情報処理学会論文誌 Vol. 49 No. 9 3234–3244 (Sep. 2008)

推薦論文

PlanetLabを用いたIPv6オーバレイネットワークの構築

櫻井 覚[†1] 管 文鋭[†2,*1] 松井大輔[†3]
今井祐二[†4] 村本衛一[†5] 河口信夫[†6]

PlanetLab (Chun, et al., 2003) は多数のノードを提供し，オーバレイネットワークは柔軟な構成のネットワークを提供する．実験者は PlanetLab ノードを用いてオーバレイネットワークを構築することで，これまで実現が難しかった実験を行えるようになる．しかし，現行の PlanetLab では，カーネルカスタマイズが不可能であるという制約と，実験者が自由にオーバレイネットワークを構築する手間が問題となる．我々は，仮想マシン User Mode Linux とトンネリングを用いることによって PlanetLab 上でオーバレイネットワークを構築する手法を提案する．また，オーバレイネットワークを容易に構築する機構 Orbit を提案する．そして UML と Orbit を用いるオーバレイネットワーク構築手法が IP 層プロトコルの検証に有益であることを XCAST6 (Imai, et al., 2003) の検証用ネットワークの構築を実例に示す．

IPv6 Overlay Network on PlanetLab

Satoru Sakurai,[†1] Fumihito Kan,[†2,*1]
Daisuke Matsui,[†3] Yuji Imai,[†4] Eiichi Muramoto[†5]
and Nobuo Kawaguchi[†6]

PlanetLab overlays the current Internet, and has been currently connecting over eight hundred nodes all over the world. Using PlanetLab nodes and constructing overlay networks, experimenters can experiment network architectures or services which seemed intractable before. However, it is difficult to construct an IP network on PlanetLab because PlanetLab does not allow us to customize the kernel. In addition, experimenters have to construct overlay networks by themselves. We propose a user-friendly method to construct overlay networks on PlanetLab with the help of User Mode Linux (UML) and tunneling, and another method, Orbit, to connect multiple nodes. We have made XCAST6 overlay network on PlantLab as a working example testbed of a new network protocol.

1. はじめに

インターネットは規模を急速に拡大しながら社会基盤としての重要性を増している．それにともない，実現すべき機能や要請される品質が変化し，これに対応するための機構として多くの新しいプロトコル提案が継続的に行われている．

新しいプロトコルが現実に使用されるまでには，発案，設計，試作，検証，普及といった数多くのステップを踏むことが必要である．インターネット以前の通信プロトコルの策定では，まず仕様策定団体が合意のための議論と精緻な文書化作業を実施し，実際のシステム作成は仕様の策定を待って行われた．これに対して，インターネットのプロトコル策定プロセスは，“Rough consensus, Running Code” というフレーズが象徴する，一種の実利主義に基づいた方針で進行する．発案されたプロトコルのアイディアは簡潔な Internet-Draft 形式の文書として提起される．仕様策定団体である IETF は Internet-Draft を起点とする標準化プロセスに基づいた議論を行いつつ，試作と実際の運用によってプロトコルを実証的に確かめる作業が並行して進行する．このため実証検証の成果がプロトコル策定にフィードバックしていき，動作する確証を得ながらプロトコルを収斂させることが可能となる．

インターネットコミュニティが，このような形でプロトコルを産み出せた理由として，下

†1 東京大学大学院新領域創成科学研究科
Graduate School of Frontier Science, The University of Tokyo
†2 名古屋大学大学院情報科学研究科
Graduate School of Information Science, Nagoya University
†3 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
†4 株式会社富士通研究所 IT システム研究所システムミドルウェア研究部
IT SYSTEMS MIDDLEWARE LAB. IT SYSTEMS LABORATORIES, Fujitsu Laboratories Ltd.
†5 松下電器産業株式会社
Matsushita Electric Industrial Co., Ltd.
†6 名古屋大学大学院工学研究科
Graduate School of Engineering, Nagoya University
∗1 現在，株式会社 KDDI
Presently with KDDI Co., Ltd.
本稿の内容は 2007 年 7 月のマルチメディア，分散，協調とモバイル（DICOMO2007）シンポジウムにて報告され，DPS 研究会主査により情報処理学会論文誌ジャーナルへの掲載が推薦された論文である．

© 2008 Information Processing Society of Japan

記のようないくつかの要因が考えられる．

- インターネットは，通信の端点のコンピュータの上位層がトランスポート以上に責任を持つことで，広く配備されるルータの IP 層以下の機構が簡素化されている．新規のプロトコルは，IP 層以下の構造が変化しない限り，トランスポート論理を End-to-end に変更することで使用できる．
- インターネットは小さなネットワークの有機的な複合体として機能しており，自律的に動作するネットワークはそれぞれの方法論で運用可能である．新規のプロトコルを実験のために部分ネットワークで試行運用することが可能である．
- インターネットはその初期においては，学術機関を結び研究活動を支援するとともに，ネットワークアーキテクチャ自体を研究することが目的とされていた．そのため運用と研究開発を同一の基盤の上で行うことが可能であった．

しかし，現在ではインターネットの大部分は商業サービスを提供することを目的として運用されており，研究開発とは基盤が分離されるようになっている．この状況下でも，プロトコルが IP 層以上の機構を用いて End-to-end に実現可能な場合，もしくは部分ネットワークのみへの導入で十分な検証を行える場合には，実証による検証は依然として容易である．

しかし，新しい IP 層の機構をインターネット上に広域分散配置するプロトコルの場合には，研究開発を目的としないネットワークの協力を期待できず，十分な規模で検証を行うことは大変困難になる．

