

第 XXIII 部

実ノードを用いた大規模な インターネットシミュレーション 環境の構築

第 23 部

実ノードを用いた大規模なインターネット シミュレーション環境の構築

第 1 章 はじめに

Deep Space One ワーキンググループは実環境向けのハードウェアおよびソフトウェアを利用した大規模な実験用環境の構築・運用に関する研究に取り組んでいる。実ノードを用いた大規模な実験設備として StarBED や GARIT、仮想機械を利用した VM Nebula などの開発と運用を通して、実験設備のあり方や、実験設備の制御方法、さらに、実験の体系的分類などの議論を進めている。また、これに加え、実験を柔軟かつ容易に行うため、実験用環境への実トポロジの再現手法や、標準的な実験のテンプレート化などの研究を行っている。

2008 年度から、Deep Space One ワーキンググループで扱っている大規模実験環境の構築とは別に、大規模実験環境のユーザ視点から利用方法やノウハウの共有、実験例、新たな利用例の考案を行うため、Nerdbox Freaks ワーキンググループが創設された。Nerdbox Freaks ワーキンググループでは、大規模実験環境での実験を、より現実性のある実験評価環境にするべく大規模実験環境における模倣インターネットの構築に取り組んでいるほか、StarBED の利用方法や StarBED と類似した小規模な実験環境の構築方法を演習形式で行うワークショップを開催している。

本報告では Deep Space One ワーキンググループ及び Nerdbox Freaks ワーキンググループの本年度の主な活動について報告する。本報告の構成は以下の通りである。

- DeepSpaceOne ワーキンググループ
 - StarPOD
 - XBurner
- Nerdbox Freaks ワーキンググループ
 - 模倣インターネット環境構築への取り組み
 - * eBGP ネットワーク模倣環境構築

- * OSPF ネットワーク模倣環境構築
- イベント
 - * Interop Tokyo 2009 Cloud Computing Competition
 - * ACM SIGCOMM 2009 Demo Session
 - * インターネットコンファレンス 2009 デモ
 - * 第 1 回 Cloud Computing Developers Workshop

● 各種ツールのリリース状況
(StarBED の詳細については [130] 参照)

第 2 章 StarPOD

StarPOD (Portable Operation Domain) は、可搬型の小規模 StarBED 環境で、12 台の PC ノードとそれらを操作するための KVM スイッチ、ノード同士を接続するためのネットワークスイッチを、2つの可搬型ラックに格納している。図 2.1 は StarPOD の概観である。StarPOD は外部でのデモや小規模実験を目的として設計されており、ネットワーク、KVM、電源の各ケーブルの接続を行うだけで小規模テストベッドの設置が完了する。StarPOD の各機器の仕様は表 2.1 に示す。ラックは音響機器用によく用いられる振動吸収型のラックを使用している。



図 2.1. 可搬型小規模 StarBED (StarPOD)

●第23部 実ノードを用いた大規模なインターネットシミュレーション環境の構築

表 2.1. StarPOD の仕様

Part	Specification	Number
Nodes	DELL R200 Pentium(R) E2200 2.20 GHz 4 GByte Memory 160 GB 7,200 RPM SATA HDD	12
Switch	D-Link DGS-3450	1
Case	SKB R910U28	2

第3章 XBurner: XENebula を利用したトラフィックジェネレータプラットフォーム

ネットワーク実験を行うための実験用の環境には、対象技術によるトラフィックと周辺技術によるトラフィックが存在する。このようなトラフィックは対象技術を動作させることにより自然に発生するものと、より現実的な実験環境を構築するためや、対象技術のパフォーマンスや手順の試験を行うために意図的に生成されるものがある。意図的にトラフィックを生成するためには、現状では専用に設計されたトラフィックジェネレータを利用することが多い。ネットワーク実験は、その目的によりさまざまな環境で行われるが、物理的に構築された環境での検証のためのトラフィックジェネレータ [5, 25, 89, 90] だけでなく、ソフトウェアシミュレーション環境でもさまざまなトラフィックを生成するモジュールが利用されている。

既存のトラフィックジェネレータは、前もって定義されたトラフィック、もしくは観測されたトラフィックを再生するものが一般的である。そのため、新しいプロトコルや新種のネットワーク攻撃など、トラフィックジェネレータの定義ないトラフィックを生成することは困難であり、例えばゼロデイ攻撃を防ぐような新しいセキュリティ技術の検証など、検証対象の技術によっては既存のトラフィックジェネレータを利用できない、もしくは検証ツールとして適さない場合もある。

本章では、特殊なハードウェアやソフトウェアを利用せず、一般的な計算機と OS 上で動作するツールを利用してトラフィックを生成することで、利用者が実装したサービスの検証を含め柔軟なトラフィックを生成するためのプラットフォーム XBurner につ

いて述べる。

また、XBurner を利用しトラフィックに多様性および規模性を持たせるため、大規模実証環境の一つである StarBED [68, 130] 上に XENebula [67, 101, 133] および AnyBed [92] を利用し、AS 間トポロジを構築し、数千ノードからなるトラフィック生成専用環境の構築を目指す。

本章では、実験用の実計算機やソフトウェアシミュレータ上に模倣された実験用の計算機やネットワーク機器をノード、ネットワーク実験のために構築された実験用のノードや実験を管理するためのノードを含めた環境を実験駆動単位と呼ぶ。用語の定義については文献 [131] が詳しい。

