リンクエミュレータの多重実行の限界値測定

明石邦夫[†1] 井上朋哉[†1] 安田真悟[†1]
村本衛一[†3] 知念賢一[†1]
宇多仁[†2] 篠田陽一[†2]

ネットワーク技術の検証を行う際に，実証環境が利用されている．実証環境では容易に検証を行うことが可能であるが，リアルタイム通信や多地点通信システムのようなネットワーク特性を考慮する必要がある技術の検証を行うことは困難である．これは実証環境におけるネットワーク特性は物理特性に強く依存するためである．実証環境上に広域ネットワークを模倣した環境を構築するため，ソフトウェアによるリンクエミュレータを利用する方法が提案されている．しかし，ネットワーク環境を模倣するにも，実証環境上のリソースを必要とするため，可能な限り大きなネットワーク環境を構築するためには，効率的にネットワークの模倣を行う必要がある．本稿では，1 台のノード上で複数のリンクを多重化し，それぞれのリンク特性を模倣することにより，効率よくリソースを使用する手法を提案する．また，ソフトウェアによる模倣は，正確に模倣できているかを保証しないため，単一ノードにおけるリンクエミュレータの多重度の限界を調査する必要がある．我々は，netem を用いて複数のリンクを模倣した環境を構築し，netem が模倣可能な限界の調査を行った．そして，その結果をまとめた．

Limitation Measurement of Multiple Link Emulator

KUNIO AKASHI,[†1] TOMOYA INOUE,[†1] SHINGO YASUDA,[†1]
EIICHI MURAMOTO,[†3] KEN-ICHI CHINEN,[†1] SATOSHI UDA[†2]
and YOICHI SHINODA [†2]

The testbed is used in order to verify the network technology. Testbed is the environment on which the network applications can be easily verified. However, it is difficult to verify the network technology in which network characteristics such as real-time and multi-point communication system are necessary to be considered. Because link characteristics in testbed strongly depends on the physical properties of network entities. Network-emulator software has been proposed to emulate wide area network environment on the testnetwork in which there are few network entities. But several network entities are needed to emulate large-scale network if such software is used. It is necessary to efficiently utilize the resources when large-scale network is emulated. To emulate link characteristics by efficiently utilizing the resources on the testbed, it is necessary to realize how many link characteristics can efficiently multiplex on a single node. In this paper, we make the experiment to observe how efficient and how many link characteristics can multiplex on a single node.

1. はじめに

インターネットは重要な社会基盤として利用され，その上では様々なサービスが動作している．これに伴い，インターネットを利用することを前提とした技術が数多く開発・利用されている．通常，新しい技術を開発し実際に導入する場合，事前に十分な検証を行い製品の質を向上させる必要がある．

ネットワーク技術の検証は，インターネットの利用が理想的である．しかし，インターネットは重要な社会基盤として利用されているため，十分な検証が行われていない技術をインターネットへ導入することは好ましくない．したがって，インターネットを模倣した環境を構築し，インターネットから隔離された環境で検証を行う必要がある．このような環境を構築するための手法はいくつか提案されている．その中の一つとして，ネットワーク技術の検証を行うために用意され

†1 北陸先端科学技術大学院大学 情報科学研究科
School of Information Science Japan Advanced Institute of Science and Technology
†2 北陸先端科学技術大学院大学 情報科学センター
Center of Information Science Japan Advanced Institute of Science and Technology
†3 パナソニック株式会社 臨場感コムタスクフォース
Immersive Communication Task Force, Panasonic Corporation

た実証環境がある．

実証環境とは，実験を行うための多数のノードやネットワーク機器からなる実験設備と，実験を行うための支援ソフトウェアから構成される．このような実証環境を利用することによって，実験者は容易にネットワーク技術の検証を行うことが可能である．しかし，検証を行う対象によっては十分な検証を行うことができない場合がある．例えば，IP 電話やビデオ会議システムのようなリアルタイム通信を行うものは，実際に利用される環境は実証環境と比較し，高遅延な環境となることが予想される．また，多地点ビデオ会議システムといった各地のノードと通信を行うようなシステムの場合，多地点相互間の遅延・帯域を考慮して検証を行う必要がある．したがって，十分な検証を行う際にはネットワークの特性を模倣する必要がある．ネットワークの特性の模倣には，リンクエミュレータと呼ばれるソフトウェアを用いることで，実現可能である．本稿では，リンクエミュレータである netem[1] を用いて多数のリンクを多重化することにより，少ないリソースで仮想的な広域ネットワークを構築するための手法を提案する．そして，複数のリンクを多重化させる際に 1 台のノードで多重化可能な限界を調査した．