本稿は，新たな IP 層プロトコルの検証のカバレッジを向上させる一手法として，インターネット広域分散テストベッド PlanetLab[13] を用いた IP 層プロトコル実証のためのオーバレイネットワーク手法を提案する．本手法では，PlanetLab の機能を活用しプロトコルの試験実装を世界規模で張り巡らされたテストベッド上に実証に必要な形態で配置させる．加えて，PlanetLab だけではできなかった IP 層プロトコルのカーネル内実装試作の配備を，UML を活用して可能にした．本手法以外の UML による IP 層プロトコル試作の配備ツールにない特徴として，オーバレイシステムの作成・配備の定式化自動化および GUI での支援機構がある．これにより，試行運用システム構築作業量が減少し，検証の効率化の向上が期待できる．

以降，2 章では，IP 層プロトコル検証における問題点を説明し，3 章で関連技術として既存のテストベッドを説明する．4 章で，本稿が提案する PlanetLab を用いた IPv6 オーバレイネットワーク手法を詳述する．5 章では策定中のプロトコル XCAST6[20] の本手法を用いた実証検証について述べ，手法の有効性を評価する．

2. IP 層プロトコル検証における問題点

本章では，新たな IP 層プロトコルの検証を行う場合に問題となる点について説明する．

2.1 IP 層プロトコル検証のテストベッド実現性に関する問題

IP 層プロトコルの検証では，BSD，Linux などのカーネル中にプロトコルスタックとしてその機能を実装し，これを用いて機能検証を行う方法が行われてきた．たとえば，IPv6 の検証では，BSD 版実装である KAME スタック[1] や Linux 版実装である USAGI スタック[2] を用いて，ルータ機能，端末機能の検証が行われてきた[3]．

IP 層プロトコルは，複数のノード間でパケットの転送を行う挙動を規定したものである．プロトコルスタックの検証を行う場合，複数のノードにこれらのプロトコルスタックの実装を導入し，ネットワークを構成して検証を行う必要がある．実験室などの小さな専用ネットワークの複数ノードにプロトコルスタックの実装を導入することは容易であるが，組織にまたがって遠隔に配置した複数のノードにプロトコルスタックを導入することは，以下の理由で困難である．

遠隔に配置したノードに新たなプロトコルスタックを導入するには，複数のノードにログインし，カーネルをリコンパイル，再起動し，その機能を有効化する必要がある．これらの操作を遠隔のユーザが行うことは，失敗時にノードが制御不能になるリスクをともなう．また，作業自体にノードの管理者権限が必要となるため，セキュリティリスクが高くなる．

すなわち，IP 層プロトコル検証のためのテストベッドには，実験者による遠隔ノードのカーネル更新を支援する機能が求められる．

2.2 様々なトポロジ構成に関する問題

複数のノードが配置されたテストベッドでは，いったんノードが配置されると，そのノード間の物理的な配線がトポロジを制約するという問題がある．IP 層プロトコルのルーティング機能の検証では，様々なトポロジにノードを配置して機能の検証を行う必要がある．このため，IP 層プロトコルのためのテストベッドは，実験ノード間に論理的な配線を実現し，様々なトポロジをエミュレーションできる必要がある．

2.3 実際の広域網の遅延・損失などの影響を反映した実験網構築に関する問題

IP 層プロトコルでは，複数のノード間で，経路情報を交換する．たとえばユニキャストでは，OSPF[4),5)] や BGP[6] によってルータ間，AS 間で経路情報を交換し，IP マルチキャストでは，PIM-SM[7]，MBGP[8] や MSDP[9] を用いて経路情報の交換を実現する．

現実のインターネットでは，これらの経路情報も，パケット伝送を行う同じ回線で交換する

© 2008 Information Processing Society of Japan

ことが，一般的に行われている．したがって，IP 層プロトコルの検証では，経路情報の伝播遅延や，経路情報交換トラフィックの負荷など，現実の広域網の遅延・損失などを影響反映した検証を行う必要がある．現実のインターネットのトラフィックは，日々変化するため，IP 層プロトコルの検証では，その影響を反映した検証をできることが望ましい．しかし，実運用している網のルータを検証対象のプロトコルを導入したルータに置き換えて検証することは，危険である．

また，IP 層では通信における信頼性は保障されていないため，パケットの損失も考慮に入れる必要がある．

IP 層プロトコル検証のためのテストベッドでは，実際の広域網の遅延・損失などの影響を反映した検証をエミュレーションできる必要がある．

2.4　実験ノードの個体差を排除した実験系構築に関する問題

広域に多数のノードを配置したテストベッドを構築するコストは高い．このため，後述する PlanetLab では，複数の実験者が実験ノードを共有することで，実験者 1 人あたりのコスト負担を削減している．

しかしながら，IP 層プロトコルの検証においては，他の実験者の影響を排除した検証を行う必要が出てくる．具体的には，パケットの転送性能の検証においては，転送機能以外の他の機能ができるだけ働いていない状況で検証を行いたい．しかし，複数の実験者が実験ノードを共有するテストベッドで，複数のノード間での CPU 資源，メモリ資源が確保された実験系を構築することは，困難である．

2.5　実験用のオーバレイネットワーク構築手間に関する問題

テストベッドでは，様々なトポロジを実現するため，論理的な回線をエミュレーションする．たとえば，IEEE 802.1Q で規定される VLAN を用いて，1 つの物理的なイーサネットインタフェース上に複数の論理的なインタフェースを設定することで論理的な回線を実現したり，ネットワークのスイッチ間で MPLS[10)–12)] や ATM（Asynchronous Transfer Mode）の論理チャネルを設定することで回線をエミュレーションしたりすることで様々な論理的な回線接続を実現する．このように，テストベッドでは，物理的な回線に論理的な回線をオーバレイさせることで，ネットワークを構築する．

これらの論理的な回線は，実験者が構築するネットワーク全体で，整合性を保って設定する必要がある．実験者がオーバレイネットワークを構築する場合，実験対象のネットワークが大規模になればなるほど，この設定および確認作業は膨大になる．

