社団法人 電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE
電子情報通信学会 NS 研究会 2010-9-2

# 仮想化ノードを使用した実験用非IPプロトコルの開発

金田 泰
日立製作所 中央研究所
〒185-8601
東京都国分寺市東恋ヶ窪 1-280
E-mail: yasusi.kanada.yq@hitachi.com

中尾 彰宏*†
*東京大学大学院
情報学環・学際情報学府
〒113-0033 東京都文京区本郷 7-3-1
†情報通信研究機構
〒113-0001
東京都文京区白山 1-33-16
E-mail: nakao@iii.u-tokyo.ac.jp

**あらまし** 情報通信研究機構 (NICT) においてはネットワーク仮想化技術を使用して任意のフレーム形式をもつ非 IP プロトコルを実装可能にした 10 Gbps 級の仮想化ノード (VNode) を開発している．我々はそれを利用して実験的な非 IP プロトコル IPEC (IP Ether Chimera) を開発した．IPEC においては IP アドレスのような階層化可能なアドレスを Ethernet スイッチの学習アルゴリズムを拡張したアルゴリズムによって学習する．IPEC はつぎのような特徴をもつ．第 1 に Ethernet, IP それぞれの特徴的な機能の一部を 1 層の単純な非 IP プロトコルによって実現している．第 2 に学習をグループ単位でおこなうため，Ethernet よりスケールするうえ，グループ単位の移動が効率的に学習できる．第 3 にこのアルゴリズムはループをふくむネットワークでも使用でき，障害時にも代替経路で通信できる．グループ ID はロケータとしても使用できるため，ID/Locator 分離を拡張したアーキテクチャを実現しているとかんがえることができる．IPEC を VNode 上に実装して，グループ単位の学習や端末の移動に実際に対応できることを実験により確認した．

**キーワード** ネットワーク仮想化，非 IP プロトコル，仮想化ノード，アドレス学習，ID/Locator 分離，モビリティ．

# Development of An Experimental Non-IP Protocol Using the Virtualization Nodes

*Yasusi Kanada*
Central Research Laboratory,
Hitachi, Ltd.
Higashi-Koigakubo 1-280,
Kokubunji, Tokyo 185-8601
E-mail: yasusi.kanada.yq@hitachi.com

*Akihiro Nakao**†
*The University of Tokyo Interfaculty Initiative in Information Studies, Graduate School of Interdisciplinary Information Studies
Bunkyo-ku Hongo 7-3-1, Tokyo 113-0033
†National Institute of Information and Communications Technology
Bunkyo-ku Hakusan 1-33-16,
Tokyo 113-0001
E-mail: nakao@iii.u-tokyo.ac.jp

**Abstract** In the National Institute of Information and Communications Technology (NICT), 10-Gbps-class virtualization nodes (VNodes) that enables implementing non-IP protocols with any frame format are developed using network-virtualization technology. We have developed an experimental non-IP protocol called IPEC (IP Ether Chimera) on a virtual network using the VNodes. In IPEC, the nodes learn addresses that can be hierarchical such as IP addresses using an algorithm that extends Ethernet switch learning algorithm. IPEC has the following features. First, IPEC realizes a simple single-layer non-IP protocol that has features of both Ethernet and IP. Second, because a group is the unit of learning in IPEC, it is more scalable than Ethernet, and mobile groups can be more efficiently learned. Third, this forwarding algorithm can be used in networks with loops and it can forward packets during failure using an alternative route. Group IDs can be used as locators, so IPEC can be regarded to realize an architecture that extends ID/Locator separation architecture. We implemented IPEC on VNodes, and confirmed that it enabled group learning and group mobility by experiments.

**Keywords** Network virtualization, Non-IP protocol, Virtualization node, Address learning, ID/Locator separation, Mobility.

## 1. はじめに

インターネットはもともと単純なネットワークをめざしてきたが，さまざまな用途につかわれるようになるのにしたがって複雑化してきた．インターネットにおいてはすべてのプロトコルがインターネット・プロトコル (IP) をベースとしているため，さまざまなプロトコル機能が干渉しあい，それらを共存させるために調整が必要になって複雑化するとともに，新機能を追加するのが困難になってきている．一方でクラウド・コンピューティングの普及などによってインターネットへの要求がさらに多様化・高度化するなかで，IPv4 や IPv6 などをベースとするプロトコルでは対応が困難になっている．

このような状況のなかで新世代ネットワークのプロジェクトにおいては IP にとらわれない新プロトコルを開発し，そのうえで IP 上では困難だったさまざまなアプリケーションの実現をはかろうとしている．そのベースとなっているのが仮想化ノード・プロジェクトにおいて開発されているネットワーク仮想化技術と仮想化ノードである．このプロジェクトは情報通信研究機構 (NICT) を中心とし，東大，NTT，NEC，富士通，日立が協力してすすめている．このプロジェクトにおいては，独立かつ自由に設計された機能を実装した複数の仮想ネットワークがひとつの物理ネットワーク上で同時に動作できる環境を実現することをめざしている [Nak 10b]．実装されるべきネットワーク機能としては IP にかわるプロトコルやアプリケーション依存の機能

