

# 異速度パケットトレインペアを用いた 可用帯域のアクティブ測定法の提案

長谷川 和音 加藤 陽介 小原 仁

秋田大学 工学資源学部 電気電子工学科

〒010-8502 秋田市手形学園町 1-1

E-mail: obara@ee.akita-u.ac.jp

あらまし ネットワークの End-to-End 通信路における可用帯域を測定する方法として、プローブと呼ばれる試験用パケットをネットワーク内に送出し、プローブが被る品質からパスの状態を推定するアクティブ測定法がある。一般にアクティブ測定法では送出プローブ量（あるいは測定時間）と推定精度の間にトレードオフの関係があり、現在もこの問題に対する様々な研究がなされている。本稿では、複数のプローブを収容したトレインを基本測定単位とし、異なる速度を持つトレインのペアを送出する方法を新たに提案する。また、最初のトレインによる測定結果を次のトレイン速度に反映させることで、必要となるプローブパケット数の削減および測定時間の短縮を図る。シミュレーションによって1ホップモデルにおける性能を評価し、提案方式の有効性を示す。

キーワード 可用帯域, アクティブ測定法, プローブ, パケットトレイン

## 1. はじめに

近年、インターネット上で電話やテレビ、あるいはビデオ会議など、通信品質が重要とされるアプリケーションが展開され普及しつつある。それに伴い、ユーザ自身がネットワークサービス品質をモニタしたいという要求が高まっている。そのためにはネットワークの End-to-End 通信路における可用帯域の測定技術が必須である[1]。

ネットワークの End-to-End 通信路における可用帯域の測定法として、プローブと呼ばれる試験用パケットをネットワークに送出し、プローブが被る品質からネットワークの状態を推定するアクティブ測定法がある[2]。一般にアクティブ測定法は、送出プローブ量を増やせばそれだけ高い推定精度を得ることができるが、多量のプローブはネットワークへの余分な負荷となる。このように送出プローブ量と推定精度の間にはトレードオフの関係がある。現在、2つのプローブパケットをペアとして用いるパケットペア方式、あるいは複数のプローブからなるパケットの列をひとまとまりのパケットトレインとして送出する方式などが開発されている。しかし、上記のトレードオフの問題を解決する測定技術は十分に確立されているとはいえない[3]。

そこで本研究では、速度の異なる2つのトレインをペアで送出する新たな測定法を提案する。また、トレインをバーストモードで転送することで得られるADR (Asymptotic Dispersion Rate) の測定値を、トレインごとに送信側にフィードバックし、次のトレイン速度とすることで測定時間の短縮を図る。本稿では簡

単なシミュレーションモデルで本方式の性能の評価を行い、その有効性を示す。

以下、2章では可用帯域の定義を行い、3章ではその測定法である従来方式の特徴と問題点を述べる。4章からは本研究における提案方式であるパケットトレインペアの原理を説明しシミュレーションにより性能を評価する。5章では、パケットトレインペアの欠点を補うためのADRによる測定値の更新の説明とシミュレーションによる性能の評価を行い、6章で本稿の結論をまとめる。

## 2. 帯域幅

最初に、インターネットの End-to-End 通信路における帯域幅について定義する。各リンク  $i$  における帯域幅を  $C_i$  とすれば、End-to-End 通信路における帯域幅  $C$  は、End-to-End 通信路に沿ったパスにおける各リンク幅の最小値

$$C = \min_i C_i \quad (1)$$

で与えられる。すなわち無負荷時に得られるそのパスの最大スループットといえる。一般に End-to-End 通信路における最小の帯域幅を持つリンクを narrow link と呼ぶ[5]。

次に、可用帯域を定義する。可用帯域とは、あるリンクにおける空帯域、すなわちそのリンクにおける利用可能な帯域のことである。例えば、リンク容量  $C_i$  を持つリンク  $i$  においてある区間  $\tau$  中の可用帯域  $A_{i,\tau}$  は、

そのリンクにおけるクロストラヒックの占有帯域を  $U_{i,\tau}$  とすると

$$A_{i,\tau} = C_i(1 - U_{i,\tau}) \quad (2)$$

で与えられる。また End-to-End 通信路における可用帯域は、ホップ数を  $H$  とすればそのパス上における最小の可用帯域であり、区間  $\tau$  中の End-to-End 通信路における可用帯域を  $A_\tau$  とすると

$$A_\tau = \min_{i=0 \dots H} A_{i,\tau} \quad (3)$$

と表すことができる。一般に End-to-End 通信路における最小の可用帯域を持つリンクを tight link と呼ぶ[5]。

### 3. 従来のアクティブ測定法

アクティブ測定法は、2つのプローブをペアとして送出するパケットペア方式と、複数のパケットをひとまとまりとして送出するパケットトレイン方式に大別することができる。これまで上記の方式をベースとした様々なプローブ転送方式が研究開発されてきたが、ここでは、そのいくつかの測定法を紹介する[6,7,8]。