実験者が複数のノード間に論理回線を設定する工程においては，この作業の負担を軽減できること，また作業者のミスが混入する可能性を排除できることが望ましい．

3. 関 連 技 術

本章では本研究と関連のあるテストベッドや技術について述べる．

3.1　X6Bone

X6Bone は IP 層プロトコル XCAST を実証検証するネットワークである．検証実験開始当時には，後述するプロトコル検証テストベッドやツール類が存在しなかったため，インターネットの上の IPv6 オーバレイ網を検証協力者にネットワーク設定を依頼しながら構築し，運用・実証実験が行われてきた．

XCAST は代表グループアドレスの代わりに明示的なユニキャストアドレスのリストを宛先として使用する IP 層マルチキャストプロトコルである．XCAST は，少数受信者がインターネット上に広く分布するグループ内でパケットを同報することができる．従来のパケットフォワーデングとは異なる処理を行うため，カーネルに変更を加える形で実装されている．

オーバレイ網は目標とする使用者の状況にできる限り近い環境を使用するために，ADSL や FTTH など，様々な接続形態のユーザに中継ルータの設置が依頼された．各中継ルータは，WIDE プロジェクト慶應湘南藤沢キャンパスに設置したルータと IPv6 over IPv4 トンネルの仮想リンクで接続されている．XCAST の IP 層プロトコルをエミュレーションでなく最終的な実現形態であるカーネル内実装として実現されている．このため，送受信ノードと中継ルータのカーネルは，XCAST 対応パッチをそのソースコードに適用し，リコンパイルしたカーネルに置き換える必要がある．送受信ノードや中継ルータは実験協力者宅で，他の用途にも共用されている NetBSD マシンであり，協力者が root 権限を持つため，他の実験者は直接操作を行えない．

このような制約のため，実証環境を広域化するためには，実証に参加する団体を募り，詳細な作業手順を説明したうえで，オーバレイ網に接続してもらう必要がある．またオーバレイ網の接続やノード自身の変更は検証協力者の対応能力の範囲内で行うこととなるため，トポロジの変更やカーネルを含むプログラムの大幅な入替えは頻繁には行えず，多様な状況での検証は困難である．加えて，協力者間で実験に必要な情報が誤って伝わったり操作を間違えたりした場合には，障害の発生で検証が中断したり，意図とは異なる接続状況になっていたことがあとに判明し，検証作業をやり直したりする事故などが発生する．このため，検証環境の正当性チェックや記録など検証で本当にフォーカスしたい部分以外に労力を割く必要がある．

© 2008 Information Processing Society of Japan

3.2 PlanetLab

PlanetLab は広域分散コンピューティングの実験を行うためのテストベッドである．

新しいサービスを実験する場合，シミュレータや模擬的なクラスタシステムが用いられることが多い．しかし実際のインターネットはこうした実験環境よりはるかに規模が大きく複雑であるため，これらの実験環境の検証範囲には限界がある．しかし，一組織が数百台のノードを世界中に持つことは難しい．PlanetLab プロジェクトは，参加者が PlanetLab に各自が用意したノードを提供することで PlantLab に登録された他の参加者のすべてのノードを共同で利用できる仕組みを提供する．現在，PlanetLab は世界 35 カ国で約 800 ノードを有している．提供されたノードは PLC という PlanetLab の管理を行うホストに登録されている．このノードを用いて，PlantLab の参加者は広域で大規模な，これまで実施が難しかったサービスの実験を行える．さらに，実際にインターネットに接続されたノードを使用できるため，サービスを公開し，ユーザを巻き込んでの実験や，インターネットでの遅延や損失などの影響を含めた実験を行うことができる．ただし設計上，カーネルに変更を加えることができないため，本研究の対象のような，IP 層のプロトコルの検証には適さない．

PlanetLab では，ノードを複数の実験者で共有するために，Slice という考え方が使われている．Slice とは個人の実験単位のことで，PLC ホストによって管理されている．利用者は Slice に対して PlanetLab 上の任意のノードを割り当てることができる．1 台のノード上の 1 つの Slice を特に Sliver とよぶ．Sliver は Linux VServer によって実装され，1 台のノードは複数の Sliver を持つことができる．Linux VServer の特徴上，ネットワークインタフェースやルーティングテーブルは 1 台のノード上のすべての Sliver で共有され，実験者は PlanetLab ノードのカーネルに手を加えることができない．これらの理由から，PlanetLab 上で可能な実験はアプリケーション層での実験に限られている．**図 1** は PlanetLab ノードの構成の概念図である．

3.3 MyPLC

独自に PlanetLab 網を構築するためのソフトウェアとして，MyPLC というソフトウェアパッケージが公開されている．公式の PlanetLab のように，協力者を募って PlanetLab の亜種を作ることもでき，実験者の手元にある計算機を用いて構築し，占有可能な PlanetLab 環境を構築することもできる．PlanetLab では，実験用のノードは起動時に PLC からカーネルを入手して起動している．そのため，PLC ホストの管理権限を持つことによって，カーネルに変更を加えた PlanetLab 環境を構築することも可能になるが，すべてのノードで同一のカーネルが使用されるため，ノードや Sliver ごとに異なるカーネルを使用することはできない．また，占有可能な資源を利用して構築する場合は，広域分散させることが難しくなる．

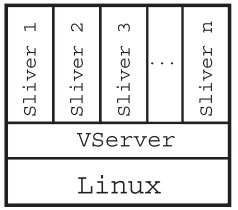

図 1　PlanetLab ノードの構成
Fig. 1　Structure of PlanetLab node.

