情報処理学会論文誌　Vol. 52　No. 3　1291–1307 (Mar. 2011)

複数の実装法に展開可能な仮想ネットワーク・モデルとその広域 VM ライブ・マイグレーションへの適用

金田　泰[†1]　垂井俊明[†1]

ネットワーク仮想化は，1 つの物理的なネットワーク上で多様なサービスを相互に干渉せず実現し，かつ単純にプログラムしカスタマイズすることを可能にする．ネットワーク仮想化は，IP，VLAN，あるいは NAT を使用するなど，多様な機構によって実現できる．我々はこれらの実現手段に依存しない統一的な仮想ネットワークのモデルをつくり，そのモデルによって記述した仮想ネットワークを基本的に変更せずに多様な実現手段のネットワーク上に展開可能にすることをめざしている．この論文では仮想ネットワークのモデルとその主要な部品を定義し，それによって広域ライブ・マイグレーションのための仮想ネットワークを記述し，さらにそれをネットワーク・パーティション（VLAN による VRF）とアドレス・マッピング（NAT）という 2 つの実現方法による仮想ネットワークに展開できることを示す．この方法を仮想ネットワーク設定プロトタイプを新規開発して評価した．その結果，アドレス範囲指定やルーティング・パラメータの指定は実現手段ごとに変更する必要があったが，モデル記述の主要部分は共通化できた．

A Virtual Network Model that Derives Multiple Implementation Methods and the Application to Wide-area VM Live Migration

Yasusi Kanada[†1] and Toshiaki Tarui[†1]

Network virtualization achieves mutually non-interfering various network services on one physical network and enables the network to be programmed and customized easily. Network virtualization can be realized by various mechanisms such as IP, VLAN, or NAT. We developed a unified virtual network model that does not depend on such specific mechanisms and deployed the model on various tools without modifying it. This paper defines a virtual-network model and major parts of the model, describes a virtual network for wide-area live migration, and shows that the model can be expanded to two virtual networks using different implementation methods, i.e., network partition (VRF by VLAN) and address mapping (NAT). We have evaluated this method using a newly-developed virtual-network configuration prototype. The results show that although address ranges and routing parameters must be specified in accordance with each mechanism, most parts of the model descriptions could be unified.

1. はじめに

ネットワーク仮想化によって，同一のハードウェアすなわち同一のコンピュータやネットワークのノードやリンクを使用しながら，複数のサービスやコミュニティが他のサービスやコミュニティに干渉されることなく通信やサービスができるように，隔離（アイソレート）することができる．ネットワーク仮想化によって各ユーザは仮想的に独自のネットワークを持つことができる．仮想ネットワークはカスタマイズ可能であり，プログラマブルである．仮想ネットワークの開発者は従来のインターネット・プロトコル（IP）に基づく従来の複雑な機能を排除して，もっと単純な仮想ネットワークをつくることができる．開発者は IP−− [14] のような単純化された IP 風のプロトコルを使用することもできるし，単純で強力かつ効率的な非 IP プロトコルを導入することもできる．それによって，複雑化する IP ネットワーク上では提供することが困難だった新規の多様なサービスを，サービス・プロバイダは仮想ネットワークを使用して提供できるようになると考えられる．

ネットワーク仮想化はさまざまな方法によって実現することができる．IP 上に IP や他のプロトコルをかさねて実現することもできるし，VLAN を使用して，あるいは MPLS を使用して実現することもできる．これらの方法にはそれぞれ長所と短所があり，使い分ける必要があるが，1 つの仮想ネットワークを他へ自由に移植できるようにすることがのぞましい．そのための方法として，これらの実装形態によらない統一的な仮想ネットワークのモデルをつくり，そのモデルに従って記述した仮想ネットワークが基本的には変更せずにさまざまな実装形態のネットワーク上に実現できるようにする方法がある．

一方，クラウド・コンピューティング環境などにおいて，仮想マシン（VM）を停止させることなくデータセンタ間で移動させる広域ライブ・マイグレーションの実現が重要な課題になってきている．クラウド・コンピューティングにおいてはユーザは CPU，メモリ，ストレージなどの計算資源がネットワーク上のどこにあるかを意識せずに利用することができ

†1 株式会社日立製作所
Hitachi Ltd.

る．そのため，計算に VM を使用し，ユーザには VM を継続的に使用しているようにみせながら VM を異なる地域にあるデータセンタ間で移動させることができる．それによって，負荷分散，障害回避，省電力などの全体最適化が実現される可能性がある．しかし，2 章において詳細にのべるが，遠地点間でのライブ・マイグレーションには，さまざまな課題を解決する必要がある．1 つの課題は，VM 移動後の設定変更に時間がかかって通信できなくなる時間を短縮することである．この課題を解決するために，我々はデータセンタ単位に複数の仮想ネットワークを用意し，VM 移動時にそれを切り替える方法を提案する．この方法においてはさまざまな実装形態の仮想ネットワークを使用することが可能であり，それらを統一的なモデルであつかうことによって，最適な仮想ネットワークの種類や方法を選択するのが容易になると考えられる．

この論文においては，Nakao（中尾）[12] による，ノードスリバーとリンクスリバーを基本要素とする仮想ネットワーク基本モデルをベースとして，VM の広域ライブ・マイグレーションの際に混乱なく経路を切り替えるために必要な仮想ネットワーク・モデルの要素を定義し，それを 2 種類の仮想ネットワークすなわち VLAN によりネットワーク・パーティション[1] に基づくそれとアドレス・マッピング（NAT）に基づくそれとに展開する方法を示す．

以下，2 章において 2 つの経路切替え法を使用した広域ライブ・マイグレーション方式について説明し，それらをネットワーク仮想化によって統一的に理解する指針をあたえる．3 章においてはそれらのマイグレーション方式を統一的に記述するためのしかけとして仮想ネットワークのモデルとその構成要素を定義する．4 章においてはそのモデルによってマイグレーション方式を統一的に記述する．5 章においてはそのモデルに基づいてマイグレーション方式を仮想ネットワークに展開する方法を説明し，6 章においてそれをプロトタイプにおいて実現した結果を示し，7 章で関連研究に言及したのち，最後に結論をのべる．

2. VM マイグレーションと 2 つのネットワーク仮想化方式

この章においては，経路の切替えによってライブ・マイグレーションを広域で実現するための 2 つの方式を示し，ネットワーク仮想化の 2 つの方式をそれらと対応づける．これらを統一的にモデル化することをめざす．

2.1 広域ライブ・マイグレーションの課題と 2 つの VM マイグレーション方式

1 章でのべたように，広域ライブ・マイグレーションを可能にするためには，さまざまな課題を解決する必要がある．1 つの課題はアドレスが突然，遠隔地に移動することによって発生する問題を解決することである．VM のマイグレーションにおいて，移動する VM のアドレスは IP アドレス，MAC（Media Access Control）アドレスともに基本的に変化しない．そのため，広域ネットワーク（WAN）上の異なる地点へのマイグレーションはネットワークを混乱させる可能性がある．すなわち，VM 移動時には VM の移動そのものにかかる時間だけでなくネットワークの設定変更に時間がかかり，一時的に通信できなくなる危険がある．これがライブ・マイグレーションの際にダウンタイムがながびく 1 つの原因である．

とくに，その WAN においてインターネット・プロトコル（IP）のように場所によって使用可能なアドレスの範囲がきめられたネットワーク・プロトコルが使用されている場合は，通常の IP アドレスを持つ VM が他の地点に移動することはできない．モバイル端末のように移動するホストのためにはモバイル IP という機構が用意されていて，ライブ・マイグレーションにおいてもたとえば Li ら[11] はモバイル IP を使用している．しかし，モバイル IP を使用する方法はオーバヘッドが大きく，ダウンタイムを短縮するのは容易なことではない．

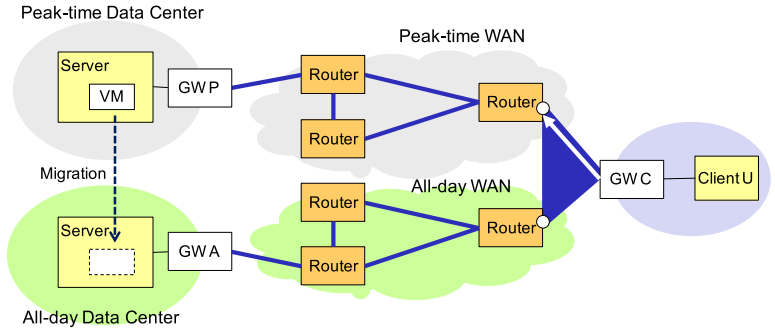
従来の研究[3),11),16)] においては WAN を経由するときは 1 分程度，それを経由しない場合でも 3 秒程度 VM が停止しているが，この研究における将来目標は VM 移動によってリアルタイム・アプリケーションが大きな影響をうけない程度，具体的には 0.1 秒以内に停止時間をおさえることである．ただし，この目標を達成するためにはネットワークの設定変更にかかる時間の短縮だけでなく VM 移動時間の短縮もあわせて必要だが，それはこの論文の範囲外である．

