

# インターネットにおける可用帯域幅の推定と伝送遅延の評価

M2004MM047 吉田 秀考

指導教員 長谷川 利治

## 1 はじめに

近年、動画や音声などのコンテンツ提供が盛んになり、帯域を消費するサービスが増えた。このようなサービスを安定して提供するには、発信と受信ホスト間で利用可能な帯域幅と遅延を把握する必要がある。これまで、帯域幅を推定するさまざまなツール [1, 2, 5] が提案されてきた。そして伝送遅延、すなわち RTT(Round Trip Time) を測定、推定するツールとして、ICMP パケットを利用した ping がある。しかし、既存ツールでは推定に時間が掛かったり、帯域幅と伝送遅延を同時に推定できるものがないといった欠点がある。

そこで本研究では、家庭などでのインターネット利用を想定し、低速で非対称なネットワークパスの可用帯域幅の推定と伝送遅延の総合評価に焦点をあてる。そして、短時間にかつネットワークへ低負荷な推定ができるツールを提案する。具体的には、パケットペアによる容量推定 [1]、ワンパケット推定法による帯域推定 [3] のアイデアを基に、ネットワークパスの容量推定と可用帯域幅の推定、そこから得られた値を基に伝送遅延を推定する機能を統合したものである。

ネットワークアプリケーションが送信すべき相手までの帯域幅と遅延を把握できることで、非 TCP アプリケーションのピーク送信レート制限や複数経路で分散伝送する際の振り分けアルゴリズムにも応用が可能である。

## 2 可用帯域幅と伝送遅延の推定

この節では、可用帯域幅の説明と既存の推定ツールの推定方法、伝送遅延の推定について説明する。

### 2.1 容量と可用帯域幅

コンピュータネットワークは複数のホストが接続され、リンクを共有するようになっている。発信ホストと受信ホスト間のデータは、それらを結んだ経路(パス)を流れる。パスに流すことができるデータ量は、どの経路を結んだかによって決まる。

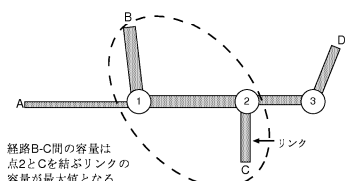


図1 パスとリンク

図1の網部分は、それらの太さによって端末AからDと点1から点3を結んだリンクの容量(capacity)を表している。パスの容量は結んだ各リンク容量なので、端末Bから端末Cのパスでは、点2とCを結ぶリンクが最小となり、これがボトルネックリンクとなる。パスの容量から共有リンクを流れるトラフィックを差し引いた分を可用帯域幅(available bandwidth)という。

容量については、パスの構成が変化しない限り変わらないが、可用帯域幅は時間によって回線の利用量が上下するため、絶えず変化する。よって、可用帯域幅の推定値は推定時間、所要時間、推定間隔との関係が深い。

### 2.2 active・passive 観測

現在研究が進められている可用帯域幅推定の手法はpassiveとactiveの2つに分けられる。

- 経路上のある地点に観測機器とツールを配置する(passive)
- 発信ホストから受信ホストへ検査パケットを送信(active)

passiveでは、測定機能を搭載した中継機器によって一定時間で流れたデータ量をカウントし、その情報から様々なネットワークの状態を把握する。SNMPをベースとしたMRTG(The Multi Router Traffic Grapher)が例として挙げられる。しかし、この情報を得られるのは限られたユーザであり、情報提供できる機器を設置しなくてはならない。また、情報は一定間隔で更新されるため、素早く把握するには困難である。また、特定の箇所しか推定できない。

ネットワークの状態を把握するため、activeではパスの端点から何らかの検査(probe)パケットを送信し、その挙動を観測して解析する。これは、レーダーが物体の影をとらえる動作やCTスキャンで断層を解析することによく似ている。activeはpassiveのように情報提供用の機器を設置する必要がないので、観測できるパスの制限が基本的に無い。また、必要な時に状態を特別なユーザでなくても把握できる。activeには、検査パケットの送信方法や送信レートなどによって、さらに2つのパケット(ペア)送信する方法と連なったパケット(トレイン)、そして単一のパケットを送信する方法に分類される。

