2005－DSM－36（6）
2005／3／18

トランスポートプロトコルの実機による性能検証における遅延・帯域とトポロジの切替時間短縮のためのシミュレーション技法の検証

小西 一暢　　村本 衛一　　米田 孝弘　　許 志彰　　川原 豊樹

松下電器産業株式会社

概　要

トランスポートプロトコルはインターネットでの運用を鑑み、様々なトポロジとリンクの遅延・帯域を想定した性能検証を行う必要がある。しかし、実機を用いて行う場合、トポロジの切替えに多大な時間を要する。その問題の解決のため、自由にトポロジを構成できる大規模実験環境 StarBED を利用する手法があるが、リンクの遅延・帯域のエミュレーションとトポロジの切替え時間の短縮に関する有効性の検証は行われていない。

我々は、Dummynet を用いた遅延・帯域のエミュレーション技法を StarBED に導入した。その上で、トランスポートプロトコルの性能検証を行い、リンクの遅延・帯域の実現とトポロジの切替え時間の短縮における有効性を検証した。性能検証においては、NS-2 と StarBED でのシミュレーション結果を比較し、同等の結果が得られることを確認した。本技法により、トランスポートプロトコルの性能検証を短時間で実現できる。

1. はじめに

インターネットの拡大に伴って、インターネットプロトコルを利用した様々なトランスポートプロトコルの研究・開発が行われており、その特性評価を行う必要がある。

インターネットは、不均一なネットワークで構成されているため、インターネットでの運用を目的としたトランスポートプロトコルの特性評価では、様々なネットワークの遅延・帯域、トポロジを想定したシミュレーションを行わなければならない。

シミュレーション手法として、論理的なネットワークシミュレータである NS-2（Network Simulator 2）[1]を用いることで、手軽に広範囲なシミュレーションを行うことができる。しかし、このような論理的な手法では実装スタックの動作検証を行うことはできない。実装スタックの動作検証を行うには、実際のマシン（以下、実機とする）を用いて実験用ネットワークトポロジを作成し、そのトポロジ上でシミュレーションを行う手法がある。しかし、実機を用いたシミュレーションの場合、これまで遅延・帯域のエミュレーション、トポロジの切替えに多大な時間を要するという課題があった。

上記課題を解決するために、様々な研究が行われている。

2. 関連研究との比較

本章では、遅延・帯域のエミュレーション、トポロジ切替え時間短縮に関するシミュレーション技法と我々が採用したシミュレーション技法について述べる。

2-1. 関連研究

実機を用いた、トランスポートプロトコルの特性評価を行う環境として、ネットワークスイッチの VLAN を設定することで、トポロジを自由に構成することができる NetBED[2]がある。NetBED では、ネットワークの遅延・帯域をソフトウェア的にエミュレートする手段として、Dummynet[3]を用いる。NetBED は NS-2 の設定ファイルを読み込み、自動的に Dummynet が組み込まれたトポロジを作成する。しかし、NetBED では実験に用いるノードの OS を入れ替えることができないという課題がある。

NetBED と同様に、ネットワークスイッチの VLAN を設定することで、トポロジを自由に構成することができる大規模インターネット実験環境 StarBED[4]が知られている。

StarBEDはNetBEDと異なり、ノードのOSを入れ替えることができるという特徴を持つ。また、StarBEDには、トポロジの作成、ノード制御、実験シミュレーションシナリオ（以降、シナリオとする）実行などシミュレーション実施のための支援プログラム群：kuroyuri[7][1]がある。これにより、トポロジの作成、シナリオ

[1] 文献[7]で、ENCD,Scenario Masterとして述べられているソフトウエアがkuroyuriである。

の実行を行うことができる。

しかし、StarBED ではネットワークの帯域・遅延を自動設定できないという課題があった。そのため、遅延・帯域のエミュレートをするためには、StarBED に Dummynet を配置したトポロジを実験者が手動で作成しなければならなかった。

