復計算が、常に真のパケットロス率に収束するための条件を解析的に明らかにする。そしてシミュレーションにより、収束するまでの反復過程をより詳細に確認・評価する。また、収束条件からもとまるサンプリングレートの条件を明らかにする。

以下、2章では提案方式の数学的意味を明確にして推定式の再導出を行い、3章では推定式が真のパケットロス率に収束するための条件を示す。この条件から4章で、サンプリングレートの条件を算出。5章で反復回数の評価を行い、6章でまとめを行う。

2. 提案手法

提案しているパケットロス率推定手法[11][12]の概要について説明する。TCP通信ではパケットロスが発生すると、同一ACK番号が連続して発生するという性質がある(通常、重複ACKと呼ばれる)。この性質を利用し、通常計測時には「重複ACK現象の発生回数 = パケットロス回数」としてパケットロスを観測することができる[13]。しかしながらサンプリング計測を行う場合、発生したすべてのACK重複を観測することは出来ない。提案手法では、発生した重複ACKのうち、サンプリング計測時に観測できる重複ACK現象の割合を検知確率 Q_n とし、(1)式を計算することで、サンプリング計測下でのパケットロス率推定を実現する。

$$\text{パケットロス回数} = \frac{\text{検出重複ACK数}}{\text{検知確率 } Q_n} \quad (1)$$

この検知確率 Q_n は事前に正確に知ることは出来ないで、文献[11][12]の提案アルゴリズムでは、はじめに検知確率 Q_n を推定し、その結果得られるパケットロス回数から、再び検知確率 Q_n を推定するという反復計算を行う。これによりパケットロス率を推定している。この反復計算はシミュレーション評価や実装評価により、経験的に真のパケットロス率に収束することが分かっていたが、理論的検証がなかった。

本章では重複ACKのモデル化の数学的意味付けを明確にすることで、文献[11][12]の提案アルゴリズムでの推定パケットロス率が、真のパケットロス率に収束することを示す。

2.1. 重複ACKのモデル化

本節では、推定式導出のために、文献[11]で行った重複ACKのモデル化(一度のパケットロスで同じACK番号を検出できる回数とその割合)の数学的意味付けを明確にする。

文献[11]では、はじめにTCPの三つの性質を指摘している。

- A) ACK重複数は、パケットロス時の輻輳ウィンドウの大きさである。
- B) 輻輳ウィンドウの大きさの平均は λ である[14]。
- C) ACK重複数の発生数はポアソン分布に従い、サンプリング計測時には「本来の値 $\times s$ 」となる(図1)。

これらの性質により、次の仮定を設定している。

【仮定】

パケットロス時ACK重複数は平均 λ のポアソン分布に従う。

$$\lambda' = \lambda \times s = s \sqrt{\frac{8}{3b} \cdot \frac{Th}{a}} \quad (2)$$

ここで b はdelay ackパラメタ、 Th はスループット、 a はパケットロス回数、 s はサンプリング確率である。

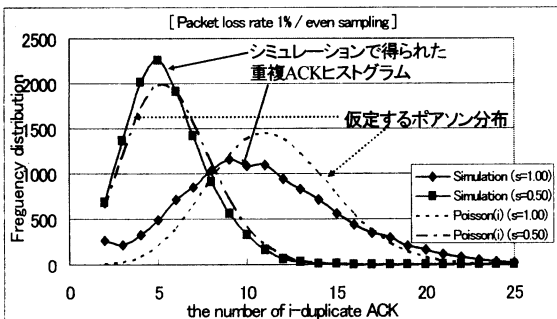


図1. サンプリング計測時の重複ACKヒストグラム

2.2. 推定式導出

本節では、前節の重複ACKのモデル化式を用いることで、パケットロス率の推定式の導出を行う。

はじめに検知確率 Q_n を求める。通信中に本来発生した重複数(パケットロス回数)を A (未知数)と置くと、前節の議論により、一度のパケットロスで同じACK番号を i 回検出できる割合は次のポアソン分布 $P_A(i)$ に従う(添え字はパケットロス回数)。

$$P_A(i) = \frac{(s \sqrt{\frac{8Th}{3bA}})^i}{i!} e^{-s \sqrt{\frac{8Th}{3bA}}} \quad (3)$$