#### 3.1. Spruce

Spruceでは、パケットペア方式を用いてプローブを送信し推定を行う。図1に測定メカニズムを示す。Spruceでは、複数個のパケットペアをポアソン過程で送出し、その到着間隔から可用帯域を推定する。例えば、End-to-End 可用帯域を持つリンク  $i$  の物理帯域幅を  $C_i$ 、送信間隔を  $\Delta_{in}$ 、到着間隔を  $\Delta_{out}$  とすると、可用帯域  $A$  は、

$$A = C_i \left( 1 - \frac{\Delta_{out} - \Delta_{in}}{\Delta_{in}} \right) \quad (4)$$

から求めることができる。パケットトレイン方式と比較プローブの送出量はネットワークに対する負荷は軽い。しかし、パス中の narrow link が既知であること、及び narrow link と tight link は同じリンクであるなどの測定条件があるため、利用範囲が制限される可能性がある。

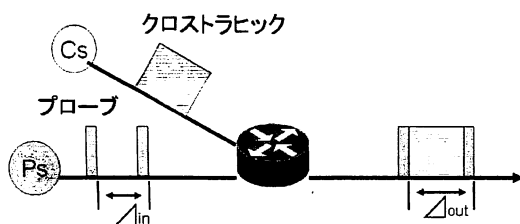


図1. Spruce の原理

#### 3.2. Pathload

Pathload ではプローブ送信レートを同一としたパケットトレイン方式を用いて推定を行う。はじめにトレイン内のプローブ送信レートを  $V1$  で送信する。測定対象パスの可用帯域を  $A$  とすると、トレイン内のプローブの片道遅延時間 (One Way Delay, OWD) が増加傾向であれば、 $A \leq V1$  と判断し、次のトレインの送信レートを  $V2 < V1$  として送信する。トレイン内のプローブに OWD 増加傾向がなければ、 $A \geq V1$  と判断し、次のトレイン送信レートを  $V2 < V1$  とする。このように Pathload では二分探索で可用帯域を狭め込むように推定していく。推定精度は非常に優れているが、測定時間が長く、それに伴いネットワークにかかる負荷も他の推定方式と比較して大きい。

#### 3.3. IGI (Initial Gap Increasing)

IGI においてもプローブ送信レートを一定としたパケットトレイン方式を用いる。送信側でのプローブ送信間隔を  $T_1$  としたパケットトレインが送出され、受信側でプローブが不均一に広がり受信される。このときの各々のプローブ間の平均受信間隔を  $T_{out}$  とする。送出するパケットトレインごとに送信間隔を徐々に広げていき、送信間隔  $T_n \approx$  平均受信間隔  $T_{out}$  となるように複数のパケットトレインを送出する。送信間隔を変化させる際の増減幅をギャップステップと呼ぶ。このギャップステップを大きく取れば送出プローブ量が少なくできるが、推定点があいまいになり、推定精度が劣化する恐れがある。Pathload と比較して軽負荷であるが精度の面で劣る。

#### 3.4. PathChirp

PathChirp では、トレイン中のプローブ間隔を指数関数的に減少させたパケットトレインを用いる。トレイン内の送信レートを次第に増加させ、ある送信レートが可用帯域を超えた場合それ以降のトレインの OWD は増加していくことになる。この OWD の増加し始めたトレイン内の臨界点のパケットペアの間隔とプローブサイズから送信レートを求め、可用帯域の推定値とする。このように、1トレイン内のプローブ間隔を変化させながら可用帯域をスキャンするように測定するため、ばらつきを無視すれば1トレインで測定可能である。他のパケットトレイン方式と比較すると、非常に軽負荷な推定を可能とする。しかし、突発的なクロストラヒックの変動などにより臨界点を誤る可能性が高く、精度の面では劣る。