我々は、実験環境として StarBED、遅延・帯域のエミュレートに Dummynet、トポロジ作成、シナリオ実行に kuroyuri といった既存手法を組合せたシミュレーション技法を用いることで、トポロジ作成、Dummynet の遅延・帯域設定、シナリオ実行といったシミュレーション作業を自動化し、大幅な時間短縮ができると考える。そのため、我々は上記したシミュレーション技法を採用した。

2-2. StarBED を用いたシミュレーション技法

本シミュレーション技法でトポロジの切替え、シナリオ実行に用いる kuroyuri は sbrm（ERM）、pickup、wipeout、master、slave といったプログラム群で構成される。表１は各プログラムの機能である。kuroyuri はトポロジの作成とシナリオの実行という 2 つの機能によりシミュレーションを行う。

具体的に、本シミュレーション技法の手順を以下に述べる。

（1）実験者は、トポロジとシナリオを記述した kuroyuri 用設定ファイルを作成する。

（2）実験ノードに配布する OS を 1 台のノード上に作成する。

（3）(2) のノードに作成した OS から、pickup により OS イメージファイルを作成する。その OS イメージファイルを用いて、wipeout により OS を実験ノードに配布する。

（4）各実験ノードにおいて、slave を起動する。

（5）マスタは実験ノードとして用いるノードを、sbrm により適切に決定する。

（6）マスタは VLAN スイッチの設定を行い、実験ノード間のトポロジを構成する。

（7）マスタで、master プログラムによりシナリオを実行する。

以上の手順により、トポロジを作成し、シナリオに従ってシミュレーションを実行する。

表１　kuroyuri を構成するプログラム群

プログラム名	機能
master	シミュレーションの自動実行を行う
slave	master と通信し、実験ノードを制御する
sbrm（ERM）	実験ノードの利用状態を管理する
pickup	OS をイメージファイル化する
wipeout	OS イメージファイルを HDD に書込む

次に、本シミュレーション技法の遅延・帯域、トポロジ切替え時間短縮に関する有効性の検証を行う。我々は、SICC（Sender Initiated Congestion Control）[5]を例に取り、本シミュレーション技法を用いて特性評価を行った。これにより、StarBED において、Dummynet、kuroyuri を用いた本シミュレーション技法の、遅延・帯域、トポロジ切替え時間の短縮に関する有効性を示す。

3. SICC

本章では、SICC の概要、評価指標、評価方法を示す。

3-1. SICC の概要

SICC とは、XCAST6[6]によるマルチキャスト通信を用いて、送信者起動の輻輳制御を行うトランスポートプロトコルである。具体的には、受信者からのフィードバック情報に基づき、送信者が各受信者の利用可能帯域を推定する。推定した帯域に適した伝送レートで通信を行うことで輻輳制御を実現する。

SICC は次の 2 つの特性を有する。

（1）TCP との公平な帯域の共有

ネットワーク上に TCP と SICC の 2 種類のプロトコルによるフローが共存していた場合、両フローが帯域を同程度（利用帯域が一方の 2 倍以内)利用した状態で安定した通信ができる。

（2）受信者間の公平性

ネットワークの帯域が各受信者で異なる場合、伝送レートが最も狭い帯域の受信者に律速されることなく、各受信者の帯域に適応した伝送レートで通信することができる。

上記 2 つの特性を評価するために、次の 2 つ

の論理トポロジを切替えてシミュレーションを行う必要がある。

（1）TCP との公平な帯域共有の評価用論理トポロジ

すべての受信者が共有するボトルネックリンクを設定し、TCP と SICC の 2 種類のフローを流す。そして、両フローがボトルネックリンクを同程度利用できるかを評価する。図１にトポロジ図を示す。

（2）受信者間の公平性の評価用論理トポロジ

各受信者の上流リンクで異なる帯域・遅延を設定し、SICC フローを流す。そして、各受信者の帯域に応じた伝送レートで通信が行われているかを測定する。図 2 にトポロジ図を示す。