2つパケット(ペア)を送信する方法では、短い間隔を開けたパケットを受信側に向けて複数回送信し、受信側でペアの間隔観測することで、パスを通過したことによる間隔の広がり具合を検出する。広がり具合の分布からパスの容量を求める。

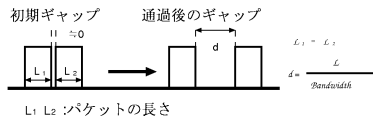


図2 パケットペアによる推定

連なったパケット (トレイン) の送信する方法では、ある長さを持つパケットの連なりを受信側に送信し、受信側でパケット間隔を観測する。パケット間隔の広がりを出し、それを手がかりに可用帯域幅等を推定するのはペアで推定するものと共通する。インターネット上はストア

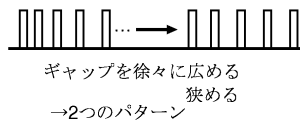


図3 パケットトレインによる推定

& フォワード方式のパケット交換網であるため、パケットを中継する際に、キューに納められる。そのため、検査パケットの送信タイミングやレートによって、中継点の処理能力を越えた場合、受信ホストに到着した検査パケットらの到着時刻に遅れが生じる。これが、推定手法の基本アイデアである。

これらの推定手法で問題となるのが、クロストラフィックとパスを構成する回線速度である。クロストラフィックが存在すると、中継ホスト到着前に検査パケットと検査パケットの間にクロストラフィックのパケットが割り込むことがある。割り込みが発生するとパケットの間隔が乱れ、推定値が過大・過小を示す。リンク速度を推定するには、クロストラフィックの影響を受けない環境で行うか、又は受けていない検査パケットペアを推定の対象にしなければならない。

パケットペアやトレインとは対称的に、検査パケットの間隔を求めず、1つの検査パケットを用いて転送速度、すなわち可用帯域幅を推定する手法がワンパケット推定法である。これは、検査パケットの往復時間 (RTT) を利用する。往復時間は、パケットサイズに応じて変化し、サイズが大きいほど大きくなる。これは、パケットの処理に掛かる時間や伝播遅延等を除く時間が、パケットサイズをパスの容量で割った時間になるからである。そのため、パケットサイズに比例する。

Click[3] は、リンク毎にサイズの異なる検査パケットでそれぞれの往復時間を測定し、そのデータの最小値を基に時間の傾きを最小二乗法によって計算する。そこで得られた傾きの逆数が、推定されるリンク速度であると考えるのがこの手法である。この手法の特長は、パケットペアでは弱点であった、クロストラフィックやパスを構成する回線速度、キューの影響を受けない点である。

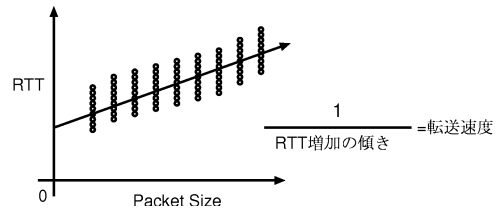


図4 ワンパケット推定法

### 2.3 伝送遅延の測定

伝送遅延の測定には、ICMP パケットを用いた ping が良く利用される。これは、ICMP エコー要求とエコー応答を組み合わせて動作する。送信側はパケットを送信する直前に時刻情報をペイロードに埋め込む。そして、受信側の応答からその情報を読み取り、応答を受信した時刻と照合してパケットの往復時間、すなわち RTT を測定する。

往復時間は ping の方法で測定できる。しかし、片道の伝送遅延を測定する場合に問題が生じる。ping では、往路、復路それぞれで送受信時刻を記録しないので片道統計が取れないからである。両ホストで時刻情報を取得する手法として ICMP や IP オプションのタイムスタンプを利用するか、独自に時刻情報を取得する方法がある。前者のタイムスタンプを使用する場合、これに対応したホストならば、特別なプログラムを利用せずに測定ができるが、単位がミリ秒なので、遅延はミリ秒単位の測定になる。

