

## N\*(NStar): ns-2 の real external interface の構想

宮地利幸<sup>†</sup> 宇夫 陽次朗<sup>†</sup>  
森島直人<sup>††</sup> 篠田 陽一<sup>†††</sup>

新しく提案されたネットワーク技術を導入する際には、実装された技術がその仕様通りの動作を示すかどうかや、その有効性を検証するためにネットワーク実験環境による検証が必要である。実ノードで構築された実験ネットワークは、実際のインターネットに近い挙動を示すが、必要な規模のネットワークを構築するのは難しい。また、ソフトウェアシミュレータで構築する場合、様々な規模のネットワークの構築が可能であるが、その動作はあくまでシミュレートされたものであるため、実ノードと全く同じ動作を示すかという点を別途検証する必要がある。我々は一般に実験ネットワークには実ノードを用いるべき部分と、ソフトウェアシミュレータで十分実現可能な部分が存在することに注目し、それぞれにソフトウェアシミュレータおよび実ノードを適切に割り当て、実ノードやソフトウェアシミュレータ単独では持ちえない特性を持ったネットワーク実験環境の構築手法を提案する。また、その実装の一つとして、VINT プロジェクトによる ns-2 を拡張し、ns によるネットワークと外部ネットワークの接合システムである N\*(NStar) の構想について論じる。

## N\*(NStar): The conception of external interface for ns-2

TOSHIYUKI MIYACHI,<sup>†</sup> YOJIRO UO,<sup>†</sup> NAOTO MORISHIMA<sup>††</sup>  
and YOICHI SHINODA<sup>†††</sup>

A new technology has to be made experiments and verified, before it is thrown into the Internet because of protection from incorrect actions and making troubles and to know effectively of the technology. A network testbed composed of live node which acts closes to the Internet, but it is difficult to compose the requested scale by experimentation. In other hand, a simulated network testbed we can compose various scale of experimental network. We must verify the acts whether close to the Internet because the network is only simulated. We take notice of an experimental network has various parts which should be composed of live node or software simulator. Then we propose the method making a network testbed allocated live node and software simulator their suitable parts and this network testbed has some special characters what a testbed composed of only live node or simulator doesn't have. And we discuss about N\*(NStar) as a one of the realization, that expand ns-2 by VINT project.

### 1. はじめに

インターネットは情報を伝達するためのネットワークとして爆発的に発展した。その結果、インターネットはある種の社会的基盤となり、その重要性は高まる一方である。

インターネットは、いままで実用ネットワークとし

ての側面と実験ネットワークとしての側面を維持したまま発展を続けてきていた。しかし、最近のインターネットは主に実用ネットワークとしての側面が強調されており、実験ネットワークとして利用することは困難になってきている。その一方で、インターネットの発展自体が、インターネットに対する新規技術を次々に提案し投入する原動力ともなっており、実験ネットワークとしての側面の重要度も同様に増す一方である。

このような矛盾を解決するために、新規技術に対する実験ネットワークとしての役割を別の領域、つまり実験用ネットワーク環境に求めるようになってきている。

現在ネットワーク実験環境を構築するには、実ノードを用いて構築する方法と、ソフトウェアシミュレータを用いる方法がありそれぞれの適応する実験はその

<sup>†</sup> 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

<sup>††</sup> 奈良先端科学技術大学院大学 情報科学研究科  
Graduate School of Information Science, Nara Institute of Science and Technology

<sup>†††</sup> 北陸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

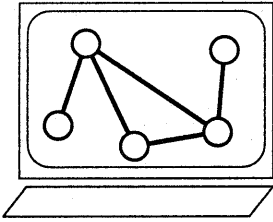


図1 ソフトウェアシミュレータの概念図

目的により使い分けられる。しかし、この2つの手法では対応できない実験が存在する。

本論文では、ネットワーク実験環境を実ノードとソフトウェアシミュレータの両者を統合し両者からお互いを透過的に扱えるネットワーク実験環境の構築手法を提案し、より柔軟なネットワーク実験環境の構築について論じる。そのために実ノードまたはソフトウェアシミュレータ単一で構築したネットワーク実験環境では要求を満たせない実験を考察する。