上記のような問題がおこるのは，1 個のネットワークの設定を変更することによって VM の移動に対応しようとしているからである．IP を使用するにせよ Ethernet のように場所によって使用可能なアドレスがきめられていないプロトコルを使用するにせよ，VM が移動すればネットワーク上で非局所的な設定変更が必要になる．すなわち，ネットワーク上の各地点において，移動後の VM との通信が可能になるように設定を変更する必要がある．大規模ネットワーク中でこの設定変更を短時間で行うことは困難である．また，もしそれが可能であるとしても，ネットワークにすくなからず負荷をあたえることになる．

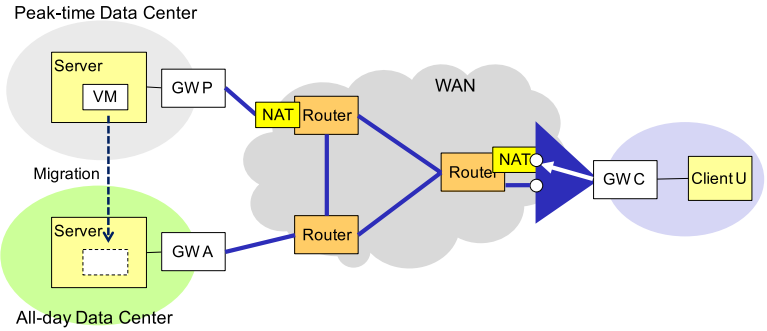
この課題を解決するためにこの論文においては，移動前と移動後の VM が共存可能なネットワークを構築し VM 移動時に必要な設定変更箇所を限定するために，次の 2 つの方法を提案する．

(1) 複数のネットワークを使用する方法

図 1 (a) に示すように，VM 移動前のデータセンタ（図の Peak-time Data Center）に



(a) 複数のネットワークを使用する方法



(b) アドレスを変換する方法

図 1　VM 移動時のダウンタイムを短縮するための 2 方法
Fig. 1　Two methods for reducing downtime on VM motion.

つながるネットワークと VM 移動後のデータセンタ（All-day Data Center）につながるネットワークとを用意し，VM が移動する際に複数のネットワークをスイッチすることによって，通信の中断なしに VM 移動を可能にする方法である．1 個のネットワークの中には同一のアドレスが複数個存在することはできないが，複数個のネットワークを用意すればそれぞれに同一のアドレスの共存が許されることを利用している．物理的なネットワークを複数個用意するのは非効率だが，次節で示すようにネットワーク仮想化によってこれらを物理的には 1 個のネットワークに収容すれば，効率的に実現することができる．

(2)　アドレスを変換する方法

図 1 (b) に示すように，VM 移動前のデータセンタと VM 移動後のデータセンタとが 1 個のネットワークに接続されているとする．この方法は，このネットワークの一方の入口（図 1 (b) においては移動前のデータセンタからの入口である GWP）においてアドレス変換することによって，移動前と移動後の VM がこのネットワーク内では異なるアドレスになるようにマップして共存を許し，すみやかな移動を可能にする方法である．ユーザからは移動前も移動後も VM のアドレスは同一にみえなければならないので，移動前はネットワークの出口においてアドレスを逆変換する．そして，VM が移動する際にアドレスを変換しないようにスイッチする．

2.2　2 つのネットワーク仮想化方式

前記の 2 つの VM マイグレーション方式をともにネットワーク仮想化アーキテクチャに基づくものとして解釈するため，記憶仮想化とネットワーク仮想化とのあいだのアナロジを利用する．すなわち，記憶仮想化の 2 つのアーキテクチャに対応する 2 つのネットワーク仮想化アーキテクチャを示す．

歴史的には仮想化技術はまずコンピュータの記憶仮想化において開発された．2 つの記憶仮想化アーキテクチャの型[20] が開発された．

(1)　セグメンテーション

記憶空間を論理的に分離された可変長のセグメントに分割し，各ユーザが 1 つまたは複数のセグメントを使用するアーキテクチャである（**図 2** (a)）．仮想記憶から実記憶へは物理記憶セグメントの先頭を指示するセグメント・レジスタを使用することによってマップされる．実記憶のアドレスはセグメントの番号（セグメント・レジスタ番号）とセグメント内の変位の組によって表現される．

(2)　ページング

記憶空間を固定長のページに分割し，そのコンピュータのすべてのユーザのページを 1 つの広いアドレス空間にマップするアーキテクチャである（図 2 (b) 参照）．仮想記憶と実記憶とは動的アドレス変換（DAT）を使用することによって相互にマップされる．実記憶のアドレスはこの広いアドレス空間の 1 点を指示する 1 個の数値によって表現される．

記憶仮想化とネットワーク仮想化とのあいだには以下のようなアナロジがなりたつため，上記の 2 つに対応する 2 つの仮想化アーキテクチャがネットワーク仮想化においても存在すると考えることができる．すなわち，記憶仮想化においては記憶装置上のデータに対して実記憶におけるのとは異なる組織構造があたえられているのに対して，ネットワーク仮想化

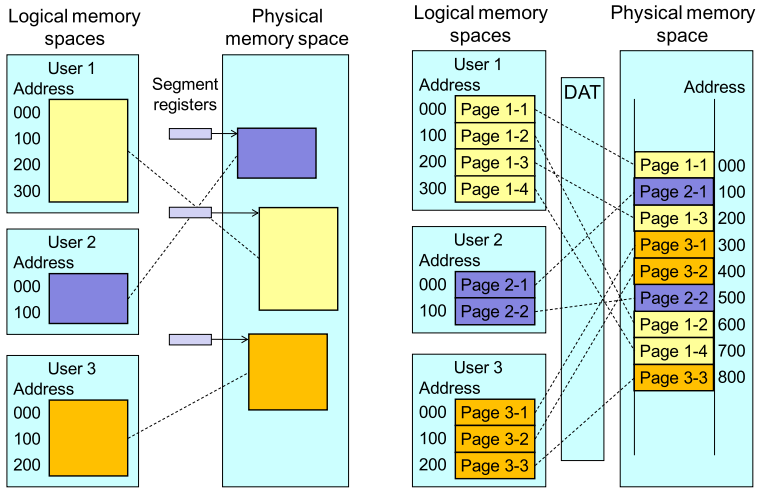


図 2　2 つの記憶仮想化アーキテクチャ
Fig. 2　Two memory-virtualization architectures.

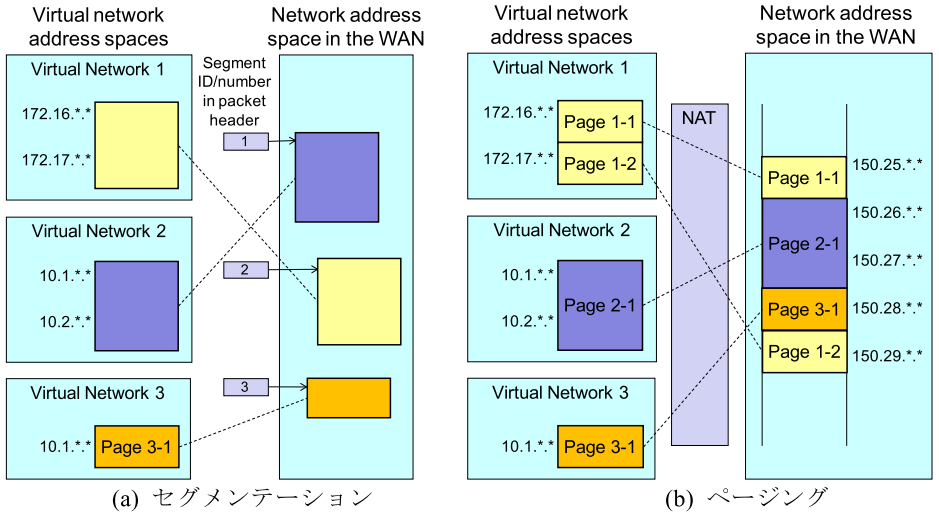


図 3　2 つのネットワーク仮想化アーキテクチャ
Fig. 3　Two network-virtualization architectures.

においてはネットワーク上にある VM などのオブジェクトに対して実ネットワークにおけるのとは異なる組織構造があたえられる．記憶仮想化によって複数の記憶空間が生成されるのと同様に，VM や仮想ネットワーク・ノードや他の種類の仮想オブジェクトが識別される複数のネットワーク・アドレス空間または名前空間がネットワーク仮想化によって生成される．また，メモリ・データもパケットも仮想アドレスによって読み書きされ，メモリ・データとネットワーク・オブジェクトのアドレス形式も類似している．したがって，ネットワーク仮想化には次のような 2 つのアーキテクチャの型が存在すると考えられる．

(1)　ネットワーク・セグメンテーション（カプセル化による仮想化）

ネットワーク内のオブジェクト（コンピュータなど）を仮想ネットワークの識別子（セグメント識別子とよぶ）と仮想ネットワーク内で一意なアドレスや名前（以下，オブジェクト識別子とよぶ）との組によって識別するアーキテクチャである．セグメント識別子としては VPN 番号，VPN 名，VLAN ID などが使用される（**図 3** (a) 参照）．実ネットワークのアドレスはこれら 2 個の識別子の組として表現され，各パケットは送信者と受信者のこの形式のアドレスをふくむ．2 つの仮想ネットワークにおいて同一のオブジェクト識別子すなわちアドレスや名前が使用されたとすると，それらはセグメント識別子がちがうがゆえに識別することができる．この型の仮想化は VPN や実験的な仮想化ネットワークなどでひろく使用されている．