## 3 PathTester の提案

本研究は、可用帯域幅の推定と伝送遅延の評価を目的としている。それらの動作を実装したプログラムを PathTester と名付ける。

### 3.1 パスの容量と可用帯域幅の推定方法

PathTester は、UDP パケットを用いる active 観測によってパス容量と可用帯域幅、そして伝送遅延を推定する。まず、ワンパケット推定法のアイデアを利用し、可用帯域幅を推定する。次に、パケットペアを用いて容量を推定したところで伝送遅延を推定する。可用帯域幅の推定には、ワンパケット推定法による active 観測を行う。パケットサイズによって、到着時間が異なることから転送速度を推定するこの手法は、パケットペアの弱点となるクロストラフィックの影響を受けずに推定が可能である。また、クロストラフィックによるパス通過時間の増加によって、転送速度が減少する。その減少した転送速度を可用帯域幅の推定値としている。パケットペアを用いる方法では、CapProbe[1] がある。これは、送信時に、間隔が最小のパケットペアを受信ホストに送信し、パスを通過後の到着時刻を受信ホストで測定する。いくつかの検査パケットで測定したものの最小、すなわち 1 番目と 2 番目の遅延が最小かつ、それらの和が最小のものから、パスの容量を推定する。

ワンパケットとパケットペア推定法を組み合わせる場合、送信回数を削減することで、パケットトレインを用いるものよりも少ないパケット量と時間での推定が期待できる。

### 3.2 伝送遅延の推定方法

伝送遅延は、伝搬遅延と中継ホストのキューによる遅延、そしてリンク速度による遅延等の合計である。パスの経路が往復で変化しないならば、伝搬遅延は変化しない。変化するとすれば、残りの遅延である。そこで、残りの遅延を計算し、往路復路の遅延を推定する。また、同時に offset の推定も行ふ。しかし、この手法でも伝搬遅延が大きく異なるパス（上りは地上回線而下りが衛星回線等）では正しい推定が難しい。リンク速度は、推定で得られた片道の容量で、キューの遅延は M/M/1 待ち行列の近似とリトルの公式より、平均待ち時間を計算して得る。offset を  $t$ 、 $D$  を観測値  $d_{fwd}$ 、 $d_{bwd}$  を真の伝送遅延、 $d$  を伝搬遅延とすると

$$\rho = \frac{(\text{Capacity} - \text{Bandwidth})}{\text{Capacity}} \quad (1)$$

$$D_{fwd} = d_{fwd} + t \quad (2)$$

$$D_{bwd} = d_{bwd} - t \quad (3)$$

となる。そして、 $t$  とキューの遅延は次の通りである。

$$t = \left( D_{fwd} - D_{bwd} + \frac{\rho_{bwd}}{1 - \rho_{bwd}} - \frac{\rho_{fwd}}{1 - \rho_{fwd}} \right) \cdot \frac{1}{2} \quad (4)$$

$$d_{fwd} = d + \frac{\text{Capacity}_{fwd} - \text{Bandwidth}_{fwd}}{\text{Bandwidth}_{fwd}^2} \quad (5)$$

$$d_{bwd} = d + \frac{\text{Capacity}_{back} - \text{Bandwidth}_{bwd}}{\text{Bandwidth}_{bwd}^2} \quad (6)$$

### 3.3 PathTester 実装上の問題

PathTester を実装するにあたり、複数の問題がある。それは、ホストに組み込まれている時計の問題と、時刻情報取得する際の問題、そしてパケット処理の問題である。また、どの時刻情報を使用するかの問題もある。PathTester は両ホストにおいて時刻情報を独自に取得する方法を取った。ICMP タイムスタンプ等の時刻情報では、ミリ秒単位の伝送遅延を測定するには利用可能であるが、可用帯域幅を推定するには単位が大き過ぎるからである。

