

インターネットにおける end-to-end パフォーマンス 測定解析システムのフレームワーク

串田高幸 (kushida@trl.ibm.co.jp)* 柴田義孝 (shibata@iwate-pu.ac.jp)**

* 日本アイ・ビー・エム株式会社東京基礎研究所

** 岩手県立大学ソフトウェア情報学部

概要

インターネットでは、途中経路の混雑や障害あるいは、その他の要因によって end-to-end でのバンド幅、遅延、パケット損失率の値が時間とともに常に変動している。本稿では、インターネットにおいて end-to-end のパフォーマンスを今までより正確に見積るための測定解析の方式とアプリケーションからの利用形態を提案する。この方式では、ネットワークのパフォーマンスを測定解析してパラメータ化する。このパラメータ化したデータと実時間で測定解析した結果のデータを組み合わせることによって、より正確にパフォーマンスの見積りを行なうことを目的としている。また、この方式をもとにしてアプリケーションから end-to-end パフォーマンスの見積りを行なうためのコンポーネントについても述べる。

Framework of end-to-end performance measurement and analysis system for Internet applications

Takayuki Kushida* and Yoshitaka Shibata**

*IBM Research, Tokyo Research Laboratory

**Faculty of Software and Information Science,
Iwate Prefectural University

Abstract

The Internet has no guaranteed performance service between end-to-end hosts. The Internet performance of the path on the way to the destination is always changing bandwidth, delay and packet loss by time. Therefore, it is very tough task to estimate the end-to-end performance. This study proposes the measurement and analysis framework to estimate the end-to-end performance on the Internet. The framework is to store the previous data which was estimated by the same Internet path. The estimation component uses both the previous data and the real time data from actual measurement to estimate the next performance on the path.

はじめに

インターネットは、社会の共通のインフラストラクチャーとして、現在、様々な日常のアクティビティーに利用されるだけでなく、企業間のビジネス取引にも利用されはじめてきている。しかし、インターネットは、もともとそのアーキテクチャがベストエフォートのネットワークサービスを提供しているだけであり end-

to-end のパフォーマンスを保証するための機構をもっていない。このようにインターネットにおいて、ネットワーク側で end-to-end パフォーマンスが保証されていないにも関わらず、インターネットは、ここ数年、急激に広まり多くの人が利用しはじめてきている。さらに社会の重要なトランザクションにおいて利用されてはじめてきている。

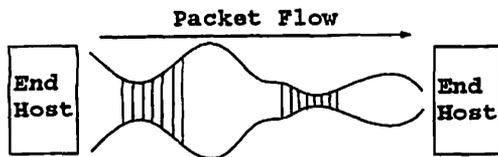


図 1: End-to-end performance

そのため、インターネットが利用される分野は、今まで以上に社会生活の中の重要な分野になっている。たとえば、インターネットが end-to-end のパフォーマンスが保証されないアーキテクチャであったとしても、アプリケーションにおいて、現在の時間での end-to-end パフォーマンスがどの程度になるのか、また、すぐ次の時間に、どの程度のパフォーマンスを得ることができるかについて見積ることが必要になっている。

図 1 は、インターネットにおいて End Host 間のパケットの流れを示している。この図において、Packet Flow は左から右に向かって流れている。これは、左側のホストから右側のホストにデータ転送されていることを示している。しかし、経路の途中においてバンド幅が異なるためにパスが狭くなっているところ（バンド幅が小さいところ）では、パケットの流れにボトルネックが生じている。そのため、end-to-end パフォーマンスの観点からみると、全体のパフォーマンスは、途中経路の最も狭いところによって、悪くなっている事がわかる。このように End-to-end のパフォーマンスにおいて経路の途中にパケット損失や混雑のためにボトルネックがあるとアプリケーションあるいはサービスからみて end-to-end のパフォーマンスが、常に変動しているように見える。

現在のインターネットのアーキテクチャでは、IP パケットを目的アドレスまで途中の経路情報の沿って単一の経路で配送しているが、将来は負荷分散やユーザーに提供されるサービスの質によって複数の経路を選択するサービスが提供される方式が確立されつつある [1]。このようなネットワーク環境が利用可能になると、ユーザーは、インターネットに対して end-to-end のパフォーマンスを見積ってから利用するパスを

