第 XXV 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 25 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 WG 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（[223]）や GARIT、仮想マシンを利用した実験環境構築ツールである VM Nebula XENebula 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、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 WG 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 WG が創設された。Nerdbox Freaks WG 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 WG 及び Nerdbox Freaks WG 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。

- 用語定義
- 仮想ノードによる BGP 網エミュレーション
- 仮想ノードを用いたトラフィックジェネレータ
- エミュレーション実験における可視化
- イベント
- 各種ツールのリリース状況

1.1 用語定義

まず本報告で用いる技術用語の定義をおこなう。

模倣インターネット：
　実インターネット上で観測されたデータセットをもとにテストベッド上にエミュレーション技術を用いて構築したインターネットを模したネットワークを示す。模倣環境と呼称する場合もある。

BGP 網エミュレーション環境：
　模倣インターネットの形式のひとつであり、実インターネットの BGP 網を模してテストベッド上に構築したエミュレーション環境を示す。

P2P 網エミュレーション環境：
　模倣インターネットの形式のひとつであり、実インターネットで利用されている P2P アプリケーションを利用してテストベッド上に構築した P2P 網のエミュレーション環境を示す。

仮想マシン：
　XEN、VMWare、KVM などによって生成される仮想的な機械を提供するホストマシンを示す。

仮想ノード：
　仮想マシン上に作成されたノードを示す。

物理ノード：
　物理マシン上に作成されたノードを示す。

実験ノード：
　仮想ノード、物理ノードを問わず、実験に利用するノードを示す。

AS：
　Autonomous System（自律システム）の略語である。一般的にインターネットにおけるネットワーク運用管理単位として用いられる。

第 2 章　仮想ノードによる BGP 網エミュレーション

本章では、仮想ノードを用いた BGP 網エミュレーションに関する研究を簡単に報告する。

2.1 解析とモデル化に基づく BGP daemon 消費メモリ算出式

テストベッドで実験を実施する際、仮想ノードを用いて利用可能な実験ノードの台数を増やすことによる実験規模拡大が行われている。その際、テストベッドの物理ノードが保持する物理メモリ資源は限られているため、仮想ノードの役割に合わせて適切にメモリ資源を割り当てなければ、仮想ノードによってメモリの過剰割り当てや割り当て不足が発生してしまう。実際に、文献 [78] で実施した 10,000 AS 規模の BGP 網エミュレーション環境構築実験では、メモリの割り当て不足による Quagga bgpd[146] の起動エラーが頻発していた。

このメモリ割り当ての問題に対し、XENebula[128] を用いた BGP 網のエミュレーション環境を構築する際に仮想ノード（bpgd）へ割り当てるメモリ量を BGP daemon のソースコードの静的解析と小規模テストベッドでの観測結果から導出した消費メモリ算出式に沿って割り当てる手法の研究開発を Nerdbox Freaks WG では実施してきた。以降、本節では研究論文（文献 [197] および文献 [198]）にまとめた BGP daemon の消費メモリ算出式に関する研究を簡単に紹介する。

文献 [128] における BGP 網エミュレーションでは、BGP のメモリ消費モデルを BGP daemon の隣接 AS 数に対し、一定の隣接 AS 数に対して単純に比例し消費メモリが増加するという単純なモデルに基づく消費メモリ算出式を導出して仮想ノードにメモリを割り当てている。しかし、BGP daemon の実装では AS 間の接続関係ごとにルートマップの記述が異なるため、隣接 AS の関係ごとに消費されるメモリ量は異なるはずである。そこで、我々は Quagga bgpd のソースコードの静的解析と、実際に物理ノード 10 台、仮想ノード 100 台程度の小規模テストベッド上で Quagga bgpd を動作させ、実際に Quagga bgpd によって消費されたメモリ量の計測から回帰式を求める動態解析を用いて Quagga bgpd のメモリ消費モデルの決定と、決定した消費メモリ算出式の妥当性や規模拡大性に関する予備検証を文献 [197] および文献 [198] にて実施した。

本手法では、基本的に静的解析と動態解析の結果をもとに bgpd の消費メモリの総量を導出する消費メモリ算出式を立式している。bgpd の静的解析と動態解析の結果、経路を保持するためのメモリ量は経路に対して比例する関係であったが、BGP セッションを保持する際に消費されるメモリ量は AS 間の接続関係（ルートマップ）ごとに異なる消費メモリ量になることが明らかとなった。