2. 研究の背景

インターネットが誕生した当初，インターネットは実験的なネットワークとして利用されていた．そのため，新しい技術の検証はインターネットそのもので行われており，検証を行うための実験環境は必要なかった．しかし，インターネットが普及した現在では，重要な社会基盤として，さまざまなユーザやサービスがインターネットを利用している．インターネットの急速な社会への浸透に伴い，インターネットを利用した数多くのソフトウェア・ハードウェアが開発されている．しかし，社会基盤として利用されている環境で何らかの実験を行えば，他のユーザやサービスへ実験者が想定しない影響が出る可能性がある．したがって，インターネットを検証の場として利用するのは好ましくなく，インターネットから隔離された環境で検証用のネットワークを構築し，その上で検証を行う必要がある．インターネットを模倣した環境にて実験を行う手法は 2 つ存在する．

1 つ目は，検証を行う実験者がネットワーク環境を構築し検証を行う方法である．この場合，実験者がネットワーク環境を構築するためには，実験トポロジを構築するために必要なノードを物理的に配置し，それぞれを接続し，ネットワーク機器の設定を行う必要がある．そのため，ノードを用意することによる経済的・空間的コストおよび，ノードの接続，ネットワーク設定のための人的・時間的コストが大きい．また，実験者がさまざまな機器を集める必要があるため，経済的・時間的コストが大きい．

2 つ目は，ネットワーク技術の検証を行うために用意された，実証環境を利用することである．実証環境は，実験設備と支援ソフトウェアから構成される．実験設備とは，多数のノードとネットワーク機器から構成される．支援ソフトウェアは，実験を行う際のノードの制御，実験トポロジの構築など，実験を支援するためのソフトウェアのことを指す．これによって，大規模な実験を行うことが可能となる．また，各ノードの制御に支援ソフトウェアを利用することにより，実験時の人的ミスを防ぐことが可能となる．実証環境を利用することにより，実験者は実験を行うためのノードを用意する必要がなく，実験を容易に行うことが可能となる．

実証環境での実験は，実験者がネットワーク環境を構築しての実験と比較し，ノード，ネットワーク機器の用意や配置が必要なく，実験ネットワークは論理的に構築するため，人的・空間的・経済的コストを押さえることが可能となる．また，支援ソフトウェアを用いることにより，実験ネットワークを容易に構築したり，実験時の人的ミスを防止することが可能であるため，大規模なネットワーク環境での実験が必要な場合には，実証環境を用いて実験を行うべきである．

しかし，実証環境で必ずしも十分な検証が行えるとは言えない．例えば，音声通信や動画配信といったリアルタイム通信を行うような技術では，高遅延な通信環境での利用も想定しなければならない．しかし，実証環境で提供されるネットワークの遅延特性は，実証環境のネットワークの物理的な特性に強く依存してしまう問題点がある．そのため，遅延や帯域といったネットワーク特性を自由に変更することが出来ない．さらに，多地点間の通信を行うようなシステムの検証を行う場合，各ノード間にそれぞれ違うネットワーク特性を持たせた状態で実験を行わなければ，十分な検証を行ったとは言えない．十分な検証を行うためには，インターネットが持つネットワーク特性を模倣した環境で，実験を行う必要がある．