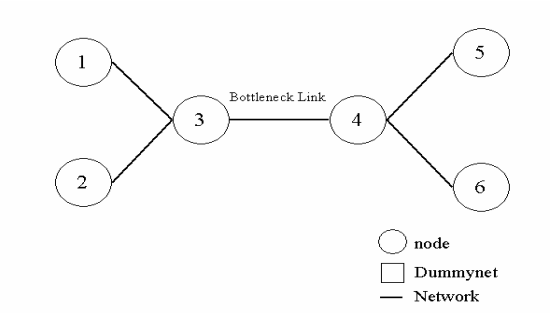

図1　TCP との公平な帯域共有の評価用論理トポロジ

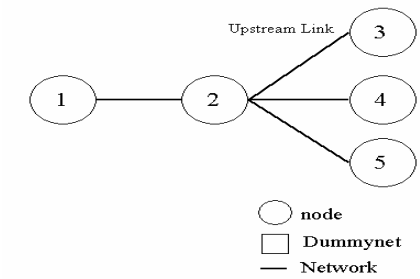

図 2　受信者間の公平性の評価用論理トポロジ

4. StarBED におけるトランスポートプロトコルの評価方法

本章では、StarBED 上での SICC の評価方法の詳細を述べる。

4-1. StarBED におけるリンクのエミュレーション技法

前章で述べた論理トポロジをエミュレートした、StarBED 上での物理トポロジを示す。StarBED 上では、図 1、図 2 のトポロジはそれぞれ、図 3、図 4 のように構成した。

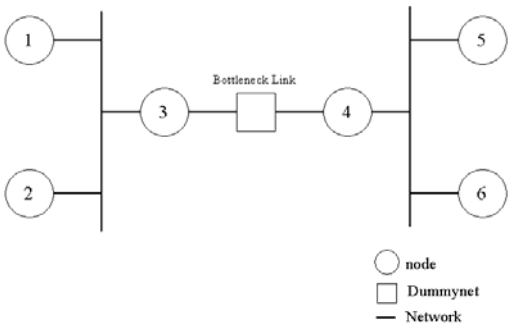

図 3　TCP との公平な帯域共有の評価用物理トポロジ

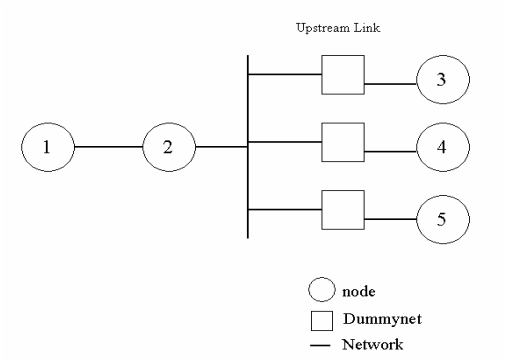

図 4　受信者間の公平性の評価用物理トポロジ

本シミュレーションでは、図 3 のノード 1、2 は送信者、ノード 5、6 は受信者、ノード 3、4 はルータを表す。それらのノードには OS として、XCAST6 を導入した NetBSD1.6.2 を用いた。また、ノード 7 の遅延・帯域をエミュレートする Dummynet ノードには OS として、FreeBSD4.9-R を用いた。図４では、１が送信者、２がルータ、３、４、５が受信者である。

本シミュレーションはDummynetを用いて、1ms単位の精度で遅延のエミュレートを行う必要がある。そのため、カーネルのスケジューリングクロックを高速化し、Dummynetの精度を上げるため、カーネルオプションとしてHZ=1000 を指定した。また、DummynetはIPv6対応ではないため、Dummynetを挟んだノード（図 3 のトポロジでは、ノード 3、4）において、IPv6-over-IPv4 トンネルを用いて、MTUが1280 バイトのIPv6 リンクをエミュレートした