(2)　ネットワーク・ページング（アドレス・マッピングに基づく仮想化）

すべての仮想ネットワークを通じて各オブジェクトに対し一意な 1 個のアドレスをあたえて識別するアーキテクチャである．仮想ネットワークにおけるオブジェクト識別子は WAN のアドレス空間（または名前空間）にかさなりがないようにマップされる．かさなりがなければ，仮想ネットワーク間で干渉が発生しないように隔離することができる．このアドレス・マッピングはネットワーク・アドレス変換（NAT）の一種である．実ネットワークのアドレスはこの 1 個のアドレスだけで表現され，各パケットは送信者と受信者のこの形式のアドレスをふくむ．仮想アドレス空間は複数のページに分割することができ，WAN における 2 個以上の連続していないサブ空間にマップすることができる（図 3 (b) 参照）．ただし，ハードウェアの制限をうけないためページサイズを固定する必要がない点は，記憶の仮想化とは異なっている．2 つの仮想ネットワークにおいて同一のオブジェクト識別子が使用されても，それらは WAN のアドレス空間における異なるアドレスに

マップされるゆえに識別できる．

従来のネットワーク仮想化法はネットワーク・セグメンテーションに基づいている．仮想ネットワークにおける各データ・フレームは下層のネットワーク（substrate network）のパケット・ヘッダによってカプセル化され，セグメント識別子はこのパケット・ヘッダにふくまれる．具体的な方法としては，IP ベースのカプセル化法としての GRE（Generic Routing Encapsulation）[6]，MPLS（Multi-Protocol Label Switching），VLAN に基づく Cisco Systems 社の VRF-lite やアラクサラネットワークス社のネットワーク・パーティション[1] などがある．これらの方法においては，GRE キー，MPLS ラベル，VLAN タグなどがセグメント識別子やセグメント識別子に対応するラベルとして使用される．これらのうちネットワーク・パーティションによる方式については 5 章において説明する．

前節における 2 つのライブ・マイグレーション方式のうち，複数のネットワークを使用する方法はネットワーク・セグメンテーションに基づいていると考えることができる．また，アドレスを変換する方法は，変換後のアドレスがかさならないようにすれば，ネットワーク・ページングによる仮想ネットワークを実現していると見なせる．すなわち，移動元のデータセンタの通信に使用されるアドレス範囲と移動先のデータセンタの通信に使用されるそれとが完全に分離され，その間の通信が基本的にできないようにすれば，それらを仮想ネットワークと見なすことができる．したがって，2 つのライブ・マイグレーション方式をともにネットワーク仮想化に基づく方法だと考えることができる．

3. 仮想ネットワーク・モデル

前章において示した 2 つのネットワーク仮想化方式のいずれにも適用でき，したがって 2 つのライブ・マイグレーション方式のいずれにも適用できる統一的なモデルを構成するため，まずこの章においては仮想ネットワークのモデルの基本および構成要素を記述する．

3.1 Nakao による仮想化ネットワーク・モデル

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行われ，さまざまなモデルが提案されている．そのなかには，PlanetLab [15),22)]，VINI [2)]，GENI [7)]，Genesis [10)] などがある．NICT の仮想化ノード・プロジェクトにおいても独自のモデルが開発されている[13),23)]．このモデルの中核部分がノードスリバーとリンクスリバーとで構成されるモデルである．

Nakao のモデル[12)] 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 4** 参照）が，これは次のようによばれる[12)]．

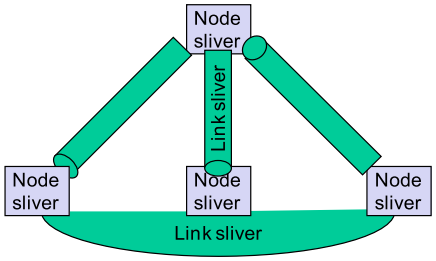


図 4　仮想ネットワークの論理構成（スライスの構造）
Fig. 4　Logical structure of a virtual network (Slice structure).

(1)　ノードスリバー（Node Sliver）

1 個の物理ノードの中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，汎用プロセッサ上の VM によるスローパス（slow path）と，ネットワーク・プロセッサや他の専用高速ハードウェアによるファストパス（fast path）の 2 種類がある．

(2)　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．通常は異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物理ノードにまたがって存在する．

ノードスリバーは 1 個以上，リンクスリバーは 2 個以上の仮想ポートを持つ．ノードスリバーの仮想ポートとリンクスリバーの仮想ポートとを結合することができる．

次節以降においては広域ライブ・マイグレーション方式のモデル化において使用する具体的なノードスリバーとリンクスリバーについて説明する．

3.2 VRF ノードスリバー

ルータや L3（3 層）スイッチが持つ IP ルーティング機能を VRF（Virtual Routing and Forwarding）ノードスリバーとしてモデル化する．Nakao のモデルにおけるノードスリバーはもともと自由にプログラムできる資源という意味を持っていたが，プログラムの自由度はスローパスとファストパスとでもちがっている．IP ルーティングにおいてはプログラム（設定）可能な範囲はかぎられているが，それでもルータや L3 スイッチにおいてはコマンドライン・インタフェース（CLI）などによってさまざまな設定をすることができる．したがって，VRF ノードスリバーはスローパス，ファストパスのいずれとも異なっているが，それ

らと同様に計算資源の一種であるから，ノードスリバーとしてあつかうのが適当だと考えられる．

VRF ノードスリバーはスライスごとの IP ルーティングのためのノードスリバーであり，1 個のスライスに対して各ノードで 1 個だけ定義できる．VRF ノードスリバーは次の機能を持つ．

(1)　IP ルーティング機能

ルーティング・テーブルを使用し，ノードスリバーに指定されたルーティング方式に従ってスライス上での IP ルーティングを行う．指定可能なルーティング方式としては，スタティック・ルーティング，OSPF，RIP などがある．IP ルーティングのために，VRF ノードスリバーに結合されたリンクスリバーには IP サブネットが割り当てられる必要がある．また，ルーティング方式の指定とあわせてルーティングのための適切なパラメータが指定されなければならない．

(2)　L2 機能

VRF ノードスリバーには自動で，または VRF ノードスリバーの属性として，MAC アドレスがきめられる．VRF ノードスリバーが受信する IP パケットのまえには Ethernet ヘッダなどの L2（2 層）ヘッダが存在しなければならない．ARP のようなブロードキャスト・パケットをのぞけば，VRF ノードスリバーが処理するパケットは前記の MAC アドレスが宛先となっているものだけである．

(3)　ARP 機能

IP パケットに指定された IP アドレスを，ノードスリバーの構成要素である ARP（Address Resolution Protocol）テーブルを使用して MAC アドレスに変換する機能を持つ．そのため，VRF ノードスリバーはスライス上の IP アドレスと MAC アドレスとの組を ARP テーブルに記録する．ノードスリバーに結合されたいずれかのリンクスリバーをとおして ARP パケットを受信すると，指定された IP アドレスがそのノードのアドレスであれば通常の応答をかえし，それがルーティング・テーブルにあるアドレスであれば代理応答をかえす．

VRF ノードスリバーが持つ属性は次のとおりである．

(1)　ノードスリバー ID

ノード内の他のスリバーとの識別およびノードスリバーに対応するスイッチ内の資源との対応づけのための，ノード内で一意な識別子である．

(2)　プロトコル

IPv4，IPv6 などのプロトコルを指定する．指定した以外のプロトコルのデータ・パケットが存在してもよいが，それはルーティングの対象にはならない[⋆1]．ただし，現在のプロトタイプにおいては IPv4 に固定されている．

(3)　ルーティング方式とパラメータ

スライス上での IP ルーティング方式としては，静的ルーティング，OSPF，RIP などが考えられる．ここでは OSPF に限定して説明する．OSPF においてはパラメータとしてサブネットとマスク，ドメイン番号，エリア番号，ルータの IP アドレス（ループバック・アドレスとして指定する）がある．なお，ルーティング方式とそのパラメータは実ネットワークと独立にきめられるとはかぎらない．たとえば，アドレス・マッピングに基づく仮想化においては実ネットワークにおいて使用されているルーティング方式と矛盾する設定はできない．これに対してネットワーク・パーティションのように実ネットワークとは独立にルーティングを設定することができるときは，実ネットワークにしばられない．したがって，この属性の値は仮想化方式の選択に依存する．

VRF ノードスリバーには，任意個のリンクスリバーを接続することができる．

3.3　Point-to-point リンクスリバー

Point-to-point（以下 PP と略称）リンクスリバー[23] は異なるノード上の 2 個のノードスリバー間を接続するリンクスリバーである．PP リンクスリバーは 2 個の仮想ポートを持つ．PP リンクスリバーの一方のポートに入力されるパケットは他方のポートから出力される．

このモデルにおいて PP リンクスリバーが持つ属性は次のとおりである．

(1)　リンクスリバー ID

スライス内の他のリンクスリバーと識別するための，スライス内で一意な識別子である．

(2)　IP サブネット

スライス上で IP が使用されるときは，PP リンクスリバーに対して IP サブネットを属性としてあたえる．2 点間をつなぐスリバーにおいてはサブネットのプレフィクスは 30 bit 以下である必要がある（端点に割当て可能なアドレスが 2 個以上存在する必要がある）．多点間の場合も，接続する点の数だけのアドレスが確保できるようにサブネットの大きさをパラメータとしてあたえる必要がある．IP サブネットを持つ PP リンクスリバーにおいては，ポート “i” にサブネットの下限 $+\,i$ のアドレスがあたえられる．このよ

⋆1 現在の仕様ではプロトコルは 1 個しか指定できないので，デュアル・スタックは表現できない．

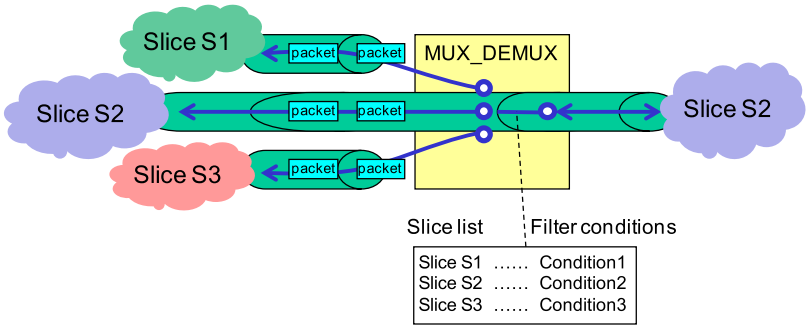


図 5　MUX_DEMUX リンクスリバーの概念
Fig. 5　Concept of MUX_DEMUX link-sliver.