3. ネットワークの模倣

ネットワークの特性を分類すると，L2 におけるリンクの特性の集合体と L3 特有の特性から構成される．L3 特有の特性とは，突発的なパスの変更や，切断な

どが挙げられる．L2 におけるリンク特性とは，ノードやネットワーク機器間の特性のことを指し，機器間の利用可能な帯域，通信における遅延やパケット損失率などが要素として挙げられる．複数のリンク特性を模倣し，1 本のパスを生成することを繰り返し，個々のパスを生成することによって，ネットワークの特性を模倣することが可能となる．本研究ではネットワークの特性の模倣を行うに当たり，L2 におけるリンク特性の模倣を行うことにより，ネットワークの特性の大部分を模倣できることから，リンク特性の模倣を最初に行った．

3.1 リンク特性の模倣

リンク特性を模倣する方法として，ハードウェアによる模倣とソフトウェアによる模倣が考えられる．

ハードウェアによる模倣は，遅延網と呼ばれる実際に模倣したい遅延が発生する長さのネットワークを構築するものと，遅延や帯域の模倣を行うために開発された専用機器を用いるもの等がある．ハードウェアを用いて模倣を行う際のメリットは，パケットの送信タイミングや，設定可能な遅延時間が細かいため精度の高い模倣が可能である事が挙げられる．デメリットとして，新たに専用ハードウェアが必要となるため，経済的コストが大きいことが挙げられる．また，物理的なノードの配置に影響を及ぼすため，ネットワーク構築のための人的・時間的コストも大きくなる．そのため，ネットワークの規模が大きくなるに従い実証環境上への導入が困難になる．

ソフトウェアによる模倣の場合には，netem や NISTNet[3]，Dummynet[2] といった Linux,BSD 系 OS で提供されているソフトウェアを用いることによってネットワークを模倣する．NISTNet は Linux2.4 から 2.6.14 までに対応したものが提供されている．netem は Linux の最新カーネル 2.6.x の一部として配布されている．Dummynet は FreeBSD の ipfw の機能の一部として実装されている．それぞれの機能の違いを表 1 に示す．ソフトウェアを用いることによるメリットは，実証環境で用意されているノードを利用して模倣が可能である事が挙げられる．また，ハードウェアによる模倣手法と比較し，柔軟にリンク特性の変更が可能である．一方デメリットとして，ハードウェアと比較し精度は劣ることが挙げられる．しかし，実験者がネットワーク構築のためのリソースを用意する必要がなく，さまざまなネットワーク環境を構築できるため，本研究では，ソフトウェアを用いた場合のメリットが大きいと考え，ソフトウェアによる模倣を用いる．この様な，リンク特性を模倣するソフトウェアは総称してリンクエミュレータと呼ばれる．

リンクエミュレータを用いてリンク特性を模倣する場合，各ノード自身がそれぞれの特性を模倣することも可能であるが，検証を行うソフトウェアと同じノード上でリンクエミュレータを動作させることは，互いに影響を及ぼす可能性があるため，検証を行うソフトウェアとリンクエミュレータは分けて動作させるべきである．

リンクエミュレータを用いることにより，実証環境上のノードを利用してリンク特性を模倣することが出来る．しかし，リンク特性を模倣するにも図 1 のようにネットワーク機器の間にリンクエミュレータを動作させるノードを用意するため，実証環境のリソースがより多く必要である．これでは，実験者が構築するネットワークの規模が大きくなるに従い，模倣するためのリソースの消費量が大きくなる．そこで，図 2 のように 1 台のノードで複数のリンク特性を模倣する方法をとる．1 台のノードで複数のリンクを模倣することにより，ネットワークの特性を模倣するために必要なリソースを抑えることが可能となる．一方，ソフトウェアでネットワークを模倣する場合には，設定通りの性能が出ているかの保証がないため，同時に 1 台のノードで多重化度の限界を調査する必要がある．模倣を行うためのパラメータとして，1) 遅延，2) 帯域が挙げられている．これは，調査の結果，リンクエミュレータに関する論文[4),7)] で採用されている．調査に使用するリンクエミュレータは，遅延や帯域の操作だけでなく，遅延揺らぎやパケットの再要求なども模倣可能な点から netem を採用した．netem では，帯域制限の機能は本来持っていないため，Token Bucket Filter(TBF) を用いて，帯域制限を行う．本稿では，以後 TBF を用いて帯域制限を行うことも netem による帯域制限と呼ぶ．NISTNet は Linux2.6.14 までで開発が終了しているため，今回は使用しない．