ところで、関連研究 [39] で述べられているように、BGP アップデートメッセージに必要なメッセージバッファの使用量を多項式時間で求めることが非常に困難である。本手法で求める消費メモリ算出式は、ハードウェアの故障等で利用可能な物理メモリの総量が突如変更されるテストベッド上での実験を想定しているため、精度は悪くとも短時間で算出できる消費メモリ算出式を想定している。そこで、BGP アップデートメッセージのメッセージバッファに関する消費メモリ算出に関しては、最悪の場合を全隣接 AS から同時にフルルートを交換する場合と想定し、この最悪の場合に必要とされるメモリ量を消費メモリの上限と設定した。そして、隣接 AS の経路フィルタごとに設定した重みパラメータで最適値に調整するという、フィッティングを用いた消費メモリ算出式を導出した。

導出した算出式の妥当性を検証するための予備実験を実施した結果、経路フィルタごとに重みを変えず、単純に隣接 AS 数に比例して BGP アップデートメッセージのメッセージバッファに関する消費メモリを算出しメモリを割り当てると、経路のコアやリーフに位置する仮想ノードにてメモリの過剰割り当てが起こることが確認された。

現在、導出した消費メモリ算出式により算出したメモリ量と実際に BGP 網構築時に Quagga bgpd によって消費されるメモリ量の差分を調べ、導出した Quagga bgpd のメモリ消費モデルの妥当性を検証している。現在検証中であるが、bgpd の起動順によって実際に消費されるメモリ量に幅があり、トポロジーの規模を変えても、隣接 AS 数 100 ノード前後の仮想ノードにて消費メモリの振れ幅が大きくなることが確認されている。検証実験の結果は国際会議に投稿予定である。

2.2 BGP 網エミュレーションの特性の考察

仮想ノードを用いた BGP 網エミュレーションを活用した実験が行われているが、まだ限定的である。より多くの研究者らに利用されるためには、仮想ノードを用いた BGP 網エミュレーションが実験に対し

てどのような反応を示すのか、実験によりどのような状態遷移をするのか、そしてそれらの遷移する様子を観測したり、あるいは制御したりすることができるかを調査によって究明する必要がある。本節では、2009年9月に行われたWIDE研究会会場で行った実験に基づいて、仮想ノードを用いたBGP網エミュレーション環境内で観測されるパラメータがどのように変動するかを観測することにより、BGP網エミュレーションの特性把握を試みた文献[202]および[203]の内容について簡単に報告する。

文献[202]にて報告した仮想ノードを用いたBGP網エミュレーションで得られた知見をもとにし、文献[203]では待機状態、初期状態、実験中状態、異常状態の4つのシステム状態に暫定的に定義し、この4状態を仮想マシン側から新たに観測した。文献[203]では、待機状態から初期状態に遷移する状態、初期状態、初期状態から実験状態に遷移する状態に注力して観測した。

実験中に観測されたCPU、メモリ、インターフェースの利用状況は、ライブなバックグラウンドトラフィックを用いた実験では、異常状態に陥ることなく稼働した。その一方で、帯域に負荷がかかると、CPUに影響がみられた。これは、今後の負荷状態の観測で詳細を明らかする。

今後は、待機状態から初期状態への遷移や、実験中状態から異常状態への遷移など、各状態について観測を引き続き行い、仮想ノードを用いたBGP網エミュレーションの特性づけについて検討を重ねる。

第3章　仮想ノードを用いたトラフィックジェネレータ

インターネットではさまざまなサービスが提供されており、非常に複雑かつ巨大な環境となってきている。我々はこれまで、ネットワーク技術の信頼性を確保するために、実環境用向けのソフトウェアやハードウェア実装そのものを用いた検証を行ってきており、このような検証では、その検証対象となる技術および、それに直接関わると考えられる周辺技術を導入した環境のみを利用することが多かった。しかし、インターネット上ではさまざまなサービスによるトラフィックが常時存在しており、それぞれのサービスやトラフィックが相互に干渉しながら現状のインターネットの挙動を創り出していると考えられる。

より現実的な検証環境の構築のため、我々がこれまでに対象としてきた実環境向けの実装の検証と同様に実環境向けの実装を用いたトラフィック生成を行うためのツール群の開発を行っている。本章では、実装を用いたトラフィック生成を行うためのツール群であるXBurnerおよびCOSMOの研究開発について報告する。