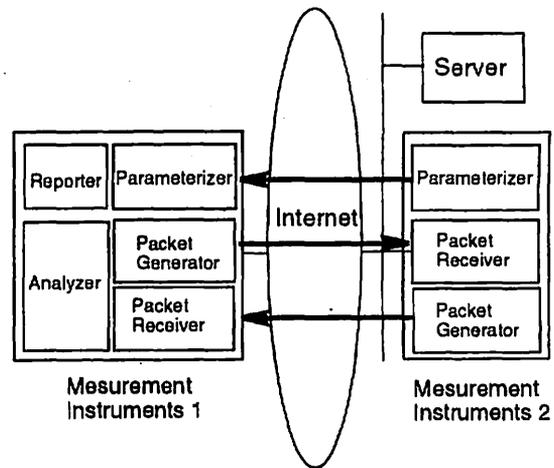


図 2: Measurement and analysis system

選択する。そのとき、end-to-end のパフォーマンスは、複数の経路の中から最もよい経路を選択する基準として用いられると考えられる。

本研究では、Quality of Service (QoS) が保証されていないインターネットにおいて、End Host からもう一方の End Host までのパフォーマンスを測定解析する方式について述べる。さらに、その測定結果を使ってアプリケーションからパラメータとして利用する方式について述べる。本稿は、以下の章から構成されている。まず、「バックグラウンド」では、インターネットパフォーマンスに関する研究について述べる。また、「フレームワーク」では、インターネットの測定及び解析方式について述べる。さらに、この方式をパラメータ化したデータと実データの両方を利用してパフォーマンスを測定するためのコンポーネントについて述べる。また、「議論」では、この方式を評価する方法及びこの方式を利用するアプリケーションについても述べる。「さいごに」では、結論と今後の方針について述べる。

バックグラウンド

インターネットにおいてパフォーマンスを測定して解析した結果は、インターネット上で、多くのアプリケーションから利用することができるようになるために、その影響力が大きい。

今までインターネットのパフォーマンス評価

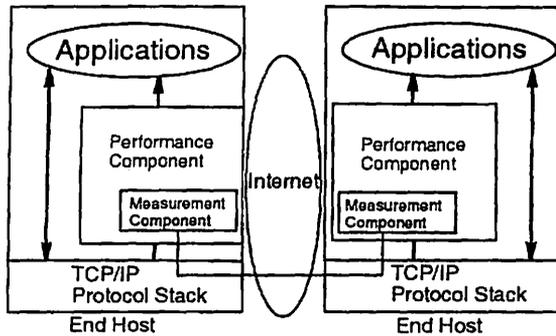


図 3: Entire system

に関連した多くの研究がなされている。例えば、End Host からネットワークのバンド幅を測定する方式には、pathchar を使う方法が一般に用いられているが、packet tailgating 方式によって、今までの方式よりも少ないパケットでバンド幅を測定できることが報告されている [8]。また関連した研究では、“pathchar” の実用性に関する研究 [2]、Default 経路以外のパスのパフォーマンスの研究 [3]、TCP パスの転送実験及び解析についての研究 [5]、ネットワーク測定及び利用アーキテクチャの提案 [7]、巨大ネットワークを管理するツールの研究 [4] がある。

このようにネットワークの end-to-end パフォーマンス及びネットワークツールに関連して、多くの関連した研究がなされているが、本研究では、これら以前の研究とは異なって end-to-end パフォーマンスを測定して、その測定解析結果をパラメータ化してアプリケーションとプロトコルスタックの間でコンポーネントとして利用することを目的としている。

フレームワーク

この研究の測定方式では、インターネットにおいて end-to-end のパフォーマンスを見積るために測定用パケットをネットワーク内部に入れるアクティブ方式を使用している。また、測定した結果を特徴としてパラメータ化して記録しておくためにリアルタイムに測定した結果と比べて end-to-end のパフォーマンスを見積る。

このフレームワークは、1. end-to-end パフ

ォーマンスを見積るためにデータの測定と、2. パラメータ化して結果を上位のアプリケーションから利用するためのコンポーネントと、大きく 2 つシステムから構成されている。

