

StarBED： 大規模ネットワーク実証環境

インターネットのような実際にさまざまなサービスが行われている環境に新たな技術を導入するには、対象技術の論理的な検証だけでなく、導入対象環境で動作する実装を、導入対象環境を考慮して構築した実験用の環境で検証することが不可欠である。筆者らは数百台の計算機からなる実験用設備 StarBED を構築し、さまざまな実験を行ってきた。本稿では StarBED および支援ソフトウェアである SpringOS の解説を行うとともに、現在進行中であるユビキタス、センサネットワークを対象とした StarBED2 プロジェクトの概要について述べる。

宮地利幸^{†,††} 中田潤也^{†,††} 知念賢一^{††,†}

Razvan Beuran^{†,††} 三輪信介^{†,††} 岡田 崇^{††,†}

三角 真[†] 宇多 仁^{††,†} 芳炭 将^{††,†}

丹 康雄^{††,†} 中川晋一^{†,††} 篠田陽一^{††,†}

[†] 情報通信研究機構 ^{††} 北陸先端科学技術大学院大学

実ノードを用いた実験の必要性

インターネットはそもそも実験的に研究機関を接続するネットワークとして構築され、多くの技術はインターネットそのものの上で検証されてきた。しかし、昨今の一般家庭への普及によりさまざまな重要なサービスがインターネットを用いて提供されるようになってきており、このようなサービスへの影響を考慮すると、今やインターネットを実験用のネットワークとして利用することは適切でない。そこで、さまざまな手法でインターネットのような実運用が行われている環境(実環境)を模倣し、その上で実験が行われている。

ソフトウェアシミュレータは最も広く利用されている手法の1つであり、ソフトウェアを用いて対象環境の計算機やアプリケーションソフトウェア、ネットワークをモデル化および抽象化し実験用の環境を構築する。ソフトウェアシミュレータには ns-2¹⁾ や SSFnet²⁾, OPNet³⁾ などのように、ネットワークの基本要素を提供し、実験実行者は必要な部分のみを新たに実装するといった汎用的に利用できるものと、実験実行者がシミュレータ全体を構築した対象技術専用のものがある。ソフトウェアシミュレータを利用すれば、実験実行者の理想的な環境を構築することができるため、プロトコルやアルゴリズムの論理的な検証に向いている。しかし、一般的に実環境用の実装を用いることはできず、対象技術をモデル化・抽象化したソフトウェアシミュレータ専用の実装を用意

する必要がある。

実環境用の実装の検証を行う場合には、実環境に存在する要素そのものを用いた実験用の環境が必要となる。小規模な環境であれば、各組織で実験用の環境を構築することもできるが、実験用のノードを用意し、維持するための金銭的コストや、構築した環境を保持しておくためのスペースなどにかかる費用は小さくない。また、小規模な環境での実験では対象技術のスケラビリティの検証を行う際には不十分な場合も多い。このような問題を解決するために、多数の計算機が用意された実験専用の設備が提案されている。

我々は、このような多数の実験用の計算機およびネットワーク機器を1カ所に集めた設備として StarBED の研究開発および運用を行ってきた。StarBED はインターネットを対象とした大規模実験設備であり、いかに多数のノードを制御し、実験実行者の意図する実験を容易かつ柔軟に実行するかに着目して研究開発を行っている。本稿では実験設備である StarBED と、StarBED に存在する実ノード群を制御するための支援ソフトウェア SpringOS の概要を述べる。また、我々は、StarBED を拡張し、ユビキタスネットワークの模倣を目的とした StarBED2 プロジェクトを進めている。このプロジェクトで研究開発している、ユビキタス環境を構成する実装を変更なく動作させるためのプラットフォームである Rune および、ネットワーク特性を模倣するためのソフトウェア QOMET についても現状を報告する。

既存研究

本章では、実環境用の計算機とネットワーク機器で構成される Emulab⁴⁾ と GARIT, 仮想化技術を利用した PlanetLab⁵⁾, VM Nebula⁶⁾ について述べ、我々が構築した StarBED の目的について述べる。

本稿では、大規模な実験設備での実験実行を支援するソフトウェアを実験支援ソフトウェアと呼び、大規模な実験設備に実験支援ソフトウェアが用意されている環境を大規模実証環境と呼ぶ。ソフトウェアシミュレータや大規模実証環境は実験を行うための環境として実験実行環境とし、実験実行環境上に構築された1つの実験を行うために必要な要素群の集合を実験駆動単位と呼ぶこととする。実験駆動単位には、実験用のノードや管理用ノードなどが含まれる。図-1に実験駆動単位と実験実行環境の概念図を示す。

● Emulab

Emulab は、分散システム、実ノードによる環境、ソフトウェアシミュレータの統合環境であり、豊富な実験管理機能を持つ。物理接続が変更可能なパッチパネルのソフトウェアによる制御および、VLAN 技術の利用により柔軟な実験トポロジを構築できる。また、ns-2 を実ネットワークに接続できるように拡張した nse⁷⁾ を利用することで、ソフトウェアシミュレータと実ノードによる環境を接続し、大規模かつ柔軟な環境構築を可能とするが、ns-2 により実現している部分では、実環境と同一の実装は扱えない。また、ns-2 上と実ノードによる環境では時間の進行速度が異なり、実験に影響をおよぼす可能性もある。

ある程度の規模の PC クラスタを持つサイトを接続することで、大規模な環境を構築しており、実環境のリンク特性を導入できる。

Emulab を拡張した多くの研究が進められている。Flexlab⁸⁾ は PlanetLab を構成するノード間のリンク特性を Emulab 上に導入するための機構である。PlanetLab 上のあるノードから他のノードまでの帯域や、ジッタなどの情報を観測しておき、これを dummynet に変更を加えたソフトウェアを用いて Emulab 上に再現する。PlanetLab 上のリンク特性をリアルタイムに Emulab 上の環境に導入することも可能である。

実験の実行部分だけでなく、計画部分なども補助するため、これまで行われてきた実験の情報をデータベースに保存しておく試みも始められている⁹⁾。

DETER¹⁰⁾ は、セキュリティ実験用テストベッドとして、Emulab をベースとして開発されている。Emulab の対外接続部のセキュリティ対策を高度化し、実環境へ

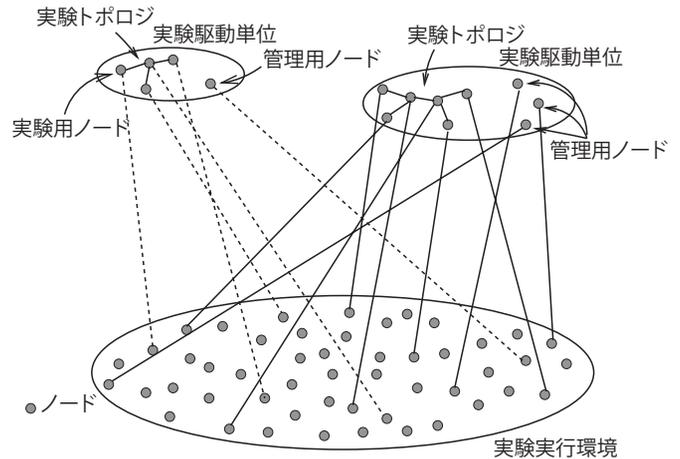


図-1 実験実行環境と実験駆動単位

の影響を排除するような工夫がなされている。

● GARIT

GARIT は、種類や性能が均一でない実ノード群からなる一拠点集中型の実験設備であり、実験支援ソフトウェア AnyBed¹¹⁾ が利用されている。AnyBed は L3 トポロジの設定を行うソフトウェアであり、まもなくハードディスクに導入されている OS、もしくはディスクレスで起動したノードで動作する経路制御ソフトウェア群 Quagga の設定を変更することにより L3 の実験トポロジを構築するが、シナリオの実行機能は備えていない。なお、Anybed は GARIT に特化したものではない。

● PlanetLab

PlanetLab はさまざまな組織に設置された計算機をインターネットを通して接続し構築した、実験用のオーバーレイネットワークである。ノードを多重化するため、Linux の Virtual Server を利用しており、利用できる OS に制限がある。利用者は、PlanetLab を構成するノードの資源の割り当てを受けることで実験用のトポロジを構築する。

● VM Nebula

VM Nebula は、PC エミュレータによるセキュリティ用途向けの実証環境で、実機による実験環境の再現精度とエミュレータによる実験環境の柔軟性、耐規模性を兼ね備えることを1つの目標とした環境である。4台の高性能な実ノードを用いており、論理的に大規模な実験駆動単位を構築できる。PC エミュレータを利用しているため、実環境で動作する OS 実装やサーバ実装、攻撃ツールなどをそのまま利用することができる。