このとき、通信中に本来発生した重複数が A 回であること、と確率分布の性質(0から ∞ までの総数が1)、より(4)式が成り立つ。

$$A \cdot \sum_{i=0}^{\infty} P_A(i) = A \quad (4)$$

しかし実際には、サンプリングにより検出した重複回数が0($i=0$)となった場合には、その現象を認識することはできない。また、検出した重複回数が1($i=1$)となった場合には、正常なACKが重複したACKを見分けることができない。このため、我々が認識できるのは、 $i \geq 2$ の場合のみである。よってサンプリング計測時に観察することができる、一度のパケットロスで同じACK番号を検出できる回数は、以下である。

$$A \cdot \left\{ 1 - \sum_{i=0}^{k-1} P_A(i) \right\} \quad (k \geq 2) \quad (5)$$

(4)式と(5)式から、検出できる重複ACKの割合(検出できる重複ACK数 / 重複ACKの総数)、つまり「検知確率 Q_n 」は以下となる。

$$1 - \sum_{i=0}^{k-1} P_A(i) \quad (k \geq 2) \quad (6)$$

この検知確率 Q_n を正確に知るためには、ポアソン分布 P_A の平均値が必要となる。つまり未知数であるパケットロス回数 A が判明している必要があり、事前に知ることはできない。このため、パケットロス回数を推定(X_n)し、その推定ポアソン分布 $P_{X_n}(i)$ により、真のパケットロス回数を求める。

(5)式と(6)式の推定値を(1)に代入すると、(7)式を導出することができる。ここで(6)の推定値が正しいければ(7)式の解は A 、つまり真のパケットロス回数となる。

$$F(X_n) = A \cdot \frac{\{1 - \sum_{i=0}^{k-1} P_A(i)\}}{\{1 - \sum_{i=0}^{k-1} P_{X_n}(i)\}} \quad (k \geq 2) \quad (7)$$

実際には検知確率 Q_n の推定値が正しい値である可能性は極めて低い。しかしながら関数 $F(X_n)$ を、 $F(X_n) = X_{n+1}$ である無限数列と考えると、 $k=2$ の場合、その極限は A に収束する性質がある($F^\infty(X_n) \rightarrow A$, 次節証明)。このため、 $k=2$ として反復計算を繰り返すと、任意の初期値 X_1 を与えても、推定値が正しい値に収束する。この反復計算により、真のパケットロス回数 A を推定することが可能となる。

パケットロス率推定アルゴリズム

1. (5)式に相当する k 回($k \geq 2$)以上同じACK番号が重複した回数をカウントする。
2. 推定パケットロス回数から(6)式の計算を行う。
3. (5)式の値と(6)式の推定値から(7)式を計算、パケットロス回数を推定する。
4. 2と3の反復計算を行い、推定値を真値に近づける。

3. 推定式の収束性証明

本章では、前章のパケットロス率推定アルゴリズムが、反復計算を行うことにより、真のパケットロス率に収束していくことを示す。

今、推定関数 $F(X_n)$ を使った反復計算を、 $F(X_n)=X_{n+1}$ である無限数列と考える。無限数列が収束するためには以下の定理を満たしている必要がある。

【利用する定理】

単調増加数列は、各項が一定の数 K を超えないときは収束。

$X_1 \leq X_2 \leq \dots \leq X_n \leq \dots \leq K$ ならば、 $\lim_{i \rightarrow \infty} X_i = \alpha \leq K$

この定理を利用し、推定関数が以下の性質を満たしていることを証明する。

- I. 初期推定パケットロス回数が真のパケットロス回数よりも小さい場合 ($X_1 \leq A$)、数列 $F(X_n)$ は単調増加し、 A を超えない ($X_n \leq F(X_n) \leq A$)。
- II. 初期推定パケットロス回数が真のパケットロス回数よりも大きい場合 ($X_1 \geq A$)、数列 $F(X_n)$ は単調減少し、 A を下回らない ($A \leq F(X_n) \leq X_n$)。

3.1. 条件 I の証明

条件 I ($X_1 \leq A$ の場合、 $X_n \leq F(X_n) \leq A$) を証明するために、(8)式を用いて変数変換を行う。

$$Z_{X_n} = \sqrt{\frac{8Th}{3}} \frac{s}{\sqrt{bX_n}} \quad (8)$$

このとき証明すべき問題 ($X_n \leq F(X_n) \leq A$) は以下に換わる。

$$\frac{Z_A^2}{Z_X^2} \leq \frac{\{1 - \sum_{i=0}^{k-1} \frac{Z_A^i}{i!} e^{-Z_A}\}}{\{1 - \sum_{i=0}^{k-1} \frac{Z_X^i}{i!} e^{-Z_X}\}} \leq 1 \quad (9)$$