などがある．ここで開発された仮想化ノードは 2010 年度に研究開発用テストベッド・ネットワーク JGN2plus に導入され，新プロトコルなどの研究開発にひろく使用できるようになる予定である．

この報告においては，この仮想化ノードの試作機によって構成された実験ネットワーク上で単純で汎用性のある非 IP プロトコルを確立するための第 1 歩として開発した，Ethernet と IP の利点をあわせもつ実験用の非 IP プロトコル IPEC (IP Ether Chimera) について報告する．なお，この開発の過程においてえられたプロトコル開発に関する経験的知識などについては別途報告する [Kan 10]．

このプロトコルの開発目標はつぎのとおりである．第 1 に，新プロトコルの研究に関する目標は，単純で汎用性のある非 IP プロトコルの確立をめざした研究にふみだすことである．この目標はさらに 2 つの副目標にわけられる．

- IP 上で実現すると多層化し複雑化する機能を，Ethernet や IP の長所をあわせもつ 1 層の単純な非 IP プロトコルにより実現する．
- Ethernet スイッチの学習アルゴリズムを拡張し，ループをふくむ任意の構造のネットワークにおいて使用可能な転送アルゴリズムを実現する．

第 2 に，仮想化ノードの開発に関連した目標は，仮想化ノード使用のネットワーク上で非 IP の新プロトコルが開発でき動作するのを実証することである．すなわち，仮想化ノードによって構成されたネットワークの動作検証とユーザビリティの検証をおこない，仮想化ノードが新プロトコルの実験に適していることをデモし，今後の仮想化ノード使用による新プロトコル開発のテストケースをつくる (開発者にノウハウを提供する) ことをめざしている．

この報告はこれらの目標のうちプロトコル研究の面に焦点をあてる．まず 2 章では NICT の仮想化ノードを中心として仮想化ネットワークと仮想化ノードについて説明し，3 章では実験用プロトコル IPEC の仕様を説明する．4 章では今回開発した IPEC の実装にもとづく実験について説明し，5 章において結論をのべる．

## 2. 仮想化ネットワークと仮想化ノード

この章においては仮想化ネットワークとそれを構成する仮想化ノードについて説明する．

### 2.1 仮想化ネットワーク

コンピュータやネットワークにおけるかぎられた資源を複数のユーザや複数のシステムで共用するとき，それぞれがあたかも他から隔離された固有の資源を使用しているようにみせるのが仮想化技術である．仮想化技術はまずコンピュータ本体における記憶の仮想化や CPU などの分割使用 (タイムシェアリング) の技術として発展してきたが，最近は仮想マシン (VM) というかたちでコンピュータじたいが仮想化されている．

ネットワークに関しては VPN (Virtual Private Network) による WAN の仮想化がすすめられてきた．すなわち，ひとつのネットワークを複数の企業などで共用しながら，あたかも専用線であるかのように，安全かつ快適に使用できる環境がつくられてきた．最近ではデータセンタ内のネットワークも VM と連携しつつ仮想化されてきている．新世代ネットワークの研究においては，こうしたネットワーク仮想化技術を発展させて，あたらしいプロトコルを他のユーザに影響をあたえることなしに開発できる環境 (ユーザの組織ごとにきちんとアイソレートされた環境) をつくることがもとめられている．

このような潮流のなかで NICT においては上記のような仮想化ネットワークを高性能かつもっとも完全なかたちで実現することをめざした仮想化ノード・プロジェクトを開始している．仮想化ノード・プロジェクトにおいてはまず研究者が自由にプロトコルを設計できる環境をつくることを目標としているが，さらにはそれを商業的展開も可能なようにすることもめざしている．

ネットワーク仮想化においては，仮想化前のネットワークと仮想化後のネットワークとが共存する．仮想化前の下層のネットワークを**仮想化ネットワーク** (virtualized network) と呼び，仮想化後の上層のネットワークを**仮想ネットワーク** (virtual network) とよぶ．

### 2.2 仮想化ノード・プロジェクトにおける仮想化ネットワーク

仮想化ネットワークに関してはすでにさまざまな研究がおこなわれ，さまざまなモデルが提案されている．そのなかには，PlanetLab [Pet 02] [Tur 07]，VINI [Bav 06]，GENI [GEN 09]，Genesis [Kou 01] などがある． 仮想化ノード・プロジェクトにおけるモデル [Nak 10b] は仮想ネットワークの管理に重点をおいている．

このモデルにおいては，物理的なネットワーク (**図 2.1**) は 1 個または複数個のドメインによって構成される (ただし，2010 年現在はドメイン 1 個だけで構成される)．各ドメインはドメイン・コントローラ (DC) によって管理される．ドメインのなかには仮想化機能をもつつぎの 2 種類のノードが存在する．

- **仮想化ノード** (VNode): 仮想ネットワークにおける中継機能をもつノード．
- **アクセス・ゲートウェイ** (AGW): 仮想ネットワークとユーザ端末やユーザのネットワークとのあいだの中継機能をもつノード．