仮想 PC の構成やマルチレイヤスイッチの設定などを

保存、配布する機能を持っているため、一度構成した実験駆動単位を再度構成することや再利用することが容易にできる特徴があり、頻繁に再実験を必要とするウィルスやワームの解析などにも用いることができる。

● StarBED の提案

このような大規模実証環境を利用することで、ソフトウェアおよびハードウェアの実装中のバグまでを含めて検証できる。ただし、それぞれの環境は目的や実装方法が異なるため、その上に構築される実験駆動単位の性質が異なる。

PlanetLab のようなインターネットにまたがるオーバーレイ環境では、各地にある余剰資源および物理的スペースの有効利用や、実験駆動単位へのインターネットのリンク特性の導入などの利点がある。しかし、実験トラフィックがインターネットの実トラフィックの影響を受けるため、再現性が確保できない点、挙動の把握が困難である点、各ノード間の接続部が実験実行者の管理下にないことから観測が不可能である点などの問題が存在する。特に PlanetLab は、インターネットの他のサービスとの分離がなされていないことから、インターネット上の既存のサービスに影響を与える危険性もある。したがって PlanetLab を用いた実験は、別の実験実行環境を用いて対象技術の安全性がある程度確認された後に行われるべきである。

Emulab や PlanetLab は分散したサイトを接続して大規模な実験駆動単位を構築する。このため、インターネットのトラフィックなどのさまざまな影響を受け、実験の再現が困難だけでなく、問題が発生した際の問題の切り分けは困難である。

PlanetLab や VM Nebula では実験資源の有効利用や、ノードの容易な制御のために仮想機械を利用している。VM Nebula では VMWare を利用しているため、x86アーキテクチャ以外のハードウェアを対象にしたソフトウェアを利用できない。PlanetLab では Linux VServer を利用しており、カーネルの変更ができないだけでなく、採用されている Linux OS で動作するアプリケーションソフトウェアしか利用することができないという問題点も存在する。

GARIT はさまざまな性質のノードを利用でき、現実的な実験駆動単位を構築できるが、その一方管理上のコストが高くなるおそれがある。

StarBED が設置された、2002 年当時はここで挙げた大規模実証環境のほとんどは存在しておらず、実ノードを用いた大規模な実験を行うことは不可能であったため、我々は多数の計算機を 1 カ所に隔離した状態で設置した実験専用の設備である StarBED の提案・実装を行った。

StarBED では、実験に利用できる OS やカーネルなどの制限の解消、対象のハードウェアを接続することによるハードウェアの検証、隔離された環境の利用による外乱の排除と再現性の向上を実現している。2007 年現在でも、それぞれの大規模実証環境が持っている機能や特性は異なっているため、実験目的による選択が必要である。

次章で StarBED について述べる。

StarBED

StarBED は、実ノードを用いた大規模な実験を行うための実験設備として、石川県能美市の情報通信研究機構北陸リサーチセンターに設置されている、一拠点集中型の PC クラスタベースの実験専用の環境である。どのようなテストベッドにも対応できることを意味する、* (ワイルドカード) BED という意味と、星のように輝く新たなアイデア (Star) を創成するゆりかご (Bed) として利用されることを願ってこの名がつけられた。

StarBED を用いれば大規模な実験を行うことが可能となるが、StarBED に存在する多数のノードすべてを 1 台ずつ制御することは現実的でない。したがって、各ノードを恒常的に制御できるような接続手段を前提とし、支援ソフトウェアによる自動的な制御が可能となるよう、設備面として以下の事項を遠隔操作により実現するように設計・実装された。

- ノードの起動方法の切り替え
- 電源管理
- 対象トポロジ構築
- ノードへの OS およびアプリケーションソフトウェアの導入

また、これ以外に、実験の精度と柔軟性の向上のため、実験用トラフィックと管理用トラフィックの分離、対外線提供も考慮されている。

本章では、StarBED の設備と、上記の要求を満たすためのアプローチについて述べる。

● 実験用ノード

2007 年 10 月現在、StarBED は、680 台の実験専用の計算機ノード (図-2, 3) とそれらを接続し、実験用トポロジを構築する 7 台のスイッチノード (図-4, 5) から構成されている。

実験用トポロジでは、実験実行者が求める任意の IP アドレスや経路などのネットワーク設定が利用できることが好ましい。この要求を満たすためには、実験駆動単位の設定時に、各ノードに接続できる必要がある。そこで、計算機ノードに最低 2 つのネットワークインタフェースを用意し、そのうち 1 つを管理に利用し、そのほか



図-2 StarBEDの計算機ノード(2002年導入)

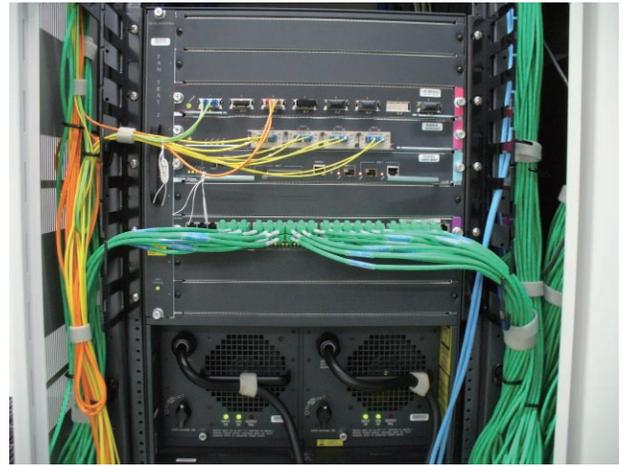


図-4 StarBEDのスイッチノード(2002年導入)



図-3 StarBEDの計算機ノード(2006年導入)



図-5 StarBEDのスイッチノード(2006年導入)

のネットワークインタフェースは実験に用いる。管理用のインタフェースおよびそれが接続されているネットワークは固定的な設定がなされており、常に各ノードへの接続ができることとする。これにより、管理用のネットワークを通して実験駆動単位構築のための設定が行える。

ネットワークを通じての電源の投入を行うため WoL もしくは IPMI に対応した計算機ノードを導入した。StarBED 設置当時 (2002 年度) に導入した計算機ノードは IPMI を搭載しておらず、WoL にのみ対応している。WoL では電源の投入のみ実現されており、電源停止や再起動などの制御が行えないが、IPMI や iLO では上記制御も行える。

計算機ノードの起動方法を切り替えるため、すべての計算機ノードは、起動に必要なファイルを起動時にダウンロードすることによるネットワーク経由での起動を行い、各ノード向けのファイルを変更することで、ノード

の起動方法を変更する。この方式に対応するため、計算機ノードはネットワーク経由で起動できるように PXE に対応したものを導入した。

自動的に対象トポロジを構築するために実験用ネットワークスイッチは VLAN および ATM に対応した製品を導入した。VLAN および ATM の Virtual Path/Virtual Channel を利用することにより対象トポロジを仮想的に構築できる。StarBED の設備としては L2 トポロジの変更に対応する。

計算機ノードの構成を表-1 に示す。これらのノードは構成によりグループに分けられており、グループ A からグループ F までの 6 種類の構成のノードが用意されている。

●ネットワーク構成

実験用ノードの各種管理および制御を行うために、各

		A	B	C	D	E	F
モデル		NEC Express 5800					
		110Rc-1		120Ra-1	110Rc-1		110Rg-1
チップセット		ServerWorks LE					Intel E7230
CPU	タイプ	Pentium3					Pentium4
	クロック	1GHz					3.2GHz * 2
メモリ		512Mbyte					2Gbyte
HDD	タイプ	IDE		SCSI	IDE		SATA
	容量	30Gbyte		36Gbyte	30Gbyte		80Gbyte*2
ネットワーク インタ フェース	ATM	0	1	1	0	0	0
	FE	0	1	4	1	4	0
	GbE	1	0	0	0	0	4
電源管理機構		WoL					WoL/IPMI
台数		208	64	32	144	64	168
導入時期		2002年4月					2006年4月