## 2. ネットワーク実験環境

システムに新たに技術を追加する場合、システムが複雑であるほどその影響を予測することは難しい。インターネットは様々な技術が複合して動作している非常に複雑なシステムであるため、インターネットに対する新たな技術投入についても同様の議論が成立する。

そこで、新たな技術がどのように既存のシステムに影響を及ぼすか、様々な視点から検証する必要がある。一般にシステムに対する影響を検証および評価する方法は極めて多岐にわたり評価対象に応じた選択肢が存在する。

このような評価および検証を目的として構築されたネットワークを『テストベッドネットワーク』と呼ぶ。テストベッドネットワークを構築するための手法には大きく分けて以下の2種類がある。

**実ノードによる環境構築** 実際にネットワーク中で稼働している実装群を用いて、ネットワーク評価用の閉鎖ネットワークを構築し、その中で検証対象の技術を検証する。

**ソフトウェアシミュレーションによる環境構築** ネットワークの挙動を抽象的に表現し、ソフトウェア環境中で検証対象の技術の動作を検証する(図1)。

これらのテストベッドネットワーク構築技術は、検証の対象となる技術が属しているクラスが異なることから利用される領域はほとんど重ならない。

### 2.1 実ノードによる環境構築

実ノードによる環境構築では実際に運用されるものと同じ機器およびソフトウェアを利用できるという点で、実際のインターネットに極めて近い動作が期待できる。

しかし、実ノードを用いた実験および検証は、ネットワーク実験環境を構成する要素数が実験の規模に比例するため、実験の種類によっては構築および運用に膨大なコストが必要となる。

また、実験環境を構築するために必要となる機材が多くなる傾向がある。そのコストを低減するために、同じプラットフォームを利用して複数の実験を共有する場合が多い。

このように実験環境の規模によって生じる問題が多く、実験規模が制約されることが多い。

### 2.2 ソフトウェアシミュレーションによる環境構築

ソフトウェアシミュレータによって構築された実験環境では、ソフトウェア的に仮想環境を構築し、その内部に必要なネットワークエンティティの動作をシミュレートする。そのため、ネットワーク規模はシミュレータの性能に比例し、一般には1台のシミュレータ環境で10~1000000ノード程度のネットワークをシミュレートできる。たとえば、COMPASSプロジェクト[1]によるParallel/Distributed NS (PDNS)[2]は32システムで32000ノードのネットワークでテストされている。

また、そのネットワークトポロジ等の設定を一元的に管理、記述できるため、実ノードを利用した環境構築と比較してネットワーク実験環境の構築にかかるコストは小さい。設定次第で様々な規模の実験環境が構築でき、そのネットワークが示す挙動についても設定は容易である。

しかしソフトウェアシミュレータは、実際にネットワーク上で使われる機器やソフトウェアを模倣したものに過ぎないため、シミュレートされたノードやネットワークが実ネットワークと同様の挙動を示すかどうかは保証されず、実験系としての正当性を別途検証する必要がある。

また多くのソフトウェアシミュレータ上であるソフトウェアを動作させるためには、そのソフトウェアがソフトウェアシミュレータの要求する形式である必要がある。したがって実験対象および、利用する必要のあるソフトウェアをソフトウェアシミュレータで動作をするように改変を行う必要がでてくる。これでは動作検証をするための実装と実際に導入する実装の2つの実装を作成しなくてはならないという点で大きなコ