$Z_A \leq Z_X$ の場合に(9)式を満足することが、反復計算で推定値が真のパケットロス率に収束するための条件である。

(i) 不等式 $[F(X_n) \leq A / (9)式の右側不等式]$ の証明

$$G(Y) = \{1 - \sum_{i=0}^{k-1} \frac{Y^i}{i!} e^{-Y}\} \quad (10)$$

ここで(9)式の分子・分母は同じ関数であることに着目し、その関数(10)式の性質を調べる。関数 $G(Y)$ は、 $Y > 0$ の範囲において、その傾き関数は $G'(Y) > 0$ となる。これは関数 $G(Y)$ が単調増加関数であり、 $Y_1 \leq Y_2$ の場合には $G(Y_1) \leq G(Y_2)$ となることを意味する。よって(11)式、つまり(9)式の右辺不等式が成立する。このため不等式 $F(X_n) \leq A$ が成立する。

$$\frac{G(Z_A)}{G(Z_X)} \leq \frac{G(Z_X)}{G(Z_X)} = 1 \quad (Z_A < Z_X) \quad (11)$$

(ii) 不等式 $[X_n \leq F(X_n) / (9)式の左側不等式]$ の証明

(9)式の左側不等式を(12)式に変形する。

$$\frac{\{1 - \sum_{i=0}^{k-1} \frac{Z_X^i}{i!} e^{-Z_X}\}}{Z_X^2} \leq \frac{\{1 - \sum_{i=0}^{k-1} \frac{Z_A^i}{i!} e^{-Z_A}\}}{Z_A^2} \quad (12)$$

ここで(12)式の右辺・左辺が同じ関数であることに着目し、その関数(13)を調べる。

$$H(Y) = \frac{\{1 - \sum_{i=0}^{k-1} \frac{Y^i}{i!} e^{-Y}\}}{Y^2} = \frac{G(Y)}{Y^2} \quad (13)$$

この関数 $H(Y)$ の傾きは、(14)式となる。

$$H'(Y) = Y^{-3} [-2 + e^{-Y} \{2 \sum_{i=0}^{k-1} \frac{Y^i}{i!} + \frac{Y^k}{(k-1)!}\}] \quad (14)$$

この関数 $H'(Y)$ は $k=2$ の場合には、常に $H'(Y) < 0$ が成立する。また $k \geq 3$ の場合には、(15)を満足する場合のみ、 $H'(Y) < 0$ が成立する。

$$I(Y) = -2 + e^{-Y} \{2 \sum_{i=0}^{k-1} \frac{Y^i}{i!} + \frac{Y^k}{(k-1)!}\} < 0 \quad (15)$$

$H'(Y) < 0$ が成立する条件では、 $H(Y)$ が単調減少関数となり、 $Y_1 \leq Y_2$ の場合、 $H(Y_1) \geq H(Y_2)$ が成り立つ。つまり(9)式の左辺不等式が成立する。このため $X_n \leq F(X_n)$ が成立する。

(iii) 極限 $[\lim_{i \rightarrow \infty} Z_{X_i} = \beta \geq Z_A]$ の証明

(i) と (ii) より、 Z_{X_n} は単調に減少し、 $Z_A \leq Z_{X_n} (n=1,2,3,\dots)$ であるから、 $\{Z_{X_n}\}$ は収束する。従ってその極限を β とすると、漸化式から(16)式となる。

$$\frac{\{1 - \sum_{i=0}^{k-1} \frac{Z_A^i}{i!} e^{-Z_A}\}}{Z_A^2} = \frac{\{1 - \sum_{i=0}^{k-1} \frac{\beta^i}{i!} e^{-\beta}\}}{\beta^2} \quad (16)$$

$X_n \leq F(X_n)$ を満足する条件では、 $H(\beta)$ は単調減少関数となることから上式の解は一つとなる。また、 $\beta = Z_A$ は明らかに(16)式の解である。よって、 $\lim_{i \rightarrow \infty} Z_{X_i} = Z_A$ となる。

