

# 輻輳を考慮した可変ウィンドウ方式の信頼性解析

## Reliability Analysis of a Window Flow Control Scheme considering Network Congestion

木村 充 位      今 泉 充 啓      安 井 一 民  
Mitsutaka Kimura      Mitsuhiro Imaizumi      Kazumi Yasui

### Abstract

As use of the Internet has spread quickly, a high quality traffic for networks has been urgently needed for the Internet. Recently, the flow of data traffic on the Internet has been worsened due to packets loss, which have been generated by network congestion. Several authors have studied window flow control mechanisms for dissolving packets loss. This paper considers a stochastic model where data are transmitted by a window flow control scheme considering network congestion. The mean time until the data transmission succeeds is analytically derived and an optimal policy which maximizes the throughput is discussed. Finally, numerical examples are given.

keyword : Internet, Packets Loss, Network Congestion, Window Flow Control, Reliability

### 1 はじめに

近年、インターネットの高信頼性を実現するため、さまざまなプロトコルが提案されている [ 1 ], [ 2 ]. 最近では、データ転送時に発生するパケット損失が問題となっている。すなわち、インターネットにおいて、輻輳が発生した場合、各ルータに停滞しているパケットは破棄され、パケット損失が発生して通信効率が損なわれる問題などが提起されている [ 3 ].

データ通信における一般的な高信頼化の方法として、可変ウィンドウ方式がある。これは、輻輳が発生したときに、ウィンドウサイズを半分に減少して送信し、輻輳が解除された場合には徐々に増加させる方法である [ 4 ].

ここでは、可変ウィンドウ方式を適用したデータ通信における信頼性の問題を考察する。すなわち、WWW サーバはウィンドウサイズに相当するパケット数を順次クライアントへ送信し、誤ったパケットは再送する。また、再送が  $k$  回失敗したならば、輻輳が発生しているものとみなし、ウィンドウサイズからその半分に当たるパケット数を再送すると仮定した信頼性モデルを設定する。そのとき、送信成功となるまでの平均時間を解析的に導出し、さらに、スループットを最大にする最適方を議論する。

### 2 モデルの設定と解析

複数のクライアントと WWW サーバで構成される通信システムを考える。各クライアントは WWW サーバへある大きさのデータ要求を行い、その要求データのうちのウィンドウサイズに相当するパケットを順次受信する。ウィンドウサイズに相当するパケットを受信完了後、要求したすべてのデータの受信が完了するまで次のウィンドウサイズに相当するパケットを順次受信する。

ここでは、あるクライアント（受信側）から送信要求を受信した時点から一つのウィンドウサイズに相当するパケットが送信成功となるまでの動作に着目してモデル化を行う。

- ( 1 ) サーバはクライアントからデータ要求を受信した時点でコネクション確立処理を開始する。データ要求にはウィンドウサイズ（送信パケット数  $n_i$ ）に関する情報を含む。サーバはコネクション確立要求を受信した後、ウィンドウサイズに相当するデータ（送信パケット数  $n_i$ ）を送信する。サーバがコネクション確立処理に要する時間分布を  $A(t)$ （平均  $a$ ）とする。
- ( 2 ) クライアントはすべてのパケット（ $n_i$ 個）が正常に受信できていれば ACK を返し、欠落パケットまたは誤りパケットがある場合は不足しているパケットまたは誤りパケットを NAK を用いてサーバへ通知する。サーバがあるパケットを送信し、確認応答パケットを受信するまでの経過時間分布を  $D(t)$ （平均  $d$ ）とおく。1つのパケットを損失する確率は  $p_i$  とする。
- ( 3 ) サーバが一つのパケットを編成し送出するのに要する経過時間分布を  $B(t)$ （平均  $b$ ）とおく。 $n_i$ 個のパケットに対し、ACKを受信したならば送信成功。