3.1 XBurner

XBurner[129, 224, 225]はAnyBed[167]とXENebula[128]をベースとして、多数のXenノードを起動し、その上でBGP網エミュレーション環境を構築し、それぞれの仮想ノード上で実環境向けのアプリケーションを動作させることで現実的なトラフィックを生成するトラフィック生成ツール群である。

XENebulaはXen[190]を利用して、多数の仮想ノードを起動する。その一方で、最近Xenだけでなくさまざまな仮想ノード実現技術が利用されており、そのなかでもKVM[102]はLinuxカーネルに正式に導入され、Linuxカーネルをアップデートすることで常に最新版が利用でき、その導入コストも低いといえる。これにともなって、今年度は、KVMを利用することでその導入コストを下げ、また、XENebulaをXenだけでなく、さまざまな仮想ノード実現技術への対応可能性を模索する取り組みとして、XBurnerが採用するXENebula部分の再実装を行った。

XENebulaは、ttylinux[175]を利用することを前提として実装されており、各ホストOSでのゲストOSのための設定生成やコマンド実行はsshによって制御されている。新実装ではSpringOS[223]を利用して、ホストOSでの各種処理を実現した。

また、XBurnerと同様に、各種処理を抽象化し、それぞれに対応するモジュールとしてシェルスクリプトを用意して、SpringOSで起動している。本手法では、シェルスクリプトを置き換えることで、KVM以外の仮想ノード実現技術にも対応可能となっている。

3.2 COSMO

COSMOもXBurnerと同様に模倣インターネット環境を利用するが、XBurnerが、任意のアプリケー

ションを動作させることによる現実的なトラフィック生成を行うのに対し、COSMO は実ネットワーク上で観測されたトラフィック情報からトラフィックを生成するツールである。具体的には、tcpdump などで取得されたデータを解析して以下の動作を行う。

1. パケットの送信元 IP アドレスを利用し送信元 AS 番号の割り出し
2. パケットの受信時間に応じて送信元 AS を担当するノードからのパケット送出

大規模なインターネット接続を模倣する際には、AnyBed[167] および XENebula[128] を利用し AS レベルでの BGP 網エミュレーション環境を構築することが多い。このため、送信元 IP アドレスが所属している AS からトラフィックを発生させることになる。

実環境のデータを元にしたトラフィックの送出だけでなく、受信についての対応も必要である。前述の通り、利用する BGP 網エミュレーション環境では、AS 内のネットワークの模倣は行わないため、宛先の IP アドレスに相当するノードが存在しない可能性が高い。このようなパケットがある AS に到着した場合に、そのルータはエラーメッセージを返す。このようなメッセージによるパケットは COSMO によるトラフィックの現実性を低下させる要因となるため、これらを抑制する必要がある。これについては、さまざまな手法があるが、今後検討を要する。また、インターネット上のノードはさまざまな種類のリンクによって接続されており、それらのリンクの特性がトラフィックの性質にも影響を及ぼすことは明白である。今後 COSMO を利用して発生させたトラフィックの現実性について検証し、適切なリンク特性を導入することを検討していく。

第 4 章　エミュレーション実験の可視化

本章では、Nerdbox Freaks WG および Deep Space One WG で開発してきた、エミュレーション実験における可視化技術に関して報告する。

4.1 LNView

LNView は BGP 網エミュレーション環境など、大規模なネットワークエミュレーション実験を行う際に、主に網の構築状況やメッセージのやり取りなど、ネットワークトポロジーに対して動的に状態を重ね合わせて実験状態の把握を行うための可視化ツールとして開発された。図 4.1 は後述する Interop クラウドコンピューティングコンペティション [84] に参加した際にライブデモンストレーションで実施した、LNView を用いた BGP 網エミュレーション環