3.2. 条件 II の証明

条件 II の証明は、条件 I と同様に進めることができる。

条件 II を(8)式を用いて変換すると、

$$1 \leq \frac{\{1 - \sum_{i=0}^{k-1} \frac{Z_A^i}{i!} e^{-Z_A}\}}{\{1 - \sum_{i=0}^{k-1} \frac{Z_X^i}{i!} e^{-Z_X}\}} \leq \frac{Z_A^2}{Z_X^2} \quad (17)$$

$Z_X \leq Z_A$ の場合に(17)式を満足することが、反復計算で推定値が真のパケットロス率に収束するための条件である。

(iv) 不等式 $[A \leq F(X_n) / (17)式の左側不等式]$ の証明

(i) と同様に(10)の関数が単調増加となることから、 $Z_X \leq Z_A$ より、 $G(Z_X) \leq G(Z_A)$ が成立。これにより(17)式の左辺不等式が成り立つ。

(v) 不等式 $[F(X_n) \leq X_n / (17)式の右側不等式]$ の証明

(ii) と同様に(13)の関数が(15)を満たしている場合には、単調減少することから、 $Z_X \leq Z_A$ の場合、 $H(Z_A) \leq H(Z_X)$ が成立。これより(17)式の右辺不等式が成り立つ。

(vi) 極限 $[\lim_{i \rightarrow \infty} Z_{X_i} = \beta \leq Z_A]$ の証明

(iii) と同様に(16)の関数の解を求める。(16)の関数の解は唯一であり、 $\beta = Z_A$ は明らかに(16)式の解であるため、 $\lim_{i \rightarrow \infty} Z_{X_i} = Z_A$ となる。

以上により、推定アルゴリズムの次の性質を示した。

- ・ $k=2$ の場合には、反復により、推定パケットロス率が真のパケットロス率に収束する。
- ・ $k \geq 3$ の場合には、(15)を満足する条件で、反復により、推定パケットロス率が真のパケットロス率に収束する。

$k=2$ の現実的な意味は、検出重複 ACK 数のカウントアップ条件が、二回以上同じ ACK 番号を検知した場合に相当する。

4. 収束条件からのサンプリングレート条件の導出

前章の推定式の収束性の証明により、 $k=2$ の場合には無条件に、 $k \geq 3$ の場合には(15)の条件を満たす範囲で、推定アルゴリズムが真のケットロス率に収束することを示した。一方、(7)の推定式は、その設計パラメタに (s, X_p, k) を持つ。

このため、 $k \geq 3$ の場合でも、 (s, X_p) に何らかの制約を持たせることで、推定アルゴリズムを真のケットロス率に収束させることが出来ると考えられる。

本章では、この $k \geq 3$ の場合に、真のケットロス率を求めるための s の条件を明確にすることで、サンプリングレート条件を導出する。

4.1. k の収束条件

$k \geq 3$ の収束条件は(15)を満たす Y が存在することである。この(15)式の微分関数は(18)となる。

$$I'(Y) = \left\{ -\frac{Y^{k-1}}{(k-1)!} e^{-Y} \right\} \{Y + 2 - k\} \quad (18)$$

このことから、 $Y=k-2$ が唯一の変極点で、 $Y > (k-2)$ では単調減少関数となる。 $I(Y)=0$ かつ、 $Y > (k-2)$ を満たす最小の Y を Y_{min} と定義すると、 $Y > Y_{min}$ の領域では常に $I(Y) < I(Y_{min}) < 0$ が成立する。このため、 $k \geq 3$ での収束条件は $Y > Y_{min}$ となる。