- (4) サーバはクライアントからの返信パケット (ACK または NAK) を確認し, NAK 対象のパケットがある場合はそのパケットを再送する.
- (5) 再送が  $k$  回失敗したならば, ネットワーク系に輻輳が発生しているものとみなし, 送信を一定時間中断する. この場合, 初期状態のウィンドウサイズからその半分に当たるパケット数 (送信パケット数  $n_1 < n_1$ ) にして再送する. 送信パケットの再編集に要する時間分布を  $W(t)$  (平均  $w$ ) とおく. なお, このとき 1 つのパケットを損失する確率は  $p_1 < p_1$  に低減するものとする.
- (6) 再び, 再送が  $k$  回失敗したならば, 送信を中断して伝送系をチェックし一定時間後, 初期状態からやり直す. このやり直すまでの経過時間分布を  $G(t)$  (平均  $\mu$ ) とおく.
- (7) 1 回目の  $n_2$  のパケットに対し, ACK を受信したら 2 回目のパケット ( $n_2$  個) を送信する. すべてのパケット ( $n_1$  個) に対し, ACK を受信したら送信成功.

以上の仮定のもとでシステムの状態を次のように定義する.

- 状態 0 : クライアントとのコネクションの確立処理開始.
- 状態 1 :  $n_1$  個のパケットを連続的に送信開始.
- 状態  $F_1$  : 再送が  $k$  回失敗し, 一時中断.
- 状態  $R_1$  : 送信パケットを  $n_2$  個にして再送やり直し開始.
- 状態  $R_2$  : 1 回目の送信パケット  $n_2$  個の送信成功, 2 回目の送信パケット  $n_2$  個の送信開始.
- 状態  $F_2$  : 送信パケットを  $n_2$  個のうち,  $k$  回の再送が失敗し, 一時中断.
- 状態  $S$  :  $n_1$  個のパケット送信成功.

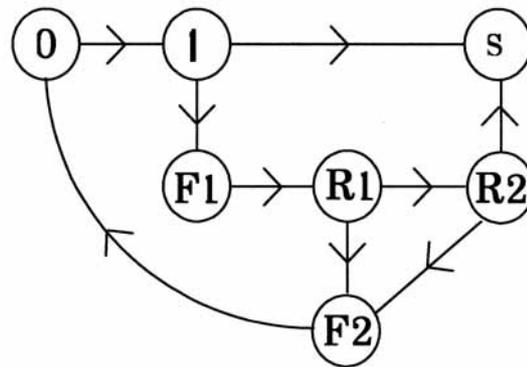


図 1 状態推移図

各状態は, 状態  $S$  を吸収状態とするマルコフ再生過程 [ 5 ] を形成し, 各状態間の推移は図 1 のように表される.

最初に, 送信成功までの平均時間  $\ell_{0,S}$  を考える. システムが状態 0 から出発して, 送信成功となるまでの経過時間分布  $H_{0,S}(t)$  は次式で与えられる.

$$\begin{aligned}
 H_{0,S}(t) = & \sum_{i=1}^{\infty} \left[ A(t) * Q_{1,F_1}(t) * W(t) * \left[ Q_{R_1,F_2}(t) + Q_{R_1,R_2}(t) * Q_{R_2,F_2}(t) \right] * G(t) \right]^{(i-1)} \\
 & * A(t) * \left[ Q_{1,S}(t) + Q_{1,F_1}(t) * W(t) * Q_{R_1,R_2}(t) * Q_{R_2,S}(t) \right].
 \end{aligned} \tag{1}$$