MyPLC は Linux の RPM パッケージの形で公開されており，インストールすることで，PLC ホストとして動作させることができる．PLC ホストの場合は，一般的な Linux を OS として利用できるが，実験用のノードの場合は，専用の OS をインストールする必要がある．カスタマイズした OS を実験ノードに配布することは可能となるが，実験用のノードのためのインストーラは各 PLC ごとに異なるものを生成する必要があり，これを PLC ホストの Web サーバ上に配置して配布したり，CDROM などのインストールメディアを生成して配布したりする．多数の実験用のノードで構成される環境を構築する場合，遠隔の協力者への通達やインストールメディアの用意などが問題となる．

つまり，実験用のノードを配置する場合，実験者が独自に配置する必要があるため，多数のノードを配置して，様々なトポロジを構成したり，広域に配置して，インターネット上で発生する損失や遅延を加えた実験トポロジを構成したりするコストは高くなる．

3.4 StarBED

IP 層プロトコルの検証では，様々なトポロジを再現して実験を行う段階がある．StarBED [19] はそのような柔軟な実験環境の構築に適したテストベッドであり，約 800 台の計算機を備えた実験施設である．StarBED の各ノードはスイッチによって接続されており，スイッチの VLAN を変更することで自由にトポロジを変更できる．使用するノードは実験者が占有し，自由に利用できるため，新しいプロトコルを実装するためにカーネルを変更することも可能である．また，1 つの施設内にノードがあるため，カーネルの変更を行い，起動に失敗した場合なども対応が可能である．さらにノードを占有できるため，ノードの性能差に関する問題も解決できる．また，一部の実験ノードを折り返し回線を用いて遅延を再現することや，トンネル接続を用いて仮想的に遠隔配置することが可能であり，トンネル接続を管理する機能が検討されている．

© 2008 Information Processing Society of Japan

表 1　関連技術と提案技術の比較
Table 1　Comparison of the related technology and the proposed technology.

	X6Bone	PlanetLab	StarBED	PL-VINI	Orbit
カーネル変更	×	×	○	△ 配布用 Web サーバが必要	○
トポロジ変更	×	×	○	○	○
実際の遅延・損失	○	○	×	△ PlanetLab などの上で構築が必要	△ 同左
ノードの占有	×	×	○	△ 占有可能な環境の上で構築が必要	△ 同左
オーバレイ自動構築	×	×	×	△ トポロジ記述が必要	○

しかし，実験ノードは，同じ施設に設置されているので，インターネット上の回線の日々のトラフィックの変動やユーザの挙動をそのまま反映した実験を行うことは困難である．

3.5　VINI

実験用のオーバレイネットワークを構築するテストベッドとして，VINI[17] がある．VINI は MyPLC によって構築された PlanetLab の亜種であり，研究者にルーティングプロトコルの実験用のオーバレイネットワークを提供する．

VINI ではユーザにルーティング機構を提供するため，XORP というルーティングソフトウェアを持つ UML を使用する．XORP に対応させるため，UML イメージの転送は，各々のノードが VINI のサイトから UML イメージをダウンロードすることで実現している．UML どうしの通信に関しては，Click というソフトウェアを用いて，UDP トンネルで接続する．この仕組みを用いて VINI はユーザに対してルーティング可能なオーバレイネットワークを提供する．

PlanetLab で VINI と同様の機能を実現する PL-VINI[21] がある．PL-VINI を用いることで PlanetLab ノード上に UML イメージ，XORP，Click を配置してオーバレイネットワークを構築することができる．

PL-VINI は，設定ファイルからスクリプトを生成し，各ノードで実行し，オーバレイネットワークを構築する．PL-VINI はほとんどの作業を make コマンドから実行する形で実装されている．

現行の PL-VINI において，IPv6 はサポートされていない．カーネルイメージの変更も可能ではあるが，Web サーバから入手する形で配布しているため，各々のノードから接続可能な Web サーバが必要になる．

表 1 にそれぞれのテストベッド特徴をまとめる．本稿の提案手法である Orbit については後章で述べる．また PL-VINI は，PlanetLab 以外にも様々なテストベッドへ適用可能である．このため，目的に応じて適用するテストベッドを変更することで，ノードの占有や実際の遅延・損失などの特性を取り入れることができる．

4.　PlanetLab を用いた IPv6 オーバレイネットワーク手法の提案

本章では，IP 層プロトコルの検証のカバレッジを向上させる実験手法を提案する．インターネットの広域性を考慮した実験用ノードの確保できる PlanetLab に着目する．PlanetLab 上でのオーバレイネットワークの構築を半自動化する Orbit を提案する．

4.1　User-Mode Linux

新たな IP 層プロトコルスタックを実装する場合，カーネルへの変更が必要となる．しかしこれまでに述べたように，PlanetLab のような環境では，実験者がカーネルに手を加えることはできない．

カーネルを変更するための手段として，本研究では User-Mode Linux[16]（以下 UML とする）を使用する．UML として動作している OS はゲスト OS，UML を動作させている OS はホスト OS とよばれる．UML はユーザの 1 プロセスとして動作するため，PlanetLab のような環境でも動作させることができる．

4.2　UML_UDP

UML を用いることで，カーネルの変更がともなう実験も行いやすくなる．ネットワークプロトコルの実験の場合は UML どうしの通信を，インターネットの遅延や損失を反映させるには，インターネットを経由しての通信を行う必要がある．このため，ネットワークプロトコルの検証のためには，異なるノード上の UML どうしで通信を行う必要がある．

UML どうしの通信を実現するための方法として，UML_SWITCH というソフトウェアがある．しかし，UML_SWITCH は同一ノード上の UML どうしの通信を実現するソフトウェアであり，異なるノード上で動作している UML どうしの通信には適用できない．本研究では異なるノード上にある UML どうしの通信を実現するために，UML_SWITCH を変更し，UML_UDP として実装した．UML_UDP では，UML の仮想イーサネットインタフェースから送られるイーサネットフレームを UDP パケットで包み込んで送受信を行う．これによって，物理的な L2 リンクの実現を可能にした．実際にホスト OS から送信されるフレームは，ゲスト OS の送信したフレームに UDP ヘッダ・IP ヘッダが付与されるため，実際のホスト OS の MTU を超えてしまう．この問題については，UML 上での MTU を調