ネットワークのパフォーマンスから得られた情報で測定解析されたデータは、「1. 過去の測定解析データ」(一般的な特徴量も含む)と「2. 現在の測定解析データ」の 2 つに分かれる。最初に「1. 過去の測定解析データ」を見積ってパラメータ化して記録するため、「測定解析システム」を構成する。また、1. のパラメータ化した値と「2. 現在の測定解析データ」を使用してアプリケーションに対して結果を知らせるために「パフォーマンス・コンポーネント」を構成する。この「パフォーマンス・コンポーネント」は、アプリケーションとプロトコルスタックの間で動作して、インターネットの end-to-end のパフォーマンスをアプリケーションに通知する。

図 3 は、提案しているシステムの全体の構成図である。通常、インターネットの Application は、TCP/IP の Protocol Stack をソケット・インターフェイスを経由して利用し、もう一方の End Host と通信リンクを生成してデータの交換を行なっている。このとき、通常のサービスとして信頼性のある TCP を使用するか、あるいはパケット転送を行なう UDP を使うかのどちらかとなる。一方、図 3 の提案しているシステムは、End Host の中のプロトコルと Application の間に Performance Component を置いて、その Performance Component を経由してインターネットを利用する。この Performance Component は、Application に対して end-to-end のパフォーマンスを見積って、通知する機能を持っている。

基本の測定に関して

一般に end-to-end パフォーマンスの評価を行なう場合、必要な測定項目は、1. バンド幅、2. パケット損失率、3. パケットの遅延の 3 つの値である。これらの 3 つの測定値を測定システムによって取得して、そのデータを解析して

パラメータ化する。これらのパラメータから一般的な特徴量を取り出す。

ネットワークの両端に測定装置を置いた場合、パケットの測定方式は、図4で示される。図4 a. は、時間を横軸に表わして、 t_s が一方の End Host からのパケットの送信時間であり、また t_r が、もう一方の End Host でのパケットの受信時間である。この図で、一方向の遅延は、 $t_r - t_s$ で測定できる。また、ここで受信したパケットにジッターがある場合は、 t_r の周囲の t_{d1} から t_{d2} の間で遅延の変動が起こることになる。図4 b. は、バンド幅測定を表わしている。測定時間 t (sec) 当たり転送されたパケット B (bytes) を測定する事でバンド幅を B/t (bytes/s) で算出することができる。このバンド幅は、パケットの転送によって測定するバンド幅でありアプリケーションでのバンド幅は、TCP バンド幅となる。これは、一般に (TCP バンド幅) < (パケットバンド幅) とパケットのバンド幅の方が常に大きい。この理由は、TCP バンド幅では、図4 c. で示しているようにセッションの開始と終了の処理があり、パケットだけで測定したバンド幅に比べて大きな Overhead となっているからである。さらに TCP では、両端のホストでのバッファによる流量制御や混雑を軽減されるために、Window 制御を行ないネットワークが転送できる容量としてのバンド幅を表わしていないからである。つまり、アプリケーションから利用できるバンド幅 (TCP のバンド幅) とネットワークで転送できるデータ量のバンド幅は、値が一致していないが依存関係がある。また、パケット損失は、送信側から転送されたパケットのうち、受信側に届いたパケットと届かなかったパケットの比率になる。パケット損失は、時間当たりの損失率となるために測定した時間に依存していることとなっている。

測定解析システム

図2が、測定解析システムである。この測定解析システムの目的は、End-to-end のパフォーマンス測定を行なって、測定解析結果をパラメータ化することである。図2の Measure-