ここで, 一般に,

$$\begin{aligned}
Q_S(t|n, p) &\equiv (1-p)^n B^{(n)}(t) * D(t) \\
&+ \sum_{m_1=1}^n \binom{n}{m_1} p^{m_1} (1-p)^{n-m_1} B^{(n)}(t) * D(t) * (1-p)^{m_1} B^{(m_1)}(t) * D(t) \\
&+ \sum_{m_1=1}^n \binom{n}{m_1} p^{m_1} (1-p)^{n-m_1} B^{(n)}(t) * D(t) \\
&\quad * \sum_{m_2=1}^{m_1} \binom{m_1}{m_2} p^{m_2} (1-p)^{m_1-m_2} B^{(m_1)}(t) * D(t) * (1-p)^{m_2} B^{(m_2)}(t) * D(t) \\
&+ \\
&\vdots \\
&+ \\
&+ \sum_{m_1=1}^n \binom{n}{m_1} p^{m_1} (1-p)^{n-m_1} B^{(n)}(t) * D(t) \\
&\quad * \sum_{m_2=1}^{m_1} \binom{m_1}{m_2} p^{m_2} (1-p)^{m_1-m_2} B^{(m_1)}(t) * D(t) * \dots \\
&\quad * \sum_{m_{k-1}=1}^{m_{k-2}} \binom{m_{k-2}}{m_{k-1}} p^{m_{k-1}} (1-p)^{m_{k-2}-m_{k-1}} B^{(m_{k-2})}(t) * D(t) \\
&\quad * (1-p)^{m_{k-1}} B^{(m_{k-1})}(t) * D(t), \tag{2}
\end{aligned}$$

$$\begin{aligned}
Q_F(t|n, p) &\equiv \sum_{m_1=1}^n \binom{n}{m_1} p^{m_1} (1-p)^{n-m_1} B^{(n)}(t) * D(t) * \dots \\
&\quad * \sum_{m_{k-1}=1}^{m_{k-2}} \binom{m_{k-2}}{m_{k-1}} p^{m_{k-1}} (1-p)^{m_{k-2}-m_{k-1}} B^{(m_{k-2})}(t) * D(t) \\
&\quad * [1 - (1-p)^{m_{k-1}}] B^{(m_{k-1})}(t) * D(t), \tag{3}
\end{aligned}$$

と定義すると、

$$Q_{1,S}(t) \equiv Q_S(t|n_1, p_1), \tag{4}$$

$$Q_{R_1, R_2}(t) = Q_{R_2, S}(t) \equiv Q_S(t|n_2, p_2), \tag{5}$$

$$Q_{1, F_1}(t) \equiv Q_F(t|n_1, p_1), \tag{6}$$

$$Q_{R_1, F_2}(t) = Q_{R_2, F_2}(t) \equiv Q_F(t|n_2, p_2). \tag{7}$$

となる。ここで、\*は分布関数のたたみ込みを表し、 ${}^{(i)}(t)$ は $(t)$ の $i$ 重たたみ込みを表す。すなわち、 ${}^{(i)}(t) = {}^{(i-1)}(t) * (t)$ 、 $(t) * (t) = \int_0^t (t-u) d(u)$ 、 ${}^{(0)}(t) = 1$ である。

式(1)をラプラス・スチルチェス(LS)変換し、方程式を解くことによって、

$$\begin{aligned}
h_{0,S}(s) &= \frac{a(s) \left[ q_{1,S}(s) + q_{1,F_1}(s) w(s) q_{R_1, R_2}(s) q_{R_2, S}(s) \right]}{1 - a(s) q_{1,F_1}(s) w(s) \left[ q_{R_1, F_2}(s) + q_{R_1, R_2}(s) q_{R_2, F_2}(s) \right] g(s)} \\
&= \frac{a(s) \left[ q_S(s|n_1, p_1) + q_F(s|n_1, p_1) w(s) \left[ q_S(s|n_2, p_2) \right]^2 \right]}{1 - a(s) q_F(s|n_1, p_1) w(s) q_F(s|n_2, p_2) \left[ 1 + q_S(s|n_2, p_2) \right] g(s)} \tag{8}
\end{aligned}$$

を得る。ここで、一般に $(t)$ のLS変換を $(s)$ とおく。

一般に式(8)において送信成功までの平均時間を求めることは困難なため、特に $k=2$ のとき、すなわち、再送信が2回失敗したとき、初期状態のウィンドウサイズの半分に当たるパケットの再送、または、中断して初期状態からやり直す場合の平均時間を求めよう。