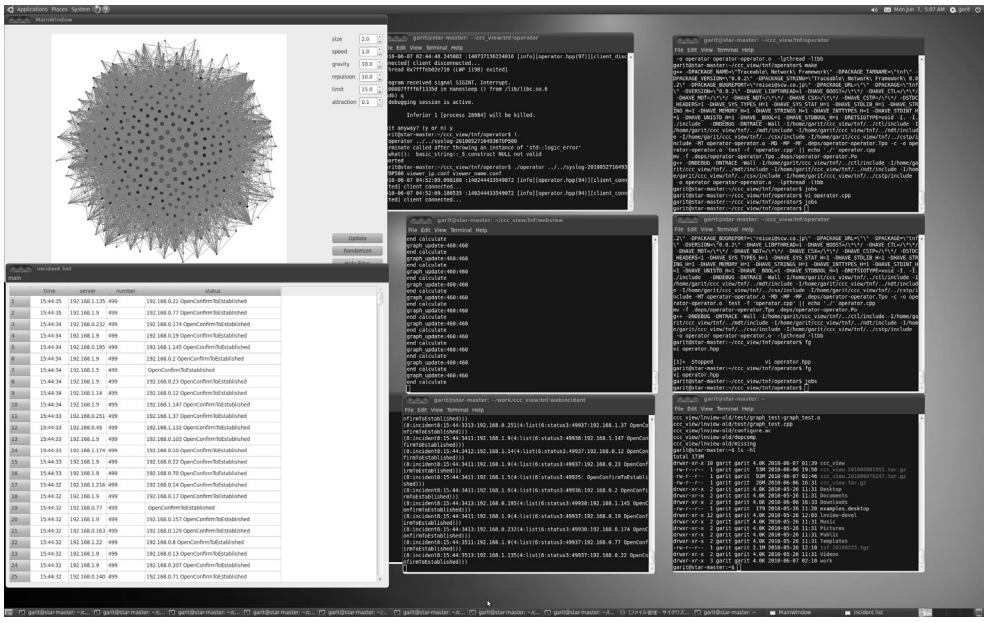

図 4.1.　LNView による BGP 網エミュレーション環境構築の可視化

境の構築状況を可視化した際の PC 上でのスクリーンショットである。LNView のアーキテクチャや性能に関しては文献 [218] にまとめ電子情報通信学会インターネットアーキテクチャ研究会にて発表した。

4.2 StarBED-Viz

模倣インターネットやクラウドコンピューティング技術の実験など、実験の内容が多数の実験ノードに跨った複雑なものになる場合には、障害の発見や実験状況の監視を支援する仕組みが必要となる。特に、テストベッド運用者から見た場合、必ずしも実験や検証の内容について詳細を知っているわけではないが、問題が生じた場合にはすぐ把握し、解決する必要があり、可視化は重要な支援手法となる。

そこで、実験ノードの通信を外部観測し、リアルタイムでのトラフィック量の変化を可視化し、通信上の問題が発生した場合に特定し易くなる StarBED-Viz を開発した（図 4.2 を参照）。

StarBED-Viz は、SNMP 等により情報を一定間隔で取得し、入力した絵の指定領域に対して、加工した数値と直線を描画する。StarBED-Viz 上に表した StarBED のスイッチ間リンクに対して、トラフィック量の変化が数値と直線の太さの両方で示され、トラフィック量の異常な変化があった場合には、可視化された情報から異常個所を読み取ることが可能となる。

第 5 章　イベント

今年度は、開発者向けのワークショップおよび実験のためのワークショップを開催し、開発技術だけではなく実験技術についても技術交流を行った。また、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、模倣環境のデモンストレーションを実施した。

5.1 MENS201003

StarBED 上での模倣環境に関する実験の連携実施と、実験技術の共有を目的として、情報通信研究機構北陸リサーチセンター（StarBED）にて、MENS201003（Multiple Experiments for Nerdbox/Nebula on StarBED）と称した連携実験を開催した。Nerdbox-freaks の有志が参加した。

MENS201003 は 3 月の 8 日から 19 日までの期間で実施された。参加した実験プロジェクトは BGP 網エミュレーション、P2P 網エミュレーション、