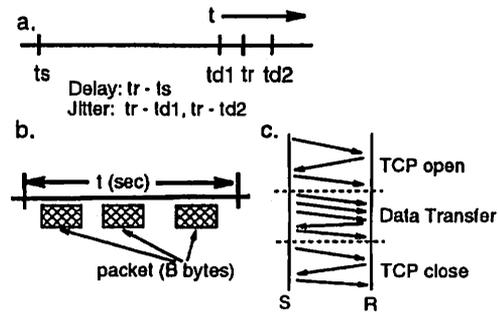


図4: Measurement Method

ment Instruments (1) は、“Packet Generator”, “Packet Receiver”, “Parameterizer”, “Analyzer”, “Reporter” があり、また、Measurement Instruments (2) は、“Packet Generator”, “Packet Receiver”, “Parameterizer” がある。Measurement Instruments (1) の “Packet Generator” は、測定用パケットを生成して、その測定パケットを Measurement Instruments (2) の “Packet Receiver” に送信する機能を持つ。また、Measurement Instruments (2) の “Packet Receiver” は、“Packet Generator” から送られてきた測定用パケットを受信して、測定に必要なデータを “Parameterizer” に送る。また、“Packet Receiver” は、“Packet Generator” に対して Measurement Instruments (1) に測定データを送るように指示を出す。Measurement Instruments (2) の “Parameterizer” では、“Packet Receiver” からデータを受け取ってパラメータ化して Measurement Instruments (1) に転送する。一方、Measurement Instruments (1) の “Parameterizer” は、Measurement Instrument (2) の “Parameterizer” から送られてきたパラメータを受信して、Measurement Instruments (1) のパラメータとともに “Analyzer” に送る。“Analyzer” では、それぞれの Parameterizer から来たパラメータを解析する。また、“Reporter” は、測定解析結果をユーザーに報告する機能及び解析結果を次のデータのために保管する機能を持つ。

測定用のパケットは、図5に示した形式を End Host 間で利用する。この測定パケットには、“Time Stamp” 及び “IP address for Time



図 5: Measurement packet

stamp”の情報が入る。この測定用パケットを使って複数の経路を測定する場合、通過した経路の Time Stamp 及び IP address をパケットのなかに書き込んでおいて、それぞれフィールドを作ってゆく。また、測定用パケットの固定長パケットの長さを一定にするため、測定のためにバイト数が足りない場合、最後尾に Data Padding を挿入して同じ長さのデータとして揃えることを行なう。

パフォーマンス・コンポーネント

アプリケーションプログラムからインターネットのパフォーマンスを利用する場合、測定解析結果をパラメータ化してライブラリとしてアプリケーションから利用できるように構成する。ここでは、過去の測定解析結果のパラメータと実時間で測定解析したパラメータを適切な形で組み合わせてパフォーマンスの見積りを行なう。図 6 が、見積りを行なうパフォーマンス・コンポーネントの構成図である。この図において Historical Parameters は、前節で説明した「測定解析システム」で各パラメータが記録されているので、そのパラメータを入力として使用する。また、Historical Parameters を Characteristics Understanding コンポーネントに入力して、その特徴量を Estimation/Prediction コンポーネントに出力する。一方、リアルタイムで End-to-end パフォーマンスが測定されたデータでは、Realtime Data Collection に集められて Realtime Analysis コンポーネントで解析される。この解析された結果は、Estimation/Prediction コンポーネントに出力される。2つの結果が入力された Estimation/Prediction コンポーネントでは、他のコンポーネントあるいはアプリケーションに適切なパラメータとして通知する。通知するパラメータは、3つの基本パラメータに加えて、そ

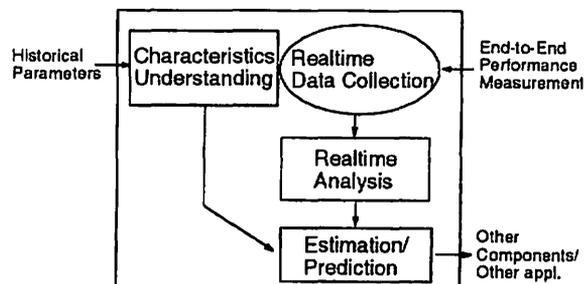


図 6: Performance Component

れぞれの分散を利用する。

- バンド幅とその分散 (Kbps)
- パケット損失とその分散 (%)
- パケットの遅延とその分散 (msec)

パフォーマンス・コンポーネントでは、ネットワークの過去の測定結果から得られた特徴量と現在の状況を比較することにより適切な値を他のコンポーネントあるいは、アプリケーションに対して通知する機能を持っている。

議論

本研究では、End-to-end パフォーマンスを測定するために両端に測定装置を設置しているが、pathchar や packet tailgating のような方式を組み合わせてもパフォーマンスの値を測定することが可能である [8]。