ここで、式(2)(3)をLS変換すると、

$$q_S(s|n, p) = [(1-p)b(s)]^n d(s) \left\{ 1 + d(s) \left[ 1 + pb(s) \right]^n - 1 \right\}, \quad (9)$$

$$q_F(s|n, p) = [b(s)]^n [d(s)]^2 \left\{ [(1-p) + pb(s)]^n - [(1-p)[1 + pb(s)]]^n \right\}. \quad (10)$$

従って、 $k=2$ のとき、送信成功までの平均時間 $\ell_{0,S}$ を次のように求めることができる。

$$\ell_{0,S} = \frac{\left[ \begin{array}{l} a + b \left[ n_1(1+p_1) + n_2[1 - (1-p_1^2)^{n_1}][1 + (1-p_2^2)^{n_2}](1+p_2) \right] \\ + d \left[ 2 - (1-p_1)^{n_1} + [1 - (1-p_1^2)^{n_1}][1 + (1-p_2^2)^{n_2}][2 - (1-p_2)^{n_2}] \right] \\ + w[1 - (1-p_1^2)^{n_1}] + \mu[1 - (1-p_1^2)^{n_1}][1 - (1-p_2^2)^{2n_2}] \end{array} \right]}{1 - [1 - (1-p_1^2)^{n_1}][1 - (1-p_2^2)^{2n_2}]} \quad (11)$$

また、 $k=3$ のとき、送信成功までの平均時間は前と同様にして、

$$\ell_{0,S} = \frac{\left[ \begin{array}{l} a + b \left[ n_1(1+p_1+p_1^2) + n_2[1 - (1-p_1^3)^{n_1}][1 + (1-p_2^3)^{n_2}](1+p_2+p_2^2) \right] \\ + d \left[ \begin{array}{l} 3 - (1-p_1)^{n_1} - (1-p_1^2)^{n_1} \\ + [1 - (1-p_1^3)^{n_1}][1 + (1-p_2^3)^{n_2}][3 - (1-p_2)^{n_2} - (1-p_2^2)^{n_2}] \end{array} \right] \\ + w[1 - (1-p_1^3)^{n_1}] + \mu[1 - (1-p_1^3)^{n_1}][1 - (1-p_2^3)^{2n_2}] \end{array} \right]}{1 - [1 - (1-p_1^3)^{n_1}][1 - (1-p_2^3)^{2n_2}]} \quad (12)$$

さらに、 $k=4$ のとき、送信成功までの平均時間は、

$$\ell_{0,S} = \frac{\left[ \begin{array}{l} a + b \left[ n_1(1+p_1+p_1^2+p_1^3) + n_2[1 - (1-p_1^4)^{n_1}][1 + (1-p_2^4)^{n_2}](1+p_2+p_2^2+p_2^3) \right] \\ + d \left[ \begin{array}{l} 4 - (1-p_1)^{n_1} - (1-p_1^2)^{n_1} - (1-p_1^3)^{n_1} \\ + [1 - (1-p_1^4)^{n_1}][1 + (1-p_2^4)^{n_2}][4 - (1-p_2)^{n_2} - (1-p_2^2)^{n_2} - (1-p_2^3)^{n_2}] \end{array} \right] \\ + w[1 - (1-p_1^4)^{n_1}] + \mu[1 - (1-p_1^4)^{n_1}][1 - (1-p_2^4)^{2n_2}] \end{array} \right]}{1 - [1 - (1-p_1^4)^{n_1}][1 - (1-p_2^4)^{2n_2}]} \quad (13)$$