3.1 XBurner の問題意識

本節では、現状のトラフィックジェネレータの問題点および、我々が開発を行う XBurner での問題の解決手法について述べる。

3.1.1 既存のトラフィックジェネレータの問題点

文献 [5, 25, 89, 90] のような物理的に構築された環境での検証のためのトラフィックジェネレータの利用ケースでは、以下のような問題が存在する。

1. 任意のトラフィックパターンが生成できない
2. 新たなサービスに対応できない
3. インタラクティブな通信ができない

新しいサービスの検証や、既存のサービスの転用などのための検証には、任意のパターンでのトラフィック生成が必要になる。一方、既存のトラフィックジェネレータでは、ある特定のプロトコルのある特定のパターンのトラフィックの生成は可能であるが、任意のパターンのトラフィックを生成することは困難であることが多い。また、同様の理由から新しいサービスの検証のために利用者が定義したプロトコルや手順にしたがったトラフィックの生成も困難である。

記録されたトラフィックを単純に再生するようなトラフィックジェネレータでは、任意の観測点で観測されたトラフィックの送出は行えるが、その応答を得ることは困難である。単にトラフィックを再生する場合には、送信されたパケットはあるポイントで破棄されればよいが、インタラクションを含めた通信を行うためには、パケットを受信したノードがそれに対して適切に対応する必要がある。このためには送信側と受信側の処理を行う要素が適切に実験

表 3.1. 既存のトラフィックジェネレータの問題点

用途	問題点
対象技術 トラフィック	任意のトラフィックパターンが生成できない 新たなサービスに対応できない
周辺技術 トラフィック	任意のトラフィックパターンが生成できない 新たなサービスに対応できない インタラクティブな通信ができない

駆動単位に用意されることが求められる。

表 3.1 に、トラフィックの発生目的とそれに対する既存のトラフィックジェネレータの問題点をまとめる。

XBurner はこれらの問題に柔軟に対応し、より現実的なトラフィックを実験駆動単位に導入し、より現実的なネットワーク実験を実現することを目的とする。

3.1.2 XBurner でのアプローチ

前項で説明した内容を解決するために、XBurner は以下の方針にそって開発を行う。

利用者独自のソフトウェア導入を許容 新たなサービスを実装した際には、その検証のための多様性をもったトラフィックを生成することが必要である [123]。XBurner では、利用者が実装したソフトウェアによるトラフィックの生成を許容するように設計を行う。これにより、トラフィックパターンについても、利用者の希望を反映したものを導入できる。

パケットダンプによるトラフィック再生機能の提供 利用者が任意の観測データを再生する場合には、tcpdump や Wireshark[96] で観測したパケットダンプを利用し、トラフィックを再生することでバックグラウンドトラフィックを生成するケースも多い。このような用途に対応するため、パケットダンプとして保存されたトラフィックを容易かつ現実的に再生する機能を提供する。このような手法は現状でも利用されているが、XBurner を用いることで多数のノードから分散してトラフィックを生成することが可能となる。

現実的なリンク特性の導入 インターネット上を流れるトラフィックは、送信元から宛先までさまざまな特性を持ったリンクを経由することにより、さまざまな通信特性を持つことになる。XBurner で生成するトラフィックも同様の特性

を持つようにリンク特性を導入する。

トラフィック量の調整機能 実験によって、必要とされるトラフィックの総量やプロトコル毎の割合は異なる。利用者の要望によりトラフィックの量の変更を容易に行えるインターフェースを提供する。トラフィック送途中にもその割合を変更できることを目標とする。

テンプレートの提供 ここまで、利用者の任意のトラフィックの導入について述べたが、典型的なプロトコルやトラフィックパターンを生成できるよう、XBurner でテンプレートを提供する。

3.2 XBurner の設計

3.1 節では XBurner が対象とする問題をまとめた。本節では、これらの要求を満たすために、XBurner で採用する技術や手法をまとめる。

3.2.1 XBurner への要求事項

前節で説明した XBurner のアプローチをまとめるために、以下の機能を実装する。

Foundation Builder インターネット向け実装を動作させるために、一般的な OS が導入された多数のノードからなる環境を構築する。

Burner Application Trigger 各ノードでのソフトウェアの起動制御を行い各種設定やトラフィックの生成を行う

Neighbor Interferer 現実的なリンク特性を導入するために、複雑なトポロジの導入やリンクエミュレータを利用することでリンク特性を変更する

Burner Factor Organizer XBurner が提供する各機能のユーザインターフェースを提供する

3.2.2 XBurner が採用する技術

本項では前項で述べた機能を実現するために採用する技術についてまとめる。

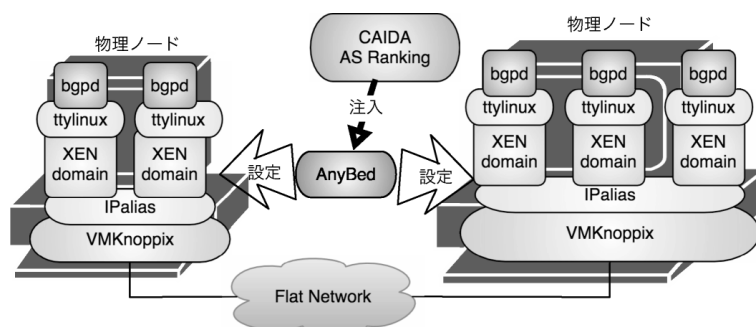


図 3.1. XENebula の L3 環境構築の概念図

Foundation Builder 単純なトラフィック生成は、実験用ノード 1 対 1 で対応が可能であるが、要求されるトラフィックが大規模化・多様化するにつれて、トラフィックを生成する環境自体も複雑化させる必要がある。このような環境構築のために、XBurner では PC ベースのクラスタと XENebula を利用した大規模な仮想化環境を利用する。