4. netem の性能調査

リンクエミュレータとして netem を用いて模倣可能なリンク特性の限界の調査を行う．netem の性能調査を行うための実験は，遅延の挿入のみを行う単機能実験および遅延の挿入・帯域の制限を行う複合機能実験を行う．実験を行う環境は，多数のノードを同時に制御する必要と実験試行回数が多くなることから，以下の特徴を持つ StarBED を実験環境として採用した．調査を行う項目は，遅延生成のみを行う単機能実験と遅延生成，帯域制限の 2 つを行う複合機能実験を行う．

- 実験ノードが一カ所に集中している

表 1 Dummynet，NISTNet，netem の特徴

	Dummynet	NISTNet	netem
OS	FreeBSD	Linux 2.4〜2.6.14	Linux 2.6
時間解像度	system clock	real time clock	system clock
割り込み箇所	送受信時	受信時	送信時
パケットドロップ	あり	あり	あり
パケット再要求	なし	あり	あり
パケット重複	なし	あり	あり
パケット不正	なし	あり	あり

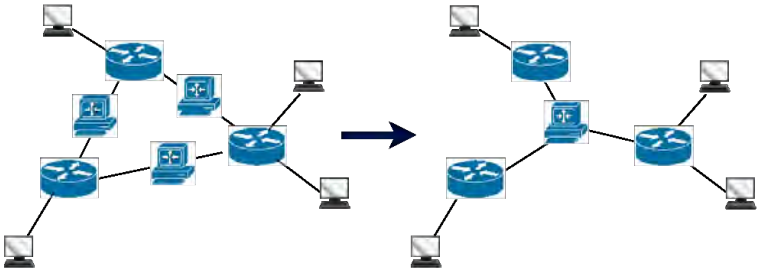

図 2 リンクエミュレータによるリンク特性の模倣の多重化

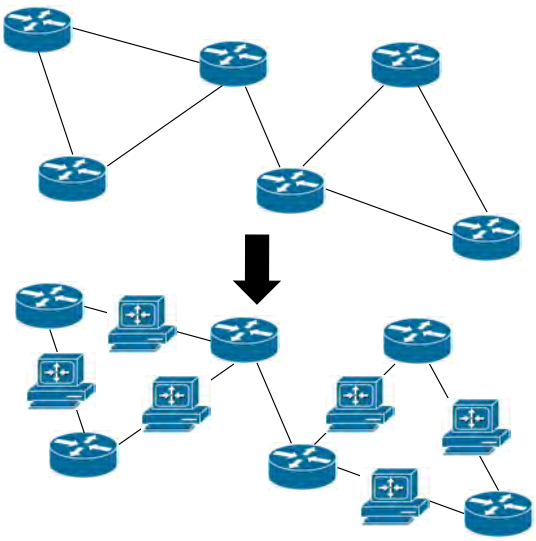

図 1 リンク特性の模倣

(トラブル発生時の修正が容易)

- 一括処理で実験を行うことが可能
- 支援ソフトウェアを使用することにより繰り返しの実験を行うことが容易

4.1 単機能実験

単機能実験として遅延生成のみの性能を測るため，図 3 の構成で実験を行った．トラフィック生成・計測装置 SmartBits 600 と PC ルータとして用意したルータノードを接続し，ノード間にはスイッチやルータといったネットワーク機器は接続されていない．ルータノードの仕様は表 2 の通りである．実験は，SmartBits 600 から送信されるトラフィックに対し，netem で遅延の生成のみを行う．実験時に使用する通信プロトコルは UDP を使用し，送信トラフィックは 14Mbps としている．

netem で設定した遅延と SmartBits 600 で観測される遅延を比較し，リンクの多重度を増やしていったときの模倣可能な限界を調査した．

Linux カーネルのデフォルト設定では netem を有効にした場合の性能に問題があるため，表 3 に示すようなカーネルのパラメータチューニングを行う．net.core.rmem_default および net.core.rmem_max の値は，受信用ウィンドウ・サイズのデフォルト値と最大値を定義している．net.core.wmem_default および net.core.wmem_max の値は，送信用ウィンドウ・サイズのデフォルト値と最大値を定義している．これらの値を大きくすることによって，1Gbps のトラフィック量でも安定して通信を行うことが可能となる．