うにサブネットをきめるだけで IP アドレスを自動的にあたえることができるので，設定を簡素化することができる．

3.4　MUX_DEMUX リンクスリバー

MUX_DEMUX リンクスリバーとは，1 個または複数個のスライスまたは IP ネットワークなどの外部ネットワーク上のパケットのうち指定された条件にあうものを 1 個のスライスにとりこむ（マルチプレクスする）とともに，それと逆向きにながれるパケットを前記のネットワークにもどす（デマルチプレクスする）ためのリンクスリバーである（**図 5** 参照）[*1]．それらのネットワークのうちの 1 個に関しては MUX_DEMUX リンクスリバーは入口と出口をあわせ持ち，その入口から入力されたパケットのうち他のネットワークにとりこまれなかったパケットはそのままその出口から出力される．マルチプレクスされた側（図 5 においては右側）の口は外部ノードに接続され，デマルチプレクスされた側（左側）の口はノードスリバーに接続される．

MUX_DEMUX リンクスリバーは複数のスライス間をつなぐものであり，今回は経路切替えのために使用する．しかし，複数スライスをつなぐ他の目的にも使用できるように汎用化されている．

複数のスライスをつなぐとき，それらが重複するアドレスを持たなければ，MUX_DEMUX リンクスリバーのような特殊な機構は必要ない．スライス間にまたがる IP ルーティングのような通常の機構によって必要な機能を実現することができる．しかし，ライブ・マイグレーションにおけるように VM の移動前と移動後のスライスに同一の IP アドレスや MAC アドレスが存在するときは通常の IP ネットワークや Ethernet の機構，あるいはそれらと同等の機構を使用することができない．MUX_DEMUX リンクスリバーはこのようなときに使用するためのものである．

MUX_DEMUX リンクスリバーは次のような属性を持つ．

(1)　リンクスリバー ID

一意な識別子としてのリンクスリバー ID を指定する．リンクスリバー ID は設定対象のルータやスイッチのコマンドにおいて使用される識別子を生成する際に使用されるかもしれない．

(2)　スライス・リスト

MUX_DEMUX リンクスリバーによって選択されるスライスのリスト $S_1, S_2, \ldots, S_N$ を指定する（図 5 においては $N = 3$ である）．通常のリンクスリバーは特定のスライス定義の中で定義されるため，スライスを指定する必要はない．しかし，MUX_DEMUX リンクスリバーは複数のスライスにまたがるため，それらのリストを指定する必要がある．

(3)　フィルタ条件名リスト

スライス・リストにふくまれるスライス S ごとに，(ユーザからとどいた）パケットをそのスライスにふりわけるフィルタ条件のリスト Cs:（$c_{s1}, c_{s2}, \ldots, c_{sns}$）の名称 Cs を指定する．フィルタ条件 c_{si}（$i = 1, 2, \ldots, ns$）は MUX_DEMUX リンクスリバーの外部で記述する．このリストの要素数はスライス・リストの要素数 N とひとしい．条件はこのリスト中でのならびの順に評価され，最初に適合したスライスが選択される．

(4)　IP サブネット

MUX_DEMUX リンクスリバー内で消費される IP アドレスをまかなうためにサブネットを指定する．消費する IP アドレス数は実装によって異なる可能性がある．

フィルタ条件名リストに指定されたフィルタ条件の内容を切り替えることによって，スライスが切り替えられる．

4.　仮想ネットワークによる広域ライブ・マイグレーションのモデル化

この章においては前章において記述した仮想ネットワークのモデルを使用して，2 つの広域ライブ・マイグレーション方式を統一的に記述することをこころみる．

⋆1 MUX_DEMUX リンクスリバーは複数個のスライスに属するノードスリバーをつなぐ機能を持つため，リンクスリバーと見なしている．しかし，このような高機能のスリバーはノードスリバーと PP リンクスリバーとの組合せによって表現するべきかもしれない．

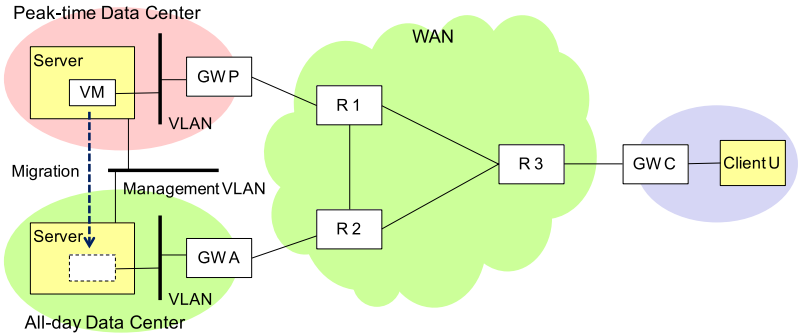


図 6　ネットワークの物理構造
Fig. 6　Physical structure of the network.

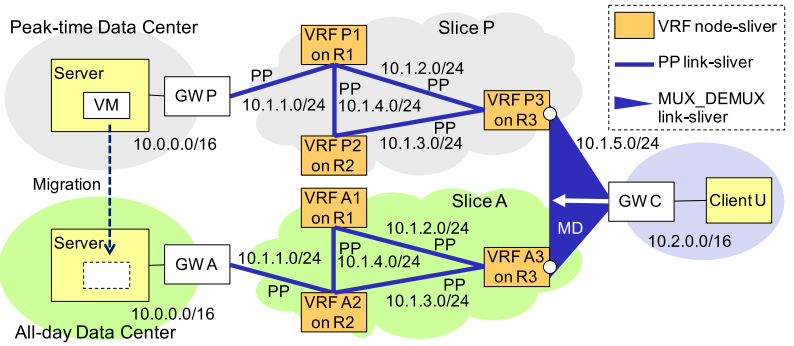


図 7　広域ライブ・マイグレーションのための仮想ネットワーク・モデル
Fig. 7　Virtual network model for wide-area live migration.

4.1　物理ネットワーク構成

広域ライブ・マイグレーションを行うためには，2 つのデータセンタ間にまたがる管理が必要であり，また VM 移動の際には両者をつなぐネットワークが必要である．VMWare, Xen をはじめとする現在の VM 管理ソフトウェアにおいてはデータセンタ間が L2 ネットワークまたは L2 トンネルによってむすばれている必要がある．そのため，ここでもこのような L2 パスが存在することを仮定する[⋆1]．しかし，データセンタ外との通信はこの L2 接続にしばられない，すなわち基本的には大域的なルーティングに従って，この（局所的な）L2 接続を使用せずに通信するものとする．それは，この L2 接続を使用して通信しつづけると，VM を移動したあとも WAN におけるトラフィックのながれが基本的にかわらず，省電力化をさまたげるからである．経路を切り替えることによって，使用されなくなったデータセンタにつながる WAN ルータの電源を遮断して省電力化をはかることができる．

このような条件のもとで，ここでは 1 個の VM の移動が完了したときに仮想化ネットワークを使用してその VM のトラフィックの経路を高速に切り替える方法について説明する．ここでは単純化されたケースだけを考察するものとする．そのため，想定するネットワークの物理構成は**図 6** のとおりである．ここでは，2 つのデータセンタすなわち Peak-time Data Center（PDC）と All-day Data Center（ADC）とに分散されていた VM を All-day Data Center に集中させ，Peak-time Data Center 全体とそれにつながる WAN ルータ R1 の電源を遮断する方法を考える．ただし，WAN を構成する各ノードがネットワーク仮想化機能を持っていることを前提とする．

4.2　仮想ネットワークの構成

図 7 に前章のモデルに基づくこの方式に必要な仮想ネットワークの構成を記述している．移動前の VM をふくむ PDC 内のサーバに接続されたスライス P と，移動後の VM をふくむ ADC 内のサーバに接続されたスライス A とを用意する．いずれのスライスも 3 個の物理ルータと対応する 3 個の VRF ノードスリバー（仮想ルータ）P1，P2，P3，A1，A2，A3 とそれらをむすぶ PP リンクスリバーによって構成されている．IP ルーティングを行うため，PP リンクスリバーにはサブネット 10.1.2.0/24，10.1.3.0/24 などの属性をあたえている．これらの属性によって，PP リンクスリバーと接続された VRF リンクスリバーの仮想ポートに自動的に IP アドレスが設定され，ルーティングなどに使用される．