XENebula は XEN[99] を利用した大規模実証環境を構築するツールであり、XBurner は XENebula 上に現実的な AS 間トポロジ環境を構築し、さらに対象トラフィックを生成するクライアント、サーバも XENebula 上に用意する。これにより、ネットワークトラフィックを生成するソフトウェアを起動できる多数のノードと、より複雑な経路を環境中に導入できる。XENebula を利用して AS 間トポロジを作成する概念図を図 3.1 に示す。

Burner Application Trigger 各実験用ノードでの、トラフィック生成用のソフトウェアや、リンクエミュレータを外部から制御するために SpringOS を利用する。

Neighbor Interferer 現実的なリンク特性を導入するため、XENebula および AnyBed を利用した Inter-AS ネットワークの導入を行い、各パケット毎に別々の経路を通るよう設定を行う。これにより各通信毎に経路やホップ数が変化し、より複雑な特性を持たせられると考えられる。また、各リンクには必要に応じてリンクエミュレータにより [28, 73] 遅延やジッタ、帯域の制御を行う。

Burner Factor Organizer これらの機能へのユーザインターフェースとして、SpringOS のシナリオをテンプレートとして提供する。

XBurner では、利用者が SpringOS の言語記述に不慣れな場合でも利用できるよう、SpringOS はメッセージ交換によるノードのタイミング同期のみを担当する。すなわち、利用者は以下の役割を持ったそれぞれのプログラムを用意して、SpringOS ではそれらを起動するだけとする。

- 実験の前準備として必要なファイルのダウンロードやネットワーク設定
- 必要であれば利用するアプリケーションソフトウェアの設定ファイルの用意とアプリケーションソフトウェアの起動
- トラフィック発生
- トラフィック生成の停止およびアプリケーションソフトウェアの終了

以上のような手法により構築した環境に一般的なサービスを提供するサーバを用意することで利用者が導入したアプリケーションの補助を行う。たとえば、通信を開始するノードは一般的に DNS サービスを用いて、通信相手の IP アドレスの解決を行うため、DNS などのサービスを XENebula 上に構築し、その現実性を高めることも考慮する。特に DNS については、DNS 網を模倣しその応答を生成するといった試み [134] も行われており、このようなサービスを導入する手法も考えられる。

最終的には XENebula を利用して StarBED 上に数千から数万 AS およびノードからなる環境を構築し利用する。図 3.1 からわかるように XENebula は XEN を利用して一台の実ノード上で多数の仮想ノードを起動する。そして、XEN の DomU 上で動作する ttylinux で Quagga[83] の bgpd を起動し実際に経路の交換を行う。AS 間接続のデータは CAIDA の AS Relationship[17] で提供されているものを利用し、AnyBed[92] を用いて、各 bgpd の設定ファイルを生成している。

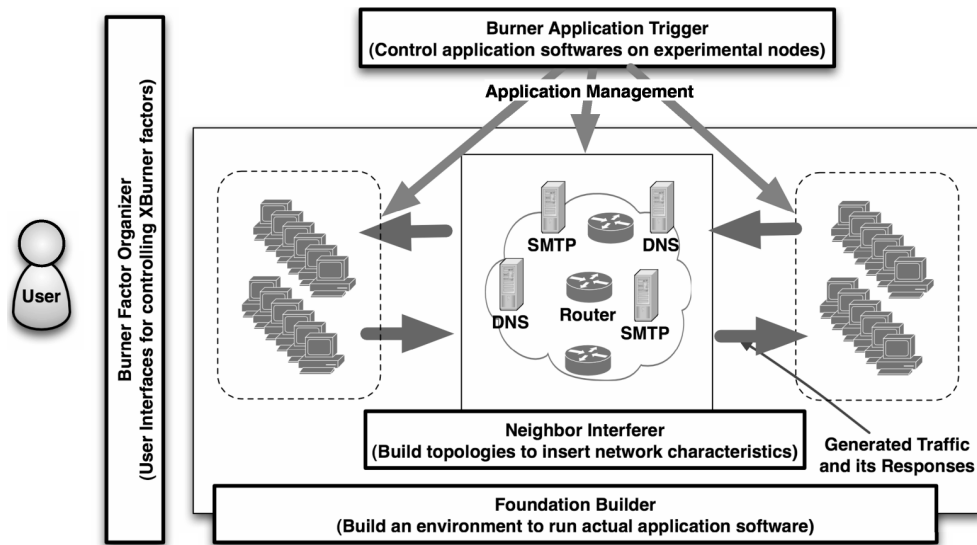


図 3.2. XBurner の概念図

ただし、SpringOS は、StarBED と同様の構成を持ったクラスタ環境での利用が可能であり、すでにいくつかの拠点で利用されている。したがって、XBurner は StarBED での利用に限定されるわけではなく、さまざまな PC クラスタ上での利用が可能である。