単機能実験の結果を図 4 に示す．これは，多重化したリンク数と模倣可能であった遅延の限界値をプロットしている．リンク多重度 1〜30 までの結果では，10,000ms の遅延生成でも模倣可能であった．そして，リンク多重度が 80 以上にした場合，総トラフィック量が 1Gbps を超えるため，今回の実験結果には載せていない．リンク多重度が 70 のとき 1,200ms までの遅

表 2 ルータノードの仕様

OS	CentOS 5.3 Kernel2.6.18
CPU	Intel Xeon X3350 2.66GHz x2
memory	8GB
NIC	Intel PRO/1000 x4

表 3 Linux のカーネルシステムパラメータの設定

設定パラメータ名	値
net.core.rmem_default	262144
net.core.rmem_max	262144
net.core.wmem_default	262144
net.core.wmem_max	262144

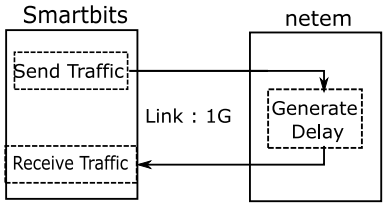

図 3 単機能実験の構成

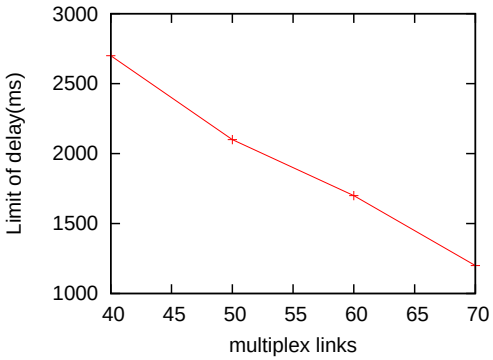

図 4 遅延生成実験

延生成であれば，設定通りの遅延を生成することが可能であった．衛星通信における RTT が約 500ms であることを考えると，1200ms の遅延生成が可能であれば，インターネットの特性を十分に模倣可能である．より高遅延な環境を想定する場合では，リンクの多重度を落とすことにより，模倣可能な最大遅延を高くすることが可能である．

4.2 複合機能実験

遅延生成だけでなく帯域制限も行った状態での複合機能実験を行う．netem の性能を調査するため図 5 の構成で調査を行う．L2 トポロジは図 6 の通りである．各トラフィック生成ノードは，VLAN を使用し個別のネットワークに属しており，netem を動作させるルータノードは，トラフィック生成ノードが所属している VLAN 全てを集約している．ルータノードとトラフィック生成ノードの間には Foundry Networks 製 (現 Brocade Communications Systems) の BigIron RX-32, RX-16，MG8 と Cisco 製 Catalyst 6509 が接続されている．

ルータノードの周りにトラフィックを送信するノードと受信するノードの組を配置する．トラフィック生成に用いたノードのスペックは表 4 の通りである．各ノードはトラフィック送信時にルータノードをゲートウェイとし通信を行う．トラフィック送信・受信ノードは 1 対 1 の組とし，netem での多重度の増加に合わせ，ノードの組も増やしていく．トラフィック生成ノードは Distribute Internet Traffic Generator(D-ITG) を用いて UDP トラフィックの生成を行う．

netem の帯域制限値はデジタル放送に相当する 12Mbps とする．送信ノードのトラフィック量は，netem の帯域制限値を超えた値でなければ，動作確認が取りにくいため，2Mbps 高い 14Mbps のトラフィック量とした．