以上より、上述の解析から、一般の $k$ における平均時間を次のように求めることができる。

$$\ell_{0,S} = \frac{\left[ \begin{array}{l} a + b \left[ n_1 \sum_{i=0}^{k-1} p_1^i + n_2[1 - (1-p_1^k)^{n_1}][1 + (1-p_2^k)^{n_2}] \sum_{i=0}^{k-1} p_2^i \right] \\ + d \left[ \begin{array}{l} k - \sum_{i=1}^{k-1} (1-p_1^i)^{n_1} + [1 - (1-p_1^k)^{n_1}][1 + (1-p_2^k)^{n_2}][k - \sum_{i=1}^{k-1} (1-p_2^i)^{n_2}] \\ + w[1 - (1-p_1^k)^{n_1}] + \mu[1 - (1-p_1^k)^{n_1}][1 - (1-p_2^k)^{2n_2}] \end{array} \right] \end{array} \right]}{1 - [1 - (1-p_1^k)^{n_1}][1 - (1-p_2^k)^{2n_2}]} \quad (14)$$

### 3 最適方策

ここでは一つのウィンドウサイズ分が送信成功するまでの単位時間当たりのデータ転送量を最大にする最適方策を考察する。最初に単位処理データを $n_i$ 個のケットに分割して送信する場合、 $n_1, 2n_2$ としてスループット $E(n_2)$ を次のように定義する。

$$E(n_2) \equiv \frac{2n_2}{\ell_{0,S}(n_2)}. \quad (15)$$

ここで、

$$\ell_{0,S}(n_2) \equiv X(n_2) - \mu, \quad (16)$$

$$X(n_2) \equiv \frac{\left[ \begin{array}{c} a + bn_2 \left[ 2 \sum_{i=0}^{k-1} p_1^i + [1 - (1 - p_1^k)^{2n_2}][1 + (1 - p_2^k)^{n_2}] \sum_{i=0}^{k-1} p_2^i \right] \\ + d \left[ k - \sum_{i=1}^{k-1} (1 - p_1^i)^{2n_2} + [1 - (1 - p_1^k)^{2n_2}][1 + (1 - p_2^k)^{n_2}][k - \sum_{i=1}^{k-1} (1 - p_2^i)^{n_2}] \right] \\ + w[1 - (1 - p_1^k)^{2n_2}] + \mu \end{array} \right]}{1 - [1 - (1 - p_1^k)^{2n_2}][1 - (1 - p_2^k)^{2n_2}]} \quad (17)$$

このとき、 $E(n_2)$ を最大にする最適なウィンドウサイズ  $n_2^*$ を求める。ここでは、 $A(n_2) = 1/E(n_2)$ とおき、 $A(n_2)$ を最小にする  $n_2^*$ を求める。不等式  $A(n_2 + 1) - A(n_2) \geq 0$  とおくと、

$$n_2 X(n_2 + 1) - (n_2 + 1) X(n_2) + \mu \geq 0, \quad (18)$$

を得る。式(18)の左辺を  $L(n_2)$ とおくと、次式を得る。

$$L(n_2 + 1) - L(n_2) = (n_2 + 1) D(n_2). \quad (19)$$

ここで、

$$D(n_2) \equiv [X(n_2 + 2) - X(n_2 + 1)] - [X(n_2 + 1) - X(n_2)]. \quad (20)$$

従って、 $X(n_2)$ は凸関数かつ  $D(1) > 0$  ならば、 $L(n_2)$ は  $L(1)$  から  $\infty$  までの  $n_2$ の単調増加関数となる。

以上より、次のような結論を得ることができる。

- (i) もし、 $D(1) > 0$  かつ  $L(1) < 0$  ならば、式(18)を満たす有限で唯一の  $n_2^*$ が存在する。
- (ii) もし、 $D(1) > 0$  かつ  $L(1) \geq 0$  ならば、 $n_2^* = 1$ である。

## 4 数値例

3. で求めた最適なウィンドウサイズ  $n_2^*$ について具体的な数値を求める。一つのペケットを編集し、送信開始するまでの平均時間  $b$  をシステムの単位時間とし、コネクション確立に要する平均時間を  $a/b = 10$ 、ペケットを送信し応答確認ペケットを受信するまでの平均時間を  $d/b = 2 \sim 32$  (可変)、初期状態のウィンドウサイズからその半分に当たるサイズに再編集するまでの平均時間を  $w/b = 10$ 、中断し初期状態からやり直すまでの平均時間を  $\mu/b = 30$ 、ペケットを損失する確率  $p_1 = 0.1 \sim 0.3$  (可変)  $p_2 = 0.05 \sim 0.3$  (可変) とおく。