© 2008 Information Processing Society of Japan

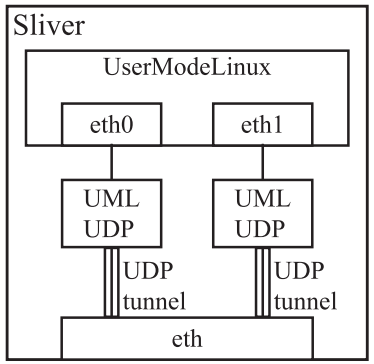

図 2　Sliver 上の状態
Fig. 2　Structure on Sliver.

整して対応した．MTU の値は ADSL などでの適正値とされている 1454 からイーサネットフレーム（26），IP ヘッダ（20），UDP ヘッダ（8）の値を減算し，1400 にした．これらの構成の概念図を**図 2** に示す．

4.3　Orbit

4.3.1　オーバレイネットワークの構築の具体的な問題点

新しいプロトコルを実装した UML と UML_UDP を用いることで，UML によるオーバレイネットワークを構築し，新しいプロトコルの実験を行える．構築のためには以下の作業を行う必要がある．

- 各ノードへの UML イメージおよび UML_UDP の転送
- UML と UML_UDP の起動
- UML の各ゲスト OS でのネットワークの設定

使用するノードおよびゲスト OS が増えるにつれ，これらの作業にともなう時間も増加する．ノードの数が n 倍になれば，とりうるリンクの数は n の 2 乗オーダの組合せがあり，誤った設定を行う可能性も高くなる．このため，UML と UML_UDP によるオーバレイネットワークであっても，実験環境を人手で構築することは困難である．シェルスクリプトなどを作成することで，構築にともなう手間は軽減されるが，実行以外にもスクリプトを生成し，各ノードへ配置する必要があり，多数のノードを使用する場合は構築が困難になる．また，誤った設定を行う可能性がある点は解決できず，ノードの追加やトポロジの変更などを行う場合はスクリプトも変更しなければならない．

```
nodes:
 node1: pl_user@node1.example.com
 node2: pl_user@node2.example.com
 node3: pl_user@node3.example.com
 node4: pl_user@node4.example.com
node1:
 tun:
  eth0:
   ipv4:
    addr: 192.168.0.1/24
   addr: 2001:100::1
   plen: 64
   port: 5000
   connect:
    node: node2
    intf: eth0
  eth1:

   ...

node2:
 tun:
  eth0:

   ...
```

図 3　YAML を用いた設定ファイルの一部
Fig. 3　A part of configuration file sample in YAML format.

4.3.2　Orbit による構築

UML と UML_UDP によるオーバレイネットワークの構築における構築の手間の軽減と，誤った設定を防ぐため，本研究では構築支援機構の Orbit を提案する．Orbit は，前章で述べたオーバレイネットワークの構築に必要な処理を，1 台のノード上からコマンドを実行することで行える．

Orbit ではオーバレイネットワークのトポロジを記述した設定ファイルをもとに，構築に必要なスクリプトを生成し，各ノードへ配置，実行する．設定ファイルはノードのインタフェースのつながりや使用する実験用 IP アドレスなどを YAML [18] 形式で記述する．**図 3** に YAML 形式でのトポロジの設定例を示す．

また，設定ファイルの作成にあたっても誤った設定を防ぐため，我々は GUI によってトポロジを記述し，それをもとに YAML 形式の設定ファイルを生成するツールを作成した．GUI ツールではノードどうしを線で結び，トポロジを記述する．実験用 IP アドレスについては自動的に割り当てられる．**図 4** にツールによるトポロジの設定例を示す．ツールによって，トポロジの記述された設定ファイルが生成され，実験者は使用するノードのユーザ名と IP アドレスを，設定ファイルの node フィールドに追記し，この設定ファイルを用いて次に述べる各種コマンドを実行することでオーバレイネットワークを構築できる．

各種コマンドは make コマンドによって実行する形で実装した．オーバレイネットワークの構築に利用するコマンドと実行される処理を以下に示す．

© 2008 Information Processing Society of Japan

- make
  スクリプトの生成を行う．生成されるスクリプトは UML の起動のように，ホスト OS で実行されるものと，ゲスト OS で実行されるものの両方が生成される．
- make setup
  UML，UML_UDP，UML を操作するためのソフトウェア，UML のためのファイルシステムイメージなど，更新する頻度が低く，サイズの大きいファイルを各ノードに対して転送する．
- make sync
  UML の起動スクリプトなど，ノードごとに必要になる作業用のスクリプトを転送する．作業用のスクリプトは make の実行段階で自動生成される．
- make start
  UML，UML_UDP の起動を行う．
- make include
  ゲスト OS へファイルを転送する．
- make config
  ゲスト OS 上で設定用のスクリプトを実行する．
- make zebra
  手作業による経路の設定の負担を減らすため，ダイナミックルーティングを実行する．現状は Zebra パッケージに含まれる OSPFv3 を用いて動的な経路制御を行う．

Orbit を用いることによって，オーバレイネットワーク上のノードの追加や，トポロジの変更などを行う場合でも，ノードの設定作業が増えることはなく，設定ファイルの記述を変更するだけで行える．

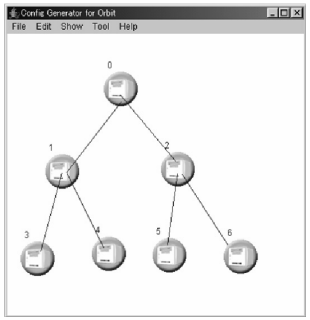

図 4　GUI を用いた設定ファイル生成
Fig. 4　Generation of the configuration file by the GUI application.