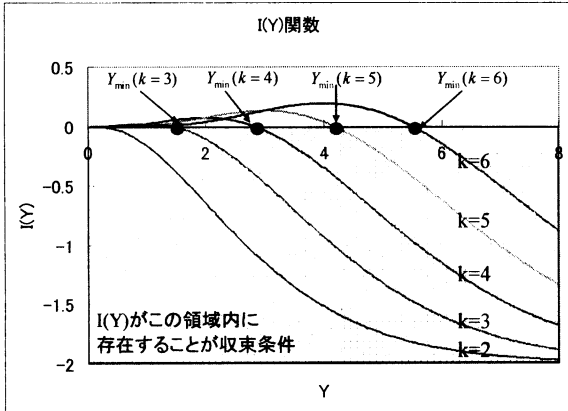


図2. k の Y に関する収束条件

(15)式から $k=2 \sim 6$ の Y_{min} を数値計算(精度 10^{-4} を切上)すると以下となる(図2参照)。

$$k=2, Y_{min} < Y'_{min} = 0 \quad (19)$$

$$k=3, Y_{min} < Y'_{min} = 1.452 \quad (20)$$

$$k=4, Y_{min} < Y'_{min} = 2.838 \quad (21)$$

$$k=5, Y_{min} < Y'_{min} = 4.184 \quad (22)$$

$$k=6, Y_{min} < Y'_{min} = 5.503 \quad (23)$$

Y は推定ケットロス率 $x (x = X_p/Th)$ と、サンプリング確率 s の関数であり、(24)式の関係を持つことから、(19)~(23)を(25)~(29)と変換することができる。これが $k \geq 3$ の収束条件を満たすための、サンプリング確率 s と推定ケットロス回数 x の関係となる。

$$x = \frac{8}{3Y^2} \cdot \frac{s^2}{b} \quad (24)$$

$$k=2, (s, x) \text{ 全領域で収束する} \quad (25)$$

$$k=3, x \leq 0.6325 s^2 \quad (26)$$

$$k=4, x \leq 0.1655 s^2 \quad (27)$$

$$k=5, x \leq 0.0762 s^2 \quad (28)$$

$$k=6, x \leq 0.0044 s^2 \quad (29)$$

(s, x) の範囲から、サンプリング確率 s を与えたときの推定ケットロス率の探索範囲が求まる。反復計算により、真のケットロス率を求めるには、推定ケットロスの探索範囲内に真のケットロス率が収まっている必要がある。よって、 $k \geq 3$ において、推定式が真のケットロス率に収束するための条件は、推定ケットロス率(特に初期推定値)と真のケットロス率が共に(25)~(29)を満足している必要がある。図3に推定アルゴリズムが収束するための(サンプリング確率、ケットロス率)の範囲を示す。

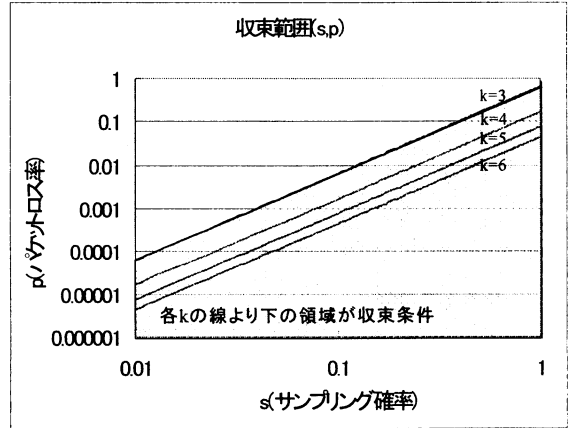


図3. (サンプリング確率、ケットロス率)の収束範囲

4.2. サンプリング性能条件

前節の議論により、推定式(7)がケットロス率を推定するための (s, x) の条件を明確にすることが出来た。本節ではその条件からサンプリング確率とケットロス率の関係を導く。

現実的なネットワークを考慮した場合、ケットロス率は通常0%から数%程度であり、0%から100%の全範囲を採る可能性は極めて低い。ネットワーク監視の目的を考慮すると、異常値の有無が検知できることが重要である。このため、監視想定のケットロス率 (x) の範囲を決定すれば、設定可能なサンプリング確率 (s) の範囲、つまり、サンプリング性能 $(1/\text{サンプリング確率 } s)$ の限界性能条件が決定される(図4)。

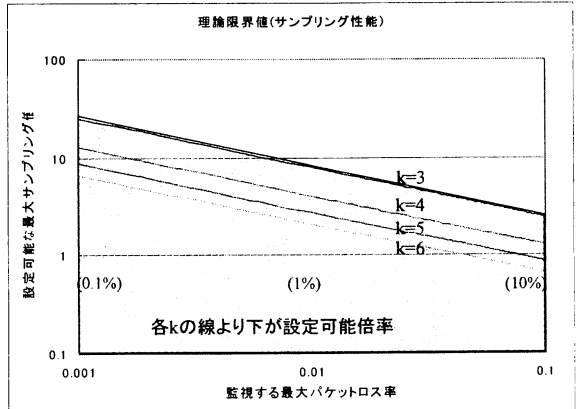


図4. サンプリング性能の限界値

$k=2$ は全倍率(最大サンプリング倍率 ∞)となる。ただし本章での性能限界値は収束条件から求まるものであり、それ以外の条件(統計量や仮定の成立範囲、許容可能な反復回数等)からこの性能限界値は制約を受ける可能性はある。