また，ドメインのなかには仮想化機能をもたない通常のルータやスイッチがふくまれていてもよい．VNode どうしは GRE (Generic Routing Encapsulation) のようなプロトコルを使用したトンネルによってむすばれるため，途中のルータやスイッチに依存せず，自由なトポロジーをもつネットワークを構成することができる．仮想化ノードとしても従来のルータやスイッチの機能を拡張したものを使用することができるため，既存のネットワークを拡張するかたちで配備 (deploy) することができる．

各 VNode はつぎの 3 種類の構成要素によって構成されている．

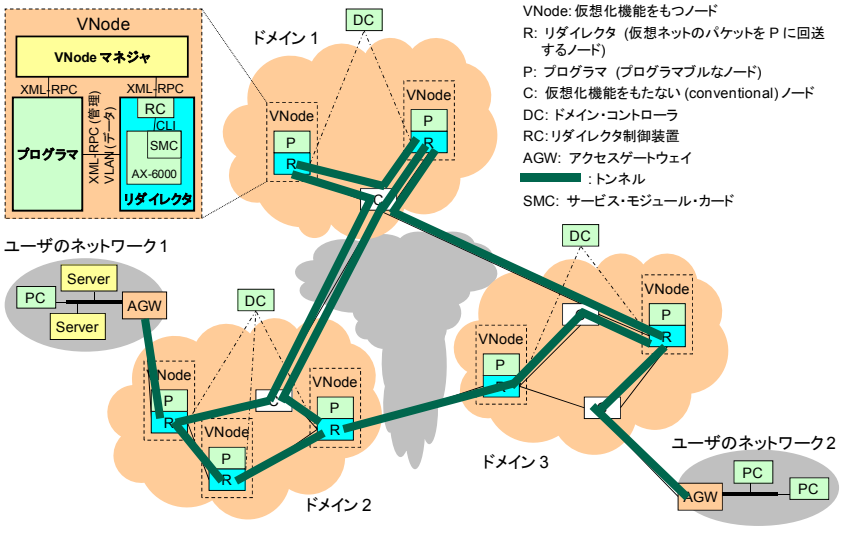


図 2.1 仮想化ノード・プロジェクトにおけるネットワークの物理構成

- **プログラマ** (Programmer): パケットに対する処理をおこなう構成要素である．ハードウェアとソフトウェアとで構成される．パケットに対する処理を仮想ネットワークの開発者がプログラムできるため，プログラマとよばれる．1 個の仮想化ノードのなかに複数個存在することができる．
- **リダイレクタ** (Redirector): 通信データ (パケット) を他の仮想化ノードや他の種類のノードから受信したり，それらに通信データを送信したりする転送機能をもつ構成要素である．
- **VNode マネジャ** (VNode Manager): 仮想化ノード全体の管理をおこなう構成要素である．1 個の仮想化ノードのなかに 1 個だけ存在する．通常，ソフトウェアだけで構成される．ネットワーク全体を管理する管理サーバであるドメイン・コントローラ (DC) からの指示にもとづいて動作する．

また，このモデルにおいては，PlanetLab にならって，仮想化ネットワーク上につくられる仮想ネットワーク (または仮想ネットワークの構成要素の集合) を**スライス** (slice) とよぶ．スライスは複数の仮想ノードとそれらをつなぐ仮想リンクとで構成される (**図 2.2** 参照) が，このモデルにおいてはこれらをつぎのようによぶ [Nak 10a].

- **ノードスリバー** (Node Sliver): 1 個の仮想化ノードのなか (プログラマのなか) に存在する計算資源である．プロトコル処理やノード制御などを実行するのに使用する．大別すると，Linux (Ubuntu 9.1) を搭載した VM である**スローパス** (slow path) と，ネットワーク・プロセッサによる**ファストパス** (fast path) の 2 種類がある．
- **リンクスリバー** (Link Sliver): ノードスリバー間を結合する仮想リンクを意味する．通常はことなる物理ノード内にあるノードスリバーを結合する．リンクスリバーは複数の VNode や AGW にまたがって存在し，リダイレクタが設定し資源管理するトンネルによって実現される．

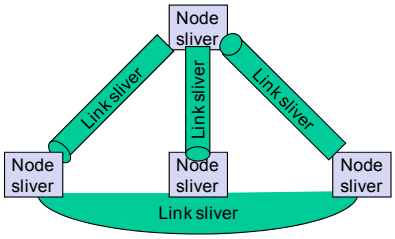


**図 2.2 仮想ネットワークの論理構成 (スライスの構造)**

## 3. 実験用プロトコル IPEC

この章においては新規開発したプロトコル IPEC (IP Ether Chimera) について説明する．

### 3.1 設計方針

この節においては IPEC の設計方針についてのべる．我々は VNode の開発にたずさわり，そのテストとデモのためのプログラムを開発する必要があった．とくに我々にあたえられた課題は非 IP プロトコルのテストおよびデモをおこなうことだった．開発期間は 5 カ月程度あったが，開発にかけられる人員はかぎられていた．そのため，1 章においてのべたように，第 1 の目標は Ethernet と IP の長所をあわせもつ 1 層の単純な非 IP プロトコルを開発することとした．