ストがかかる。この改変によるソフトウェアの動作の確かさも問題となる。

### 3. 実験環境の特色と適応する実験

上で述べたように、それぞれの実験環境構築技術にはそれに応じた適用領域がある。ここでは、各技術が持つ特性を分類し、各技術に適応する実験に関して考察する。

#### 3.1 実験環境構築技術の特性

ネットワーク実験環境の特徴として以下の指標がある。

**ネットワークの挙動** 実験ネットワークの挙動は、その実験ネットワークを構築する技術によって異なる。実ノードを利用した実験環境では、ノードの挙動レベルでは実際の環境に近い結果を得ることができるが、大域的な挙動に関しては追従できない可能性が高い。一方、ソフトウェア環境を利用した場合は、それとは逆になる場合が多い。

検証および評価に必要な挙動を得られない場合は、その結果の信頼性は著しく低下する。

**実験環境の規模** 現在のインターネットで利用する技術の検証を考えた時には、擬似インターネット環境と言えただけの大規模なネットワーク環境を必要とする実験も存在する。したがって、大規模な実験環境の構築を要求されたときに適切な規模の実験環境を構築できる柔軟性が必要となる。

**変更の容易性** ネットワーク実験は一般的に、トポロジやノード、ネットワークの設定など少々の変更とともに似たような実験を何度も繰り返すことが多い。したがって変更にも柔軟に対応でき、また変更が簡単に行えるネットワークの構築が望まれる。

#### 3.2 対象となる実験

行われる実験の内容に応じて、利用される実験環境構築技術は使い分けられる。本節では実験環境の特性とその特性に応じた実験について考察する。

各技術によって構築される実験環境の長所および短所を表1にまとめる。

実ノードを利用する場合には、実際の機器を利用することによってその系の動作を検証するのが容易であるが、実験の規模を大きくしたり、動的にトポロジの変更をおこなうことなどは困難である。

一方、ソフトウェアシミュレータを用いた場合、その系が正しい実験環境となっているかの検証が必要ではあるが、様々な規模のネットワークを構築でき、トポロジ等の設定の変更も比較的容易に行える。

このような特徴からそれぞれの実験環境は以下のよ

うな実験に用いられる。

**実ノードによる実験環境** 実ノードによるネットワーク実験環境では高い信頼性が得られ利用したいソフトウェアを改変することなく利用できることから、新しく実装したソフトウェアの動作確認等に使用されることが多い。

**ソフトウェアシミュレータによる実験環境** 一方ソフトウェアシミュレータは、大規模なネットワークの構築およびネットワークポロジや設定の変更が容易なためルーティングなどノードをあまり意識せず、大規模な網単位での動作の検証に多く用いられる。

しかし、ネットワーク全体の挙動に関して詳細な動作を検証しなければならない実験を考えた場合、これらのどの技術を用いても実験環境を構築し結果をだすことは困難である。

そこで、これらの技術を組み合わせることを考える(図2)。実ノードである必要があるノードとそうでないノードを何らかの方法で区別し、それぞれに実ノードとソフトウェアシミュレータを割り当てることでネットワーク実験環境全体ではないが、実ネットワークに近い挙動を示すネットワークが構築できる。またソフトウェアシミュレータの部分で規模を調節することにより、様々な規模のネットワーク実験環境を構築できる。

## 4. 実ノードとソフトウェアシミュレータを接合した実験環境

一般に実験ネットワークに要求される機能は、『妥当な実験結果を生成するための環境を提供すること』である。既存の実験環境構築技術だけでは対応できないような実験を扱うために、実ノードによって構成されるネットワーク環境とソフトウェアシミュレータを組み合わせたハイブリッドな実験ネットワークを構築することを考える。その場合、対象となる実験ネットワークのどの部分にどの技術を適用するかが論点となる。

### 4.1 実験ネットワークの詳細度

実験ネットワークを構成する要素に要求される機能は、その実験内容に応じて異なる。極めて厳密な動作を要求される領域については実ノードから生成されたネットワーク環境が必要である一方、多数のノードによる大域的な動作が要求される場合には、ソフトウェアシミュレータによるアルゴリズムに則った大域的な挙動が向いている場合もある。また、ネットワーク中で計測対象となる部分なども同様に実ノードを利用し