表-1 StarBEDの計算機ノード構成

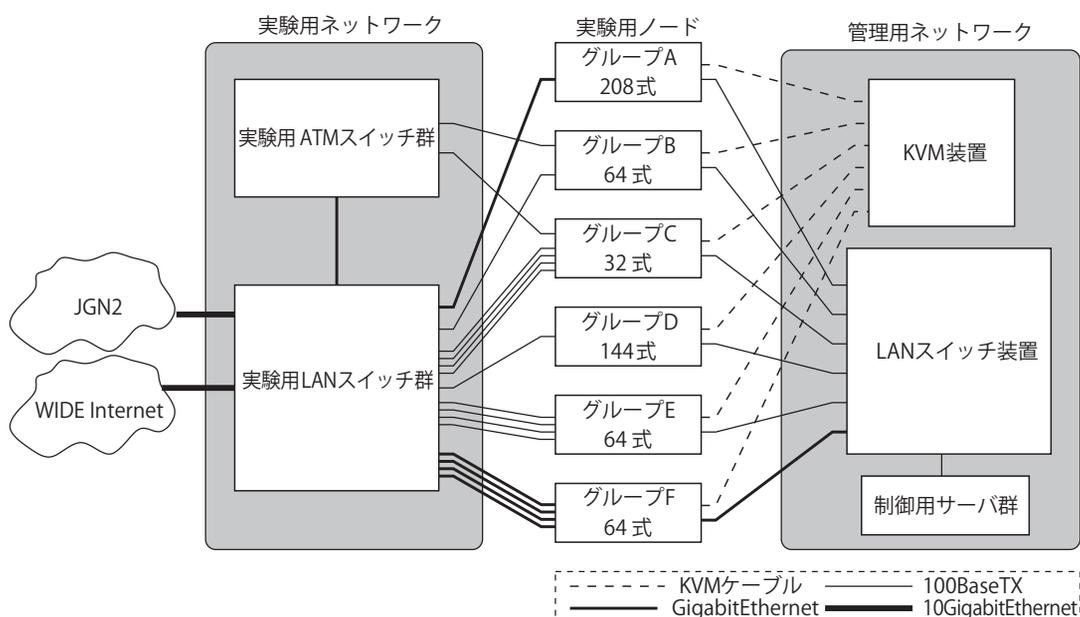


図-6 StarBEDのトポロジイメージ

実験用ノードは実験用、管理用のネットワークに接続されるとすでに述べた。つまり、StarBEDには、実験用と管理用のネットワークがそれぞれ独立に用意されている。

管理用と実験用ネットワークを分離することにより、それぞれのトラフィックの相互影響の排除も実現している。また、管理側ネットワークには、ノードの自動的な起動や電源管理のためのサーバを設置している。実験中に実験用ノードに障害が発生した場合に、実験用ノードが接続されたすべてのネットワークからの制御が不可能になることは珍しくない。このため、Ethernetによる接続以外に、コンソール表示の確認と、キーボードとマウスによる入力を一括して管理できるKVM装置を導入している。

さらに、インターネットの実トラフィックを導入する

場合や、遠隔地に存在する実験施設やサーバに接続が必要な場合には、WIDE InternetおよびJGN2による接続が可能である。

StarBEDのトポロジの概念図を図-6に示す。

管理用ネットワークと実験用ネットワークの分離と、PXEおよびWoL、IPMIに対応した計算機ノードの導入により、恒常的な実験用ノードへの接続、ノードの起動方法の切り替え、電源管理、ノードへのOSおよびアプリケーションソフトウェア導入を実現した。また、VLAN、ATMの利用により、対象トポロジ構築が可能である。さらに、StarBEDには持ち込みハードウェアを設置するためのラックが用意されており、実験実行者が

StarBED : 大規模ネットワーク実証環境

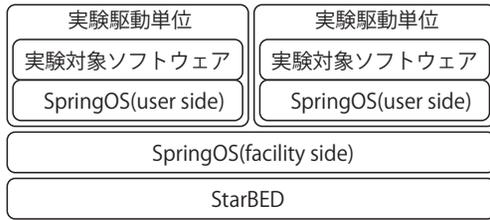


図-7 StarBED と SpringOS

持ち込んだハードウェア実装を StarBED の実験用ネットワークに組み込むことが可能である。

このような設備を用いても、実験を人の手で実行した場合には、時間的コストが大きだけでなく、設定ミスなどによる実験精度および実験の再現性の低下が発生するおそれがある。このような設備をソフトウェアで自動的に制御し、ノードへの OS の導入や、実験シナリオの制御、トポロジの変更などを自動的に行うことにより、実験の精度・柔軟性・再現性の向上が期待できる。次章では我々が開発している StarBED 上での実験を支援するためのソフトウェア SpringOS について述べる。

SpringOS

SpringOS は StarBED の設備を適切に設定し、実験実行者が要求する実験を自動的に行うために設計・実装されたソフトウェアモジュール群の総称である。StarBED というベッドを支えるバネとして働くようにこの名がつけられている。図-7 に StarBED 上に構築される実験駆動単位と SpringOS の関係を示す。資源管理機構など施設全体として運用されるモジュールと、シナリオ実行機構のように各実験ごとに動作するモジュールがあり、後者は同時に複数起動される可能性がある。

実験は通常、図-8 (a) のような順序で行われ、StarBED を利用することで、手順 2 を省略できる。SpringOS は手順 3 から手順 6 を補助し自動で実行する機能を提供するため、SpringOS を利用した場合の実験手順は図-8 (b) となる。

実験実行者は SpringOS 専用の記述言語により、実行する実験の詳細を記述する (図-8 (b) 手順 ii)。この記述を SpringOS に入力し実行する (図-8 (b) 手順 iii) ことで、実験が実行される。

本章では、これらの自動化に必要な機能の詳細と、SpringOS の動作手順について説明する。

● SpringOS が提供する機能

SpringOS は入力された設定記述にもとづき、必要な資源の確保と、必要に応じてノードへの OS の導入、実験トポロジの構築、そして実験シナリオの実行を行う。

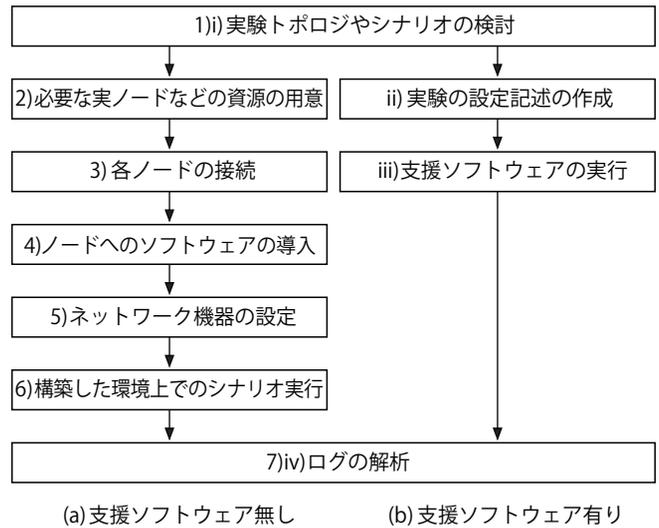


図-8 実験の実行手順

これらを実現するために必要な機能は、資源管理機構、実験用ノードへの OS やアプリケーションソフトウェアの導入および設定機構、実験側ネットワークの VLAN 設定変更機構、各ノードでのシナリオ実行機構、電源管理機構、そして、起動方法変更機構である。以下でその詳細についてまとめる。

【資源管理機構】

実験を行うためには実験実行者は、実験設備の資源の貸し出しを受け、その資源の設定を変更することで実験駆動単位を構築する。このため、実験実行者が必要とする資源の検索や割り当てを行い、貸し出された資源が他の実験実行者に割り当てられないよう排他制御を行う機構が必要である。このような排他制御を行うことで、StarBED 上で同時に複数の実験駆動単位を構築できる。

実験資源の情報として、各ノードの MAC アドレスや接続先・種類・数などのネットワークインタフェースの情報がある。また、ディスクの種類や状態監視に利用できる手法なども情報に含まれる。SpringOS では実験トポロジの構成に VLAN などを用いているため、VLAN を設定するために必要な VLAN 番号なども資源として扱われる。

【実験用ノードへの OS およびアプリケーションソフトウェアの導入および設定機構】

実験用ノードの設定には、OS やアプリケーションソフトウェアなど、複数のノードで同一のものが利用できるものと、IP アドレスなどのネットワーク設定のように各ノードで固有のものが存在する。SpringOS では、OS やアプリケーションソフトウェアなど、複数のノードで共通して利用できるものを導入し、固有の設定については OS およびアプリケーションソフトウェアの導入後、シナリオの一部としてソフトウェアを実行して設

定を行う。

ノードへの OS やアプリケーションソフトウェアの導入には、(1) まえもって実験実行者が作成したハードディスクのイメージを対象ノードのハードディスクに書き込むことによる方法と (2) ディスクレスシステムとして各ノードを動作させる方法、(3) 標準的にインストールされている OS を利用する方法に対応している。

(1) では、実験実行者は StarBED の 1 台のノードの割り当てを受け、そのノードに必要な OS やアプリケーションソフトウェアのインストールと複数のノードで共通して利用できる設定を行う。その後、このノードのハードディスクの内容をバイナリイメージ（ディスクイメージ）としてファイルサーバに保存する。ここで作成したディスクイメージを各ノードのハードディスクに書き込むことで、実験実行者が最初に設定したノードと同様のソフトウェア構成、設定を持つノードを複製する。