PDC とスライス P とはゲートウェイ GWP によって接続され，ADC とスライス A とはゲートウェイ GWA によって接続されている．これらのゲートウェイはデータセンタやクライアントのネットワークから出力される通常の IP パケットを仮想ネットワークにおけるパケット形式に変換するとともに，仮想ネットワークから出力されるパケットを通常の IP パケットの形式にもどす働きをする．すなわち，仮想ネットワークの 2 つの実装方式に対応して，次のうちのいずれかの処理を行う．

⋆1 移動する VM がデータセンタ内のストレージや他の VM と通信しているときは，移動にともなってデータセンタ間での通信が必要になるが，この場合もデータセンタ間の L2 パスを使用して通信すると仮定する．

(1)　ネットワーク・セグメンテーションのとき

仮想化ネットワークにむかうパケットにはセグメント ID をふくむヘッダをつけ，外部にむかうパケットからはそのヘッダをはずす．ただし，ゲートウェイ GWC は MUX_DEMUX リンクスリバーと接続されていて，スライスを選択するのはその MUX_DEMUX リンクスリバーであるため，ヘッダをつけないと解釈することもできる[⋆1]．

(2)　ネットワーク・ページング（アドレス・マッピングに基づく仮想化）のとき

仮想化ネットワークにむかうパケットは仮想化ネットワーク内部のアドレスづけに従ってアドレス・マッピングを適用する．外部にむかうパケットは外部の IP ネットワークのアドレスづけに従ってアドレス・マッピングを逆に適用する．ただし，ゲートウェイ GWC においてはそれに接続された MUX_DEMUX リンクスリバーがアドレス・マッピング機能を持つ必要があるので，ゲートウェイ GWC がアドレス・マッピング機能を持つ必要はかならずしもない[⋆2]．

これらのゲートウェイと VRF ノードスリバーとを結合しているのも PP リンクスリバーであり，属性として IP サブネットがあたえられている．これらのゲートウェイはスライスの一部である．また，ユーザのネットワークとスライス P，A とは MUX_DEMUX リンクスリバー MD とゲートウェイ GWC によって接続されている．この MUX_DEMUX リンクスリバーがデータセンタとクライアントとのあいだの経路をスイッチする．したがって，VM 移動が完了したときにこのリンクスリバーに経路切替えのためのトリガをかける必要があるが，その実装法については 6 章において記述する．

障害対応が必要なければ仮想ルータはスライス P，A それぞれに 2 個ずつでよいが，ルータ R1 が遮断されていないときには障害時に代替経路への切替えができるように，物理構成にあわせて仮想ルータを 3 個使用し，かつスライス内で動的ルーティングを行うことにする[⋆3]．動的ルーティングのパラメータは，スライス内で矛盾なく設定するためにはスライス全体に対して定義されるべきだと考えられる．しかし，現在の実装においては VRF ノードスリバーごとに指定している．これらのスライスは図 6 にえがかれた 1 つの物理ネットワークにかさねあわせられる．これらの仮想ネットワークにおいては OSPF などの IP ルーティング・プロトコルが使用され，仮想ネットワーク内の経路はそれに従う．

これらのデータセンタに接続されたネットワークの IP サブネットはひとしく，この図においては 10.0.*.* である．移動する VM の通信用の IP アドレスはこのサブネットに属する．これらはサブネットがひとしいため 1 個の IP ネットワークに接続することはできないが，異なる仮想ネットワークに接続するのに問題はない．仮想化ノードは 2 重化された経路情報を持つ．すなわち，経路情報は一部を除いて 2 個の仮想ネットワークに共通である．共通な経路情報は実際には複写せず，2 つの仮想ネットワーク間で共有することが理論的には可能である．しかし，各ノードはデータセンタが属するサブネットに関しては異なる経路情報を持つ．必要に応じて，ほかにも異なる点があってよい．すなわち，2 つの仮想ネットワークの構造は異なっていてもよい．これによって，PDC を使用しないときに使用しないノードをあらかじめ仮想ネットワークからはずしておく，すなわち使用しないノードには VRF ノードスリバーをおかないようにすることができる．

4.3　データセンタ切替え方式

以下，VM の移動にともなってデータセンタを切り替えるための方式について説明する．仮想化ネットワークを使用する広域ライブ・マイグレーション方式においては，仮想ネットワーク（Nakao のモデルにあわせて，以下**スライス**とよぶ）を切り替えることによって，パケット転送先のデータセンタを切り替える (**図 8** 参照)．すなわち，図 8 (a) においては，PDC にある（移動前の）VM との通信には PDC に接続されたスライス P を使用する．クライアント U から VM（そのアドレスを AV とする）へのパケットは WAN への入口ルータ R3 の MUX_DEMUX リンクスリバー MD において識別され，スライス P にとりこまれる（VRF ノードスリバー P3 に転送される）．すなわち，MD においてはクライアント U からのパケットに次の転送規則を適用する．

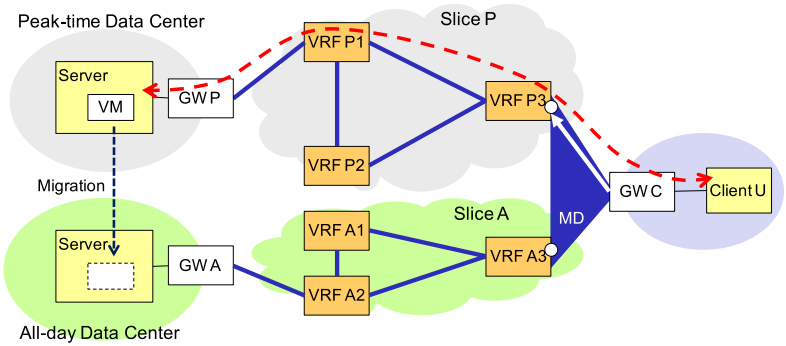
```
if destination_IP == AV then
   /* パケットの送信先 IP アドレスが VM のアドレス AV とひとしければ */
   slice = P, forward the packet /* パケットをスライス P に転送 */
if true then
   slice = A, forward the packet /* パケットをスライス A に転送 */
```

これらの規則の左辺が MUX_DEMUX リンクスリバーのフィルタ条件名リストにおいて指定されたフィルタ条件である．ここではフィルタ条件リストの要素は移動する VM のた

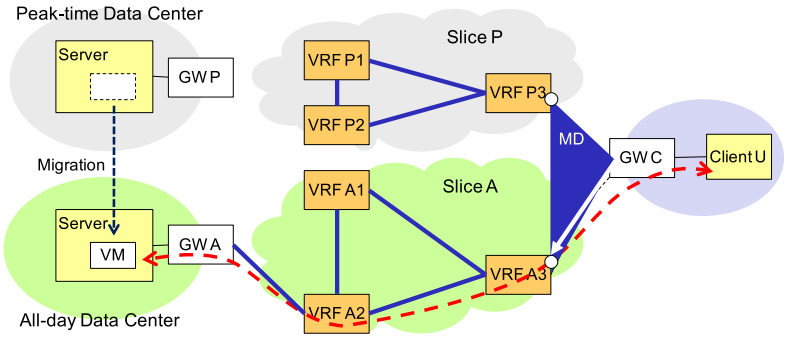
⋆1 ゲートウェイ GWC が他のゲートウェイと統一された機能を持つように定義するには，ユーザのネットワークにもセグメント ID をあたえ，ゲートウェイ GWC はそのセグメント ID をふくむヘッダをつけるようにすればよい．ネットワーク・パーティションを使用するときは実装もこれにちかくなる．

⋆2 しかし，他のゲートウェイと機能を統一するためには，もし仮想化ネットワーク内のアドレス形式が外部におけるのと異なっているときには，ゲートウェイ GWC において内部の形式に変換する必要がある．たとえば，外部では IPv4 を使用し仮想化ネットワークにおいては IPv6 を使用する場合には，外部からのパケットは IPv4 から IPv6 に変換し，外部へのパケットはその逆の変換をする必要がある．

⋆3 ただし，ルータ R1 の電源を遮断したときには VRF ノードスリバー P1 は機能しなくなり，冗長経路はなくなる．



(a) VM 移動前のネットワークの構成



(b) VM 移動後のネットワークの構成

図 8　ライブ・マイグレーションの仮想ネットワークによるモデル
Fig. 8　Method of live-migration using virtual networks.