	長所	短所
実ノード	実運用ネットワークに近い挙動が期待できる。 (運用される機器およびソフトウェアをそのまま利用可能)	環境の構築および維持にかかるコストが大きい。
ソフトウェアシミュレータ	様々な規模のネットワークの構築が比較的容易に行える。 設定等の変更に柔軟に対応できる。 用意する機器が少なくてよいためコストも小さい。	実験系として正しい挙動を示すか別途検証が必要。 必要なソフトウェアを動作させるために改変が必要。

表 1 実ノードおよびソフトウェアシミュレータの比較

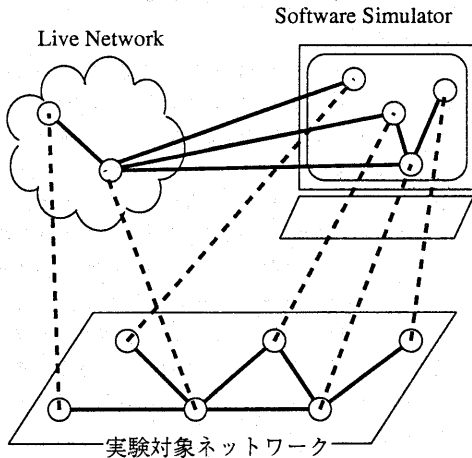


図 2 実ネットワークとソフトウェアシミュレータの接合

て構築するほうが良い場合が多い。

一方、ある挙動要求にしたがって、単にパケットを転送したり適当なトラフィックフローを要求に応じて出力するといった動作は、ある入力に対してそれに応じた出力をするような、ある特定の挙動を示すだけでよく、実ノードを使う必要は特に無い。

ここでは、ネットワークに対して要求される動作を評価するメトリックを詳細度と呼ぶ。厳密な測定を要求するような領域は詳細度が高く、それ以外の場所の詳細度は低いといったように様々な詳細度を持つ部分が実験ネットワーク中には存在する。

#### 4.2 実ノードとソフトウェアシミュレータの分配

実験が要求する実験ネットワークの詳細度は一様では無い。このため、ネットワーク実験環境を構成するすべての部分が実ネットワークと等価の動作をする必要性はない。

つまり、ネットワークは詳細度が高く実運用ネットワークと同様の挙動が求められる部分と、詳細度が非常に低いいわゆる挙動記述のみの記述により、ある入

力に対して適切な出力を行うだけで十分な部分に分けられる。

この詳細度の差異により、実ノードとソフトウェアシミュレータを割り当てる。たとえば、詳細度が高い部分には実ノードを、詳細度が低い部分にはソフトウェアシミュレータを割り当てることにより、大規模かつ部分的に必要な詳細度を持つネットワーク実験環境が構築できるようになる。

このように実験環境を構築すると以下のような利点が生まれる。実ノードで構築されたネットワークからみれば、ソフトウェアシミュレータの利用により、実ノードによる構築の欠点であるコスト消費の軽減が図れる。またソフトウェアシミュレータ上のネットワークは実ノードにより構築されたネットワークの一部であるようにみえため、実ノード側でソフトウェアシミュレータ側と接続するために特別な処理をする必要はなくソフトウェアシミュレータ上のネットワークを透過的に扱うことができる。

一方ソフトウェアシミュレータ側からみると、実運用ネットワークで用いられる要素をその一部として組み込み、実装済みのソフトウェアを実ノード上で動作させられる。これにより実験の信頼性が向上し、ソフトウェアシミュレータで利用するための改変によるコストを節約できる。また実ノード側を見た場合も同様に、実ノードで構築されたネットワークはソフトウェアシミュレータの一部として見える。この場合の利点としてはソフトウェアシミュレータ内の設定と、実ネットワークとの接合点の設定を別にすることなく、一様な設定で対応できるという利点が挙げられる。