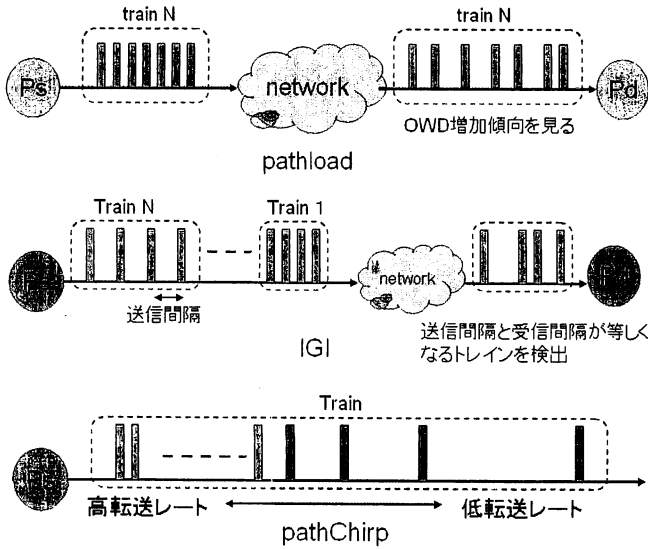


図 2. 既存のパケットトレイン転送方式

#### 4. プロブトレインペア (PTP) 方式の提案

##### 4.1. 推定原理

本提案方式においてもトレイン方式を採用する。本方式では速度の異なるトレインを2つ、ペアとして送出しその分散を受信側で測定する。トレイン内のプロブパケット間隔は一定である。図4において、最初に送出されるトレインを1stトレイン、2つ目に送出されるトレインを2ndトレインとする。送信側での1stトレイン速度を $v_1$ 、持続時間を $t_1$ 、2ndトレイン速度を $v_2$ 、持続時間を $t_2$ とし、受信側での1stトレイン持続時間を $t'_1$ 、2ndトレイン持続時間を $t'_2$ とする。 $v_a$ を可用帯域 $v_c$ をクロストラヒックによる占有帯域であるとすると、以下の連立方程式が成り立つ。

$$(v_1 + v_c) \cdot t_1 = (v_a + v_c) \cdot t'_1 \quad (5)$$

$$(v_2 + v_c) \cdot t_2 = (v_a + v_c) \cdot t'_2 \quad (6)$$

この方程式を可用帯域 $v_a$ について解くと、次式が得られる。

$$v_a = \frac{(t_1 - t'_1) \cdot t_2 \cdot v_2 + (t'_2 - t_2) \cdot t_1 \cdot v_1}{t_1 \cdot t'_2 - t_2 \cdot t'_1} \quad (7)$$

この値を可用帯域の推定値とする。

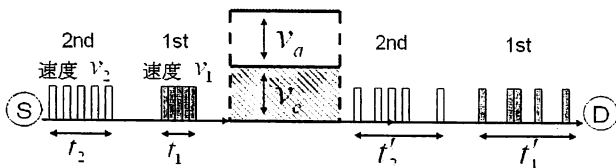


図 3. パケットトレインペアの原理

##### 4.2. 性能評価

提案方式の基本的な性能を評価するため、最も単純

な1ホップリンクでの動作を、シミュレーションによって評価した。図4にシミュレーションモデルを示す。モデルはプロブとクロストラヒックにそれぞれ送受信端があり、2つのルータには含まれたリンクを測定対象リンクとし、リンク速度は100Mbpsとする。ルータはスケジューリングをFIFO (First In First Out)とし、バッファサイズを無限大と仮定する。

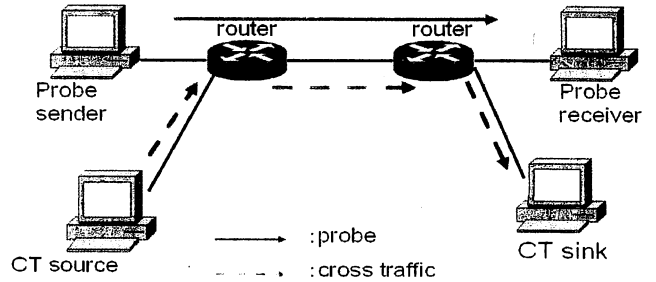


図 4. シミュレーションモデル

クロストラヒックはパケットサイズを1500bytes固定とし速度を一定レートとした。プロブサイズはクロストラヒックパケットサイズと同様に1500bytes、1トレイン当りのパケット数を10pkts/trainとし1stトレインの速度 $v_1$ を100Mbps、2ndトレインの速度 $v_2$ を90, 80, 70, 60Mbpsのそれぞれについて1回だけの測定を行った。図5に測定結果を示す。横軸はクロストラヒックによる負荷、縦軸は可用帯域である。 $v_2=90$ Mbpsの場合について着目すると、ばらつきが大きくマイナスの値を返しているプロット点まで見られる。しかし $v_2$ を小さくするにつれ、理論値に沿った推定ができていくことがわかる。しかし $v_2$ よりも大きな値を持つ可用帯域はうまく測定できていないことが読み取れる。

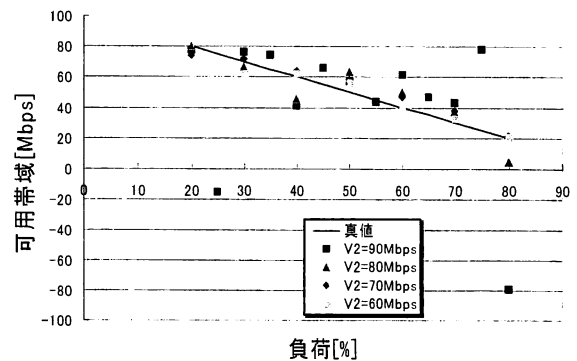


図 5.  $v_2 = \{90, 80, 70, 60\}$  Mbps, 10pkts/train

次にパケットサイズ及び、1トレイン当りのプローブパケット数に着目し、シミュレーションを行った。クロストラヒックのパケット長を1000bytes固定とし、プローブのサイズがクロストラヒックパケットサイズよりも小さい場合(500bytes), クロストラヒックパケットサイズよりもサイズが大きい場合(1500bytes) についての評価及び、1トレインあたりのパケット数を10pkts/train 及び、50pkts/trainとした場合の測定を行い、それぞれ100回の測定サンプルを取得し比較した。トレイン速度は  $v_1=100\text{Mbps}$ ,  $v_2=90\text{Mbps}$  とした。