また，IP においては IP じたいとはべつにルーティング・プロトコルがつかわれるが，IPEC においては単純化するためすべての処理をひとつのプロトコルの処理としておこなうことにした．そのため，第 2 の目標はループをふくむ任意の構造のネットワークにおいて使用可能な，学習にもとづく転送アルゴリズムを実現することとした．

以下，より詳細な設計方針を記述するまえに，Ethernet と IP の長所と短所をかんたんにのべる．Ethernet においてはアドレス (ホスト ID) が構造のない識別子としてあつかわれ，階層構造は存在しない．そのため，転送においてはアドレスが個別にあつかわれ，転送アルゴリズムはもっとも単純であり，補助的なプロトコルなしに転送できる．その反面，スケーラビリティはひくい．また，ネットワーク構造上ループが存在すると転送アルゴリズムだけでは対応できず，ループにそってパケットのコピーが爆発的に生成される．そのためループをなくす必要があり，ネットワーク構造が制約される．

これに対して IP においてはアドレスが順序づけられているため，サブネットのような階層構造をつくることができる．アドレスは単純な 2 進数であるから，階層はネットワーク設計者が自由にきめるられる．IP ルーティングにおいては Ethernet の転送とはちがってアドレスを集約することができ，転送テーブルの項目数をへらせるため，スケーラブルである．また，ネットワーク構造上ループがあってもかまわないし，それによって障害につよくなっている．しかし，IP による転送のためにはルータに繁雑な設定 (静的ルーティングの設定) が必要になる．これを手動で設定するのは現実的でないし，手動ではネットワーク構成の変化に対応するのは困難になるため，IP そのものとはべつにルーティング・プロトコルが必要であり，複雑化する．

このような長所・短所をかんがえて，前記の目標をさらに以下のような方針に展開した．

- **階層的なアドレスと学習の適用**: Ethernet よりスケールさせるため，IP のように順序のあるアドレスを使用し，階層構造をもたせられるようにした．しかし，ルーティング・プロトコルを導入して転送機構を複雑化すると短期開発は困難になるため，すべてをデータパケットから学習することにした．Ethernet は開発当初は単純なリピータ・ハブとともに使用されたが，現在は通常，学習機能をもつスイッチによってパケット転送している．この学習機能を拡張して階層的なアドレスに適用するのは興味ぶかい挑戦である．
- **1 種類のアドレスと 2 種類の転送法の適用**: 1 種類の階層化されたアドレスを Ethernet 風，IP 風という 2 種類の方法によって転送できるようにする．たとえば，LAN と WAN とが接続された環境において，LAN においては個別のアドレスを学習しその結果にもとづいて転送するが，WAN においては集約されたアドレスを学習しその結果にもとづいてスイッチング (転送) できるようにする．WAN においても集約しなければ個別に転送することもできるので，端末の移動にも対応できる．
- **ロケータ指定 / 非指定の自由選択**: アドレスを 2 階層にすれば，上位がロケータ，下位がホスト ID と解釈できる．通信相手のホスト ID だけがわかっているときには，とりあえずロケータなしで通信すれば個別学習によって通信することもできるが，ID/Locator 分離方式での通信も可能である．すなわち，サーバにロケータをといあわせることによって，効率的な通信が可能になる．

### 3.2 アドレスとフレームの実装形式

上記の設計方針においてはアドレスの階層化はネットワーク設計時に自由にきめることができるが，今回の実装においては単純化のため固定的な 2 階層とした．すなわち，IPEC においてあつかうアドレスとフレーム (パケット) の形式は**図 3.1** のようにする．アドレス (ながさ 8 バイト) の構造はつぎのとおりである．

- **ホスト ID**: アドレスの下位 (仕様上は可変長だが実装上は 4 バイトに固定している) はホスト ID をふくむ．以下単に ID と称する．ID は不可分な値 (すなわち構造がない) である．
- **グループ ID**: アドレスの上位はグループ ID すなわち複数個または 1 個のホストによって構成されるグループの ID をふくむ．グ

ループは階層化できるが，今回は階層はかんがえない．以下単に**グループ**と称する．

グループはロケータと解釈することもできる．

また，フレーム・ヘッダは 22 バイトあり，上位から順につぎのフィールドから構成されている．

- **フレーム長**: フレーム・ヘッダとペイロードのながさの総和である．
- **受信者アドレス**: フレームを受信するべきホストのアドレスを指定する．上記のアドレス形式をしている．
- **送信者アドレス**: フレームを送信したホストのアドレスを指定する．上記のアドレス形式をしている．
- **送信者グループ ID 長**: グループ ID を可変長とするときはそのながさを指定する．現在は 32 (32 ビット＝4 バイト) に固定される．
- **年齢**: スイッチ間でパケットが転送されるごとに，1 ずつ増加する．ループの存在により重複したパケットの廃棄に使用される．