以上から我々が想定しているような利用法には、それぞれの構築方を単独で利用するよりも組み合わせで利用することが有効であることがわかる。

実ノードによるネットワークとソフトウェアシミュレータ上のネットワークを相互に透過的に扱うことができれば、ネットワーク管理および設定を簡単でできるという点で、様々なコストの節約を実現でき、本来重

点を置くべき実験の実行により大きな比重を置くことができる。

### 4.3 利用例

大規模かつネットワーク上での実験が必要な問題として、分散型サービス不能 (DDoS) 攻撃を対象とした実験が挙げられる。

DDoS 攻撃を扱うネットワーク環境を構築する場合、攻撃対象となるノードと、攻撃ホストが分布している広域ネットワークを構築しなければならない。攻撃対象となるノードは、その攻撃パターンや防御に関する特性を評価するために詳細度の高い実ノードで構成されなければならない一方で、攻撃ホストが分布している広域ネットワークは、攻撃パターンを再現できれば詳細度を低くおさえることができる。

たとえば、ネットワーク内のルータ群による協調 DDoS 防御に関する実験では、それらのルータ部分に必要な詳細度で設定することで、要求される実験ネットワークを構築可能となる。

## 5. N\*(NStar)

我々は、前述の実ノードとソフトウェアシミュレータとの接合システムの一実装として、ns と外部ネットワークとの接合システムである N\*(NStar) を設計している。

### 5.1 ns

The Network Simulator(ns) は Virtual InterNetwork Testbed (VINT) プロジェクト [3] によりネットワークの研究を対象に開発されたネットワークシミュレータであり、現在ネットワークの動作の検証などに広く利用されている。現在は ns の version 2 である ns-2 が使われている。

ns-2 では、ns で記述されたホストを外部ネットワークと接続することができるように拡張されたが [4]、ns で記述されたネットワークを接合する機構はまだ存在していない。

### 5.2 N\*の概念

複合型ネットワークを構築するためには、ns 内に構築されたネットワークを実ネットワークと同様に外部に接続し、相互に透過的に動作させなければならない。

N\*は、ns 内部のネットワークを、透過的に外部ネットワークと接続する機構である (図 3)。N\*を用いることで、ns と別プロセスで動作している ns を含む外部ネットワークは、相互にそれぞれが ns 上で動作しているのか、あるいは実ネットワーク上で動作しているのかということにかかわらず、互いに接合できる。したがって、詳細度の低い領域を ns で構築し、詳細度

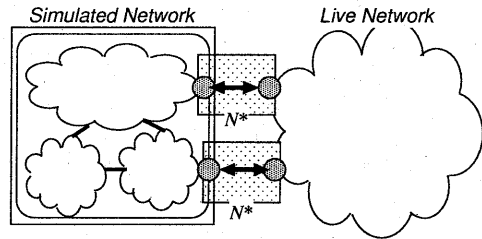


図 3 N\*の概念図

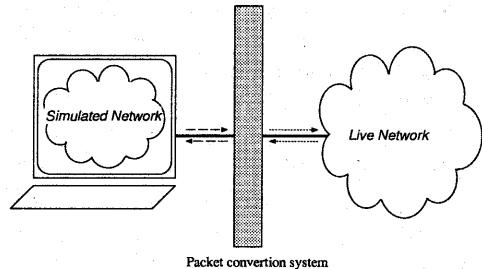


図 4 接合部におけるパケットの変換

の高い部分を実ノードを用いるような実験ネットワークを実現できるようになる。

### 5.3 N\*の設計

N\*を実現するにあたって、実ネットワークとソフトウェアシミュレータの両者がお互いに透過的に通信を行うためには、ns-2 の実装内部で扱われているパケット構造と実ネットワークで扱われているパケットの構造の統一をはかる必要がある。この統一をどのタイミングで実現するかという観点で次の 2 つの実現法が考えられる。

**接合部分における情報変換** ns 内部のパケットは特殊な方法で表現されているため、外部にでる際になんらかの手法で写像する必要がある。ns のネットワークと外部ネットワークとの接点で、パケット構造を相互に変換する機構を実装する (図 4)。