ホストに組み込まれた時計は、温度補償されない水晶によって駆動されるものがほとんどである。そのため、時刻を修正しても時間が経つにつれて、温度や水晶振動子自体の精度等の要因によって時刻のずれ (offset) が生じる。図 5 は、同一スイッチに接続されたホスト A からホスト B へパケットを送信し、ホスト A の時刻を基準とした片道遅延を測定したものである。差が次第に少なくなっていくのが分かる。このグラフの傾き、すなわち offset の 1 階微分係数を skew という。

検査パケットが送信される直前の時刻を取得した際、OS のバッファリングや NIC のまとめ処理によってパケッ

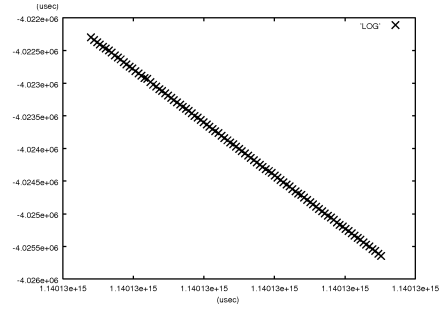


図 5 Time at host A vs. one way delay at host B

トが実際にネットワークへ送出される時刻と大きく異なる事態が起こる可能性がある。この問題を解決するには、使用するソケットのバッファサイズを小さくし、パケットの送信方法を工夫しなくてはならない。送信方法として、サイズの小さなパケットを小数で送らず、連続して送信させることで、まとめ処理やバッファリングの回避を図る。

一方、受信ホストにおいて、検査パケットの到着時刻を取得する際にも問題が生じる。これは、取得した時刻に受信関数が読み出すまでバッファに留まっている時間も含まれてしまうからである。そこで、パケットが OS からユーザに渡された時刻を `ioctl()` を用いて読み取る方法がある。この方法で、一般ユーザでも `libpcap` を用いたパケットキャプチャによるものと同様なパケット到着時刻に近い時刻を読み取ることができる。

推定の手順は以下の通りである。

- 測定する両ホストに事前に `ntp` 等で調べた初期 Skew をセットする
- Skew が正しいか確認し、必要に応じて補正する (短いパケットによる片道遅延測定)
- パケットペアを用いて容量を推定する
- ワンパケット推定法を用いて可用帯域を推定する
- 容量・可用帯域から、伝送遅延を推定する

## 4 実験結果

この節では、既存ツールと PathTester の動作実験について、結果と考察を示す。次の実験は、PathTester が他の推定ツールよりも短時間かつ低負荷で測定できるかを確認することを目標とする。実験では、小規模なネットワークを作り、クロストラフィックを生成する PC を設置した。

### 4.1 実験環境

実験で使用した環境は次の通り。使用 PC を表 1 に、ネットワーク構成を図 6 に示す。また、使用したツールは以下の通り。

- Pathload(可用帯域幅)
- PathChirp(可用帯域幅)
- Pathrate(容量)
- PathTester(可用帯域幅/容量)

表 1 実験環境 (使用 PC)

PC	CPU	RAM	NIC(module)
PC1	Celeron(400MHz)	192MB	eeepro100
PC2	Athlon XP-M(1393.2)	256MB	via-rhine
PC3	Celeron(500MHz)	128MB	e100
PC4	Pentium3(500MHz)	450MB	3c59x
Router1	Celeron(500MHz)	256MB	e100,3c59x
Router2	Pentium4(2.6GHz)	512MB	e100,3c59x

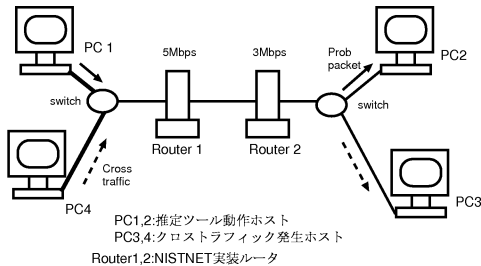


図 6 実験ネットワーク構成

クロストラフィックは、iperf[4] を用いて UDP パケットを流した。そのため、トラフィックモデルは、CBR となる。500Kbps から 2900Kbps まで、2500Kbps までは 500kb づつレートを増加させた。3000Kbps まで実験を行わなかった理由は、そこまでクロストラフィックを流してしまうと、ネットワークの輻輳はピークになり、既存ツールの動作が困難になってしまい、実験できないからである。ルータは、NISTNET による帯域制限を掛けた。それぞれ 5Mbps, 3Mbps の帯域をエミュレートさせている。