[2]。

Dummynet によって帯域を設定する場合、Dummynet を通過するパケットサイズは IPv6-over-Ipv4 トンネルにより IPv4 ヘッダ 20 バイト分大きくなる。そのため、IPv4 ヘッダの増加分を考慮して、帯域を設定しなくてはならない。設定帯域を BW[kbps]とすると、MTU が 1280[Byte]の場合、Dummynet により実際に設定する帯域 BW_{dummy}[kbps]は次式で求められる。

$$BW_{dummy} = BW \times \left((1280+20)/1280\right) \cdots (1)$$

kuroyuri のシナリオに上記変換式を導入し、シナリオ中に遅延・帯域を変更しながら繰返し実験を行うことを可能にした。

4-2. 本シミュレーション技法の計測限界

本シミュレーション技法の計測限界について述べる。図 3 のトポロジにおいて、ノード 3、4 では、1 つのノードに対して多数の受信者を接続しなくてはならない。そのため、ノード 3 とノード 4 に接続するすべてのノード数分 VLAN を作成し、仮想的に複数のネットワークインタフェースを持つルータのエミュレートを行った。StarBED のノードは 100Mbps のネットワークインタフェースを持つ。1 つのネットワークインタフェースで複数の受信者と接続する場合、受信者数 n とすると、各受信者が利用できる最大帯域 Max_bw[Mbps]は、次式で表される。

$$Max\_bw = 100/n \quad \cdots (2)$$

つまり、受信者の帯域の上限は 1 台のノードが持つネットワークインタフェースの帯域によって計測限界が決定される。

また、遅延については、先に述べたように Dummynet の遅延設定限界の 1ms の粒度に制限される。

[2] DummynetをIPv6 対応させるパッチは存在する。しかし、SICCのような新しく開発したプロトコルが独自のオプションヘッダを持つ場合、IPv6 対応の Dummynetが正常に動作しないという問題があった。

[3]従来手法による実験では、スイッチの設定に慣れていない実験者では、スイッチの設定ミスに伴うパケットのループや、スパニングツリーが原因で通信ができないといったトラブルが発生した。

4-3.トランスポートプロトコルの評価手法

本シミュレーション技法では、安定して計測するため、ひとつのトポロジ上での実験に、次の 3 つの手順をとる。

（1）トポロジの構築

VLAN の設定を行うことで、実験用トポロジを構築する。master は、VLAN の設定を行い、kuroyuri の設定ファイルに記述されたトポロジを自動で構築する。

（2）ヘルスチェックシナリオ

（1）で述べた系構築用シナリオで正しくトポロジが構築されているかを検査する。ネットワークの接続、Dummynet の遅延・帯域設定が正しく行われているかを調べるために、ping、netperf の 2 つのプログラムを実行し、そのログを調べる。最初に、Dummynet により遅延・帯域を設定し、図 3 では、ノード 1、2 からノード 5、6 に対して、図 4 では、ノード 1 からノード 5、6、7 に対して ping と netperf を実行する。

（3）実験用シナリオ

実験シミュレーションを実施する。実験用シナリオは 1 つのトポロジにおいて、遅延・帯域を切替えて複数回シミュレーションを行う。そのため、シミュレーションの 1 試行ごとに Dummynet の遅延・帯域設定を変更する。この設定変更は、マスタから実験ノードに対して送信されるメッセージと、実験ノードに配布したパラメータファイルから実験ノードが 1 試行ごとに Dummynet の遅延・帯域設定を行うことで実験をする。すべての試行が終了すれば、スレーブはマスタに対して FTP により、シミュレーションログを送信する。

5. シミュレーション時間の短縮の評価

本章では、シミュレーション時間の短縮について、従来方式と比較し考察を行う。

5-1. 従来手法

実機を用いたシミュレーションにおいて、従来方式として、トポロジの作成、遅延・帯域の設定、シミュレーションの実行などはすべて手作業で行っていた。本章では、従来手法である手作業によるトポロジの作成、ヘルスチェックに用いた時間を調べ、本シミュレーション技法と比較する。

3 人の被験者に手作業によるトポロジの作成、

ヘルスチェックを行い、その作業時間を調べた。被験者は、UNIXの知識を持ち、スイッチの設定などネットワーク構築を行った経験を持つ。被験者には、作成するトポロジ図、vlan番号とポートの対応表、スイッチマニュアル、作業手順書を配布した。