図 3.2 は XBurner の概念図である。これまでに説明したように、XBurner は StarBED もしくは同様の PC クラスタ環境もしくは XENebula による仮想ノード環境を利用しトラフィックを生成する。各 PC ノード上ではトラフィックを生成する一般的な通信用ソフトウェアや、ソフトウェアベースのトラフィックジェネレータが動作しており、その起動やパラメータの変更は SpringOS により制御される。また、同様にリンクエミュレータおよび DNS といった補助的なサービス用アプリケーションソフトウェアも XENebula 上で動作し、SpringOS によって制御される。

3.3 実装と評価

XBurner の設計の妥当性を確認するため簡単なトポロジを作った実験を行った。

実験のトポロジを図 3.3 に示す。

実機ベースのテストベッドとしては StarBED のデモンストレーション用環境である StarPOD を利用した。

実験では StarPOD の 8 台のノードを利用し、2つのネットワークを構築した。一台の物理ノード上で

25 台の仮想機械を動作させ、それぞれのネットワークを 100 台の仮想ノードで構成した。この 2 つのネットワーク間を一台の PC ルータで接続し、すべてのトラフィックがこのルータを必ず通過するように設定し、トラフィックの生成を行った。

現状の AnyBed および XENebula は一つの BGP 網もしくは OSPF 網を構築し、このときに利用される IP アドレスは自動的に決定される。XBurner の実装に当たってこのアドレスレンジを指定できるよう AnyBed の拡張を行った。また、SpringOS で各仮想ノードを制御するためには、仮想ノードの管理用 IP アドレスやマックアドレスからなるデータベースが必要となる。今回は、AnyBed が生成する仮想ノードの管理用 IP アドレスリストから各ノードに実際に ssh でログインし、自動的にデータベースを構築するスクリプトを作成した。さらに、ファイルをダウンロードするようなサービスを利用する場合には、ディスクの容量が問題となる。特に仮想ノードを利用した場合などについては利用できるディスク容量には限りがあるため、ダウンロードしたファイルを何らかの方法で定期的に削除する必要がある。このためには単純にファイルを削除するという方法と、そもそもディスクに書き込ませないという二通りのアプローチが存在する。できるだけ利用者の負担を軽減するため、Filesystem in Userspace (FUSE) [34] を利用し、ディスクにファイルが書き込まれないようなファイルシステムを実装した。

SpringOS のシナリオファイルは、必要なファイル

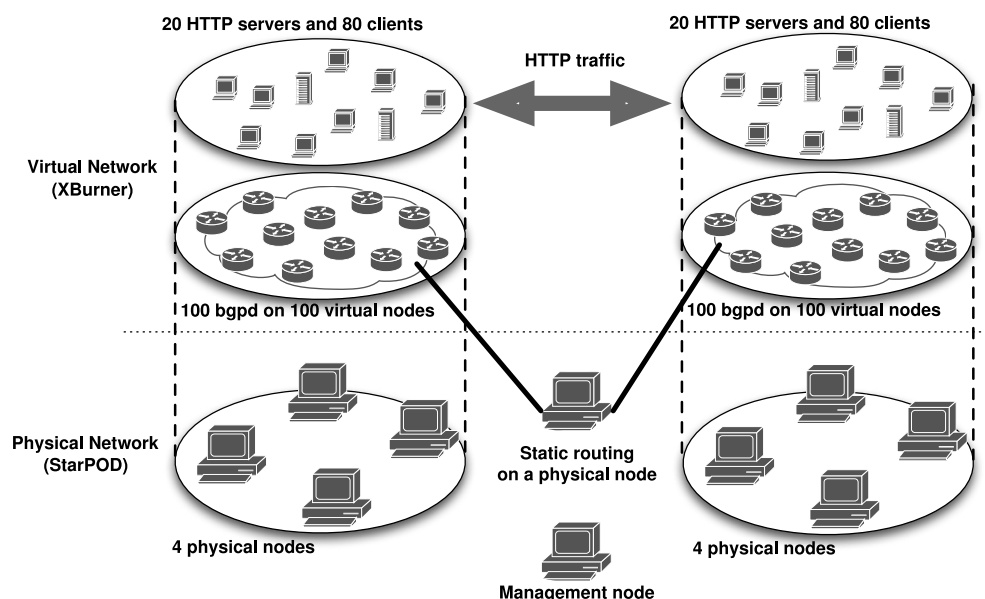


図 3.3. 評価実験用トポロジ

表 3.2. XBurner の特徴と改善点

特徴	改善点
一般的なソフトウェアの導入を許容	利用者のソフトウェアの利用が可能 利用者が定義したトラフィックパターンへの対応 ソフトウェアベースのリンクエミュレータの利用が可能 ソフトウェアベースの既存のトラフィックジェネレータの導入が可能
SpringOS によるソフトウェア制御	容易な環境整備 任意のタイミングでのコマンド実行による 柔軟なトラフィックパターンの生成
StarBED と XENebula による ノード数の確保	大規模な実験駆動単位の構築 複数の OS の導入を許容することによる現実的なトラフィックの導入 大規模な AS 間トポロジの挿入による現実的なトラフィックの導入