#### 4.2 実験結果

結果を表 2 と表 3 に示す。各ツールの推定値は、動作させた結果から得られた推定値 5 つの平均である。この実験の結果から得られたことは次の通り。

- Pathload の推定精度が他と比べて高い
- 既存ツール全般に推定所要時間が長い
- 実験環境であっても推定動作が一定でない
- PathTester はネットワーク閑散時の精度が悪い

同一環境での実験で、各ツールによって推定時間のばらつきがみられた。これは、推定値を収束させるために推定動作の回数が増減することが考えられる。また、Pathload は、回線が非常に混雑している場合において推定不能となった。その反面、PathTester はこのような場合でも推定可能であった。表 3 より、PathTester は他のツールより短時間で可用帯域幅や容量の推定ができていくことがわかる。これは、一定ペースによる検査パケット送受信と 1 回の試行につき 15 個という少量の検査パケットを用いた推定であることが要因であると考えられる。

## 5 まとめ

Pathtester によって、パスの可用帯域幅、容量、そして伝送遅延が短時間で推定できることが確認できた。また、

表 2 実験結果 (Mbps)

Traffic (kbps)	Pathload (帯域)	PathChirp (帯域)	Pathrate (容量)	PathTester (帯域:容量)
0	2.86	2.84	3-3	3.13:2.94
500	2.86	2.55	2.9-3.0	3.09:2.84
1000	1.72	2.47	2.9-3.1	1.68:2.93
1500	1.47	2.21	2.9-3.1	1.50:2.90
2000	0.98	1.37	2.9-3.0	1.23:2.91
2500	0.43	0.73	2.9-3.1	0.65:2.96
2900	-	1.03	1.8-1.9	0.66:2.92

表 3 実験の所要時間 (秒)

Traffic (kbps)	Pathload (帯域)	Pathrate (容量)	PathTester (帯域/容量)
0	91.2	68.6	8.2/52.0
500	111.4	1071	8.0/52.0
1000	164.8	1071	8.0/52.0
1500	149.2	1071	9.0/52.0
2000	260.0	1071	10.0/52.0
2500	146.9	1071	14.0/52.0
2900	218.3	222	106.6/168.6

少量検査パケットによる、低負荷な推定ができた。課題として、容量推定におけるクロストラフィックによる影響排除の対策、伝送遅延の評価方法の検討、ツールの精度向上、一層のネットワークへの負荷低減が挙げられる。

## 参考文献

- [1] Kapoor, R., Chen, L., Sun, T., Yang, G., Sanadidi, Y. M., and Gerla, M. : CapProbe: A Simple and Accurate Capacity Estimation Tool, ACM SIGCOMM Computer Communication Review, Vol. 34 , Issue 4 , pp.67-78, Oct. 2004.
- [2] Dovrolis, C., Jain, M. : Pathload : a measurement tool for end-to-end available bandwidth, <http://www.cc.gatech.edu/fac/Constantinos.Dovrolis/pathload.html> (accessed June 2005).
- [3] Downey, B. A. : Clink, <http://allendowney.com/research/clink/> (accessed January 2006).
- [4] Tirumala, A., Qin, F., Dugan, J., Ferguson, J., Gibbs, K. : Iperf 1.7.0 - The TCP/UDP Bandwidth Measurement Tool, <http://dast.nlanr.net/Projects/Iperf/> (accessed June 2005).
- [5] Ribeiro, J. V., Riedi, H. R., Baraniuk, G. R., Navratil, J., and Cottrell, L. : pathChirp: Efficient Available Bandwidth Estimation for Network Paths, Proc. of Passive and Active Measurements (PAM) workshop, Apr. 2003.