トポロジの作成は StarBED と同様、次の 2 つの手順で行う。トポロジを構成する各ノードの IP アドレスや経路表などの設定は、事前にシェルスクリプトを作成し、それを実行することで行う。

（1）トポロジ構築

スイッチの VLAN 設定をコマンドラインにより行う。

（2）ヘルスチェック

Dummynet の遅延・帯域設定を行い、その後、ping と netperf により、トポロジのヘルスチェックを行う。

5-2. 従来方式との比較

手作業による従来方式と、StarBEDにおけるトポロジ作成、ヘルスチェックに要した時間を表 2 に示す[3]。

表 2　トポロジ作成時間

	被験者 1	被験者 2	被験者 3	StarBED
時間[分]	23	40	16	7

本シミュレーション技法を用いた場合、従来方式に比べて、短時間でトポロジの作成からシミュレーション実施まで行えることが確認できた。また、このトポロジ作成時間はトポロジ上のノード数が増加すればするほど本シミュレーション技法による時間短縮の効果が得られる。

5-3. シミュレーション結果の比較

本シミュレーション手法の正当性を示すため、SICC の持つ特性の１つである TCP との公平な帯域共有について、StarBED と NS-2 で行った結果を図 5、図 6 に示す。図 5、図 6 において、縦軸 F は、TCP、SICC 両フローがボトルネックリンクを公平に帯域共有しているかを示す指標である。0.5<F<2 内に両プロトコルが収まっていれば、公平に帯域を共有している。

シミュレーションシナリオとして、TCP フローを流し始めてから、50 秒後 SICC フローを流しはじめ、550 秒間実行する。シミュレーションパラメータとして、TCP フロー数を {1,2,4,8,16,32,64,128}とし、ボトルネックリンクの遅延を 64ms、キューを RED、帯域を TCP フロー数に応じて設定した。

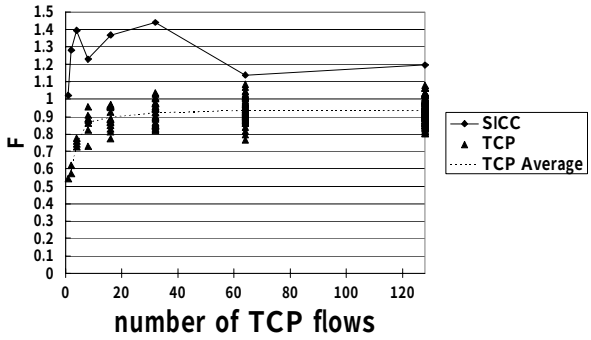

図 5　実機におけるシミュレーション結果

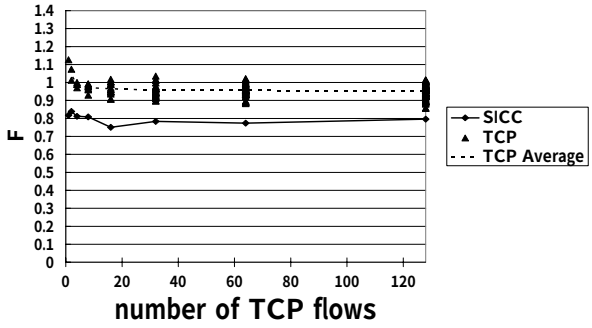

図 6　NS-2 におけるシミュレーション結果

図 5、図 6 からわかるように、StarBED、NS-2 両者とも 0.5<F<2 内に収まっており、SICC の特性である TCP との公平な帯域の共有が確認でき、SICC の特性評価において、StarBED を用いたシミュレーションと NS-2 を用いたシミュレーションで同等の結果を得た。

また、NS-2 では、シミュレーション上で生成されるパケット数が増加すると、それに比例してシミュレーション時間が増加する。例えば、最大送信レート 1Mbps、TCP フローを 128 本流す１試行で、NS-2 では約 30 分、最大送信レートが 10Mbps の場合、シミュレートするパケット数が増加するため、約 2 時間を要した。しかし、本シミュレーション技法では、両者とも約 660 秒（実時間）で実験を行えた。本シミュレーション技法を用いることで、短時間で NS-2 と同程度の結果が得られることが検証できた。