それぞれの測定結果を図6から図9に示す。横軸は測定サンプル数、縦軸は可用帯域を表す。測定値のばらつきに起因する200Mbpsを超える推定値は200Mbpsに、マイナスの推定値は0Mbpsにプロットした。測定結果より、小さいプローブよりも大きいプローブを用いた場合、ばらつきが抑えられることがわかる。これはプローブサイズが小さい場合パケット間隔が短くなり、クロストラヒックによる変動が相対的に大きくなるためと考えられる。

この傾向は従来のIGI方式を用いた実験結果と同じ傾向を示している。また、トレイン長を長くした場合全体的にばらつきが抑えられている。特に大きく外れていたプローブサイズ500bytesの場合においては、全ての推定値が100Mbps以内にプロットされている。これはトレイン持続時間を長くすることにより、クロストラヒックの重なり合いのタイミングのばらつきの影響が抑えられたためと考えられる。また全体的な傾向として、トレイン速度と可用帯域との差が大きいほどばらつきが大きくなることが読み取れる。

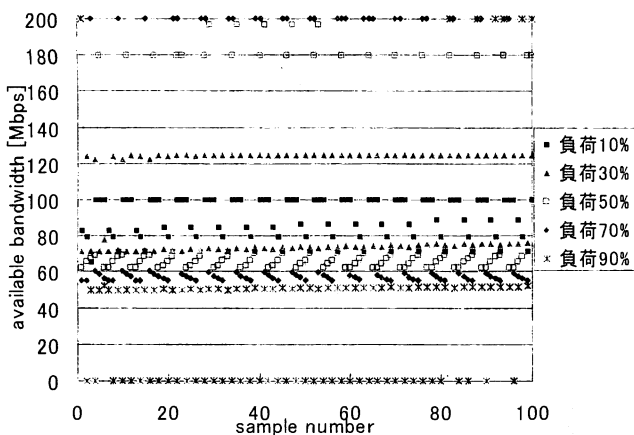


図6. probe size = 500bytes, 10pkts/train

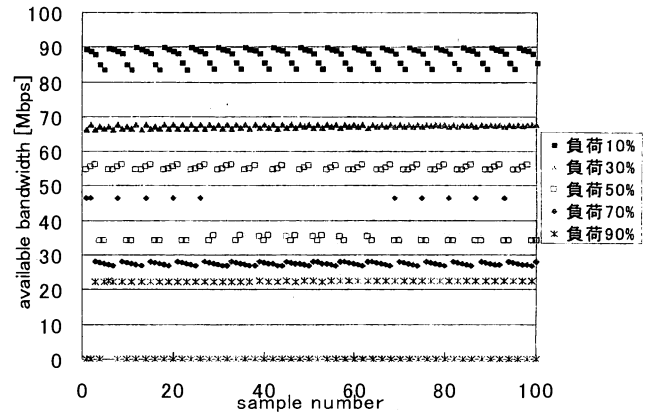


図7. probe size = 500bytes, 50pkts/train

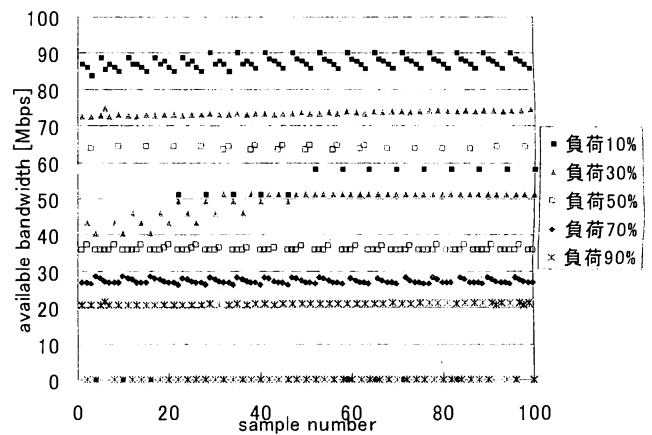


図8. probe size = 1500bytes, 10pkts/train

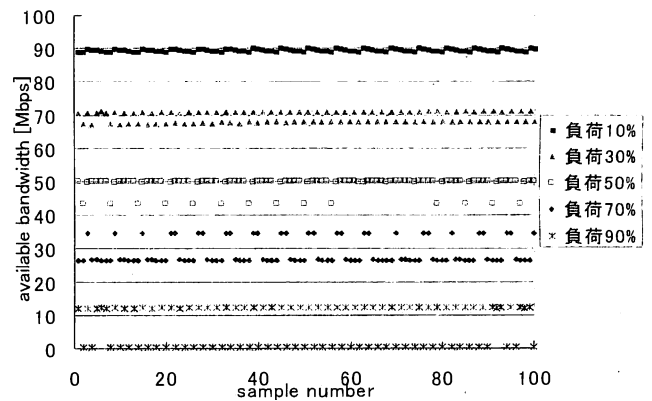


図9. probe size = 1500bytes, 50pkts/train

次に、提案方式を実環境ネットワーク上のクロストラヒックに近いリアルトラヒックモデルを用いて、提案方式を評価した。パケットサイズを100bytes及び1500bytesとし、100bytesパケットが全体の60%、1500bytesパケットが全体の40%を占める。発生分布は指数分布とした。評価に用いるパケットトレインペアは、各々のトレイン速度を  $v_1=100\text{Mbps}$ ,  $v_2=90\text{Mbps}$ , プローブパケット長は1500bytes, 1トレイン当りのプ

ローブ数を 50pkts/train とした。