5. 提案手法の評価

XCAST6 プロトコル検証に本稿提案手法を用い，評価した．

5.1 XCAST6 による通信実験

提案手法の実現可能性，有効性を評価するため，Orbit を用いて実際にオーバレイネットワークを構築した．IP 層プロトコルとしては，XCAST の IPv6 版プロトコルである XCAST6 を採用した．実験では，3 つの実験環境に Orbit を適用した．まず Orbit 機能の動作確認するため，実験室内の環境で MyPLC を用いて検証した．複数のノードからなるより複雑なトポロジでの動作検証を行うため，StarBED と MyPLC を用いて構築した環境[14] で実験を行い，最後に，インターネットの遅延・損失による影響を確認するため，公式の PlanetLab で実験を行った．

5.1.1 実 験 内 容

多数のノードがある StarBED と PlanetLab で実験を行った際には，Abiline をモデルに，**図 5** のトポロジを構築した．XCAST6 による通信実験を行い，XCAST6 のパケットが送受信されるか，ルータによる分岐配送が行われるかを検証した．経路設定に関しては，OSPFv3 を用いて動的に設定を行った．

いずれの環境でも，XCAST6 による通信が行えることを確認できた．

5.2 カーネルの変更機能に関する評価

新しいプロトコルを実装したカーネルを実験者が自由に配置できる機能を確認するため，世界各国に配置された PlanetLab ノードへの配置を試みた．**図 6** は，ドイツ，北米，

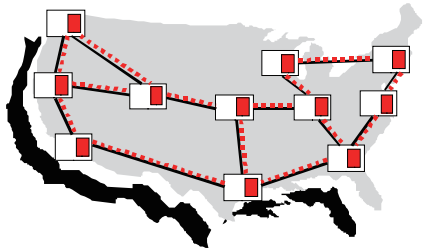

図 5　Abiline のバックボーントポロジ
Fig. 5　Backbone topology of Abiline.

© 2008 Information Processing Society of Japan

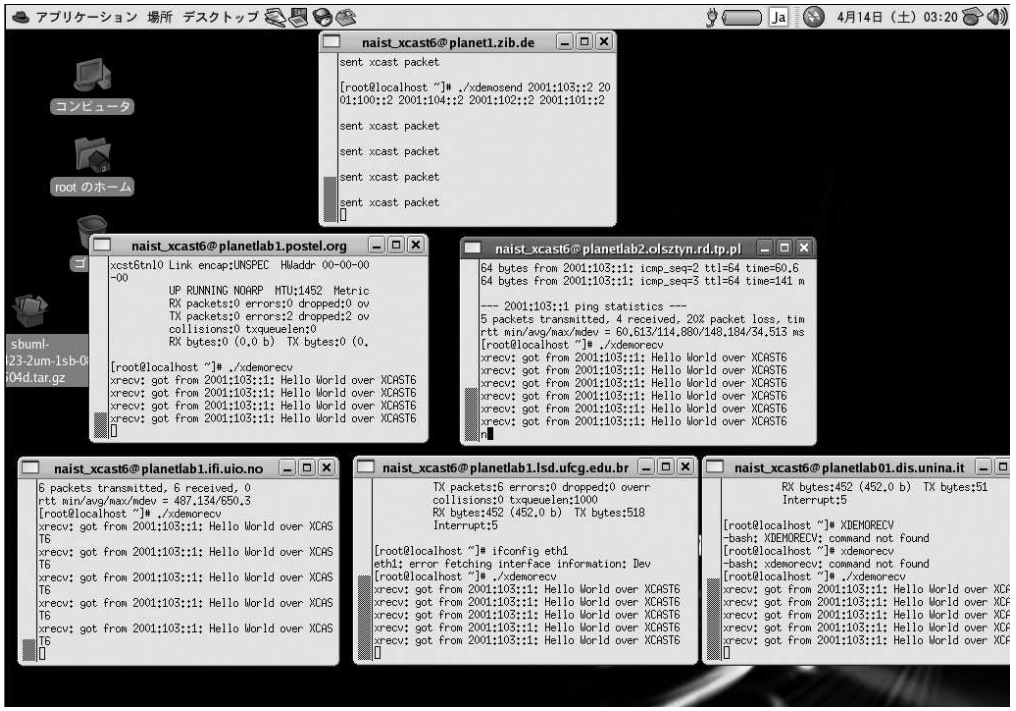

図 6　6 カ国の PlanetLab ノードへカーネルを配置
Fig. 6　Kernel deployment to the PlanetLab node in 6 countries.

ポーランド，ノルウエイ，ブラジル，イタリアの PlanetLab ノード上に XCAST6 が実装されたカーネルを配置し，XCAST6 パケットを交換している様子を示したスクリーンショットである．カーネルの配置に数時間を要したが，すべてのノードに自動的にカーネルが配置され，XCAST6 パケットが配送可能なオーバレイネットワークが構築できることを確認できた．

5.3　トポロジの変更機能に関する評価

実験においては，以下の 2 つの方法でトポロジを記述した YAML ファイルを作成し，これを用いて実際にオーバレイネットワークが構築できることを確認した．

- GUI を用いてトポロジを作成し，YAML を自動生成させる方法
- YAML を直接編集してトポロジを記述する方法

5.4　遅延・損失の考慮に関する評価

PlanetLab を用いることによって，実際にインターネットの遅延や損失の影響を受けた環境での，オーバレイネットワークの挙動を確認した．具体的には，PlanetLab 上の複数のノード上の UML 間に Orbit を用いて IPv6 オーバレイネットワークを構築した後，UML 上で OSPFv3 を動作させ，アンダレイヤの IPv4 ネットワークの遅延，損失がある状況でも，自動的に IPv6 経路情報が伝播できることを確認した．