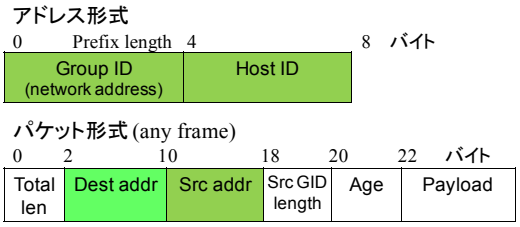


図 3.1 プロトコル・フォーマット

## 3.3 仮想ノードの転送・学習アルゴリズム

仮想化ノード・プロジェクトの仮想ネットワークへの IPEC 実装に即して説明をするため，**図 3.2** に例として後述する実験において使用したスライスの構成をしめす．実験におけるスライス構成については 4 章においてより詳細に説明する．

今回の実装においては，VNode 上の各ノードスリバーは IPEC の WAN 機能を実現する．すなわち，グループの転送と学習のアルゴリズムを実装する．VNode においては ID は参照しない．ノードスリバーは 2 個以上の任意個のポート (仮想インターフェイス) をもち，すべてのポートについておなじ処理 (対称な処理) をする．いずれかのポートにパケットが到着すると，基本的には以下で説明する学習と転送のアルゴリズムが順に実行される．

学習アルゴリズムにおいては送信者グループ (source group) の学習とパケット廃棄をおこなう．

> **if** 到着パケットの src group が転送テーブルに登録されていない **then**
> 転送テーブルに group, group length, input port, age を登録する (学習する);
> **else if** 登録要素の age > 到着パケットの age **or** 登録要素が「登録タイムアウト」している **then**
> 登録要素の age, port = 到着パケットの age, port;
> 登録要素の タイムスタンプ = 現在の時刻 (ns);
> **else if** 登録要素の age < 到着パケットの age **or** 登録要素の port != 到着パケットの port **then**
> パケットを廃棄する (転送アルゴリズムを実行しない);
> **else** 登録要素の タイムスタンプ = 現在の時刻 (ns);

学習結果は転送テーブルに記録される．学習情報は時間がたつと忘却されるが，そのためのタイムアウトには 2 種類あり，タイムアウト時間は独立に設定できる．ひとつは**登録タイムアウト**である．登録タイムアウトするまでは重複して到着したパケットを廃棄するので，ネットワークに**ループ** (複数の経路) があっても通常はちょうど 1 個だけパケットが転送される．しかし，タイムアウトすると重複したとはみなさない．これは，障害に対応するためである．すなわち障害時には代替経路で通信できる．もうひとつの参照タイムアウト時間については，転送アルゴリズムの説明のなかで説明する．登録内容にしたがってパケットが転送されたときや，登録タイムアウトして登録内容が更新されたときは，その登録要素のタイムスタンプを更新する．

転送アルゴリズムにおいては受信者グループ (destination group) にもとづいて転送する．

> **if** 到着パケット の dest group が転送テーブルに登録されていない **or** 登録要素が「参照タイムアウト」している **then**
> 到着パケットの age を増加したものをフラッドする;
> **else** 登録要素の port にだけ，到着パケットの age を増加したものを出力する;

ここではもうひとつのタイムアウトである**参照タイムアウト**が使用される．参照タイムアウトが発生すると既存の登録要素は無効になり，パケットはフラッド (flood) される．すなわち，パケットがとどいたポート以外のすべてのポートから同一のパケットが出力される．登録されていないときも参照タイムアウトが発生してフラッドされる． 参照タイムアウトは Ethernet スイッチにおけるタイムアウトに相当する[1]．

このアルゴリズムのひとつの問題点は，1 個のパケットが重複して配送される場合があることである． Ethernet とはちがって重複したパケットは通常は廃棄される．しかし，フラッドによって重複したパケットが

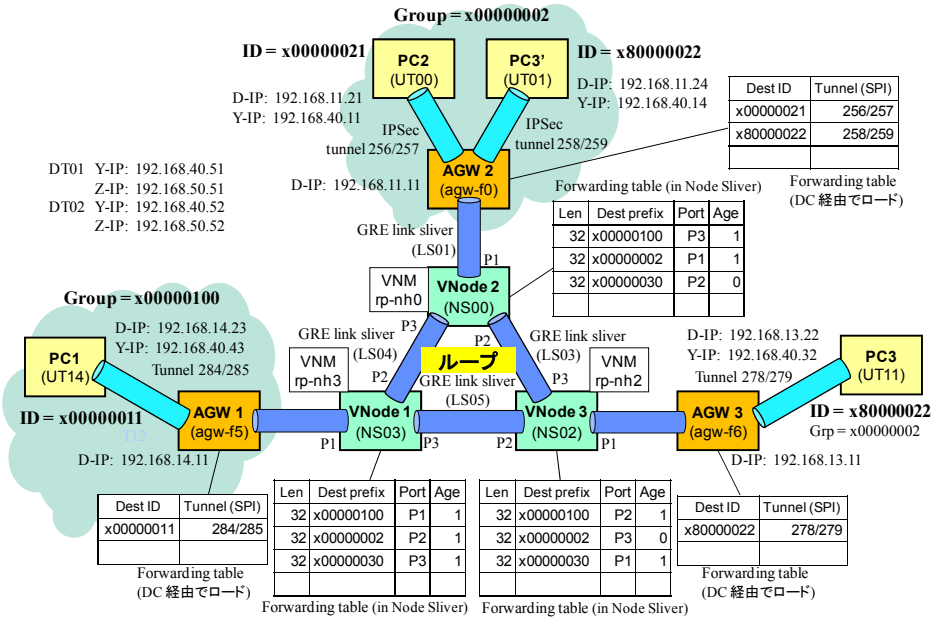


図 3.2 スライス構成 (学習後の状態)