のダウンロード、各ノードの設定、アプリケーションの実行、アプリケーションの終了に分け、それぞれが決められた名前のシェルスクリプトを呼び出すように記述した。これにより、各ノードで行わせたい各段階の作業をシェルスクリプトで記述し、ファイルサーバに配置することで自動的にファイルがダウンロードされ実行される。

本実験では PC ルータで 180 M-200 Mbps のトラフィックが観測された。本実験では Web サーバ (thttpd) や取得されるファイルサイズなどを、特に調整しなかったためこの程度の値であったと考えられる。しかし、本実験で、XBurner がトラフィックを生成する基本機能を満たしている事が確認できたと言える。

3.4 XBurner の今後の課題

本章では、柔軟かつ現実的なトラフィックを生成するために XBurner の設計を行った。XBurner は多数の一般的な OS が起動するノードからなる環境において、それぞれのノードが利用者が新たに開発したモノを含め、一般的な通信ソフトウェアによりトラフィックを生成するため、より現実的なトラフィック生成が可能であるとともに、利用者による拡張も容易である。3.1.1 項で示した既存のトラフィックジェネレータの問題は、表 3.2 にまとめた XBurner の特徴により改善が可能である。

現在 XBurner は設計を行ったすべての機能を満たしているわけではない。各仮想ノード上でのアプリケーション制御が実現できているため、リンクエミュ

レータの起動も理論的には可能であるが、実際に動作確認は取っていない。また、今回は非常に簡単な実験であり、DNS や SMTP のサービス網の提供も行っていない。今後はこれらの機能の拡充を図る。

XBurner は一般的なアプリケーションソフトウェアを一括して実行し、その結果としてトラフィックを生成するためのプラットフォームである。仮想ノードを利用し一台の物理ノードのリソースを共有するため、一台で大量のトラフィックを生成するといった用途には不向きであるといえる。この点も踏まえ、XBurner を道具として利用して、現実的なトラフィックを生成するための議論は今後の課題となる。まずはトラフィックパターンの検討や現実的なリンク特性の導入についての議論を、実際に生成したトラフィックを観察した上で検討し、典型的な通信ソフトウェアを利用したトラフィック生成のためのテンプレートを用意する。同時に大規模なネットワークを経由することでトラフィックにどのような影響があるのかについても検討し、XBurner のトラフィック生成アルゴリズムに組み込んでいく予定である。

第 4 章 模倣インターネット環境構築への取り組み

新しいプロトコルスタックやアプリケーションソフトウェアをインターネット上に導入する場合、事前にインターネットに近い環境で機能やスケーラビリティを検証することが望ましい。そこで我々は、機能や規模が実際のインターネットに近い検証・実験環境を提供するための構築手法（模倣インターネット）の研究を行っている。Nerdbox Freaks では、StarBED での実験利用方法の一環として模倣インターネット環境および模倣インターネット環境を用いた実験環境として NERDBOX（Network Emulation for Realistic Deployment and Behavior Observation eXperiments）を開発している。

4.1 模倣構築ツールの研究

3.2 節の図 3.1 でも説明したように、模倣インターネット環境とは、実インターネットの観測データなどを元にして、実インターネットのトポロジやトラフィック、ノードの振る舞いなどを実験環境上に

模倣して作成し、実インターネット上での実証実験と同じくらいリアリティおよびスケーラビリティのある実験を再現可能かつ制御、観測が可能な状態にしてアプリケーションなどの動作検証が行えるようにしようという取り組みである。

現在、Nerdbox Freaks では AnyBed[92] と XENebula を用いて StarBED[68] 上に模倣インターネットを構築する NERDBOX のプロトタイプシステムの開発を行っている。

4.1.1 eBGP ネットワーク模倣環境構築

物理ノードを用いる場合も仮想ノードを用いる場合も基本的には図 3.1 に示すような形で模倣 eBGP 網は構築される。まず、AnyBed によって CAIDA AS Relationship Data（以降 CAIDA ASRD と略記）[17] から Quagga bgpd 用の設定ファイルを作成し、次に物理ノードもしくは XENebula で作成した ttylinux 上に配布し、Quagga bgpd を起動させて模倣 eBGP トポロジを作成する。

模倣 eBGP トポロジを構築する際、トポロジの大きさや AS 間の結合密度によっては AS 間リンクが 5000 リンク以上存在する可能性がある。そのため、VLAN による仮想インターフェースを用いた設定ファイル作成では StarBED[68] のような実験環境では VLAN 番号をすぐに枯渇してしまい、模倣 eBGP 網の構築が失敗する。そこで、NERDBOX では IP alias を仮想インターフェースとして利用し、Internet Exchange ポイントのように同一レイヤ 2 ネットワーク上で eBGP ピアリングを行う形で模倣 eBGP 網を構築している。

今年度は昨年度から実施している CAIDA の ASRD の上位 10,000 AS を XEN による仮想化を用いて構築する実験において、より充実した制御や観測が行えるようにするための実験やツール開発を行っている。