図 10 に測定結果を示す。横軸は測定サンプル数で、縦軸が可用帯域である。この図より、負荷が 10% のとき以外の負荷に対しては、非常に大きなばらつきが生じる。負荷が 10% の場合は、 $v_2=90\text{Mbps}$  であり、トレイン速度=可用帯域であったため、比較的正確な測定ができたのではないかと考えられる。提案方式を用いて、より正確な推定を行うためには、可用帯域に近い速度を持つパケットトレインを用いた測定が有効と考えられる。次章において、そのアプローチに基づく改良方法について述べる。

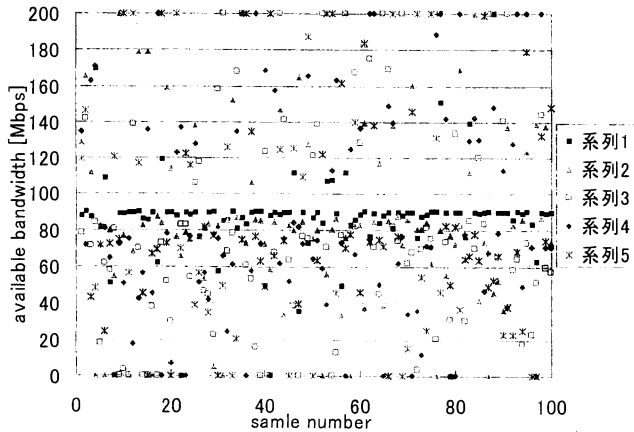


図.10 probe size = 1500bytes, 50pkts/train

## 5. ADR 測定値の更新

プローブトレインペアの速度を可用帯域に近づけるため、プローブトレインをバースト転送することにより得られる ADR (Asymptotic Dispersion Rate) というパラメータを用いて、プローブトレインペアの速度を決定する方式を提案する。

### 5.1. ADR

まず、はじめに ADR の原理について説明する。ADR は一定レートで長さ  $L$  bytes のプローブ  $N$  個で構成されたトレインを用いる。受信側において到着したトレインの最初のプローブの先頭 bit から最後のパケットの先頭 bit までの分散  $\Delta$  より、ADR は、

$$ADR = \frac{(N-1)L}{\Delta} \quad (8)$$

によって求めることができる。この ADR の値はそのまま可用帯域の値として用いることはできない。例えば図 13 のように 1 ホップパスの場合において各リンク

の帯域幅を  $C_0, C_1$  ( $C_0 \geq C_1$ ) とした場合を考える。

$$\Delta_0 = \frac{(N-1)L}{C_0} \quad (9)$$

となる。リンク 1 に流れ込むクロストラヒック速度を  $R_c$  とすると、 $\Delta_0$  の間に流れ込むクロストラヒック量を  $X_1$  は、 $X_1 = \Delta_0 \cdot R_c$  となる。受信側でのパケットトレイン到着間隔は

$$\Delta_1 = \frac{(N-1)L + X_1}{C_1} = \frac{(N-1)L}{C_1} \left( 1 + \frac{R_c}{C_0} \right) \quad (10)$$