多重度が増えていくと実験時のノードの数が増加し各ノードの制御が困難になる．そのため，実験には StarBED で提供されている支援ソフトウェアである SpringOS を使用することにより全ノードの同時制御を行う．送信ノードから送信されたトラフィックが，netem で遅延の挿入・帯域の制限が行われた後，受信ノードで受け取られる．この時に，本来受信ノードに届くはずの帯域から欠けたデータグラムの量を調査した．netem で想定通りの帯域制限が行われているのであれば，帯域制限を行った分だけのトラフィック量を受信ノードが受け取るはずである．

実験の結果を図 7 に示す．Delay Setting は設定した遅延の値を示しており，Multiple Links は多重化したリンク数を示している．fail emulate rate(パケット損失率) は，次式で表される．

$$\text{パケット損失率} = \frac{\text{帯域制限値} - \text{受信帯域}}{\text{帯域制限値}}$$

受信ノードが 12Mbps のトラフィックを受け取ったならば，帯域制限は完璧に行われており，パケット損失率は 0%となる．実験結果のグラフは，水平な面と，傾きのある面で構成されている．水平な面は，パケット損失率も少なく，設定した値に近いトラフィック量を受信できている．そのため，この範囲であればリンク特性の模倣はできている．一方，傾きのある面は，遅延の設定値が上がるにつれパケット損失率が高くなっている．傾きのある面では，パケット損失率が高く，

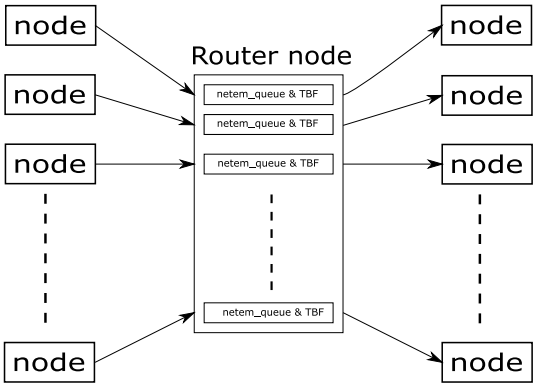

図 5 L3 トポロジ

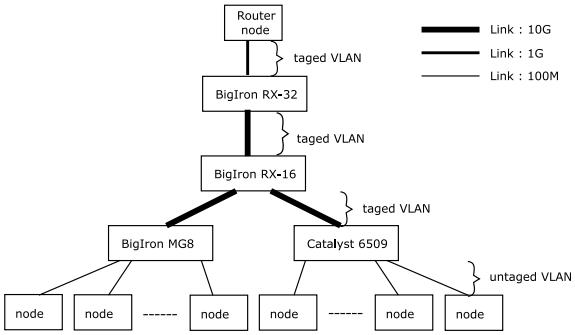

図 6 L2 トポロジ

表 4 トラフィック生成ノードの仕様

OS	Fedora Core 5 Kernel2.6.14
CPU	Intel Pentium3 1GHz
memory	512MB
NIC	Intel PRO/100 x2

設定した値よりも少ないトラフィック量しか受信できていない．そのため，この範囲ではリンク特性の模倣はできているとは言えない．このグラフより，多重度が 70 までかつ，遅延の設定が 100ms までの模倣であれば．正確なリンク特性の模倣が可能である．それ以降の遅延 200ms 以上の場合には，10 本の多重度でもパケット損失率が 10%を超えているため，リンク特性を模倣できていない．

4.3 考　　察

調査の結果，多重化 70 本までの遅延 100ms までの模倣ならば可能であると判断できる結果が得られた．遅延が 100ms を超えた値を設定すると，netem 上での遅延・帯域の模倣ができていない．しかし，広域ネットワークの模倣を 1 台で行うのではなく，複数台のノードを用いて構築すればさらに大きなネットワークを模倣することが可能である．