また、500 AS 程度の小規模な環境であれば、StarPOD（2 章参照）を用いて構築が可能である。本年度は 9 月 WIDE 合宿に置いて StarPOD を用いて作成した日本国内の AS トポロジ模倣環境（模倣 JP 環境）を合宿バックボーンネットワークに挟み込み、IPv4 ネットワークは模倣 JP 環境を通過させ、模倣環境の安定性や障害復旧などの実験を実施した。実験の詳細は WIDE 合宿ネットワーク実験に掲載されている「模倣インターネット環境の実インターネット環境への統合実験」を参照してほしい。

4.1.2 OSPF ネットワーク模倣環境構築

さらに、eBGP 網だけでなく、AS 内ネットワークの模倣環境構築として、WIDE バックボーン OSPF ネットワークの模倣環境構築を今年度は実施している。OSPF ネットワークの模倣環境構築においては、実運用されている OSPF ネットワークのトポロジを IP アドレスと OSPF コストを含めて模倣し、実験環境を構築する事を目的としている。その手法として、実運用されている OSPF ネットワーク中のルータから OSPF Link State Database (以下 OSPF LSDB) を抽出し、その OSPF LSDB から OSPF トポロジを推測した後、AnyBed のトポロジ構築機能を用いて GARIT や StarBED 上に OSPF ネットワークを模倣する。

本模倣環境構築の検証のため、50 個の OSPF ルータからなる WIDE ネットワークのバックボーンエリアトポロジを StarBED 上に構築した。その結果を論文にまとめ、インターネットコンファレンス 2009[114]にて報告を行った(詳細は、`wide-paper-nerdbox-freaks-mio-ic2009-00.txt` を参照してほしい)。

また、AnyBed を用いて作成した模倣環境を使った実験例をまとめ、研究発表として WIDE Project 以外の研究コミュニティに対しての宣伝活動も実施した。

4.2 イベント

今年度は、学会や展示会などで開催されたコンペティションに参加し、模倣環境構築のデモンストレーションを実施した。また、クラウドコンピューティングデベロッパーズワークショップと題し、開発者向けの合宿形式のハンズオンワークショップを開催し、開発合宿を通じての技術交流を実施した。

4.2.1 Interop Tokyo 2009 Cloud Computing Competition

インターネット関連技術の展示会である Interop Tokyo にて本年度から開催された Interop Tokyo Cloud Computing Competition (Interop CCC) というクラウドコンピューティングに関連する技術を StarBED 上に展開し、それを用いたデモンストレーションを競うコンペティションが開催された。この Interop CCC に“チーム模倣インターネット”として参加し、模倣環境のデモンストレーションを実施した。

参加申請した内容は、XENebula と AnyBed を用

いて日本国内および周辺 1 ホップの AS トポロジー (合計 648 AS) を StarBED グループ G のディスクレスサーバ 50 台を用いて構築し、eBGP 網が構築されていく様子の可視化などのデモンストレーションを実施した (図 4.1 を参照)。

また、慶応大黒宮らが出展した P2P オーバレイ最適化アルゴリズムの実装である“Intellisense”の実験網として構築した模倣環境を利用してもらうという模倣環境を利用した P2P オーバレイ実験も実施した (図 4.2 を参照)。

審査の結果、チーム模倣インターネットは総合 4 位という結果となった。

Interop CCC で行われた実験や、過去 AnyBed を用いて構築された模倣環境を用いて実験した内容や実験手法をまとめインターネットアーキテクチャ 9 月研究会にて発表を行った (文献 [124])。詳細は、`wide-paper-nerdbox-freaks-hazeyama-ia200909-00.txt` を参照してほしい。

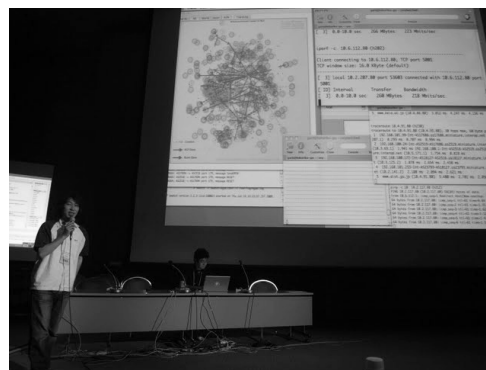


図 4.1. チーム模倣インターネットのデモンストレーション

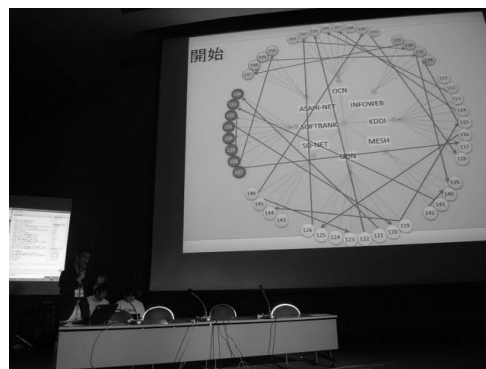


図 4.2. Intellisense のデモンストレーション

4.2.2 ACM SIGCOMM 2009 Demo Session

国外の学会でのデモンストレーションとして、ACM SIGCOMM 2009 Demo Session において“Emulating over 10K AS topology with massive VM multiplexing”というタイトルでデモンストレーションを行った。(ACM SIGCOMM 2009 Demo Session に投稿した Extended Abstract については `wide-paper-nerdbox-freaks-hazeyama-sigcomm2009-00.txt` を参照)