[1] 参照タイムアウトが発生したときは本来は登録要素を削除するべきだが，現在はアルゴリズムを単純化するために削除はおこなっていない．また，上記のアルゴリズム記述においては，実際のプログラムではおこなっているテーブルあふれの検査を省略している．

1 個の VNode にとどき，しかもホップ数のすくない経路をとおったもの (Age がわかいもの) があとでとどくと，重複したまま転送される．ネットワークの輻輳時にこのような現象の発生をふせぐのは困難だが，重複が発生することはまれだとかんがえられる．ホップ数がすくない経路のほうが距離がながくて遅延がおおきいときは，上記のアルゴリズムにおもみを導入すれば (パケットに移動距離を記録すれば) 重複をなくせる．なお，このアルゴリズムは，周辺部分をのぞけば C で記述しても 100 行程度である．

### 3.4 AGW の設定

今回の実装においては，AGW は IPEC の LAN 機能を実現する．前章の方針にしたがえば，LAN においては ID を学習するべきである．しかし，現在の AGW の機能をいかすため，学習アルゴリズムをくみこむかわりに，ID と端末識別子 (具体的には IPsec の SPI の値 (図 3.2 参照)) との対をあらかじめ設定しておくことにした．

## 4. 実験

NICT 白山リサーチラボに設置された仮想化ノードを使用して IPEC に関する実験をおこなった．

### 4.1 スライス構成

実験のために，3 個の VNode，4 個の AGW のうえに生成した図 3.2 のような構成のスライスを使用した．図にはこのスライスに接続した端末 (Linux PC) も記述し，また VNode や AGW がもつテーブルの内容も記述している．ただし，タイムスタンプなど，一部のテーブル要素は省略している．VNode がもつ転送テーブルは初期状態においては空だが，この図には学習後の状態を表示している．AGW がもつ転送テーブルの内容は端末登録の際に登録される．

スライス定義には物理装置名と論理装置名の両方が記述され，それらの対応も記述される．この図にもその両方を記述している．たとえば，論理名 AGW1 は物理名 agw-f5 に対応している．また，ノードスリバー NS00 は物理 VNode 2 内にある．

### 4.2 ノードスリバー用と端末用のプログラム

ノードスリバー用と端末用のプログラムでは，自由な形式のパケットを送受信するために Linux の promiscuous mode を使用している．そのプログラミングについては他の報告 [Kan 10] に記述する．今回の実験はデモ用をかねたプログラムによっておこなったため，ノードスリバー上で動作するプログラムは高速転送モード以外にデモ・モードで動作する．このモードにおいては，パケットが到着するごとにそれがスイッチされたかフラッドされたかパケットが廃棄されたかなどの情報が表示される．また下記の端末用プログラムが出力するパケットにはペイロードにシーケンス番号がふくまれるが，デモ・モードにおいてはその値も表示される．

端末用プログラムもデモ用と実験用をかねている．端末とノードスリバーのプログラムの出力がみやすいように 2 秒ごとにパケットを送信し，それを受信するとその内容を表示する．パケットの送信をとめて受信専用にすることもできる．

### 4.3 フラディングの実験

2 端末間で双方向に通信すると，ただちに学習してスイッチする．そのため，フラディングの動作をみるためには片方向の通信を継続的におこなう．すなわち，前記の端末用プログラムを一方は送受信モード，他方は受信モードで動作させる．ただし，IPEC のあるべき用法においてはパケット受信者はすぐ応答するべきであり，また応答がなければパケット送信を停止するべきである．したがってこのようにフラディングが継続する非効率な通信は実用上はおこらない．

**図 4.1** には図 3.2 における PC1 から PC3 に片方向の通信をおこなったときの各ノードスリバーの出力を表示している．PC1 から出力されたパケットは AGW1 を経由して VNode1 に到達する．VNode1 はまだ学習していないためフラディングが発生している．そのため，左下のウィンドウに “Flood the packet” と表示されている．フラディングによって同一のパケットが VNode2 と VNode3 に送信されている．VNode3 にとどいたパケットは同様にフラディングによって AGW3 と VNode2 に送信されるが，AGW3 にとどいたパケットは PC3 にとどいて表示される．VNode3 には VNode2 からもパケットがとどく (同一のパケットがとどいていることは，図 4.1 にシーケンス番号 68 が 2 回表示されているのでわかる) が，重複パケットは廃棄される．廃棄されたことは右下のウィンドウ上の “Packet being dropped (port)” というメッセージにより確認できる．VNode2 にもパケットが重複してとどいているが，こちらもパケット廃棄を確認できる．AGW2 にとどいたパケットは AGW2 に接続されたどの端末の ID とも一致しない受信者 ID をもっているため廃棄される．端末出力はここには表示しないが，端末にとどいていないことは受信モードのプログラムを動作させれば確認できる．