測定専用を使うパケットをネットワークに入れて、その結果によって end-to-end パフォーマンスを測定する場合、測定用のパケットがネットワーク資源を消費し、測定パケットによって、結果に影響が出てしまう。例えば、アナログモデムを利用している公衆回線のようにリンクのバンド幅が測定用パケットに比べて小さい場合には、測定用パケットの占める割合が相対的に大きくなってしまい、結果への影響の問題は大きい。しかし、End Host からアクセスのバンド幅が大きくなれば、測定用パケットの占める割合が相対的に小さくなる。そのため、End Host からのアクセス速度がより速くなれば、ネットワーク資源に対して測定用パケットの影響が小さくなってくる。

一般にネットワークの測定を行なう場合、測定したパケットの誤差について考慮する。ネットワーク測定装置の誤差に関しては、パケットカウンターとネットワークパケットを捕獲するとき CPU ロードが高くなっている場合、測定の誤差が出ることが報告されている [6]。この報告では、パケットヘッダーの中のパケットの長さのフィールドを使用してパケットを捕獲したときの誤差を除去している。また、測定の時間に関しては、ハードウェアクロックを使用して、さらに精度を高める必要がある。本研究では、CPU ロードによって測定の影響をなるべく受けないように、実際に利用するシステムとは別に測定解析システムを用意して、実際に利用するコンポーネントでは、実時間の参照のために測定解析だけを行なうようにシステムを構成している。

本来、アプリケーション側から end-to-end パフォーマンスを最初に要求して、その要求されたパフォーマンスをネットワーク側で保証する。しかし、このフレームワークでのアプリケーションからの利用は、パフォーマンスの保証ができないため、ネットワーク側からアプリケーションに対してパフォーマンスの情報を伝えるに留まっている。そのため、アプリケーションでは、ネットワークからのパフォーマンスの情報をもとにして、アプリケーションの転送方式を変更する。例えば、ビデオ転送のアプリケーションであれば、ネットワークに転送するビデオストリームのビットレートをネットワークのバンド幅にあわせて変更する。この変更では、パフォーマンスによって動的に行なわれる。

終わりに

本研究では、インターネットの end-to-end パフォーマンスの見積りをアプリケーションから利用できるフレームワークについて、その測定方式とアプリケーションからの利用方式について述べた。この方式では、end-to-end パフォーマンスの測定解析を行なうことによって、インターネット内部の特徴をパラメータとして保管しておく。この保管されたデータをもとにし

て実時間で測定解析した結果とをあわせてパフォーマンスを見積るコンポーネントについて述べた。また、このコンポーネントは、ネットワークを利用するアプリケーションが、パフォーマンスの見積り結果を過去の特徴データとしてコンポーネントの入力として使うことによって、今までの実時間での測定解析だけよりも、さらに正確な見積りが可能となる。

参考文献

- [1] T. Kushida, "A framework to support alternate paths to provide the better end-to-end performance on the Internet", In Proc. of NGITA00 workshop (ICPADS2000 workshops), Iwate, JAPAN, July 2000.
- [2] A. Downey, "Using pathchar to estimate Internet link characteristics", In Proc. of ACM SIGCOMM99, Cambridge MA, USA, September 1999.
- [3] S. Savage, A Collins, E. Hoffman, "The End-to-end Effects of Internet Path Selection", In Proc. of ACM SIGCOMM99, Cambridge MA, USA, September 1999.
- [4] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, and J. Rexford, "NetScope: Traffic Engineering for IP Networks", IEEE Network, pp.11-19, March/April 2000.
- [5] V. Paxson, "End-to-end Internet Packet Dynamics", IEEE/ACM Transactions on Networking 7(3), pp 277-292, June, 1999
- [6] S. Belenki, S. Tafvelin, "Analysis of Errors in Network Load Measurements", ACM SIGCOMM CCR, (30) 1, January 2000.
- [7] M. Stemm, R. Katz and S. Seshan, "A Network Measurement Architecture for Adaptive Applications", In Proc. of IEEE INFOCOM 2000, Tel-Aviv, Israel, March 2000.
- [8] K. Lai and M. Baker, "Measuring Link Bandwidths Using a Deterministic Model of Packet Delay", In Proc. of ACM SIGCOMM 2000, Stockholm, Sweden, August 2000.