となり、これより ADR は、

$$ADR = \frac{(N-1)L}{\Delta_1} = \frac{C_1}{1 + \frac{R_c}{C_0}} \quad (11)$$

となる。可用帯域  $A$  は  $A = C_1 - R_c$  であるので基本的には  $A \neq ADR$  である。また ADR はパケットトレインが高送信速度であれば、可用帯域よりも大きな値として測定値が得られる。例えば本稿のシミュレーションモデルの場合を用いれば  $C_0=C_1=100\text{Mbps}$ 、クロストラヒック速度を  $70\text{Mbps}$  とすると、可用帯域は  $30\text{Mbps}$  であるのに対して、ADR は、式(11)を用いて、 $58.8\text{Mbps}$  となる。また ADR の求まる範囲はパケットトレイン長  $N$  に依存しており、パケットトレイン長を長くすればするほど、測定範囲がまとまる特徴を持つ。

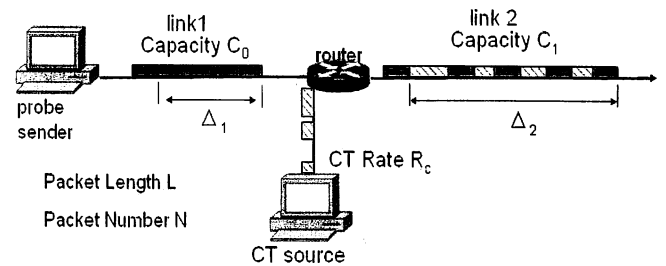


図.11 ADR の原理

### 5.2. 測定値の更新動作

上で示したように、ADR は可用帯域よりも大きな値が得られ、その値以下の範囲に可用帯域が存在するという目安を付けることができる。さらに、ADR の値を更新することによって、可用帯域に収束させる。すなわち、図 12 のように ADR の値を送信側にフィードバックし、次に送信するプローブトレインの速度として反映させる。これを複数回繰り返すことによって可用帯域に測定値が収束していくことが期待される。ADR の測定範囲のばらつきを抑えるため、プローブトレインは、ある程度の長さを必要とする。そこで 1 回の送

信につき同一転送レートを持つパケットN個で構成されたプローブトレイン M 個を持つフリート単位で送信し、フリート内の各々のプローブトレインから得られた ADR を平均して次に送信するフリート内のプローブトレインのレートとする。まず、シミュレーションを用いて単純に測定値を更新していった場合について検討を行った。

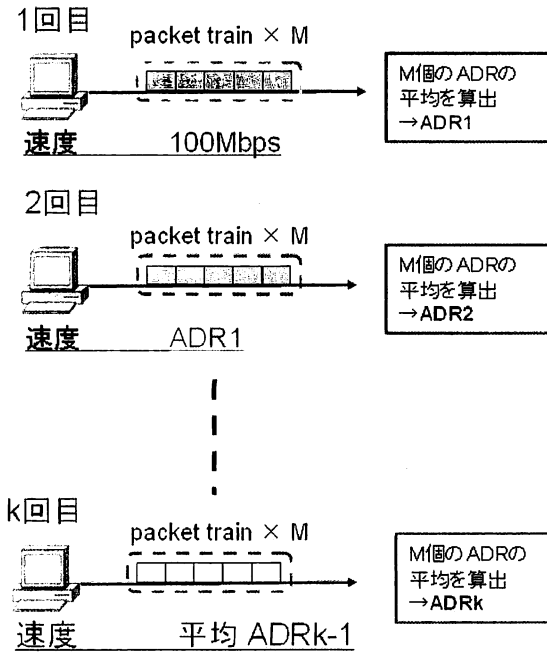


図 12. 測定値の更新動作

### 5.3.シミュレーション結果

シミュレーションモデルは図4のワンホップモデルを用いて行った。クロストラヒックは、前章で用いたリアルトラヒックモデルに従う。負荷は10%から20%刻みで90%まで行う。プローブサイズは1500bytes, 1回の送信につき1トレインあたり50個とした同一転送レートを持つパケットトレインを送信する。トレイン速度の初期値は100Mbpsとする。送信回数を15回とし、試行回数を100回とした。

値を更新していくごとに得られる ADR の測定値の平均値をプロットしたグラフを図13に示す。横軸が送信回数、縦軸が ADR である。また、可用帯域の理論値に対する誤差の変化を示したグラフを図14に示す。横軸が送信回数、縦軸が可用帯域に対する相対誤差である。負荷10%の場合は1回, 30%は2回, 50%は4回, 70%は7回, 負荷90%の場合は15回のフリート送信を行っても理論値に到達できていないことが分かる。このように負荷によって理論値に到達するまでの送信回数は大きく異なる。また理論値に到達したとしても、さらに更新していくことによって ADR の値は理論値を下回り緩やかに下降していくことが読み取れる。こ