ディスクイメージの生成を支援するモジュールも用意しており、1 台のノードに必要なシステムを構築すれば、その環境を多数のノードに複製できる。

(2) では、ディスクレスシステム用のディスクイメージとカーネルイメージを作成し、これを利用し起動する。この場合は、導入が非常に短時間でできる利点がある。ディスクレスシステムを構築する方法には、TFTP を利用し、ディスクイメージを取得した後、メモリ上のファイルシステム上に展開する方法および NFS を利用する方法が利用できる。頻繁にディスクアクセスが必要な実験では、NFS を利用しネットワークを経由でディスクにアクセスを行うと、実験結果に影響をおよぼすおそれがある。ただしログの記録程度であれば、ログ保存用スペースを実験用ノード上に確保することで回避できる。

StarBED の計算機ノードには標準的に Windows/FreeBSD/Linux が導入されており、実験によってはこれらをそのまま利用できる。(3) は、これらの OS にシナリオの一部として必要なアプリケーションソフトウェアや設定ファイル、設定補助プログラムなどを FTP サーバなどからダウンロードし導入・設定する方法である。

【VLAN 設定変更機構】

前述のとおり、StarBED では実験トポロジを実験用ネットワークのスイッチの VLAN および ATM を用いて設定するため、SpringOS は StarBED のこの機能を利用し、実験用ネットワーク上のスイッチの設定を変更することで、実験トポロジを構築する。ただし現状では VLAN のみに対応している。

【シナリオ実行機構】

実験シナリオは、管理用ノードで動作する SpringOS の管理用ソフトウェアと、実験用ノードで動作する SpringOS のクライアントソフトウェアが協調して実行

する。したがって、シナリオの自動実行のためには、実験用ノードにこの実験制御用のソフトウェアを導入し起動しておく必要がある。

管理用ソフトウェアは設定記述を読み込み、実験前に各実験用ノード用のシナリオをクライアントソフトウェアに配布する。これを受け取ったクライアントソフトウェアは、基本的にシナリオに記述されたソフトウェアを順次実行する。各実験用ノードが別のノードの挙動をトリガとしてイベントやソフトウェアを実行するといったノード間のシナリオの同期が必要な場合は、クライアントソフトウェアによる管理用ソフトウェアへのメッセージ送信や、あるメッセージを受信するまでシナリオの実行を停止するといった処理により同期を実現する¹²⁾。

図-9 にシナリオ実行の概念図を示す。

【電源管理機構】

SpringOS では、StarBED の計算機ノードがサポートする WoL によるノード起動と、各ノード上で動作させた SNMP デモンと通信することによる電源断、再起動を実現する。IPMI に関しては現在対応作業中である。

【起動方法切り替え機構】

計算機ノードの起動方法には、ローカルのハードディスクに導入されている OS を利用する場合やディスクレスシステムとして起動する方法などがあり、実験によって使い分けられる。柔軟な実験を行うためには、上記の起動方法や、ハードディスクを用いた場合には利用するパーティションを選択できる必要がある。これを実現するために StarBED のノードは常に PXE によるネットワークブートを行い、ブートローダを取得する。各パーティションから起動するためのブートローダをそれぞれ用意し、各ノード用のブートローダを変更することで利用するパーティションを変更する。これにより、DHCP オプションで指定されたファイルのシンボリックリンクを切り替えることで、ノードごとの起動パーティションを変更できる。ディスクレスで利用する場合には、設定記述中に利用するファイル群を指定し、これらを利用するように変更を行う。

【その他】

ログは実験により出力形式や出力場所などが一定でない。したがって、SpringOS では、シナリオ実行によるログの収集を行う。これは特別な機構を用意しているわけではなく、実験実行者がシナリオとして FTP などを用いたファイルの転送を実行する。

StarBED にはリンク特性の変更のためのハードウェア機器が存在しないため、dummynet や NIST Net, netem といったソフトウェアにより実現する。これらは、SpringOS の設定として記述することで自動的に実現できる。また、シナリオの一部としてこれらのアプリケー

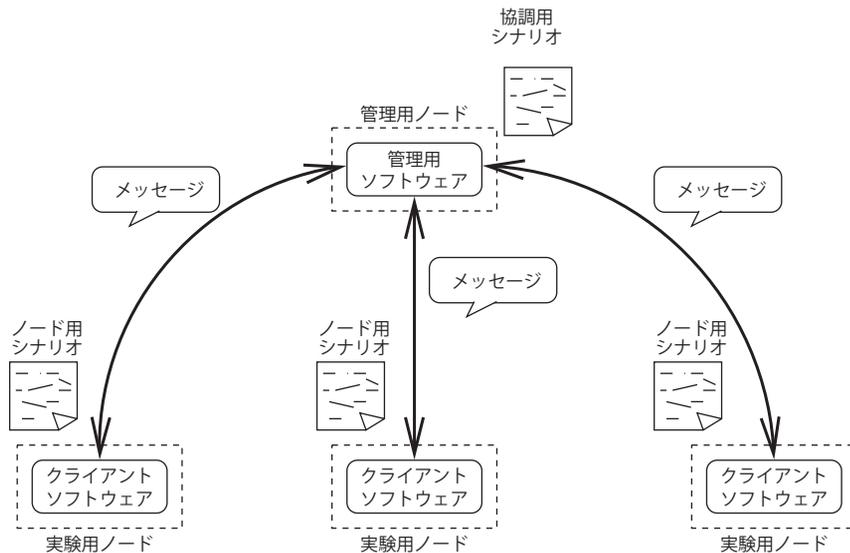


図-9 SpringOS でのシナリオ実行

ションソフトウェアを起動するように記述することも可能である。

● SpringOS の利用

SpringOS は専用に用意された言語で記述された設定に従い、利用できる物理ノードの検索や、各ノードに導入するソフトウェア情報や起動方法・シナリオなどを認識する。その後認識された内容をもとに、実験駆動単位を自動的に構築し、その後シナリオを実行する。したがって、実験手順は図-8 (b)となる。

実験用ノードでシナリオ実行のためのクライアントソフトウェアが起動した状態であれば、管理用ソフトウェアからシナリオを実行させることができるため、一度構築した実験駆動単位を利用して、繰り返し別のシナリオを実行することもできる。この機能を用いて、実験駆動単位の構築後、シナリオを分割して必要なソフトウェアの実行や、ログの収集などを行うことも可能である。

StarBED2

StarBED2 はこれまで述べてきた StarBED に拡張を加え、ホームネットワークやセンサネットワークを含むユビキタスネットワークの開発や検証を可能とするプラットフォームの構築を目指すプロジェクトである。

本章ではユビキタスネットワークの特徴、およびその大規模実証環境の要件を明らかにし、我々のアプローチについて述べる。

● ユビキタスネットワークの特徴

ホームネットワークやセンサネットワークに代表され

るユビキタスネットワークは、インターネットが「均質なノードからなる密結合なネットワーク」であるとする「不均質なノードからなる疎結合なネットワーク」である。ユビキタスネットワークでは、その用途に応じ、ハードウェア、ソフトウェアの双方の面から必要最低限の能力を持つプラットフォームの選択が行われる。また、こうしたヘテロジニアスなノードがアプリケーションの要求に見合う数だけ集合し、協調動作を行う。こうして構成されるネットワークは、場合によっては無数のノードから構成されることになる。しかし、ユビキタスネットワークにおいては、ネットワークに属する

ノードのうち、必ずしもすべてのノードが常時稼働しているとは限らない。たとえば、典型的なシンクセンサネットワークでは、電池駆動のセンサノードは電力消費をできるだけ抑えるため、アクティブ期間とスリープ期間を周期的に繰り返す間欠動作を行う。ユビキタスネットワークにおけるノード群は互いにネットワークを介して対話しながら動作を行うが、その際、媒体とプロトコルの両面から多様なネットワークが用途に応じて選択される。こうしたユビキタスネットワークが持つさまざまな意味での多様性に加え、ユビキタスネットワークの最たる特徴として、周囲の環境、たとえば温度場、電磁場などとの対話が挙げられる。つまり、ユビキタスネットワークでは、ノードが何らかの形で周囲の環境に影響を与え、または周囲の環境から影響を与えられて動作を行うことが多い。この場合の周囲の環境にはそのネットワークを利用するユーザ、つまり人間も含まれる。このような周囲の環境との対話が重要な要素であるということは、ユビキタスネットワークにおいては、ノードの地理的情報や時間的情報の重要性が相対的に高くなることを意味している。