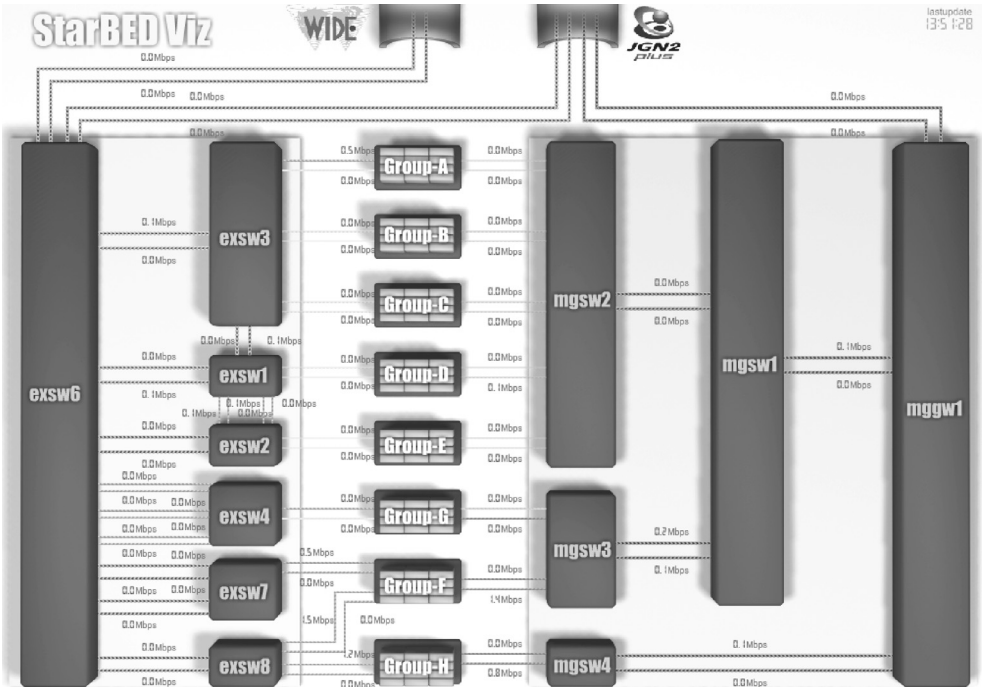

図 4.2.　StarBED-Viz

図 5.1.　MENS201003 に基づく成果の展示

XBurner、COSMO、MIMI-MAT（模倣インターネットによるマルウェア感染実験環境構築ツールセット）の 5 つのプロジェクトで、実験項目は 20 以上に渡った。また、各種の実験手法や実験関係のツールに関して、およそ 10 項目に渡る技術移転も行われた。

なお、MENS201003 における実験の成果は、「ICT システムテストベッドに関する国際シンポジウム」にてデモンストレーション展示が行われた（図 5.1 を参照）。

5.2 CCDW2010 Spring

広くクラウドコンピューティングに関連する技術開発を目的とした実装合宿である CCDW（Cloud Computing Develpers Workshop）を本年度も開催した。CCDW は 4 月 14 日から 4 月 17 日の 4 日間、大阪南港の研修センターにて実施した。参加者は 5 名であった。開発として、LNView、StarPOD を使った XBurner の評価、MENS で実施した大規模実験のログ解析、FreeBSD の vImage を用いた模倣ネットワーク実験環境の予備実験を実施し、参加者を講師役とした VRDF（Virtual Resource Definition Framework）、TNF（Traceable Network Framework）、クラウドの標準化の現状、CYBEX（Cyber security information exchange framework）とサイバーセキュリティの勉強会を実施した。図 5.2 は FreeBSD vImage に関する勉強会の様子である。

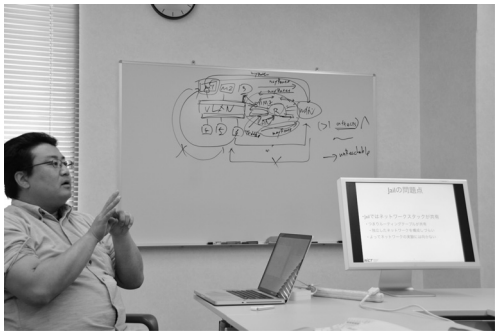

図 5.2.　CCDW2010 Spring での FreeBSD vImage の勉強会

5.3 SNDW2010 Summer

CCDW（Cloud Computing Develpers Workshop）とは別に、StarBED や XENebula を使った実験や開発を行っている人を集めた実験・実装合宿である SNDW（StarBED Nebula Developers Workshop）を実施した。SNDW は 7 月 27 日から 7 月 30 日の 4 日間、北陸リサーチセンターにて実施した。参加者は北陸リサーチセンターでの参加者が 10 名、遠隔参加 3 名であった。遠隔参加にはポリコムと慶應大学の WebEX を利用して行った。図 5.3 は北陸リサーチセンターでの SNDW 中の様子である。