本年度の ACM SIGCOMM 2009 でのデモは、VISA 2009 で三輪が発表した論文“Experiences in emulating 10K AS topology with massive VM multiplexing.” (`wide-paper-nerdbox-freaks-danna-visa2009-00.txt` 参照) と連動し、VISA2009 の発

表に含まれる ASRD トップ 10,000 AS (10k AS) トポロジー構築の追実験も兼ねて、デモンストレーションを実施した。デモンストレーションは 8 月 18 日 (12:30–15:45) Poster and Demo Session I & Community Interest Session にて実施した。(図 4.3 はデモンストレーション会場の様子)。

図 4.4 が ACM SIGCOMM 2009 で行ったデモンストレーションのポスターである。デモンストレーションでは、昨年同様 NICT 北陸リサーチセンター内にある StarBED と ACM SIGCOMM 2009 の間を VPN で接続し、StarBED 上のグループ F とグループ H を用いて実験の操作と状態監視をデモ会場から行うという形でデモンストレーションを行った。

この実験では、三輪が VISA2009 で行った実験を追実験する形式で、syslog での観測を行えるように syslog の投入や static mac の投入などを行った。

デモンストレーションでは来場者、とくにルーティングの研究者から興味を持っていただき、いくつかの質問を受けた。質問をまとめると以下の通りである。

Q.1 Xen の代わりに OpenVZ でやってみないか？

人手が足りないです。つかっているツールはオープンソースとして公開しているので、ぜひ OpenVZ 版を作成してほしい。

Q.2 BGP の代わりに Secure BGP でやってみたいんだけど？

基本的にはできます。ぜひ行ってほしい。

Q.3 ウォームアップに数日かかるというのが面白い
まだまだチューニング段階なので、もう少し短時間にできるようにしたい。

Q.4 この再現したインターネット使わせろ

NICT と国際協力で実験を申請すればできるので、利用した実験を行いたい場合は後日コンタ

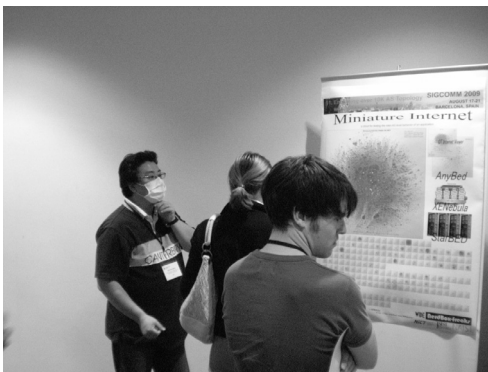


図 4.3. SIGCOMM 2009 デモ会場の様子

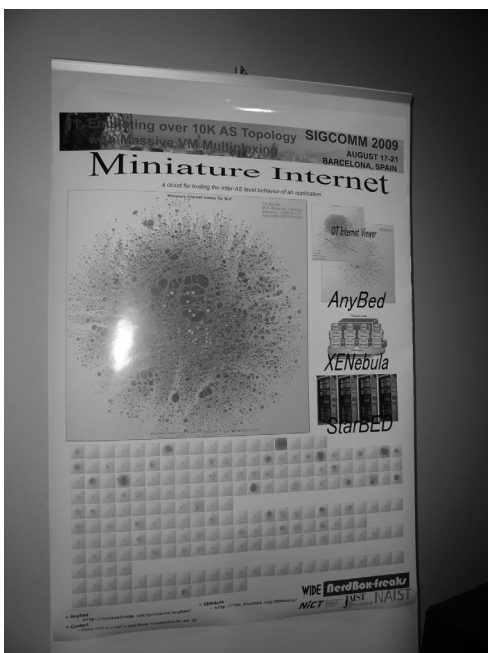


図 4.4. SIGCOMM 2009 デモポスター

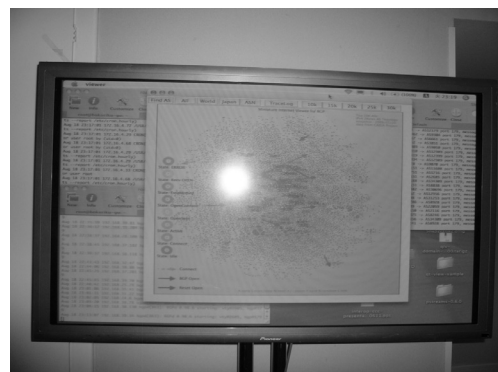


図 4.5. SIGCOMM 2009 デモのスナップショット

クトしてほしい。

Q.5 リーフレットほしいんだけど、ないかしら？

すみません。用意していませんでした。

なお、この実験では最終的に全ての bgpd ノードが起動しきらなかった。実験の最終結果や観測したエラーをまとめると以下のとおりである。

- デモ終了までに 9983 台で ttylinux の起動および ssh ログインを確認。残り 17 台はなかなか ssh に反応してくれないノードであった。
- デモ終了時には out of memory で bgpd が落ちたノードは存在しなかった。
- しかしながら、xen の netfront、netback あたりのエラーや、domU で /dev/sda1 を readonly で remount し、/var/run/bgpd.pid が作れず起動に失敗した bgpd が存在することがデモ終了後の解析と追実験で判明。
- 最終的に bgpd が上がってないノードは実際は 101 台。うち 7 台は ssh に反応しないノード、残り 94 ノードは bgpd 起動前に /dev/sda1 を readonly で remount していたために /var/run/bgpd.pid が作れず、手動で起動しても落ちるという状況であった。
- ほか、BGP で広報されるメッセージが長すぎて受け取れないという趣旨の bgpd エラーなどが見受けられた。