● ユビキタスネットワーク実験環境に対する要求

前節で述べたような特徴を持つユビキタスネットワークをシミュレートする大規模実証環境が持つべき機能としては、

- (1) 大規模ネットワークのシミュレーションにも対応可能なスケーラビリティ
- (2) さまざまなネットワーク媒体およびプロトコルをエミュレートすることが可能な機構
- (3) 多様なノードアーキテクチャをエミュレート可能な機構

(4) 周囲の環境をシミュレートする機構

などが考えられる。

この分野では多くの先行研究が行われてきているが、その多くは離散的なイベントモデルを用いたシミュレーションを行っており、実際のノードで利用される実装そのものをエミュレータ上で実行するものは、TOSSIM¹³⁾、ATEMU¹⁴⁾、Avrora¹⁵⁾ など、少数である。また、周囲の環境を含むシミュレーションが可能なものも OMNeT++¹⁶⁾、Sidh¹⁷⁾、SENS¹⁸⁾ などが部分的に対応しているのみである。

●ユビキタスネットワーク実験環境 Rune

StarBED2 では、図 -10 に示すように、StarBED のハードウェア、ソフトウェア構成の上位に Rune (Realtime Ubiquitous Network Emulation platform) と呼ばれるソフトウェア層を実装し、その上位でユビキタスネットワークにおけるあらゆるエミュレーション対象、シミュレーション対象を実行することによってユビキタスネットワークの挙動を模倣することを可能としている。

エミュレーションやシミュレーションの対象は Space という概念で扱われ、(1) のスケラビリティの確保、および (4) の周囲の環境のシミュレーションを可能とし、多目的ネットワークエミュレータ QOMET (Quality Of transforMing Environments Testbed) を利用することで、(2) のさまざまなネットワークのエミュレーションを可能としている。さらに、マルチレベルエミュレーションレイヤと呼ばれる機構を提供することで、(3) の多様なノードアーキテクチャをエミュレートすることを可能としている。

●多目的ネットワークエミュレータ QOMET

QOMET は、StarBED2 プロジェクトにおいて Rune と並行して開発が進められている多目的ネットワークエミュレータである。QOMET の基本的な動作は、通信が行われている環境に依存するパラメータに基づき、伝送品質の劣化を示す ΔQ を求める第 1 フェーズと、 ΔQ をネットワークエミュレータが解釈する記述に変換を行う第 2 フェーズの 2 段階の処理から構成されている。第 1 フェーズでは、通信を行う 2 ノードの送受信機の能力やノード間の環境などから、帯域幅、フレーム損失率、遅延などのパラメータを含む ΔQ を求める。第 2 フェーズでは、 ΔQ を dummynet や NIST Net などのネットワークエミュレータが解釈する記述に変換を行う。

●Rune の構成

Rune では、ノード、ネットワーク、物理環境、ユーザなど、あらゆるエミュレーションやシミュレーション

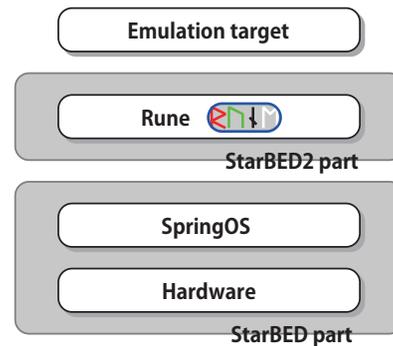


図 -10 StarBED2 の構成

ンの対象を Space という概念で扱う。また、これらの Space 間のやりとりは Conduit と呼ばれる仮想の通信路を通して行う。すべての対象を Space という概念で抽象化し、Space 間の通信を Conduit という概念で抽象化することにより、Rune は Space の実装の詳細を知ることなく統一的に扱うことが可能となり、模倣される対象の実装者に対しても、分散環境で実行されることへの特別の配慮を不要としている。

各 Space は StarBED を構成する x86 アーキテクチャの PC 上で実行される。しかし、実際にユビキタスネットワークシステムのノードで実行されるコードは必ずしも x86 アーキテクチャ用のバイナリコードではない。したがって、各ノードアーキテクチャ向けのネイティブバイナリを実行可能とする機構が必要となる。Rune ではこうした機能を提供するため、マルチレベルエミュレーションレイヤを提供している。マルチレベルエミュレーションレイヤは、インストラクションレベル、システムコールレベル、ミドルウェアレベルの 3 層から構成されている。インストラクションレベルエミュレーションレイヤはプロセッサエミュレータとなっており、インストラクションを命令ごとに解釈し実行する機能を提供している。これにより、ユビキタスネットワークのノードで実際に実行されるバイナリコードをそのまま利用したエミュレーションを可能としている。インストラクションレベルエミュレーションレイヤを利用したエミュレーションでは、実行コストは非常に高くなるものの、実際の機器の挙動に対して非常に忠実なエミュレーションが可能となっている。それほど精密なエミュレーションが必要ない場合や、実行コストを低減することによってスケラビリティを得たい場合にはインストラクションレベルエミュレーションレイヤやミドルウェアエミュレーションレイヤを利用するライブラリをリンクし、リコンパイルを行ったネイティブバイナリを利用したエミュレーションを行うことも可能となっている。

これらの機構を利用したシミュレーションは、図 -11 に示すように、各ノードに配置された Rune Manager と全体

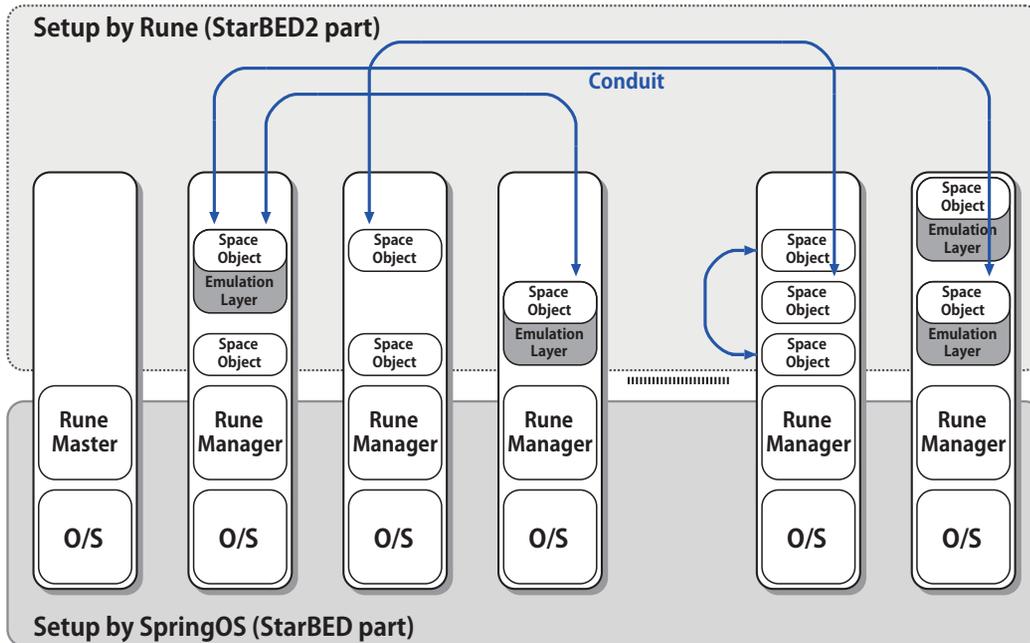


図-11 実験の全体構成

の制御を行う Rune Master の連携によって協調が行われる。

StarBED を用いて行われた実験例

これまで StarBED ではさまざまな実験が行われてきた。本章ではこのような実験のうちいくつかの例を紹介する。

● Adhoc multicast 網制御技術

本実験では、中規模の ISP の網をエミュレートし、Multicast 技術を導入した際の影響についての検証を行った¹⁹⁾。ISP バックボーンネットワーク部以外に VMWare を用いることで 1,044 台の大規模な実験駆動単位を構築し、JGN を用いて奈良先端科学技術大学院大学でトラフィックを往復させることでコアネットワークに実環境のリンク特性を導入した。図-12 にこの実験で利用したトポロジを示す。

● プロセス多重による環境の大規模化の予備実験

VM Nebula や PlanetLab では仮想機械を利用することで、実験用ノードを多重化し、より大規模な実験駆動単位を構築できる。仮想機械を利用することで、既存のアプリケーションソフトウェアへの変更がほぼ必要ないという利点があるが、ハードウェア部分をエミュレートするための計算機ノードの負荷は軽視できない。プロセスレベルでノードを仮想化し、1 台の実ノードで多数のプロセスを起動することでより多重度を上げ、大規模な実験駆動単位を構築できる可能性がある。しかしプロセス多重による仮想化にはさまざまな問題点も生じるため、