**ns 内の情報保持方法の拡張** ns 内部のパケット構造を拡張する方法である。ns ネットワークと外部ネットワークの接合部では、通常のマルチホームのコンピュータと同様、ns ネットワーク側のインターフェースから、外部ネットワーク側のインターフェースへパケットを転送する形になる。

またそれ以外にも設計するうえで解決しなければならない問題が存在する。

**実ネットワークと ns ネットワークとの時間の同期**  
実ネットワーク上のノードは実時間で動作するが、

ns ネットワーク上では、ネットワークをシミュレーションする負荷により、ns ネットワークの規模およびハードウェアの能力によっては実ネットワークとの遅延が生じる可能性がある。設計時には実ネットワークとの遅延ができるだけ小さくなるよう、ボトルネックを作らないように設計する必要がある。

**シミュレートの高速度化** 対象としているネットワーク実験環境が非常に大規模なものであるため、ソフトウェアシミュレータ部分に大きな負荷がかかる可能性がある。したがって、ns の並列処理化など高速度化についても考察する必要がある。この問題は時間同期の問題とも密接に関係する。

**相互接続性** ns の外部ネットワークという表現をした際には、別プロセスで動作する ns なども含んだ様々なネットワークを想定する。したがって、様々なネットワークと相互に通信できるような配慮が必要となる。

**拡張性** ネットワーク実験環境という性格上これまで存在しなかった新しい技術およびその実装が N\* 上で動作する必要がある。新規技術に対応するための十分な拡張性が必要である。

どちらの方法で実現するかは、今後双方の実現方法の利点、欠点を調査、考察し決定する。

## 6. 今後の課題

我々はネットワーク実験環境構築の手法の一つとして N\* を提案した。N\* はまだ構想段階のものであり実装するには至っていない。今後は ns-2 の拡張方針に関してさらに議論を重ね、以下の手順で実装をおこなう予定である。

**N\* の仕様決定** ns-2 の内部で情報がどのようにやりとりされているのかを調査し、そのうえで、どのような拡張方法が N\* には適しているのか、また、その後の拡張性を考えた仕様を決定する。

**N\* の実装** 策定した N\* の仕様にそって実装を行う。まず ns ネットワークと外部ネットワーク間でパケットのやりとりができる最低限の実装を行う。

**N\* の検証** N\* の実装を用いて動作確認を行う。まず、ns 内部から実ネットワークへ正しい構造のパケットを転送できているか、また、逆に実ネットワークから ns ネットワークへのパケットの転送が正しく行われているかを検証する。

その上で、実際の実験を N\* を用いた複合型ネットワークに適用し、その有用性を検証していこうと考えている。

## 7. おわりに

本論文ではネットワーク実験環境の必要性を論じ、実ノードおよびソフトウェアシミュレータによる実験環境の特徴とそれぞれが適応する実験について考察した。その上で実ノードまたはソフトウェアシミュレータを単一で利用することでは、実験の要求が満たされない実験について論じ、実験対象ネットワークにおける詳細度による、実ノードとソフトウェアシミュレータの割り振りにより、それらを統合したネットワーク実験環境の構築と N\* の構想について論じた。

そのうえで、ソフトウェアシミュレータ環境として ns-2 を利用した複合型ネットワークを構築する機構である N\* を提案した。N\* を実装する上での議論をおこない、実現方法や現在確認している問題点などを考察した。

今後、複合型ネットワーク構築についてさらに考察を進めていく予定である。

## 参考文献

- 1) COMPASS Project  
<http://www.cc.gatech.edu/computing/compass/index.html>
- 2) PDNS - Parallel/Distributed NS  
<http://www.cc.gatech.edu/computing/compass/pdns/index>
- 3) VINT Project  
<http://www.isi.edu/nsnam/vint/index.html>
- 4) Kevin Fall, Network Emulator in the Vint/NS Simulator, ISCC '99