### 4.4 単純なスイッチングの実験

スイッチングは双方向の通信をおこなうことでかんたんにためすことができる．**図 4.2** には PC1 と PC3 とが双方向の通信をおこなったときの各ノードスリバーの出力を表示している．VNode1 と VNode3 の出力に “Switch packet …” というメッセージがよみとれる．VNode2 の出力はこの通信よりまえに表示されたものであり，この通信においてはなにも表示されない (この通信の出力でないことはシーケンス番号がちがうことからわかる)．すなわち，パケットは VNode1 と VNode3 とのあいだだけでスイッチされている．

### 4.5 グループ単位の学習によるスイッチングの実験

図 3.2 の構成においては AGW2 に同一グループ x2 に属する 2 個の端末が接続されている．PC1 と 2 個のうちの 1 個である PC3’ とのあいだで双方向通信して VNode1，VNode2 を学習させた直後に PC1 から 2 個のうちのもう 1 個である PC2 に片方向通信すると，

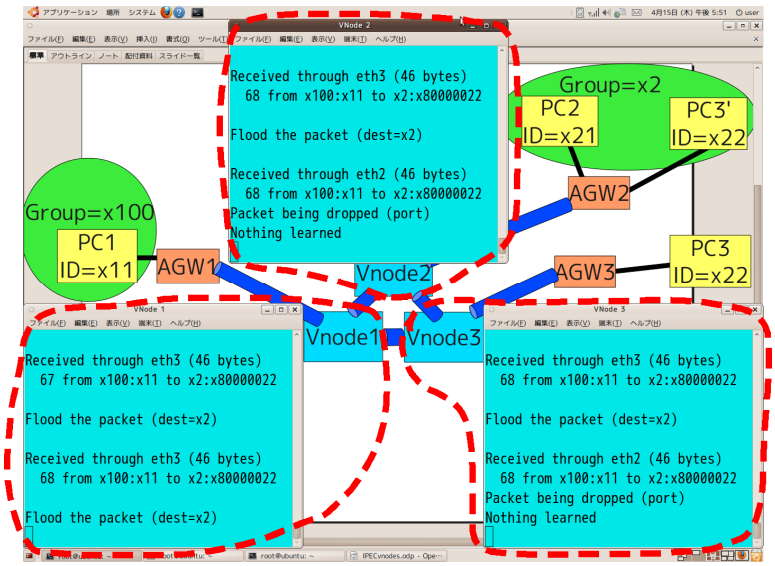


**図 4.1 フラディング時のノードスリバー出力**

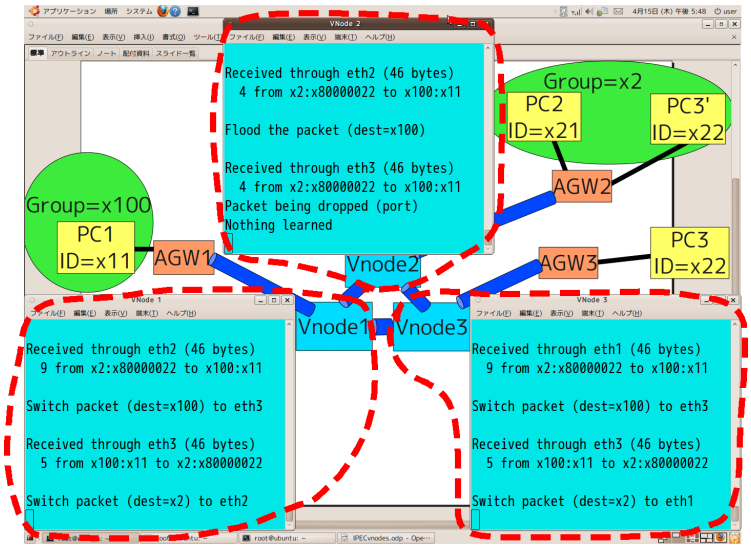


図 4.2 スイッチング時のノードスリバー出力

VNode1, VNode2 がただしくスイッチすることがたしかめられた．このように動作するのは，学習がグループ単位でおこなわれ，かつ PC2, PC3' が同一のグループに属しているからである．

### 4.6 端末の移動とグループ単位の移動の実験

現在の仮想化ネットワークにおいては AGW が学習機能をもたないため，端末をべつの AGW につなぎかえるポータルと端末の双方に設定変更が必要になる．そのため，端末を移動させるのは実用上困難であり，またデモのように時間がかぎられているときにも移動は困難である．しかし，2 つの AGW に同一のアドレスをもつ端末を接続し，どちらか一方だけで送受信プログラムを動作させることによって，モバイル端末をシミュレートすることはできる．

図 3.2 の構成においては PC3 と PC3' に同一のアドレスをあたえている．そこで，まず PC1-PC3 間で双方向通信し，つぎに PC1 から PC3' への片方向通信をこころみた (PC1 は同一の動作を継続するだけである)．VNode1, VNode3 が学習しているため VNode2 にはすぐにはパケットが到達せず，PC3' はパケットを受信できない．しかし，参照タイムアウトをまてばフラディングがおこり，PC3' との通信が可能になるのを確認した．もちろん，参照タイムアウトのまえに PC3' がパケットを送信して VNode を学習させれば，よりはやく通信可能になるはずである．