図 5.3.　SNDW2010 Summer の様子

5.4 Interop Tokyo 2010 Cloud Computing Competition

2009 年度に続き、2010 年度においても Interop 2010 Cloud Computing Competition（InteropCCC）[84] に参加した。InteropCCC はクラウドコンピューティングに関する技術開発の競技会であり、Interop 2010 Tokyo クラウドコンピューティングコンペティション実行委員会が主催した。Deep Space One WG 及び Nerdbox Freaks WG では、StarBED とその運用技術をコンペティション参加者に提供する技術協力を行った。また、Nerdbox Freaks WG の有志による競技会への参加も行われた。

5.4.1 運用

InteropCCC では、StarBED をコンペティション参加者に提供し、各参加者は各々のクラウドコンピューティングに関連する技術を検証し、ライブデモンストレーションを実施した。今年度は、特に運用方式の改善を行うとともに、可視化による監視を行った。詳細に関しては文献 [228] にて報告している。

5.4.2 模倣インターネット構築デモ

StarBED の提供とともに、InteropCCC では実際に競技会に参加してライブデモンストレーションを実施した。ライブデモンストレーション “ミニチュアインターネット 3 分クッキング” と題して、2009 年 7 月 20 日に取得された CAIDA AS Relationship[27] のトップ 500 AS のトポロジーを用い、同じく CAIDA で公開されている IPv4 プレフィックスを広報している AS のデータセットである pfx2as のうちデータセット 2009 年 7 月 20 日のデータセットに含まれている実 IPv4 アドレスをマッピングした BGP 網エミュレーション環境をライブデモンストレーション中に構築し、構築する様子を可視化するという内容で実施した。可視化には情報通信研究機構で開発されている TNF（Traceable Network Framework）を用いて成形した BGP メッセージを StarBED の実験環境からデモンストレーション会場まで飛ばし、可視化用 PC にて CCDW2010 で開発した LNView[218] を用い、BGP 網が構築される様子をライブで描画した。図 5.4 は可視化した AS トポロジーの様子、図 5.5 はコンソールで実際に netstat -nr を実行したライブデモンストレーションを収めた写真である。総評としては、ライブで構築される様子をもっと見せてほしかったという意見が多かった。

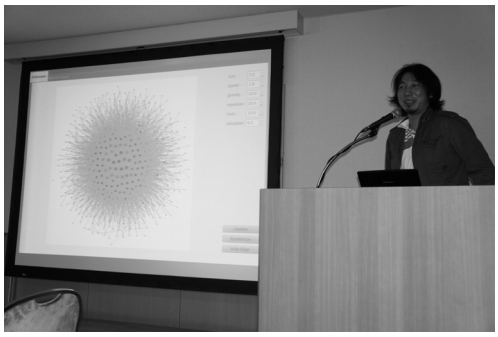

図 5.4. InteropCCC でのデモ（その 1）

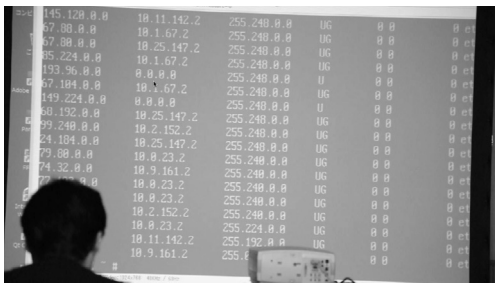

図 5.5. InteropCCC でのデモ（その 2）

第 6 章　各種ツールのリリース状況

現在、AnyBed[6]、XENebula[191]、SpringOS[165] が公開されている。また、LNView に関しては SIGCOMM 2009 Demo セッション [78] で用いたバージョンに関して [83] で公開されている。

第 7 章　おわりに

本報告では、今年度の Deep Space One WG および Nerdbox Freaks WG の活動報告を行った。本年度は、Deep Space One WG および Nerdbox Freaks WG と合同で SpringOS や XENebula など昨年度リリースしたツールに対する大規模検証実験や CCDW/SNDW などの開発ワークショップを実施し、現行のテストベッド管理ツールや実験補助ツールの利用を促進するとともに、次期テストベッド管理ツールや実験補助ツールのグランドデザインを行った。

来年度も Deep Space One WG と Nerdbox Freaks WG 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平行して行っていく。また、LACE（Long range Access to Collaborative Environments）Project において米国 Deterlab とのテストベッド連携実験およびテストベッド連携環境でのエミュレーション実験を実施する予定である。