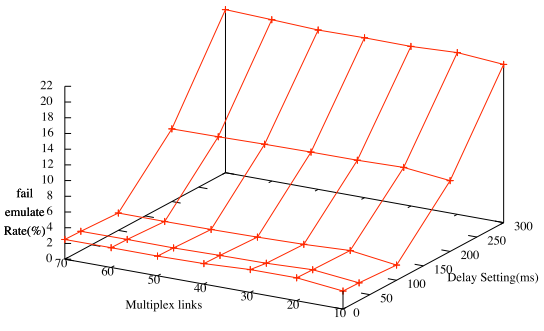

図 7 複合機能実験

今回 netem を動作させるのに用いたノードは StarBED に 240 台用意されているため，それら全てで模倣を行った場合，16800 本のリンク特性の模倣が可能である．より高遅延な環境を構築するためには，複数台のリンクエミュレータを用いて 1 本のリンクを模倣することにより，さらに大規模なネットワークを模倣することも可能である．

現状での注意点として，リンクエミュレータで帯域制限を行う際には，パケットは一度キューに溜められてから送信される．帯域制限値を超えた分のパケットは，次の送信を行うタイミングまでキューの中に溜められる．そのため，帯域制限値を超えたトラフィック量を受信した場合には，設定した遅延よりも長い遅延が観測される．キューのサイズを超えた分のパケットは破棄されるため，帯域制限を行うための最低限の長さにキューを設定する必要がある．実験者が帯域の模倣を行う際には，キューの長さによる遅延の増加を考慮する必要がある．

5. おわりに

本稿では，実証環境上にリンクエミュレータを用いて実証環境上のリソースを効率よく使用するための手法として，リンク特性の模倣を多重化する手法を提案した．複数のリンクを多重化し，各リンク特性の模倣を行った．実験者が予め想定しているネットワーク特性を抽象化し，自動的に実験環境を構築する機構があれば，実験者はネットワーク構築のための人的・時間的コストの削減が期待できる．

実証環境で用意されているノードをネットワークの

模倣に利用することにより，実験者が負担するコストを抑えることが可能になる．そして，用意されているノードが模倣可能な限界について調査することにより，実証環境に構築可能なネットワークの規模の限界を推定した．

今後の課題として，リンクエミュレータの設定・構築を自動化する機構の設計を行う．リンクエミュレータが模倣可能な範囲で実験者が想定する広域ネットワークを自動的に構築することにより，検証を行う際に実験者がリンクエミュレータの設定を行う必要がなくなる．そのため，検証の際の人的・時間的コストをさらに抑えることが可能となる．また，実験中の動的なネットワーク特性の変更を可能とすることにより，検証の幅を広げることが可能となる．現在，StarBED 上に広域ネットワークを構築することを想定しており，StarBED で提供されている支援ソフトウェアである SpringOS と連動して動作可能な実装を考えている．

参 考 文 献

1) Net:Netem, http://www.linuxfoundation.org/en/Net:Netem
2) Luigi Rizzo, Dummynet: a simple approach to the evaluation of network protocols, ACM SIGCOMM Computer Communication Review Volume 27 , Issue 1 , PP.31-41 JAN 1997
3) Mark Carson, Darrin Santay, NIST Net — A Linux-based Network Emulation Tool, ACM SIGCOMM Computer Communications Review Volume 33,Number 3 PP.111-126 July 2003
4) Pramod sanaga, Jonathon Duering, Robert Ricci, Jay Lepreau, Modeling and Emulation of Internet Paths, Sixth USENIX Symposium on Networked Systems Design and Implementation (NSDI), pages 199212, Boston, MA, Apr. 2009
5) Toshiyuki Miyachi, Ken-ichi Chinen and Yoichi Shinoda, StarBED and SpringOS large-scale general purpose network testbed and supporting software , International Conference on Performance Evaluation Methodlogies and Tools (Valuetools) 2006, ACM Press, ISBN 1-59593-504-5, Pisa, Italy, Oct.2006
6) Mio Suzuki, AnyBed, http://sourceforge.net/projects/anybed/
7) Lucas Nussbaum, Olivier Richard A Comparative Study of Network Link Emulators , 12th Communications and Networking Simulation Symposium (CNS) SpringSim'09
8) Emulab, http://www.emulab.net/
9) PlanetLab, http://www.planet-lab.org
10) Eduardo Ciliendo, Tuning Red Hat Enterprise Linux on IBM Eserver xSeries Servers, http://www.redbooks.ibm.com/redpapers/pdfs/redp3861.pdf, 2005