なお、10k 実験は後日 9 月 WIDE 合宿にて、SIGCOMM2009 のバージョンからさらにチューニングを施した XENebula にて再度追実験を実施したが、こちらは 8,800 ノード起動した段階で StarBED の利用時間終了となり、最終的な起動ノード数は確定できないまま終了した。

このように、データ観測可能で且つ安定した 10k AS 以上の模倣環境を構築するためにはまだまだ多くの課題が残されている。

4.2.3 インターネットコンファレンス 2009 デモ

本年度のインターネットコンファレンス 2009 (IC2009) でのデモは、“模倣インターネット——実 OSPF ネットワークの模倣——”[115] というタイトルで、IC2009 で鈴木が発表した論文である“ネットワークエミュレーションテストベッドを用いた実 OSPF トポロジ模倣システム”[114] と連動して 10 月 26 日にデモンストレーションを実施した。(IC2009 に投稿した論文については wide-paper-nerdbox-

freaks-mio-ic2009-00.txt を参照) デモンストレーションでは、StarBED のノード 50 台を用いて、50 のルータで構成される WIDE プロジェクトの AS 内 OSPF ネットワークの模倣を行った。

4.2.4 クラウドコンピューティングデベロッパーズワークショップ

クラウドコンピューティングに関する技術やオープンソースの習得を目的として、三重県津市磨洞温泉涼風荘にて 10 月 13 日から 16 日の 3 泊 4 日にて実装合宿形式のワークショップを開催した。CCDW の主催は株式会社ソリューション・クルー (scw) であるが、CCDW 開催に nerbox-freaks に参加する有志で協力する形で実施した (図 4.6 はワークショップの様子)。参加者は 7 名である。

CCDW で実施した内容は以下のとおりである。

- 「How to port/how to hack XENebula」
講師として、NICT 三輪、クルウィット安田、NAIST 榎本ら XENebula 開発者を招き、XENebula の porting 方法や XENebula のチューニング方法に関する講習を受けた。
- 「できる XDT — xdt による multithread programming —」
講師として scw 藤原を招き、sourceforge にて公開されているマルチスレッド C++ テンプレートライブラリ xdt[98] に関する講習を受けた。
- 「ADMD にみる、xsd と configure を使った、C++ XML システム開発手法」
講師として scw 藤原を招き、codesynthesis で公開されている XSD[102] を使った XML システム開発手法に関する講習を受けた。
- 「XEN hacking, tuning のいろは (仮)」



図 4.6. CCDW の様子

NICT 安藤および NICT 三輪により XEN[99]を用いた開発やチューニング方法に関する講習を受けた。

- 各自開発と成果発表

CCDW では単に講習を受けるだけでなく、参加者に期間中の開発目標を立ててもらい、3 日目夜の閉会時に開発成果を発表してもらう形式をとった。自己評価は 50 点から 115 点となったが、参加者からは「充実した内容だったので、ぜひ第 2 回、第 3 回と定期的に開催してほしい」との要望を受けた。

第 5 章 関連ツールのリリース状況

これまで Deep Space One ワーキンググループおよび Nerdbox-freaks では、CAIDA や route view などが計測したトポロジ情報やルータの LSDB から Quagga の設定ファイルを生成する AnyBed、Xen を用いて多数の仮想ノードからなるトポロジを構成する XENebula、そして StarBED における実験支援ツールである SpringOS などのさまざまなツールを開発してきた。

これらのツールについて、オープンソースソフトウェアとしてリリースを開始した。AnyBed は、2007 年 7 月 7 日にリポジトリを SourceForge のプロジェクトページ [9] から公開を開始し、現在、OSPF 対応を含めたソースコードが公開されており、同ページからダウンロードが可能である。XENebula は、2009 年 7 月 10 日より XENebula のホームページ [101] およびリポジトリの公開を開始し、同 7 月 16 日に 1.0 版をリポジトリにてリリースした。リポジトリからダウンロードが可能である。また、同時に、チュートリアル文書である「できる XENebula」の正式版も XENebula のホームページにて公開を開始した。SpringOS は、2010 年初頭に 1.5 版をリリースすべく現在作業中であり、α 版が StarBED のダウンロードページ [91] からダウンロード可能である。

第 6 章 おわりに

本報告では、今年度の Deep Space One ワーキンググループおよび Nerdbox Freaks ワーキンググループの活動報告を行った。本年度は、Deep Space One ワーキンググループとしては SpringOS や XENebula などのツールのリリースを行い、また新たに XBurner の開発を行った。Nerdbox Freaks ワーキンググループとしては、昨年度同様デモンストレーションやワークショップを開催し、模倣インターネット環境の構築を実施した。

来年度も Deep Space One ワーキンググループと Nerdbox Freaks ワーキンググループとの協調により、柔軟な実験環境を構築するためのツール群の研究開発とともに、それらを利用しより現実的な実験を行うための研究開発も並行して行っていく。また、ドキュメントの整備およびチュートリアルの開催による利用者のサポートも引き続き行っていく予定である。