5. シミュレーション評価

本章では Network Simulator2(以後 ns2)[15]を用い、明らかにした推定アルゴリズムの反復過程を確認し、解析結果と比較する。次にシミュレーションを通じて、推定アルゴリズムの推定パケットロス回数の初期値に関する考察を行う。最後に提案方式が一定精度のパケットロス率を推定するためには、どの程度の反復回数が必要になるかの検証を行う。

5.1. 評価環境

シミュレーション環境について説明する。以降の評価では、本節の変動パラメータを変化させて、反復による推定精度の変化を評価する。

5.1.1. 固定パラメータ

a. ネットワーク構成

図5で示すネットワーク構成で評価した。End-to-EndでTCP通信が行われている状況である。

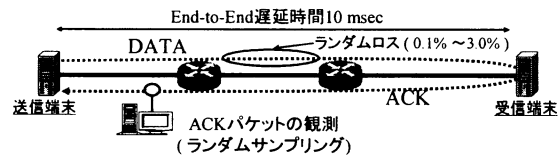


図5. 評価ネットワーク環境

b. 観測場所

提案方式ではTCP通信のACKパケットを観測して、データ側のパケットロス率を推定する。このため、ACK側パケットのみを観測する。ACK側なら、観測場所を問わず、ほぼ同じ結果を得ることができる。

c. 回線遅延

End-to-End遅延時間として10ms、RTT約20msの場合の評価を行った。これは東京-大阪間程度の通信に相当する。提案手法では、End-to-Endの遅延時間が同じであれば、その中のノード数やノード間の遅延時間の分配等に関わらず、同じ結果となる。

d. 観測時間

パケットロス率を算出するために、どの程度の時間のデータを元に示す指標である。この期間が長いほど、正確な値を求めることができる。本評価では、標準的なMIBと同程度の5分とした。

5.1.2. 変動パラメータ

e. パケットロス率(実際のロス回数A/スループットTh)

TCP通信の送信端末から受信端末中に向けてのデータ側パケットに対してロスを設定した。このときのロスパターンは、ランダムとし、0.1%、0.2%、0.5%、1.0%、1.5%、2.0%を評価した。

f. サンプリング倍率(1/サンプリング率s)

サンプリング率sは、観察頻度である。本評価では、サンプリング倍率1($s=1.0$)~10($s=0.1$)の条件で行った。このときのサンプリングパターンとしてはランダムサンプリングを行う。

g. 推定パケットロス回数の初期値(X_1)

推定式(7)に代入する X_1 である。この値が、推定アルゴリズムが真のパケットロスに収束する反復回数に影響を与える。

5.2. 解析結果とシミュレーションの比較評価

本評価では、推定パケットロス回数の初期値を与えた後の、(7)の解析式とns2でのシミュレーションとの反復過程を比較する(図6)。これにより、実際に反復計算結果が、解析式通りに変化しているかを確認する。

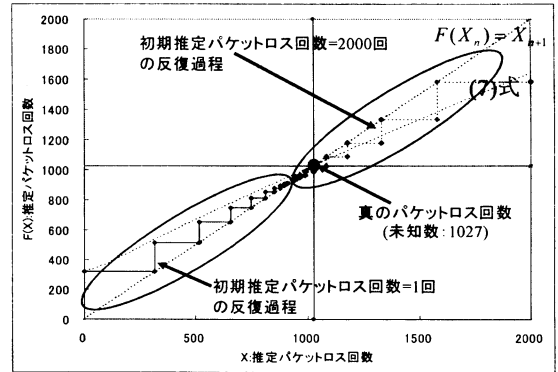


図6. シミュレーション/解析式(7)の反復過程

図6は、パケットロス率1%の環境をサンプリング確率10%で観測した結果である。真のパケットロス回数は1027回(実際には未知)であった。このとき初期推定パケットロスを1回にした場合と2000回にした場合のシミュレーションでの反復過程を記載している。どちらも数回の反復で926回に収束している。図からシミュレーション結果が、解析式(7)と反復条件 $F(X_n)=X_{n-1}$ の範囲におおよそ収まっていることが分かる。