問題の把握と解決のための予備実験が行われた^{20), 21)}。

本実験では、StarBED 設置時 (2002 年) に導入された計算機ノード 1 台あたり、マルチキャストパケットを送信するプロセスを 5,000 個起動することに成功しており、そのためのさまざまな要求事項が明らかになった。

● 複数の大規模実証環境の統合実験

StarBED は汎用的な実験向けの大規模実証環境であるが、VM Nebula や SIOS²²⁾ といったある分野の実験に特化して設計されているものも存在する。VM Nebula はすでに述べた通り、攻撃対象ノードを実現するための環境である。これに対し SIOS は攻撃トラフィックに関するデータベースを持っており、このデータベースに従って攻撃トラフィックを再生できる。SIOS にも実ノードによる攻撃対象ノードが用意されているが、攻撃対象ノードを実現するために VM Nebula を利用すれば、さまざまな点で効率が良い。さらに、実環境を考えた場合、攻撃元と攻撃対象のノード間には複雑な網が存在し、この網が攻撃トラフィックにさまざまな影響をおよぼすと考えられる。そこで、StarBED とこの 2 つの環境を協調利用し、SIOS からの攻撃トラフィックを StarBED 上に構築したトポロジを通して、VM Nebula に用意した被攻撃ノードに対して攻撃を行い、実環境により近い実験駆動単位を作り出した²³⁾。図-13 にこの実験のトポロジを示す。

● Web ロードバランサ装置の性能測定

150 台の高性能ノードを StarBED の利用者用ラックに設置し、HTTP サーバを動作させ、グループ A の 208

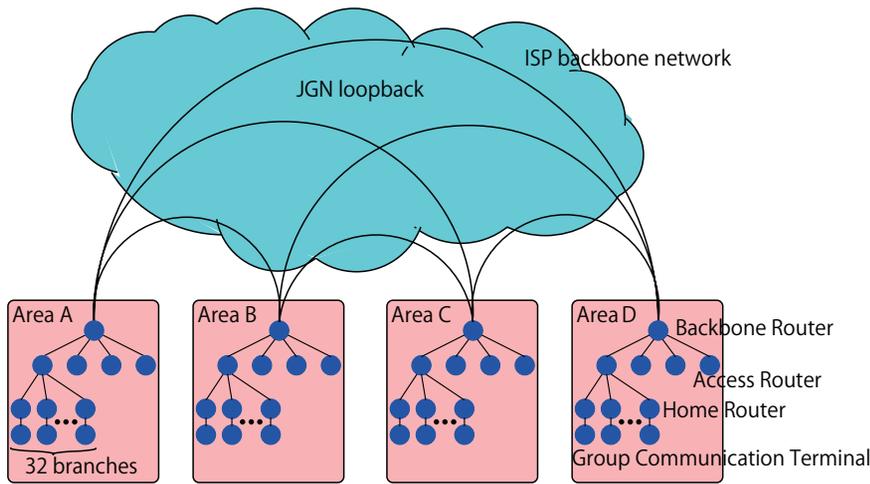


図-12 Adhoc multicast 網制御技術用実験トポロジ

● AS 間接続の再現

実験によっては、実験駆動単位に大規模かつ複雑な L3 トポロジを導入する必要がある。このようなトポロジを構築するために、Xen を用いて実ノードを多重化し、それぞれで Quagga を動作させることで大規模な AS 間接続を再現した²⁴⁾。本実験では CAIDA が提供する AS 接続に関するデータベース²⁵⁾を用い、2007 年 4 月 30 日時点の隣接 AS 数が多い上位 5,000AS を 100 台のグループ F の実ノードを用いて模倣することに成功している。

● 宅内における無線 LAN 通信の干渉の計測

StarBED2 プロジェクトの成果物を利用した実験のうち、代表的なものとして、図-14 に示すように、無線 LAN を利用する温度監視システムと利用者の通信の干渉をシミュレートする実験²⁶⁾が行われた。この実験では、利用者の PC、エアコン、熱センサ、無線 LAN アクセスポイント、4 つの居室における温度場をそれぞれ Rune の Space として実装を行った。また、無線 LAN による通信のエミュレーションには QOMET を利用した。この構成において、利用者の PC から発生する無線 LAN 上のトラフィックがどれだけ温度監視に影響を与えるかの確認を行った。

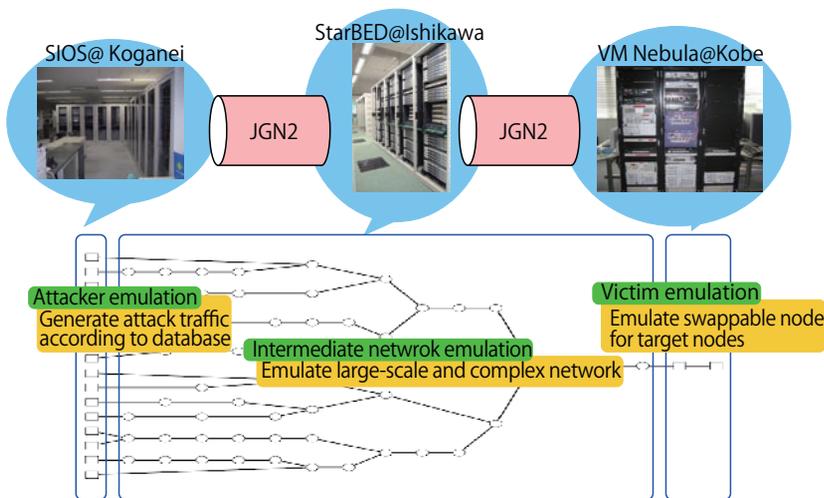


図-13 大規模実証環境の協調実験

台をクライアントとして利用し、その間に同様に利用者用ラックに設置したロードバランサ装置を設置した場合と設置しない場合での性能比較を行った。クライアントノードでは複数のクライアントプログラムを起動することでセッション数を増やし、対応できるセッション数や、1 秒ごとに対応できるリクエスト数などについての計測を行った。

本実験では Web ロードバランサ装置の検証であったが、このようなベンチマーク試験はさまざまなアプリケーションソフトウェア、サービスで行われ、サーバノードおよびクライアントノードで動作するアプリケーションソフトウェアを変更することにより、さまざまなケースに対応できる。また、本実験はすべてのノードをディスクレスシステムとして起動し、起動コストの低減を図った。

● モーションプランニングロボットのシミュレーション

Rune や QOMET を利用した実験のうち、もう 1 件代表的なものとして、モーションプランニングロボットのアルゴリズムの検証(図-15)がある。この実験では、マップマネージャとロボットを Rune の Space として実装した。マップマネージャは、ロボット、障害物など、実験に含まれる全オブジェクトの地理的情報の管理を行う。ロボットは、ビジュアルセンサと無線 LAN インタフェースを持ち、ビジュアルセンサは障害物の、無線 LAN インタフェースはロボット同士の衝突の回避に用いられる。この実験では、少数のロボット間では正確に動作するとされているモーションプランニングアルゴリズムが