めの 1 個のフィルタ条件だけだが，PDC にほかにも VM が存在するときは，その VM のためのフィルタ条件も必要である．スライスの指定のしかたは仮想化の方式に依存するが，VLAN や GRE などのトンネルを使用する場合はパケット上のラベル（VLAN ID またはトンネル ID）によって指定され，GMPLS を使用する場合は波長によって指定したり，異なる光パスを使用することなどによって指定される．

また，VM からクライアント U へのパケットは WAN からの出口（P3）においてクライアント U にむけられる．このときは必要に応じてパケットのラベルを削除するなどの操作が必要になるが，パケットのふりわけは必要ない．

VM 移動後は図 8 (b) のようにする．すなわち，クライアント側出口における転送規則を次のように変更する．

```
if true then
  slice = A, forward the packet /* パケットをスライス A に転送 */
```

すなわち，2 個の転送規則のうちの最初の 1 個を無効にする．この変更は，スライス P に関する唯一のフィルタ条件を削除することによって実現される．

これによって，クライアント U が VM のアドレス AV を指定してパケットを送信すると移動後の VM と通信することになる．この切替えによって WAN 内部の設定やルーティングは変化しないので，切替えによってただちに VM 移動前と同一のアドレスを使用して移動後の VM とクライアント U とが通信できるようになる．

この論理構成において注意するべき点は，クライアント U のネットワークの経路情報をスライス P，スライス A の両方から参照できるようにする必要があるということである．そうしないと，VM が移動したとたんにクライアント U との通信ができなくなる．そのため，WAN 内の経路情報はスライス P とスライス A とで共通だが，それとあわせてクライアント U のネットワークの経路情報も共通情報にいれるのがよい．もし WAN 内の経路情報をスライス P とスライス A とで重複して持つのであれば，クライアント U のネットワークの経路情報も重複して持つ必要がある．

5.　モデルから仮想化ネットワーク方式への展開法

この章では前章の仮想ネットワーク・モデルを 2 つの実装方式に展開する方法を記述する．モデルの主要な要素は VRF ノードスリバー，PP リンクスリバー，MUX_DEMUX リンクスリバーであるから，これらについてそれぞれ展開法をしめす．

5.1　ネットワーク・パーティションによるカプセル化に基づく実装への展開法

アラクサラネットワークス社のスイッチ AX6000 シリーズはネットワーク・パーティション[1] という VLAN に基づくネットワーク仮想化機構を持っている．この仮想化機構を使用することによって，VLAN 機能によって，したがって一種のネットワーク・セグメンテーションによって生成された仮想ネットワークごとに独立な IPv4/v6 ルーティングを設定することができる．ネットワーク・パーティションは VLAN を利用するため，パケットにあらたなヘッダを挿入するわけではない．しかし，標準の Ethernet ヘッダとくらべると VLAN タグを追加しているため，ネットワーク・セグメンテーションによる仮想化方式と見なすことができる．この節においては前章のモデルをネットワーク・パーティションによる仮想

ネットワークに展開する方法を 4 項目にわけて記述する．

第 1 に，スライスへの資源の割当てについてのべる．このネットワーク・パーティションによる実装においては物理リンクとして通常は Ethernet を使用し，そのスライスが使用する各物理リンクにそのスライスに対応する VLAN ID を割り当てる必要がある．スライスが使用する VLAN ID は物理リンクごとに異なっていてもよいが，現在の実装においてはどの物理リンクにおいても同一の VLAN ID を（セグメント番号として）使用するようにしている．すなわち，ドメイン内のすべてのスイッチにおいて同一の VLAN ID をそのスライスの実装属性として予約する．

現在の実装においてはこの VLAN ID をモデル記述におけるスライスの宣言においてパラメータとしてあたえるようにしている．VLAN ID を指定することによって，実装依存のパラメータが導入されるが，管理やデバグ，物理リンク上のパケットの解析は容易になるという長所がある．しかし，生成可能なスライス数がおさえられるという短所がある．なお，この実装においては後述するように VLAN ID をリンクスリバーにも割り当てていて，そのほうがはるかに多数になる．

第 2 に，VRF ノードスリバーの展開法についてのべる．このネットワーク・パーティションによる実装においては VRF ノードスリバーにルーティング方式とそのパラメータとを指定することができるが，これらが CLI（vrf コマンドと ospf コマンド）によってスイッチに設定される．リンクスリバーによって結合された VRF ノードスリバー間のルーティング方式やパラメータの整合性を自動的に検査するのがよいが，現在は行っていない．

第 3 に，PP リンクスリバーの展開法についてのべる．前記のように物理リンク上ではスライスに対応する VLAN ID を使用するが，ネットワーク・パーティションにおいてはスイッチの内部において方路ごとに異なる ID を使用する必要がある．そのため，スイッチ内ではこの ID を PP リンクスリバーに自動的に割り当てている（対向のスイッチにおいては同一のリンクスリバーにこれとは独立に ID があたえられる）．**図 9** のようにパケットがスイッチから出力される際に（物理インタフェースの設定において）ID を静的にきまる他の ID にスイッチするように設定している．この図から分かるようにこの仮想化方式においてはスライスに割り当てられる VLAN ID より PP リンクスリバーに割り当てられる VLAN ID のほうが通常は多数になる．

PP リンクスリバーが VRF ノードスリバーどうしを結合するときは，その両端に IP アドレスが必要である．PP リンクスリバーの属性として IP サブネットを指定すれば，そこから両端のアドレスを自動的に生成することができる．

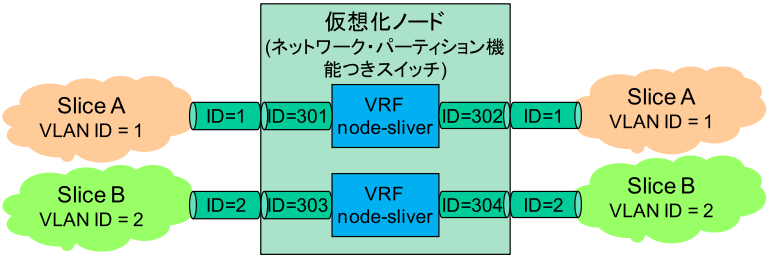


図 9　PP リンクスリバーの VLAN ID とスライスの VLAN ID
Fig. 9　VLAN IDs of PP link-slivers and VLAN ID of slice.

第 4 に，MUX_DEMUX リンクスリバーの展開法についてのべる．ネットワーク・パーティションによる実装においては，MUX_DEMUX リンクスリバーが動的に VLAN ID をスイッチする必要がある．スライス間でデータをやりとりしない通常の仮想化ネットワークや VPN においてはこのような機能は必要とされないため，既存のスイッチにおいては動的な VLAN ID のスイッチすなわちパケットの内容から VLAN ID を決定するのに使用できる機能はかぎられている．ポリシベース・ルーティング（PBR）はそのために使用できる方法の 1 つである．ここで PBR とは，パケットの内容に関するフィルタ条件に基づいて転送先 IP アドレスをきめる機能である．

スライスの切替え機能の実装に PBR を使用することによって，やや複雑な方法をとる必要があった．すなわち，PBR においてはパケット転送先として隣接ノードを指定する必要がある．これは通常のルーティングにおいて隣接ノードを指定するのとおなじである．ところが，前記のモデルにおいては仮想化ノード内（仮想化ノードからの出口）にこの機能を実装する必要がある．そのため，**図 10** に示すように，両端が同一のスイッチにある物理リンクを使用している（図では物理リンクが 2 本あるように見えるが，VLAN を使用するので 1 本でよい）[*1]．この実装においては最低 3 個のアドレスが必要であり，この構造を実現するためにプレフィクスが 29 bit 以下のサブネットを指定する必要がある．また，上記の物理リンクを用意して，そのインタフェース番号もパラメータとして指定する必要がある．

モデル上はフィルタ条件を変更するとそれを参照しているリンクスリバーの動作が変化す

⋆1 MUX_DEMUX リンクスリバーの機能をノードスリバーとしてモデル化し，VRF ノードスリバーとは異なる仮想化ノードにおくことにすれば，モデルと PBR による実装の構造をあわせることができる．しかし，このモデルは他の実装法においては不要な制約を課す可能性がある．

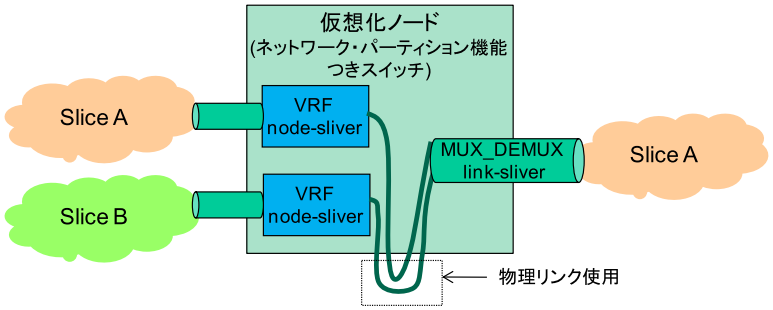


図 10　VRF ノードスリバーと MUX_DEMUX リンクスリバーとの結合
Fig. 10　Binding of VRF node-sliver and MUX_DEMUX link-sliver.