5.5　ノードの個体差の問題の解決に関する評価

StarBED のノードと MyPLC を用いた実験では，ノードを占有できるため，ノードの個体差による影響やインターネットの伝播遅延を排除した環境で，XCAST6 パケットの転

© 2008 Information Processing Society of Japan

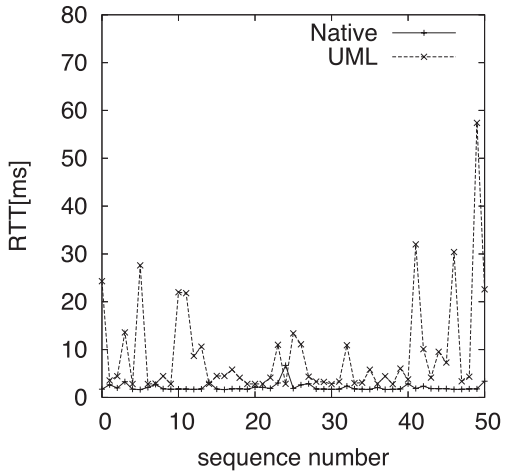

図 7　実際の RTT とオーバレイネットワーク上の RTT の比較
Fig. 7　Comparing of the RTT in the native network and the overlay network.

送性能の評価が可能となる．我々は，StarBED のノードで MyPLC を動作させた状況で，Orbit を稼動させ，オーバレイネットワークを構築した．この上の XCAST6 パケットの転送処理において，伝送遅延のジッタが発生するかどうか，評価を行った．

この結果，インターネットの伝播遅延を排除した環境で，XCAST6 パケットの転送時間にばらつきが発生することを確認した．

図 7 は，転送時間のばらつきが発生する様子を示したグラフの一例である．planetlab1.sfc.wide.ad.jp から planetlab1.iii.u-tokyo.ac.jp に対して実ネットワーク，およびオーバレイネットワーク上で ping を用いて通信し，測定を行ったときのものである．

オーバレイネットワーク上の結果では速度にばらつきが見られた．原因としては，UML を使用しているため，ユーザ空間とカーネル空間のコンテキストスイッチが頻繁に起こるためと考えられる．

5.6　オーバレイ構築の問題の解決に関する評価

Orbit はオーバレイネットワークの構築に特化した実験支援機構であり，カーネルの配置，UML 間のインタフェース設定，UML 上の経路設定は，自動的に進行する．このため，オーバレイネットワーク構築時に作業ミスによる不具合は，発生しなかった．**表 2** は，Orbit と PL-VINI の IPv6 オーバレイネットワーク構築に関する項目を比較したものである．Orbit

表 2　提案手法 Orbit と PL-VINI の比較
Table 2　Comparing of Orbit and PL-VINI.

	Orbit	PL-VINI
UML の利用目的	カーネル変更	ユーザごとの経路表
UML の変更	○	△
IPv6 の設定への対応	○	×
GUI ツール	○	×
外部との通信	×	○

がより柔軟に IP 層プロトコルの検証に対応できていることが分かる．また，PL-VINI と同様に，実験の目的にあわせてアンダレイヤのテストベッドを変更することで，表 1 にあげた他のテストベッドの特性を取り入れることができる．

6. ま と め

IP 層プロトコルの実証検証実施に必要な世界規模のテストベッドを効率良く構築するため，PlanetLab を用いた IPv6 オーバレイネットワーク構築手法を提案し，テストベッドの構築と，実際の IETF 提案中のプロトコル検証作業を行い，手法の有効性を示した．

本手法は，これまでの同種の試みの成果を統合したうえで，テストベッド作業の詳細を見直し，定式化と自動化が可能なポイントを追加するツールを整備することで実現した．PlanetLab に参加する地理的にもネットワーク運用主体としても分散している場所への実証に必要な機構の配備を行い，UML により IP 層を改造した Linux kernel をルータとして動作させ，UML_UDP によって仮想的な IPv6 オーバレイ網を構築する．これらの機構の組合せで，ネットワークバックボーンルータの置き換えや改造が難しい状況でも，実環境に近い状況下での IP 層プロトコルの広域実証検証が可能になる．

これに加えて，テストベッドオーバレイネットワークの形式的記法，Orbit による GUI 設計支援，IP 層論理を含む UML カーネルの自動配備，UML_UDP オーバレイ設定作業自動実行，経路制御デーモン起動などを新たに整備した．これによりテストベッドの設計と構築にかかる開発手番が短縮された．さらに，ルータの配備や設定を検証協力者の手を煩わせることなく遠隔から自動的に行えるため，テストベッドの構成変更が自由に行える．オーバレイネットワーク作成作業のための指示誤解や運用作業人為ミスが排除されるので，設計とテストベッド構造が必ず一致することになり，検証環境自身の状況確認や記録などが不要となり，本来フォーカスしたい検証作業により多くの工数をかけることが可能となった．また今後の課題として，L2TP などを用いてインターネットとの接続を可能にし，ユーザが

© 2008 Information Processing Society of Japan

オーバレイネットワークに参加できるようにすることが考えられる．

謝辞　本研究の一部は，総務省戦略的情報通信研究開発推進制度（SCOPE）（061106004）の支援を受けて行われた．本研究にあたり，実験や提案に協力してくださった方々に感謝します．VINI の著者，Andy さんには関連研究の調査に協力いただきました．StarBED の利用の際には，北陸リサーチセンターの皆様にお世話になりました．日頃からご指導いただいている加藤明教授，篠田陽一教授，そして研究の機会を与えていただいた WIDE プロジェクトの皆様に感謝します．