また、遅延・帯域をシミュレーションパラメータとして、様々な設定で繰返しシミュレーシ

ョンを行う場合、従来方式では、すべての Dummynet の設定を手動で行い、実験プログラムを実行しなければならない。本シミュレーション技法を用いることで、遅延・帯域を切替えながら繰返しシミュレーションを行うことができ、NS-2 等の論理的な手法と同等の幅の広いシミュレーションを短時間で実行できる。

つまり、トランスポートプロトコルの性能検証に、本シミュレーション技法を用いることで、従来手法に比べて大幅にシミュレーション時間を短縮することが可能となり、実験者にかかる負荷を軽減できる。

6. まとめ

遅延・帯域、トポロジの切替え時間について、StarBED において、遅延・帯域のエミュレートに Dummynet、トポロジ切替えに kuroyuri を用いたシミュレーション技法と従来方式の比較を行った。その結果、従来方式では、約 35 分、本シミュレーション技法では、約 7 分で遅延・帯域、トポロジを切替えられることがわかり、本シミュレーション技法の時間短縮に関する有効性が確認できた。また、シミュレーション結果において、NS-2 と比較し、同等の結果を得た。

また、シミュレーションの時間を NS-2 と StarBED で比較した結果、NS-2 では、約 2 時間、StarBED においては、約 660 秒要することを確認した。

本シミュレーション技法を用いることで、NS-2 などの論理的な手法と同等の様々なシミュレーションパラメータによる、広範囲なトランスポートプロトコルの特性評価を短時間で実施することが可能となる。

7. 今後の課題

今後の課題として、次の 2 つを挙げる。

（1）ネットワークインタフェースの性能によって実験可能な最大帯域が決定される問題

1 つの 100Mbps のネットワークインタフェースを複数のノードで共有するため、本シミュレーションで用いたトポロジでは、広帯域での実験が行えない。ギガビットイーサネットのような高速なネットワークインタフェースの導入が求められる。

（2）リンクの遅延・帯域の自動設定

StarBED では、Dummynet による遅延・帯域を実験者が設定しなければならないが、様々なシミュレーションにおいて、遅延・帯域のエミュレートが必要になる。そのため、より実験者が低負荷で実験するために kuroyuri の基本機能として自動で Dummynet をトポロジに組込む仕組みが求められる。

謝辞

実験を行うにあたり、北陸先端科学技術大学院大学の知念賢一助手、博士後期課程　宮知利幸氏ほか JAIST メンバー、北陸 IT 研究開発センターの StarBED スタッフには数多くのご助言、ご助力いただきました。

参考文献

[1] The Network Simulator http://www.isi.edu/nsnam/ns/

[2] B.White, J.Lepreau, L.Stoller, R.Ricci, S.Guruprasad, M.Newbold, C.bard, and A.Joglekar, “An Integrated Experimental Environment for Distributed Systems and Networks.” In Proceedings of PADS ’98, May 1998.

[3] Luigi Rizzo, “Dummynet: A Simple Approach to the Evaluation of Network Protocols.”, Computer Communications Review, 27(1):31-41, January 1997.

[4] StarBED Project http://www.starbed.org/

[5] 村本 衛一, 米田 孝弘, 鈴木 史章, 鈴木 良宏, 中村 敦司, "送信者起動マルチキャストにおける輻輳制御方法の提案", インターネットコンファレンス 2003 論文集 pp5-10, 2003/10.

[6] Y. Imai, M. Shin and Y. Kim, "XCAST6: eXplicit Multicast on IPv6", IEEE/IPSJ SAINT2003 Workshop 4, IPv6 and Applications, Orland, Jan.2003.

[7] Toshiyuki Miyachi, Ken-ichi Chinen, Yoichi Shinoda: Automatic Configuration and Execution of Internet Experiments on an Actual Node-based Testbed, Proceeding of TridentCOM2005, Trento, Feb, 2005.