るようになっている．しかし，実装上はリンクスリバーに対応する PBR の設定の中に条件がとりこまれている（コピーされている）ため，フィルタ条件が変更されたときに PBR の設定変更の手続きが起動されるようにしている．

5.2　アドレス・マッピングに基づく実装への展開法

前章のモデルをアドレス・マッピングに基づく仮想ネットワークに展開する方法をしめす．説明を簡単にするため，ここでは物理ネットワークと仮想ネットワークの両方で IPv4 を使用する．仮想ネットワークを生成するのに適したアドレス変換機構はまだ商用化されていないと考えられるので，ここでは変換プログラムを搭載した PC をふくむネットワークへの展開を考える．しかし，モデルを変換する方法はアドレス・マッピングに基づく仮想ネットワークに一般的に適用可能である．

以下，4 項目にわけて記述する．第 1 に，スライスへの資源の割当てについてのべる．ネットワーク・セグメンテーションに基づく実装におけるセグメント番号の割当てに相当するのが，スライスへのアドレス範囲の割当てである．スライス定義においてそのスライス上でのアドレス範囲を指定する[*1]．現在の実装ではスライス上で IPv4 アドレスを使用することを仮定しているが，一定範囲の整数値と対応づけられれば他の種類のアドレスを使用することも可能である．このアドレス範囲はネットワーク・セグメンテーションに基づく実装においては必須ではないが，指定されていてもよい．仮想化ネットワークの管理システムはこの指定に基づいてスライスに物理ネットワークのアドレスを対応させる．

第 2 に，VRF ノードスリバーの展開法についてのべる．アドレス・マッピングによる実装においては，ネットワーク・パーティションにおけるように自由にスライス上でのルーティング方式やパラメータをきめることはできない．現在の実装においては，OSPF についていえば，物理ネットワークに対してあらかじめ設定されたドメイン番号，エリア番号，ループバック・アドレスなどをそのまま使用するため，これらはパラメータとして指定することはできない．経路のサブネットとマスクは指定する必要があるが，これらはスライス定義からきめられたアドレス変換に従って変換し，指定されたアドレスが宣言された範囲内かどうかを検査したうえで設定する．

第 3 に，PP リンクスリバーの展開法についてのべる．スライス定義の PP リンクスリバーの部分において指定されたサブネットは，宣言された範囲内かどうかを検査したうえ，アドレス変換して設定される．

第 4 に，MUX_DEMUX リンクスリバーの展開法についてのべる．現在の実装においては，このリンクスリバーは仮想化ノードに付加されたアドレス変換機構によって実現される．すなわち，アドレス変換機構にパケット・ヘッダを参照して実行する規則を選択できる機能をくみこみ，フィルタ条件が変更されるとアドレス・マッピングの設定を変更してそれを反映させる．今回の実装においてはアドレス・マッピングがソフトウェアによって実現されているため，設定変更は容易である．実用的な性能を得るにはすくなくとも一部をハードウェア化する必要があり，設定にもくふうが必要だと考えられる．

6.　プロトタイプ開発と評価実験

この章においては，提案した広域ライブ・マイグレーション方式の効果を確認するために行ったプロトタイプ開発と評価実験について，まずその方法についてのべ，次にその結果についてのべる．

6.1　管理アーキテクチャ

スライスを物理ネットワークにマップし，その管理とくにスライスを設定するためのアーキテクチャを記述する．

複数の仮想化ネットワークが存在する一般の場合におけるネットワークとその管理システムの全体構成を**図 11** にしめす．仮想化ネットワークは仮想化機能を持つノードとネットワークに接続されたデータセンタなどを**統合管理システム**（Integrated Management System, **IM**）が管理する．各仮想化ノードはルータまたはスイッチと，それを管理する**ノード管理**

⋆1 アドレス範囲はスライス定義上できめるのでなく，管理サーバが適切な方法に従ってきめることも可能だと考えられる．

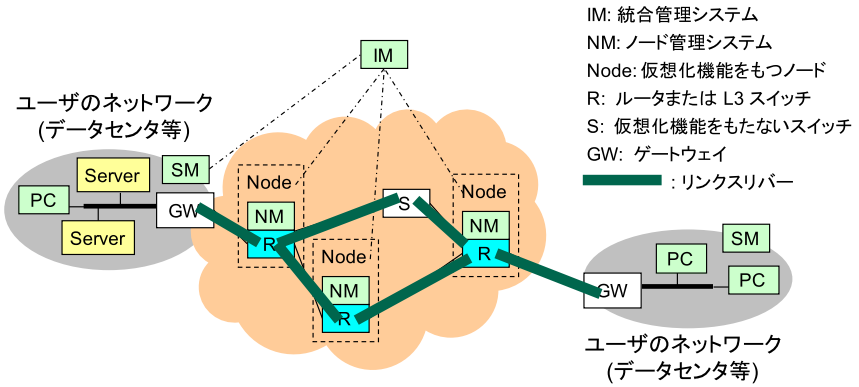


図 11　仮想化ネットワークの全体構成
Fig. 11　Whole structure of virtualized networks.

システム（Node Management System，**NM**）とで構成されている．

管理ドメイン内には仮想化機能を持たないルータやスイッチ（以下，非仮想化ノードとよぶ）が存在していてもよい．仮想化ノード間にこれらの仮想化されていないノードが存在していても，ある条件をみたせば，これらをトンネルするかたちでリンクスリバーを生成することができる[⋆1]．また，直接接続されたドメイン間はもちろん，WAN などを介したドメイン間でも，同様の条件をみたすことによって仮想化ノード間にリンクスリバーを生成することができる．その際には 2 つのドメインの IM どうしの交渉が必要である．

プロトタイプにおいてはドメインをまたがる機能は実装していない．スライスを生成するときは，IM にその定義をあたえる．IM は Perl によって実装されていて，スライス定義は Perl のデータとしてあたえられる．特定のプログラミング言語に依存しないかたちにするにはその構文を XML によって表現すればよいが，Perl によるスライス定義構文は容易に XML に変換することができる．

IM はドメイン全体の定義を仮想化ノード単位に分割し，各仮想化ノードに配布する．プロトタイプにおいては，この配布のために XML に基づく独自の設定用プロトコル XConf を使用している．ここで NETCONF[5] や XML-RPC[24] のような標準プロトコルを使用することもできるが，実装の容易さのために独自プロトコルを使用している．

6.2　評価の方法

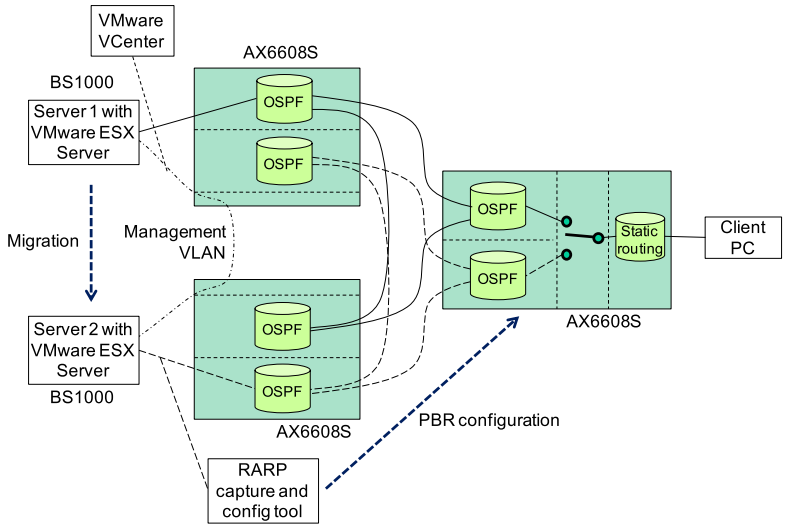
図 7 のような仮想ネットワークをモデルに従って定義し，これをもとに 2 種類の実装に基づく設定を生成して，それぞれ広域ライブ・マイグレーションのための設定の実験を行った．ネットワーク・ノードとしては 3 個の L3 スイッチ（アラクサラ AX6608S）を 10 Gbps のリンクでつないで使用した．サーバ・システムとしては，VMware ESX Server を搭載した 2 セットのブレード・サーバ（日立 BS1000）と 1 台のクライアント PC を使用した．実験ネットワークの構成を**図 12** に示す．ネットワーク・セグメンテーションに基づく構成が図 12 (a)，ネットワーク・ページングに基づく構成が図 12 (b) である．いずれにおいても VLAN 設定によって各スイッチを論理的に分割して使用している．とくに，図 12 (b) においては後述するアドレス・マッピングのための NAT Box とサーバやクライアントを接続するために 1 台のスイッチを 2 台以上のルーティング機能を持つ論理的なノードに分割して使用している．このような構成にしたおもな理由は，実験ネットワークをかぎられた台数のスイッチによって構成する必要があったことである．

図 12 (a) には記述していないが，ネットワーク・セグメンテーションに基づく構成の設定のためには，前章に記述したように，Perl によって記述した IM と NM のプロトタイプを使用している．IM にスライス定義をあたえると，NM 経由で L3 スイッチが設定される．すなわち，L3 スイッチが持つネットワーク・パーティション機能に基づく 2 個のスライスが生成され，それぞれのスライス上で OSPF による動的ルーティングが独立に動作するように設定される．

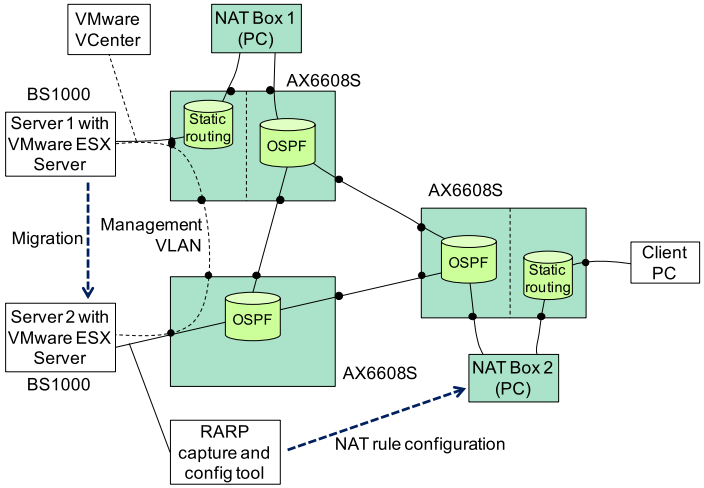
一方，ネットワーク・ページングに基づく構成（図 12 (b)）のためには，スライス定義に基づいてアドレス変換の機能を設定し，仮想化ネットワークにおける OSPF の 1 つのインスタンスが両方のスライスに機能する．ただし，現在の NM はネットワーク・パーティションによる実装だけに対応しているため，アドレス・マッピングによる実装に関しては手変換にたよっている．アドレス・マッピングは図の NAT Box 1 および 2 によって実現されるが，これらは複数個のネットワーク・インタフェースを持つ 2 台の PC に搭載された C のプログラムによって実現されている．NAT Box にはコマンド入力によって変換規則を設定することができる．NM が対応していないため NAT Box にはあらかじめ変換規則を設定してあるが，NAT Box 2 は VM 移動後にスライス切替えのために変換規則がスイッチされる．