れは、可用帯域を下回る速度でプローブトレインを送信したとしても、クロストラヒックにより多少間隔が広げられ、ADR の値は減少するためであると考えられる

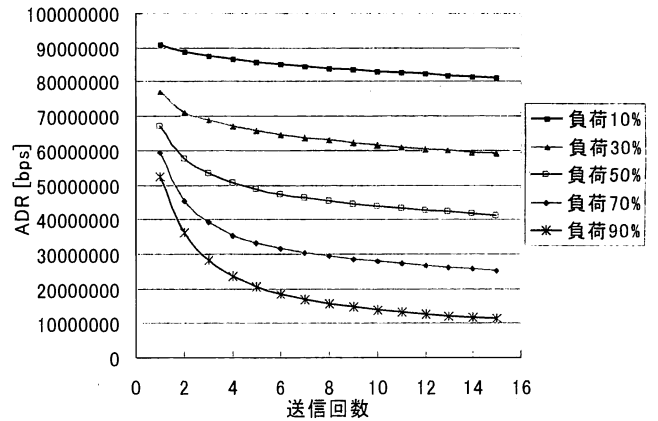


図.13 ADR の測定値の平均

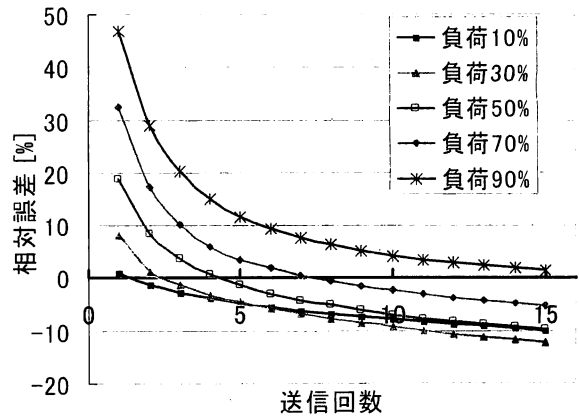


図.14 可用帯域に対する ADR 測定値誤差の遷移

このように単純な ADR 測定値を用いた更新のみによる可用帯域の測定は、精度の面、負荷の面どちらにおいても有益ではない。しかし、数回のフリートの送信による測定の更新により、ADR の値を可用帯域に近づけることは可能である。

### 5.4 ADR と PTP の併用

フリートからは、プローブトレインの平均 ADR 及びプローブトレイン平均持続時間の二つのパラメータを得ることができる。例えば、最初のフリートの平均トレイン持続時間と K 回目に送信したフリートの平均トレイン持続時間を異速度 PTP の分散として式(7)を適用することにより可用帯域が決定される。また K 回目に送信する場合、K-1 回目の ADR を送信速度として反映させるため、可用帯域に近い速度としてフリートを送信できるため、重い負荷であっても高精度な推定が期待できる。

### 5.5 シミュレーションによる評価

最初に、シミュレーション条件を4章の図10における測定条件と同様に設定した。すなわちフリート当りのトレイン数は1とし、1トレイン当りのパケット数を50とした。トレインの送信回数は2回とし、送信パケット総量を等しくした。測定方法は、1回目に得られたADRの値を2回目のトレイン速度として送信する。2回のトレイン送信から得られた持続時間から式(7)を用いて可用帯域を求める。この方法による測定結果を図15に示す。4章の図10における結果と比較すると、まだ大きなばらつきがあるものの、それぞれの理論値に近い値にまとまった測定ができています。しかし、重い負荷の場合には、トレインと可用帯域との値に差があるため、ばらつきが大きくマイナスの値を返す測定点も多い。

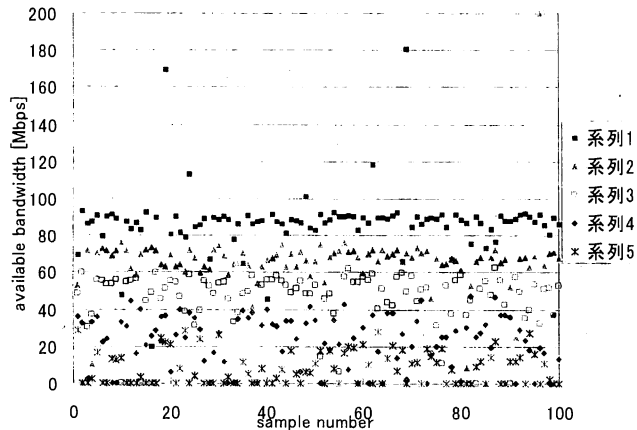


図15. 送信回数2回, M=1

次に、送信するパケット数を2倍とし、送信回数Kを増やした場合、及びトレイン数Mを増やした場合についてそれぞれ評価を行った。送信回数2回, M=2とした場合の測定結果を図16に、送信回数4回, M=1とした場合の測定結果を図17に示す。図16を見ると、軽負荷に対する推定値は全体的にまとまってきた。しかし、更新回数が少ないため重い負荷に対する測定結果はマイナスの値が多く、測定結果が不安定である。一方、図17を見ると、更新回数を多くすることによって、重い負荷の場合でも比較的安定した測定結果となっている。これは更新回数を増やすことによって、得られるADRの測定値が可用帯域に近くなり、そのADR測定値でトレインを送出したため、重い負荷にも対応できたと考えられる。ただし、軽負荷に対しては、得られるADR測定値が可用帯域を下回るため、理論値より過小評価となることが分かる。このため提案方式では、1度に送信するトレインのパケットを費やすかあるいは更新回数の方にパケットを費やすかを、負