参　考　文　献

1) KAME project homepage. http://www.kame.net/
2) USAGI project homepage. http://www.linux-ipv6.org/
3) TAHI project homepage. http://www.tahi.org/
4) Moy, J.: OSPF Version2, RFC 2328 (Apr. 1998).
5) Coltun, R., Ferguson, D. and Moy, J.: OSPF for IPv6, RFC 2740 (Dec. 1999).
6) Rekhter, Y., Li, T. and Hares, S.: A Border Gateway Protocol 4 (BGP-4), RFC 4271 (Jan. 2006).
7) Estrin, D., Farinacci, D., Helmy, A., Thaler, D., Deering, S., Handley, M., Jacobson, V., Liu, C., Sharma, P. and Wei, L.: Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification, RFC 2362 (June 1998).
8) Bates, T., Rekhter, Y., Chandra, R. and Katz, D.: Multiprotocol Extensions for BGP-4, RFC 2858 (June 2000).
9) Fenner, B. and Meyer, D.: Multicast Source Discovery Protocol (MSDP), RFC 3618 (Oct. 2003).
10) Rosen, E., Viswanathan, A. and Callon, R.: Multiprotocol Label Switching Architecture, RFC 3031 (Jan. 2001).
11) Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T. and Conta, A.: MPLS Label Stack Encoding, RFC 3032 (Jan. 2001).
12) Andersson, L., Doolan, P., Feldman, N., Fredette, A. and Thomas, B.: LDP Specification, RFC 3036 (Jan. 2001).
13) Chun, B., Culler, D., Roscoe, T., Bavier, A., Peterson, L., Wawrzoniak, M. and Bowman, M.: PlanetLab: An Overlay Testbed for BroadCoverage Services (2003). http://www.planet-lab.org/
14) 松井大輔，櫻井　覚，管　文鋭，今井祐二，村本衛一，河口信夫，篠田陽一：新広域サービスの段階的検証手法への PlanetLab の適用，*DICOMO 2007* (2007).
15) Kawaguchi, N., Imai, Y., Muramoto, E., Sakurai, S., Matsui, D. and Kan, F.: XCAST on PlanetLab, *Workshop on Peer-to-Peer Multicasting 2007* (*P2PM'07*), Demonstration (2007).
16) Dike, J.: User-Mode Linux. http://user-mode-linux.sourceforge.net/
17) Bavier, A., Feamster, N., Huang, M., Peterson, L. and Rexford, J.: In VINI Veritas: Realistic and Controlled Network Experimentation, *Conference on Computer Communications* (*Sigcomm*) (2006).
18) YAML. http://www.yaml.org/
19) Miyachi, T., Chinen, K. and Shinoda, Y.: StarBED and SpringOS: Large-scale General Purpose Network Testbed and Supporting Software, *International Conference on Performance Evaluation Methodologies and Tools* (*Valuetools*) *2006* (Oct. 2006).
20) Imai, Y., Kishimoto, H., Shin, M.-K. and Kim, Y.-H.: XCAST6: eXplicit Multicast on IPv6, *IEEE/IPSJ SAINT2003 Workshop 4 IPv6 and Applications* (2003).
21) PL-VINI homepage. http://www.vini-veritas.net/documentation/pl-vini/user

(平成 20 年 1 月 6 日受付)
(平成 20 年 6 月 3 日採録)

推　薦　文

世界中に分散した 800 近いノードを有する実験環境である PlanetLab を題材に，オーバレイネットワークを容易に構築する機構と，マシンを相互接続させるしくみ（Orbit）を提案．オーバレイネットワーク上で XCAST6（IPV6 のプロトコル）を動作可能とした．本手法は User Mode Linux を用いカーネルレベルの研究を可能としたことに意義があり，意欲的な取り組みである．論点を整理することで論文として十分にまとまる内容であり，推薦に値する．

（マルチメディア通信と分散処理研究会主査　櫻井紀彦）

櫻井　　覚

2008 年東京大学大学院新領域創成科学研究科修士課程を修了．DNS，コンピュータネットワークの実験環境の研究に従事．

© 2008 Information Processing Society of Japan

管　　文鋭

2008 年名古屋大学大学院情報科学研究科修士課程修了．ユビキタスコンピューティング，オーバレイネットワーク，スマートルームに関する研究に従事．現在，KDDI（株）ソリューション事業統轄本部所属．

松井　大輔

2006 年稚内北星学園大学情報メディア学部卒業．現在，北陸先端科学技術大学院大学博士前期課程に所属．PlanetLab，StarBED を用いた検証環境の構築に関する研究に従事．WIDE Project メンバ．

今井　祐二（正会員）

株式会社富士通研究所 IT システム研究所システムミドルウェア研究部．1992 年 3 月名古屋大学工学研究科博士課程前期課程情報工学専攻修了．データセンタ基盤の研究開発に従事．XCAST fan club，Japan エバンジェリスト．WIDE Project メンバ．

村本　衛一

1991 年 4 月松下電器産業株式会社に入社，生産技術本部でシステム開発を担当．1998 年 4 月北陸先端科学技術大学情報科学研究科博士前期課程入学（企業派遣）．2000 年 3 月同課程修了後，帰社し本社技術部門でグループ通信の研究開発に従事．WIDE Project メンバ．

河口　信夫（正会員）

1990 年名古屋大学工学部電気電子工学科卒業．1995 年同大大学院工学研究科情報工学専攻満了．同大助手，講師，助教授を経て 2007 年より同大工学研究科准教授．ユビキタスコミュニケーション，スマートネットワークの研究に従事．2004 年より大学発ベンチャー（有）ユビグラフ取締役兼務．工学博士．ACM，IEEE，電子情報通信学会，日本ソフトウェア科学会，電気学会，人工知能学会，日本音響学会各会員．

© 2008 Information Processing Society of Japan