以上の仮定の下で、スループットを最大にする最適なウィンドウサイズ  $n_2^*$  ( $k = 2 \sim 4$ ) の数値例を表1～表3に示す。

表1～表3によれば、スループットを最大にする最適なウィンドウサイズ  $n_2^*$ は同一のペケット損失確率  $p_1, p_2$ の下でペケットを送信し応答確認ペケットを受信するまでの平均時間  $d/b$  が大きくなるにつれて増大し、さらに同一の  $d/b, p_1$ の下で、ウィンドウサイズを半分にしたときのペケット損失確率  $p_2$ が大きくなるにつれて減少することが示された。

また、同一の  $d/b, p_2$ の下で、 $p_2$ が小さいときは、 $p_1$ が大きくなるにつれて概ね増大し、逆に  $p_2$ が大きいつきは、 $p_1$ が大きくなるにつれて減少する傾向を示した。

スループット  $E(n_2^*)$ は同一の  $p_1, p_2$ の下で  $d/b$  が大きくなるにつれて減少し、さらに同一の  $d/b$ の下で  $p_1$ が大きくなるにつれて小さくなることが示された。また、同一の  $d/b, p_1$ の下で  $p_2$ が大きくなるにつれて減少し、同一の  $d/b, p_2$ の下で、 $p_1$ が大きくなるにつれて減少することが示された。

さらに、文献[3]にあるような数値例をもとに考察する。例えば、 $k = 2, p_1 = 0.3, p_2 = 0.1, d/b = 8$ のようなネットワーク環境のとき、スループット  $E(n_2)$ を最大にする  $n_2^*$ は  $n_2^* = 19$ である。ここでは  $n_2 = 19$ に固定したときの再送回数  $k$ の大きさについて考察する(図2)。

図2から  $k$ がある程度大きくなると、例えば、 $k \geq 7$ で  $E(n_2)$ は大略一定の値に収束する傾向を示している。このことは、再送回数が  $k \geq 7$ ではネットワークに輻輳が発生しているとみなして早急に対策等を実行すべきであることを示している。

輻輳を考慮した可変ウィンドウ方式の信頼性解析

表1  $E(n_2)$ を最大にする最適ウィンドウサイズ  $n_2^*$  ( $k=2$  のとき)  
 Table 1 Optimal window size  $n_2^*$  to maximize  $E(n_2)$  when  $k=2$ .

$p_1$	$p_2$	$d/b$	$n_2^*$	$\ell_0(n_2^*)$	$E(n_2^*)$	
0.1	0.05	2	66	48.6803	2.71157	
		4	70	58.2817	2.40213	
		6	74	68.5127	2.16018	
		8	78	79.3791	1.96525	
		16	90	123.814	1.45379	
		32	104	215.185	0.96661	
		0.1	2	36	37.9711	1.89618
	4		38	45.475	1.67125	
	6		39	52.1523	1.49562	
	8		41	60.5302	1.3547	
	16		45	91.1515	0.987368	
	32		49	152.224	0.64379	
	0.2		0.1	2	18	42.6974
		4		19	51.8385	0.733045
6		20		61.6159	0.649183	
8		21		72.0376	0.583029	
16		23		110.693	0.415563	
32		26		195.772	0.265615	
0.2		2		9	32.7915	0.548923
		4	10	41.8558	0.477831	
		6	10	47.3457	0.422425	
		8	10	52.8356	0.378533	
		16	11	81.9202	0.268554	
		32	12	140.981	0.170236	
		0.3	0.1	2	15	54.8185
4				17	71.027	0.478691
6	18			84.4996	0.426038	
8	19			98.8436	0.384446	
16	23			164.933	0.278902	
32	28			306.821	0.182517	
0.2	2			6	35.9464	0.33383
	4		6	41.7099	0.287701	
	6		6	47.4735	0.252773	
	8		6	53.237	0.225407	
	16		7	88.2839	0.158579	
	32		7	141.076	0.0992371	
	0.3		2	4	31.3912	0.254849
4			4	36.2887	0.220454	
6		4	41.1863	0.194239		
8		4	46.0839	0.173597		
16		4	65.6742	0.121814		
32		4	104.855	0.076296		