5.3. 推定パケットロス回数の初期値による反復回数評価

提案手法では推定パケットロス回数の初期値を適当に選定しても真値に収束するが、選定する値によって収束するまでの反復回数が異なってくる。本評価ではシミュレーションを通して、適切な初期パラメータの選定方法について検討する。

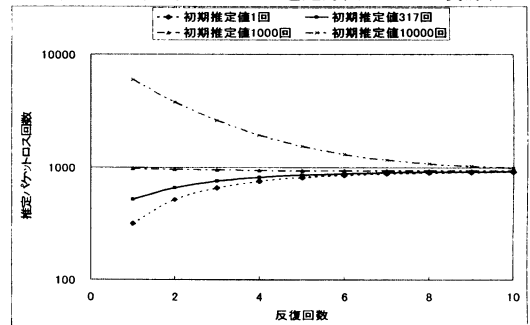


図7. 各初期推定値による反復過程

図7は、パケットロス率1%の環境をサンプリング確率10%で観測した場合に、四種類の初期推定値を用いて反復回数を評価した結果である。真のパケットロス回数は1027回である。

- I. 初期値1回: 最小値の代表値(パケットロス率0.001%)
- II. 初期値317回: 検知重複ACK数
- III. 初期値1000回: 直前のパケットロス回数
- IV. 初期値10000回: 最大値の代表値(パケットロス率10%)

この結果、収束までの反復回数が少ない順に、III<II<IVである。IIIは直前の値を使うため、ロス変動が大きい場合には使えない可能性がある。このため、事前にある程度の予測ができる場合にはその値を使用し、予測が出来ない場合には検知重複ACK数を使用することが望ましいと考えられる。

5.4. 収束するまでの反復回数評価

真の値に収束するまでの反復回数は、サンプリング倍率(サンプリング確率の逆数)やパケットロス率によっても変化する。本節では、提案方式の推定値がある程度の精度を確保するまでに、どの程度の反復計算が必要になるかを評価する。

評価条件1：推定パケットロス回数の初期値

提案手法での収束までの反復回数は、推定パケットロス回数の初期値 X_1 に大きな影響を受ける。この X_1 は前節での評価により、実際の値が予測できる場合にはその値を、予測出来ない場合には検知重複 ACK 数を使用するのが望ましいことが分かった。一般に予測値を使える状況は限定されるため、本評価での初期値 X_1 は「検知重複 ACK 数」を用いる。

評価条件2：収束について

反復計算で実際に真値に収束するためには多数の反復を必要とする。これは真値近辺では反復による推定値の変化率が少ないからである。このため、本評価では十分に収束した値を基準値として、その値の90%以上まで収束するのに必要な反復回数と、95%以上まで収束するのに必要な反復回数とを評価する。

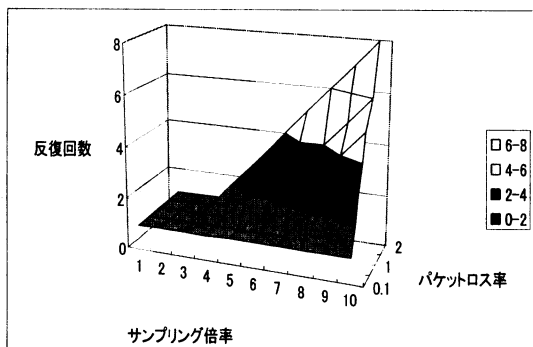


図8. 収束するまでの反復回数 (精度90%以上)

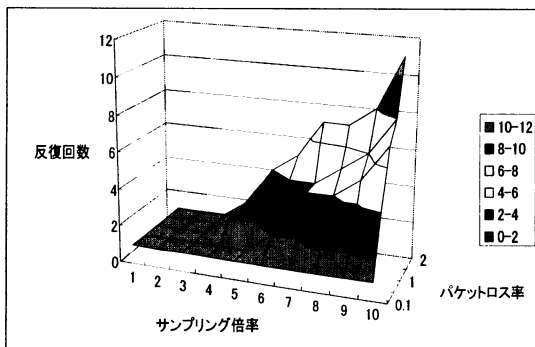


図9. 収束するまでの反復回数 (精度95%以上)