今回は使用可能な端末用 PC が 4 個だけだったため実際にはためせなかったが，AGW3 にも 2 個の端末をつなげれば，これらを同時に AGW2 のもとに移動させた (移動をシミュレートした) ときには，1 回の学習で両方の端末への通信が可能になるはずである．

### 4.7 広域での実験とデモ

上記の実験はいずれも NICT 白山リサーチラボラトリ内でおこなったが，はじめての広域での実験を 6 月 7 ～ 11 日に幕張においてひらかれた Interop Tokyo 2010 において実施した．図 3.2 における 3 台の仮想化ノードのうち 2 台を幕張に設置し，もう 1 台は白山に設置したものを使用して，前節まででのべたのとほぼおなじ通信実験をおこなった．この環境でもおなじ実験結果がえられた．仮想化ノードは今後，実験用ネットワーク JGN2plus に導入されることになっているので，今後はさらに広域での実験をおこなっていきたい．

また，8th GENI Engineering Conference (GEC8) においては，中尾 [Nak 10c] が仮想化ノードの応用例のひとつとして IPEC を紹介し，デモビデオを Web に掲載している．

## 5. 結論

この研究開発においては非 IP プロトコル確立にむけた第 1 歩をふみだすことを一目標としてきたが，結果としてつぎのような特徴をもつ非 IP プロトコル IPEC を開発することができた．

- Ethernet, IP それぞれの特徴的な機能の一部を 1 層の単純な非 IP プロトコルによって実現した．
- Ethernet スイッチの学習アルゴリズムを拡張して，ループをふくむネットワークで使用でき障害にも対応できる方法を実現した．
- 学習をグループ単位でおこなうため，Ethernet よりスケールする．また，グループ単位の移動が効率的に学習できる．

グループはロケータとしても使用できるため，ID/Locator 分離を拡張したアーキテクチャを実現しているということができる．

IPEC を VNode 上に実装して，グループ単位の学習や端末の移動に実際に対応できることを実験により確認した．今回の実験ではノード機能の実現のためにスローパスすなわち汎用 CPU (x86, x86-64) を使用している．しかし，実用レベルにちかづけるにはファストパスすなわちネットワーク・プロセッサを使用して同一のアルゴリズムをより高速に実現することがひとつの課題となる．VNode はすでにファストパス機能をもっているので，これはすぐに実験可能になっている．

## 謝辞

このプロトコルの開発にあたって NEC の 元木 顕弘 氏，富士通の 渡辺 紀一 氏，アラクサラの 木谷 誠 氏ほか，NICT 仮想化ノード・プロジェクト参加者各位の意見や助力をいただいたので感謝する．

## 参考文献

[Bav 06] Bavier, A., Feamster, N., Huang, M., Peterson, L., and Rexford, J., "In VINI Veritas: Realistic and Controlled Network Experimentation", *2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications* (*SIGCOMM'06*), pp. 3–14, 2006.

[GEN 09] The GENI Project, "Lifecycle of a GENI Experiment", GENI-SE-SY-TS-UC-LC-01.2, April 2009, http://groups.geni.net/geni/attachment/wiki/ExperimentLifecycleDocument/ExperimentLifeCycle-v01.2.pdf?format=raw .

[Kan 10] 金田 泰, "NICT 仮想化ノードを使用した非 IP プロトコル開発法と経験", 電子情報通信学会発表予定.

[Kou 01] Kounavis, M., Campbell, A., Chou, S., Modoux, F., Vicente, J., and Zhuang, H., "The Genesis Kernel: A Programming System for Spawning Network Architectures", *IEEE J. on Selected Areas in Commun.*, vol. 19, no. 3, pp. 511–526, 2001.

[Nak 10a] Nakao, A., "Network Virtualization as Foundation for Enabling New Network Architectures and Applications, IEICE Trans. Commun., Vol. E93-B, No. 3, pp. 454–457, March 2010.

[Nak 10b] 中尾 彰宏, "ネットインフラを用途別に "スライス" 柔軟な機能拡張の実現に効果", 日経コミュニケーション, June 2010.

[Nak 10b] Nakao, A., "Update on CoreLab and VNode", http://groups.geni.net/geni/wiki/Gec8Agenda .

[Pet 02] Peterson, L., Anderson, T., Culler, D., and Roscoe, T., "A Blueprint for Introducing Disruptive Technology into the Internet", *ACM SIGCOMM Computer Communication Review*, Vol. 33, No. 1, pp. 59–64, January 2003.

[Tur 07] Turner, J., Crowley, P., Dehart, J., Freestone, A., Heller, B., Kuhms, F., Kumar, S., Lockwood, J., Lu, J.,Wilson, M., Wiseman, C., and Zar, D., "Supercharging PlanetLab - High Performance, Multi-Application, Overlay Network Platform", *ACM SIGCOMM Computer Communication Review*, Vol. 37, No. 4, pp. 85–96, October 2007.