輾轉を考慮した可変ウィンドウ方式の信頼性解析

表2  $E(n_2)$ を最大にする最適ウィンドウサイズ  $n_2^*$  ( $k=3$  のとき)  
 Table 2 Optimal window size  $n_2^*$  to maximize  $E(n_2)$  when  $k=3$ .

$p_1$	$p_2$	$d/b$	$n_2^*$	$\ell_0(n_2^*)$	$E(n_2^*)$	
0.1	0.05	2	356	113.609	6.26711	
		4	403	136.379	5.90999	
		6	446	159.028	5.60908	
		8	486	181.7	5.34947	
		16	626	273.808	4.57255	
		32	851	467.241	3.64266	
	0.1	0.1	2	193	73.1701	5.27538
			4	214	87.8539	4.87172
			6	233	102.621	4.54097
			8	249	116.834	4.26245
			16	301	173.718	3.46539
			32	369	286.206	2.57856
	0.2	0.1	2	81	79.5201	2.03722
			4	92	100.347	1.83363
6			102	121.844	1.67427	
8			111	143.691	1.54498	
16			140	233.7	1.19812	
32			184	433.339	0.849219	
0.2		0.2	2	36	49.6148	1.45118
			4	40	63.1054	1.26772
			6	42	74.4301	1.12858
			8	45	88.3077	1.01916
			16	52	140.41	0.740645
			32	60	247.418	0.485009
0.3		0.1	2	78	130.478	1.1956
			4	98	179.289	1.09321
	6		116	228.487	1.01538	
	8		132	277.059	0.952866	
	16		178	454.745	0.782856	
	32		233	778.081	0.598909	
	0.2	0.2	2	20	52.2225	0.765953
			4	22	67.0239	0.656482
			6	24	83.303	0.576209
			8	25	97.2409	0.514187
			16	29	160.006	0.362487
			32	34	295.144	0.230396
	0.3	0.3	2	12	41.4807	0.578582
			4	13	53.0082	0.49049
6			14	65.6304	0.426632	
8			15	79.3752	0.377952	
16			16	122.855	0.26047	
32			18	222.302	0.161942	

輻輳を考慮した可変ウィンドウ方式の信頼性解析

表3  $E(n_2)$ を最大にする最適ウィンドウサイズ  $n_2^*$  ( $k=4$  のとき)  
 Table 3 Optimal window size  $n_2^*$  to maximize  $E(n_2)$  when  $k=4$ .

$p_1$	$p_2$	$d/b$	$n_2^*$	$\ell_0(n_2^*)$	$E(n_2^*)$	
0.1	0.05	2	1247	311.466	8.00729	
		4	1469	374.963	7.83545	
		6	1672	435.006	7.68725	
		8	1859	492.06	7.556	
		16	2496	699.541	7.1361	
		32	3496	1067.45	6.55021	
		0.1	2	751	198.094	7.34882
	4		862	234.595	7.34882	
	6		961	268.925	7.14698	
	8		1050	301.365	6.9683	
	16		1347	420.856	6.40124	
	32		1778	631.394	5.63199	
	0.2		0.1	2	227	153.589
		4		272	194.98	2.79003
6		314		236.565	2.65466	
8		354		278.692	2.54044	
16		499		451.905	2.20843	
32		750		824.066	1.82024	
0.2		2		109	87.4244	2.49358
		4	124	108.459	2.28657	
		6	137	129.205	2.12066	
		8	148	149.261	1.9831	
		16	183	229.034	1.59801	
		32	229	387.951	1.18056	
		0.3	0.1	2	203	250.335
4				357	461.799	1.54613
6	475			635.324	1.4953	
8	560			769.825	1.45488	
16	781			1168.15	1.33716	
32	1059			1785.69	1.1861	
0.2	2			54	85.3963	1.26469
	4		62	110.01	1.12717	
	6		70	136.949	1.02228	
	8		76	161.954	0.938535	
	16		96	267.195	0.718576	
	32		121	482.141	0.501928	
	0.3		2	31	60.3706	1.02699
4			35	78.4006	0.892851	
6		38	95.8283	0.793085		
8		41	114.612	0.715456		
16		49	188.386	0.520209		
32		58	338.976	0.342207		