図8に収束値の90%以上を推定するまでの反復回数、図9に収束値の95%以上を推定するまでの反復回数、を記載している。

図8と図9を比較すると、90%までの反復回数と95%までの反復回数とを比較すると約1.5倍回数がかかっている。また図より、多くの場合数回以内の反復で、収束値の95%以上に近づくことが分かる。ただし、パケットロス率2%以上の領域では10回程度の反復が必要となる。

6. まとめ

本研究では、サンプリング技術を用いることで、高速ネットワーク環境においてTCPの品質、パケットロス率、を監視する方式を提案した。本稿では特に、提案方式が真のパケットロス率に収束するための条件を解析的に明らかにした。また、この収束条件から設定可能なサンプリング確率の範囲も明らかにした。この結果、二回以上同じACK番号を検知した場合に、検出重複ACK数のカウントアップ(k=2)をすれば、常に真のパケットロス率に収束することが分かった。さらにシミュレーション評価も行い、推定方式が真のパケットロス率に収束する過程を明らかにした。

今後は、仮定条件のより理論的な考察と、統計量の影響(推定精度と観測時間)等について検討していく予定である。

参考文献

- [1] 独立行政法人情報処理推進機構, “情報セキュリティ対策研究開発評価事業 高トラフィック観測・分析法に関する技術調査”, 2004年4月.
- [2] P.Pheal, S.Panchen, and N. McKee, RFC3176 “InMon Corporation’s sflow: A Method for Monitoring Traffic in Switched and Routed Networks”, IETF, September 1992.
- [3] 菅内公徳, 樋口秀光, 土屋一暁, 水野潤, “高トラフィックネットワークにおけるトラフィック監視技術”, 信学技報, NS2003-312, IN2003-267, 2004年3月.
- [4] 森達哉, 内田真人, 川原亮一, 後藤滋樹, “サンプルパケットから構成されるフロー統計の評価”, 2004年電子情報通信学会ソサイエティ大会 BS-3-1.
- [5] 森達哉, 内田真人, 川原亮一, 後藤滋樹, “サンプルしたパケットからエレファントフローを特定する方法”, 信学技報, NS2004-13, 2004年4月.
- [6] C.Estan and G.Varghese, “New Directions in Traffic Measurement and Accounting”, In Proceeding of ACM SIGCOMM, August 2002.
- [7] N.Duffield, C.Lund, M.Thorup, “Properties and Prediction of Flow Statistics from Sampled Packet Streams”, ACM SIGCOMM Internet Measurement Workshop, Marseille, France, November, 2002.
- [8] N.Duffield, C.Lund, M.Thorup, “Estimating Flow Distributions from Sampled Flow Statistics”, In Proceedings of ACM SIGCOMM, August 2003.
- [9] 川原亮一, 石橋圭介, 森達哉, 阿部威郎, “サンプルパケット情報を用いたTCPフローレベル性能劣化検出法”, 信学技法, TM2004-34, 2004年7月.
- [10] 川原亮一, 石橋圭介, 森達哉, 阿部威郎, “サンプルパケット情報を用いたTCPフローレベル性能劣化検出法とその評価”, 2004年電子情報通信学会ソサイエティ大会 B-7-7.
- [11] 山崎康広, 下西英之, 村瀬勉, “サンプリング計測でのパケットロス率推定手法の提案”, 信学技法, NS2004-159, TM20046-62, 2004年11月.
- [12] 山崎康広, 下西英之, 村瀬勉, “サンプリング計測でのパケットロス率推定手法の実装評価”, 2005年電子情報通信学会総合大会 B-14-4.
- [13] 大岸智彦, 井戸上彰, 長谷川亨, 加藤聡彦, “片方向IPトラフィックからTCPレベルの統計情報を収集するパフォーマンスモニタの設計”, 2000年電子情報通信学会総合大会 B-7-3.
- [14] Jitendra Padhye, Victor Firoiu, Don Towsley, Jim Kurose, “Modeling TCP Throughput: A Simple Model and its Empirical Validation”, In ACM SIGCOMM’98.
- [15] “Network simulator-ns(version2)..” available from <http://www.isi.edu/nsnam/ns/>.