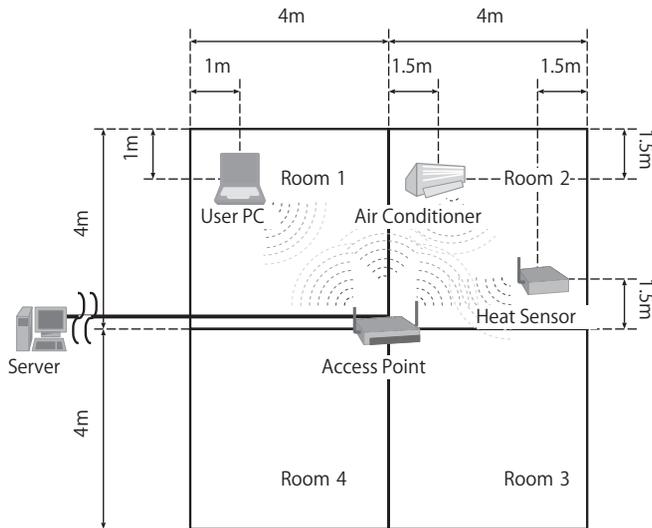


図-14 無線センシングシステムのシミュレーション

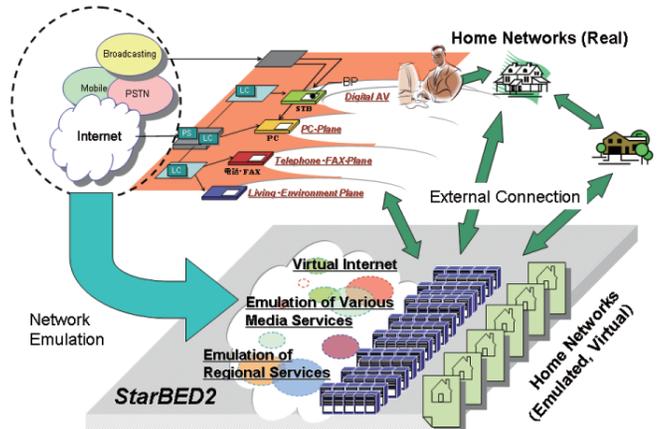


図-16 タウンネットワークシミュレーション

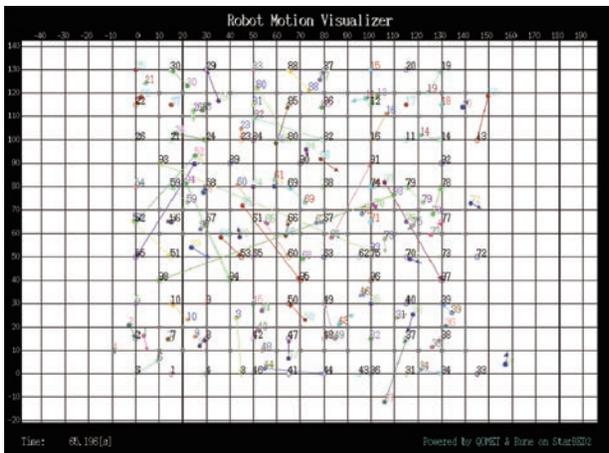


図-15 モーションプランニングロボットのシミュレーション

実時間性を保ったまま、多数のロボットが活動する環境において正常に動作するかの確認を行った。本実験の結果、110 台余りのロボットを用いた場合²⁷⁾の正常動作が確認された。

まとめと今後の予定

StarBED は、これまで困難であった多数の実ノードを利用した一拠点集中型の実験駆動単位の構築を可能とし、さまざまな実験が行われ、多くの成果を上げてきている。実験例として紹介した、ロードバランサ装置の検証は単純であるが、多数の実ノードが存在する環境に対象装置を接続して実験を行う必要があり、StarBED 以外での実現は困難である。また、インターネットにより近い現実的な環境を構築するために、他の大規模実証環境との協調や AS 間接続の再現実験、プロセス多重の実験も行われ、実際の実験駆動単位に組み込まれてきている。

このような実験により、実環境上での対象実装の現実的な挙動を観測することが可能になってきており、実環境への導入の前に実装の検証や問題点の修正を行うことができる。

StarBED2 プロジェクトは現在進行中であるが、すでに数々のシミュレーションが行われている。今後、StarBED2 プロジェクトでは、プロジェクトのグランドチャレンジであるタウンネットワークシミュレーション(図-16)の実現に向けて開発を進めていく。これは、StarBED2 上で1つの街規模のネットワークをシミュレートし、この環境内においてさまざまなサービスの検証を行おうとするものである。さらに、この実験を行う際に、閉じたシミュレーションとして検証を行うのではなく、一部に実環境を組み込んだ形でのシミュレーションを行うことも目標とする。この形態でのシミュレーションを可能とするため、StarBED2 プロジェクトでは、さらなるスケーラビリティの確保、実環境に対するインタフェースの実装、厳密な実時間性の確保などの課題を克服するための開発を続けていく。

StarBED2 以前には StarBED の多数の実ノードを制御することに着目した研究開発を進めてきた。この活動は StarBED2 においても非常に重要な部分を占め、特にユビキタス環境を実現するためにさらなる拡張が必要である部分も少なくない。このためユビキタス環境のエミュレーション部分だけでなく、これまで行ってきた実験設備の制御についての研究も引き続き進める。

実験駆動単位の現実性や実験結果の保証なども大きな問題の1つである。実環境のリンクをそのまま使う、もしくは、計測された特性の導入や、実環境のトポロジの導入が提案されており、これは実験駆動単位の現実性向上のための1つの手法である。しかし、インターネット

の特性は対象とするサービスや、場所によって異なり、単純に実環境のリンク特性を導入しただけでは、目的の環境は構築できないと考えられる。したがって、個々の実験にとっての現実的なトポロジや実験内容、観測内容などを統合的に議論することが必要である。

さらに、セキュリティ実験など特殊な実験の実行を考えた場合には、他の実験実行者との資源分割などの点が不十分であるため、随時設備の構成変更および支援ソフトウェアでの対象領域の拡大を進める。

実験例で挙げたように StarBED を単一で利用するよりも、他の実証環境と協調して利用することにより、機能面、規模面での強化が期待できる実験も数多い。このような実験を行うには、さまざまな実証環境が協調できるフレームワークが必要であり、これについての議論も進めている²⁸⁾。また、他の実験実行環境と SpringOS の設定記述を相互変換するため、実験実行に必要な機能および設定ファイルへの記述方法についての議論も行っている²⁹⁾。これにより、ns-2 などの設定記述を SpringOS の設定記述に変換することが可能となり、他の実験実行環境で行った実験を StarBED で、StarBED で行われた実験を他の実験実行環境で実現できると考えている。

本来、実験は新たな技術を開発するための補助的な役割を果たすものであり、実験を行うこと自体は目的ではない。むしろ、開発者や研究者が構築した実装もしくは理論を入力すれば、必要な結果だけ得られることが理想である。また、統一した実験駆動単位および実験内容で評価を行うことにより、実験対象技術の比較の易化や、統一された基準による認定を行うことができる。これには実験の実行部分の支援だけでなく、実験計画の自動生成や実験結果の自動解析などが必要となる。現在は実験実行の支援に主眼を置いているが、実験計画の支援の一環として実験トポロジや、さまざまな実験実行環境を持つ性質の整理を進めている。

謝辞 本研究プロジェクト推進に多大なご貢献をいただいた元総務省北陸総合通信局局長寺崎明氏、松井房樹氏ならびに北陸総合通信局各位に深謝する。なお、本プロジェクトは、北陸先端科学技術大学院大学インターネット研究センターと情報通信研究機構の共同研究であり、多くのユーザと研究者にご協力をいただいている。この場を借りて謹んで謝意を表す。

参考文献

- 1) The VINT Project : The ns Manual, <http://www.isi.edu/nsnam/ns/ns-documentation.html>
- 2) Network, S. R. : Scalable Simulation Framework, <http://www.ssfnet.org/>
- 3) OPNET Technologies, Inc. : OPNET User Forums, <http://forums.opnet.com/>
- 4) White, B., Lepreau, J., Stoller, L., Ricci, R., Guruprasad, S., Newbold,

- M., Hibler, M., Barb, C. and Joglekar, A. : An Integrated Experimental Environment for Distributed Systems and Networks, *OSDI02*, Boston, MA, USENIXASSOC, pp.255-270 (2002).
- 5) Peterson, L., Anderson, T., Culler, D. and Roscoe, T. : A Blueprint for Introducing Disruptive Technology into the Internet, *HotNets-I '02* (2002).
- 6) Miwa, S. and Ohno, H. : A Development of Experimental Environments "SIOS" and "VMNebula" for Reproducing Internet Security Incidents, *Journal of the National Institute of Information and Communications Technology*, Vol.52 numbers 1/2, pp.23-34 (2005).
- 7) The VINT Project : Network Emulation with the NS Simulator, <http://www.isi.edu/nsnam/ns/ns-emulation.html>
- 8) Ricci, R., Duerig, J., Sanaga, P., Gebhardt, D., Hibler, M., Atkinson, K., Zhang, J., Kasper, S. and Lepreau, J. : The Flexlab Approach to Realistic Evaluation of Networked Systems, *Proceedings of the Fourth Symposium on Networked Systems Design and Implementation (NSDI 2007)*, Cambridge, MA, pp.201-214 (2007).
- 9) Eide, E., Stoller, L. and Lepreau, J. : An Experimentation Workbench for Replayable Networking Research, *Proceedings of the Fourth Symposium on Networked Systems Design and Implementation (NSDI2007)*, pp.215-228 (2007).
- 10) Members of the DETER and EMIST Projects : CYBER DEFENCE TECHNOLOGY NETWORK AND EVALUATION - Creating An Experiment Infrastructure for Developing Nextgeneration Information Security Technologies, *COMMUNICATIONS OF THE ACM*, Vol.47, No.3, pp.58-61 (2004).
- 11) Suzuki, M., Hazeyama, H. and Kadobayashi, Y. : Expediting Experiments Across Testbeds with AnyBed : A Testbed-independent Topology Configuration Tool, *Proceedings of 2nd International IEEE/Create-Net Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom)* (2006).
- 12) Chinen, K., Miyachi, T. and Shinoda, Y. : A Rendezvous in Network Experiment - Case Study of Kuroyuri, International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (Tridentcom) (2006).
- 13) Levis, P., Lee, N., Welsh, M. and Culler, D. : TOSSIM : Accurate and Scalable Simulation of Entire TinyOS Applications, *Proceedings of the First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003)* (2003).
- 14) Polley, J., Blazakis, D., Mcgee, J., Rusk, D. and Baras, J. S. : ATEMU : A Fine-grained Sensor Network Simulator, *Proceedings of the First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON 2004)*, pp.145-152 (2004).
- 15) Titzer, B. L., Lee, D. K. and Palsberg, J. : Avrora : Scalable Sensor Network Simulation with Precise Timing, *Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks (IPSN 2005)*, pp.477-482 (2005).
- 16) Mallanda, C., Suri, A., Kunchakarra, V., Iyengar, S. S., Kannan, R., Furesi, A. and Sastry, S. : Simulating Wireless Sensor Networks with OMNeT++, *Proceedings of the Sixth International Symposium on Autonomous Decentralized Systems (ISADS 2003)* (2003).
- 17) Carley, T. W. : Sidh : A Wireless Sensor Network Simulator, No. Institute for Systems Research Technical Reports 2005-88 (2005).
- 18) Sundresh, S., Kim, W. and Agha, G. : SENS : A Sensor, Environment and Network Simulator, *Proceedings of the Thirty-seventh Annual Simulation Symposium (ANSS2004)*, pp.221-228 (2004).
- 19) Muramoto, E., Yoneda, T., Nakamura, A., Misumi, M., Miyachi, T. and Shinoda, Y. : Report on a Method of Simulating Multicast Group Communication on the Internet, Towards Peta-bit Ultra Networks (2003).
- 20) 三角 真, 中川晋一, 我如古津世史, 知念賢一, 篠田陽一 : メガノード IP マルチキャストセンサネットワークに関する実験的検討, 信学技報 CQ2006016, pp.75-81 (2006).
- 21) 三角 真, 中川晋一, 我如古津世史, 知念賢一, 篠田陽一 : メガノードマルチキャストシミュレーションに向けての送信実験報告, 信学技報 MoMuC2006-16, pp.85-90 (2006).
- 22) 大野浩之, 武智 洋, 永島秀己 : インターネットの脅威に対抗する脆弱性データベースと検証システムの構築, 分散システム/インターネット運用技術(DSM)シンポジウム, 情報処理学会(2001).
- 23) 三輪信介, 宮地利幸, 大野浩之 : 不正アクセス等再現実験環境の統