輻輳を考慮した可変ウィンドウ方式の信頼性解析

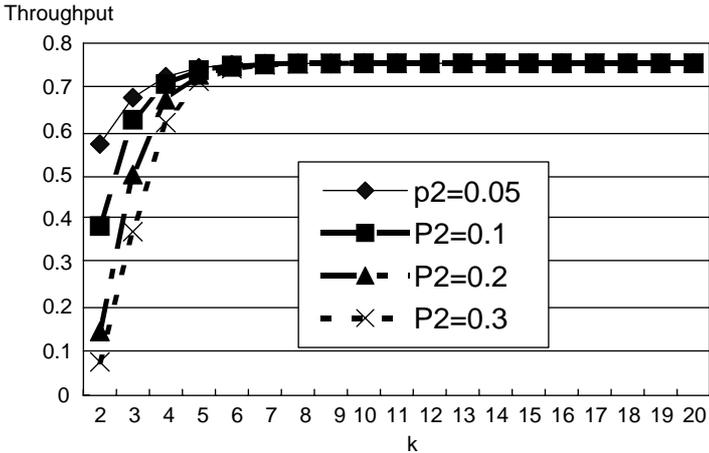


図2 p1=0.3, p2=0.05, 0.1, 0.2, 0.3, d/b=8, n2=19のときのスループット E(n2)の推移

5 おわりに

ネットワーク上に発生する輻輳を考慮して可変ウィンドウ方式のモデルを設定し、その信頼性解析を行った。すなわち、WWWサーバはウィンドウサイズに相当するパケット数を順次クライアントへ送信し、誤ったパケットは再送する。再送がk回失敗したならば、輻輳が発生しているものとみなし、ウィンドウサイズからその半分に当たるパケット数を再送すると仮定した信頼性モデルを設定した。ネットワーク上に輻輳が発生した時のパケット損失確率等を考慮して、送信成功となるまでの平均時間を解析的に導出し、さらに、スループットを最大にする最適方策を議論した。

数値例による考察から、スループットを最大にする最適なウィンドウサイズは、輻輳が発生した時のパケット損失確率が大きくなるにつれて減少し、また、再送回数がある程度大きくなるとスループットは大略一定の値に収束する傾向を示した。

このようなネットワーク上に発生する輻輳に対応したデータ通信の高信頼化の問題は、今後ますます重要な課題となることが考えられ、この方面に対する多くの研究が期待される。

参考文献

- [ 1 ] H. T. Kung and A. Chapman, "Credit-based flow control for ATM networks: Credit update protocol, adaptive credit allocation, and statistical multiplexing," *ACM SIGCOMM Computer Communications Review*, vol. 24, pp. 101-114, October 1994.
- [ 2 ] H. Ohsaki, M. Murata, H. Suzuki, C. Ikeda, and H. Miyahara, "Rate-based congestion control for ATM networks," *ACM SIGCOMM Computer Communication Review*, vol. 25, pp. 60-72, April 1995.
- [ 3 ] 田沢力, 山内長承 "Forward Error Correction による損失パケット回復の性能評価", インターネットコンファレンス, 2001.
- [ 4 ] 情報処理学会編, "新版情報処理ハンドブック", オーム社, P2513, 1995.
- [ 5 ] S. Osaki, "Applied Stochastic System Modeling", Springer-Verlag, Berlin, 1992.

(提出期日 平成15年12月10日)