荷に応じて選択する必要がある。これらの問題については今後の検討課題である。

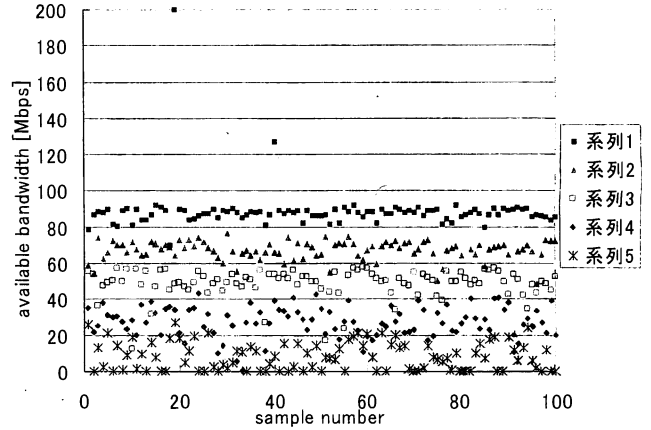


図16. 送信回数2回, M=2

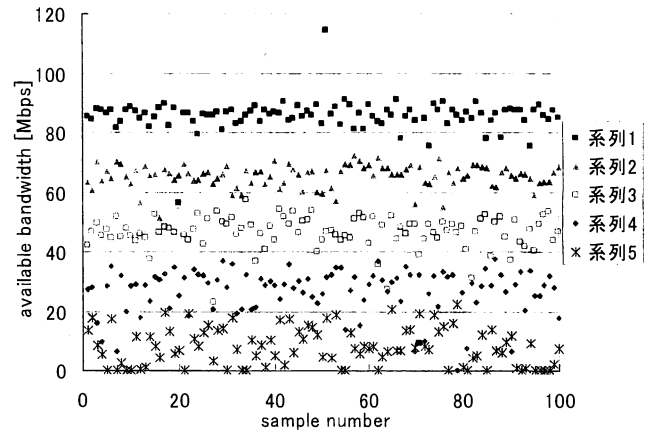


図17. 送信回数4回, M=1

### 6. まとめ

本稿では、アクティブ測定法を用いた新たなEnd-to-End通信路における可用帯域測定法を提案した。提案方法は、2つの異速度のプロブトレインのみを用いて測定が可能のため、非常に軽負荷な測定が可能である。本稿では提案方式の基本的な性能を明らかにするため、1ホップモデルについてシミュレーションを行った。

今回のシミュレーション結果から明らかになった本方式の特徴を以下にまとめる。

- 2つのトレイン速度の差をある程度取ったほうが、実際の可用帯域に近い推定ができる。
- トレイン速度よりも大きな可用帯域は測定できない。
- 理論値に対するばらつきを抑えるためにはある程度のパケットトレイン長が必要である。
- ADRを用いて測定レートを更新する方法は、測定結果の収束の様子が負荷によって異なる。また、更新回数が増えると誤差が増大する場合がある。

## 文 献

- [1] 鶴正人, 尾家祐二, "インターネット品質管理における計測技術の最新動向" 信学技報, NS2003-153, pp.37-42. 2003
- [2] 石井純, 川原憲治, 鶴正人, 尾家祐二, "プローブ遅延時間の相関を利用したアクティブ計測による可用帯域推定手法の提案とその評価," 信学技報, IN, Vol.103, no. 573, pp.73-78, 2004
- [3] Dovrolis, P. Ramanathan, and D. Moore. What do packet dispersion techniques measure? In *Proc. IEEE infocom*, Anchorage, April, 2001
- [4] M. Jain and C. Dovrolis. End-to-end available bandwidth: measurement methodology, dynamics, and relation with tcp throughput. In *Proc. ACM SIGCOMM*, Pittsburg, Aug. 2002.
- [5] N. Hu and P. Steenkiste. Evaluation and characterization of available bandwidth probing techniques. *IEEE J-SAC*, 21, Aug. 2003.
- [6] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, and L. Cottrell. Pathchirp: Efficient available bandwidth estimation for network paths. In *Proc. PAM workshop*, La Jolla, California, Apr. 2003.
- [7] J. Strauss, D. Katabi, F. Kaashoek, "A Measurement Study of Available Bandwidth Estimation Tools", *ACM SIGCOMM Internet Measurement Workshop*, 2003.
- [8] R.S. Prasad, M. Murray, C. Dovrolis, K. Claffy: "Bandwidth estimation: metrics, measurement techniques, and tools" . *IEEE Network Magazine* ,2003