StarBED : 大規模ネットワーク実証環境

- 合実験, マルチメディア, 分散, 協調とモバイル(DICOMO2005)シンポジウム論文集, 情報処理学会, pp.393-396 (2005).
- 24) 三輪信介, 宮地利幸, 大野浩之: 不正アクセス等再現実験環境の統合実験, インターネットコンファレンス 2007 論文集, 日本ソフトウェア科学会, pp.41-48 (2007).
- 25) CAIDA : AS Relationships, <http://www.caida.org/data/active/as-relationships/>
- 26) Nakata, J., Beuran, R., Miyachi, T., Chinen, K., Uda, S., Masui, K., Tan, Y. and Shinoda, Y.: StarBED2 : Testbed for Networked Sensing Systems, *Proceedings of the 4th International Conference on Networked Sensing Systems (INSS07)*, pp.142-145 (2007).
- 27) Okada, T., Beuran, R., Nakata, J., Tan, Y. and Shinoda, Y.: Collaborative Motion Planning of Autonomous Robots, 3rd International Conference on Collaborative Computing (CollaborateCom 2007) (2007).
- 28) 宮地利幸, 三輪信介, 知念賢一, 篠田陽一: ネットワーク実験支援ソフトウェアの汎用アーキテクチャの提案, 情報処理学会論文誌, Vol.48, No.4, pp.1695-1709 (Apr. 2007).
- 29) 宮地利幸, 知念賢一, 篠田陽一: ネットワーク支援システムに必要な機能と設定記述, マルチメディア, 分散, 協調とモバイル(DICOMO2006)シンポジウム論文集, 情報処理学会, pp.769-772 (2006).

(平成 19 年 12 月 5 日受付)

宮地利幸(正会員)
miyachi@nict.go.jp

2007 年北陸先端科学技術大学院大学博士後期課程修了。同年情報通信研究機構北陸リサーチセンター研究員。ネットワーク実験に関する研究に従事。博士(情報科学)。電子情報通信学会, 日本ソフトウェア科学会各会員。

中田潤也(学生会員)
jnakata@nict.go.jp

2000 年北陸先端科学技術大学院大学情報科学研究科博士前期課程修了。2006 年情報通信研究機構北陸リサーチセンター研究員。2007 年 12 月現在, 北陸先端科学技術大学院大学情報科学研究科博士後期課程在学中。

知念賢一(正会員)
k-chinen@jaist.ac.jp

1998 年奈良先端科学技術大学院大学情報科学研究科博士後期課程修了。1998 年同研究科助手。2003 年北陸先端科学技術大学院大学情報科学研究科助手およびインターネット研究センター研究員。2006 年情報通信研究機構北陸リサーチセンター短期研究員。各種ネットワークサーバやネットワーク実験環境に関する研究開発に従事。博士(工学)。IEEE, 電子情報通信学会, 日本ソフトウェア科学会各会員。

Razvan Beuran
razvan@nict.go.jp

2004 Ph.D. from University "Politehnica" Bucharest (Romania) and University "Jean Monnet" Saint Etienne (France). 2006 researcher, NICT, Hokuriku Research Center. Research interests : application performance in wired and wireless networks. Doctor (engineering).

三輪信介
danna@nict.go.jp

1999 年北陸先端科学技術大学院大学博士後期課程修了。同年同大情報科学センター助手。2001 年通信総合研究所(現情報通信研究機構)研究員。インターネットのセキュリティおよびその再現実験環境の構築方

法の研究開発に従事。博士(情報科学)。

岡田 崇(学生会員)
tk-okada@jaist.ac.jp

2007 年北陸先端科学技術大学院大学情報科学研究科博士前期課程修了。同年情報通信研究機構北陸リサーチセンター特別研究員。2007 年 12 月現在, 同大同研究科博士後期課程在学中。

三角 真(正会員)
misumi@nict.go.jp

2004 年北陸先端科学技術大学院大学情報科学研究科博士前期課程修了。同年情報通信研究機構特別研究員。2006 年同機構技術員。インターネット応用技術に関する研究に従事。修士(情報科学)。

宇多 仁(正会員)
zin@jaist.ac.jp

2004 年北陸先端科学技術大学院大学博士後期課程修了。博士(情報科学)。同年同大情報科学センター助手。2006 年同助教。情報通信研究機構北陸リサーチセンター特別研究員。ネットワークアーキテクチャならびにインターネット運用技術に関する研究に従事。

芳炭 将
yoshizumi@jaist.ac.jp

2002 年北陸先端科学技術大学院大学情報科学研究科博士前期課程修了。同年(株)NTTドコモネットワーク研究所研究員。2007 年情報通信研究機構北陸リサーチセンター特別研究員。同年 12 月現在, 同大同研究科博士後期課程在学中。

丹 康雄(正会員)
ytan@jaist.ac.jp

1993 年東京工業大学理工学研究科博士後期課程修了。博士(工学)。1993 年北陸先端科学技術大学院大学情報科学研究科助手。同情報科学センター助教。情報科学研究科助教を経て 2007 年情報科学研究科教授。国立情報学研究所連携教授。情報通信研究機構研究マネージャ併任。総務省情報通信審議会専門委員, 同 ITU-T 部会 HN 合同 WG 主任, 次世代 IP ネットワーク推進フォーラム HN-WG 主査, エコネットコンソーシアムアドバイザーフェロー, TTC 次世代 HN システム WG 委員等を務め, ホームネットワークの技術開発, 標準化に取り組んでいる。

中川晋一(正会員)
snakagaw@nict.go.jp

1988 年滋賀医科大学医学部卒業, 1996 年京都大学大学院医学研究科修了。同年国立がんセンター, 1998 年郵政省通信総合研究所主任研究員。2004 年現職。国際(APII)・国内(JGN)テストベッド構築・実験, インターネット技術の応用研究に従事。医師, 博士(医学), 日本内科学会会員。

篠田陽一
shinoda@jaist.ac.jp

1988 年東京工業大学理工学研究科博士後期課程修了。同年同大工学部助手。1991 年北陸先端科学技術大学院大学情報科学研究科助教。2001 年同大情報科学センター教授および同大インターネット研究センタープロジェクトリーダー。2006 年情報通信研究機構北陸リサーチセンタープロジェクトリーダー。同年同機構情報通信セキュリティ研究センター長。2007 年内閣官房情報セキュリティセンター情報セキュリティ補佐官。情報環境, ネットワーク分散情報システム, ソフトウェア開発環境の研究に従事。工学博士。