上記のプロトタイプにはフィルタ条件を変更してスライスを切り替える機能も実装した

⋆1 みたすべき条件の例をあげる．仮想化ネットワークを構成するのに VLAN が使用される（VLAN リンクスリバーによって構成する）ときは，非仮想化ノードは VLAN スイッチであり，VLAN リンクスリバーを通過させるように構成されている必要がある．また，仮想化ネットワークを構成するのに IP が使用されるときは，非仮想化ノードは IP ルータである必要がある．



(a) ネットワーク・セグメンテーションによる方法の実験ネットワーク構成



(b) アドレス・マッピングによる方法の実験ネットワーク構成

図 12　評価のための実験ネットワーク構成
Fig. 12　Experimental network structure for evaluation.

表 1　仮想ネットワークの定義がふくむパラメータ数
Table 1　Numbers of parameters that the virtual network definitions contain.

スライス定義の構成要素	スライス定義のパラメタ数			ネットワーク・パーティション実装のパラメタ数	アドレス・マッピング実装のパラメタ数
	実装非依存のパラメタ	ネットワーク・パーティション用パラメタ	アドレス・マッピング用パラメタ		
スライス	3	3	3	9	3
ノードスリバー	56	6	6	111	62
リンクスリバー	66	4	0	63	33
その他	140	0	0	0	0
総計	265	13	9	183	98

が，現在のところ IM および NM は VM 移動終了を検出してこの機能にトリガをかける機能を持っていない．そのため，今回の実験においては VMware において VM 移動終了時に生成される RARP（Revere Address Resolution Protocol）メッセージをトリガ・イベントとして検出して，NM が生成するのと同じスライス切替えのコマンドを生成する独立のプログラムを使用した．このプログラムは上記の仮想ネットワークとは独立の “WAN” 管理ネットワークを経由してクライアント側のスイッチや NAT Box の設定をコマンドによって変更する．

6.3　結　　果

この節においては評価結果をしめす．第 1 に，現在のプロトタイプにおいてどれだけ実装方式への依存性がのこっているかを検討する．スライス定義には実装方式に依存するパラメータの一部がふくまれていて，完全に実装非依存にはなっていない．これらのパラメータは他の実装方式をとるときには不要であり，指定できない．しかし，スライス定義の大半の部分は実装非依存である．

表 1 はスライス定義における実装非依存および 2 つの実装のために追加された実装依存のパラメータ数を示す．表 1 にはスライス定義を各実装方式に従って変換したあとの CLI などのコマンドがふくむパラメータ数も右 2 列に示す．パラメータのなかには配列のかたちに構造化されているものもあるが，その場合は配列要素ごとにカウントしている．表 1 にはスライス定義の構成要素としてスライス（スライス自体の記述），ノードスリバー，リンクスリバーをあげている．スライス定義はこのほかにノードスリバーとリンクスリバーを結合するためのワイヤやフィルタ条件をふくんでいるが，これらには実装依存の部分がないため，表 1 においては 1 行にまとめている．パラメータの総計で見れば実装依存のものは全

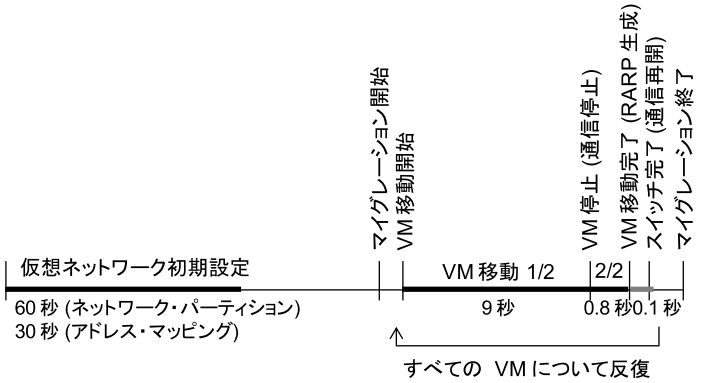


図 13　実験におけるネットワーク設定とマイグレーションのタイム・チャート
Fig. 13　Time-chart of network configuration and migration in the experiment.

体の 5%未満であることが分かる．

スライス定義における実装依存のパラメータとしては，ネットワーク・パーティション用には VLAN ID の指定，アドレス・マッピング用には使用可能なアドレス範囲の指定，MUX_DEMUX リンクスリバーの物理インタフェース指定などがある．また，両方がふくんでいるものとしてルーティングに関するパラメータがある．一部の実装依存パラメータは，今後，実装方式をくふうすればなくせると考えられるが，アドレス範囲指定のように省略困難なものもある．

第 2 に，初期設定性能測定結果についてのべる．**図 13** にはネットワークの初期設定からライブ・マイグレーション終了までのタイム・チャートを記述したが，その最初の部分すなわち左端が初期設定である．IM から NM への定義の配布は 1 秒未満の時間で完了するが，各 NM から L3 スイッチへの設定配布には 30〜60 秒の時間がかかる．その理由は，設定のために L3 スイッチに投入するコマンドが多数あり，L3 スイッチにおいて実行に時間がかかるためである．本来は各スイッチに並列に設定を配布することができるはずだが，現在のプロトタイプにおいては実装とデバグを容易にするため，逐次的に配布するようになっている．並列化は比較的容易に実現することができるが，1 台あたりの設定時間を短縮するには CLI のような逐次的設定方式によらないスイッチ設定が必要であり，その実現は容易でない．

第 3 に，ライブ・マイグレーション自体の性能についてのべる．図 13 に示したように，この実験においては VMware が 1 個の VM を移動させるのに約 10 秒かかっているが，VM が停止するのは移動終了の約 0.8 秒前である．移動終了にともなってトリガ・イベント（RARP）が生成されるが，それから通信再開までの時間は 0.08〜0.1 秒であり，目標値が実現されている．VM が停止してから通信が再開するまでに 0.9 秒かかっているので，このままではリアルタイム・アプリケーションに適用することはできない．しかし，ネットワークの設定変更の時間は短いので，VM 移動時間が短縮されればリアルタイム・アプリケーションにも適用できると考えられる．なお，この実験を反復すると数回に 1 回は通信再開後，約 4 秒で再度 1 秒間程度，通信が中断する現象が観察された．その原因は分かっていない．

7. 関 連 研 究

仮想ネットワークをモデル化するにはそれを仮想ノードと仮想リンクとの組合せによって表現するのが自然である．したがって，3 章に記述した Nakao のモデル以外に GENI の RSpec[17]，ProtoGENI[18]，G-lambda[19] などにおいてもこのようなモデルが使用されている．G-lambda においてはノードのプログラマビリティはあつかわれていない．それに対して GENI においては仮想ノードをプログラマブルにすることをめざしているため，仮想ノードに対して CPU，ディスク，ネットワーク・デバイスなど，さまざまな設定が可能である．しかし，プリミティブ（計算要素）を組み合わせて仮想ノードの機能をくみあげる手段はない．プリミティブを組み合わせて機能をくみあげるノード・モデルとしては Click[9] があるが，Click はネットワーク全体をモデル化することを目的とはしていない．

また，ネットワーク仮想化に直接関係はないが，ITU-T においてはトランスポート・ネットワークを記述するためのモデリング言語の標準 G.805[4),8)] がさだめられている．

さらに，論理イーサネット・モデルに基づくネットワーク設定変更の容易化などに関する吉澤らの研究[25)] がある．本研究においては現在のところ新規の設定ないし再設定だけをあつかっているが，吉澤らの研究においてあつかっている設定の変更や削除にはそれにはない複雑さがあり，今後，本研究においても設定変更・削除に対象をひろげていく必要がある．その一方で，吉澤らの研究が対象をイーサネットに限定しているのに対して，本研究においては多様なネットワークに展開可能な汎用的なモデルをあつかっている．

8. 結　　論

広域ライブ・マイグレーション中に通信を 2 つのデータセンタ間で切り替える方法を Nakao のモデルをベースとした仮想ネットワーク・モデルに基づいて実現した．このモデル化に

よって，ネットワーク・セグメンテーションに基づく仮想ネットワークとネットワーク・ページングに基づく仮想ネットワークとを統一的に記述することができる．ただし，現在のプロトタイプにおいてはモデルがふくむパラメータの 5%以下の実装依存のパラメータをモデルとともに管理する必要がある．

前記のプロトタイプを使用してネットワークを設定し，切替え実験を行った．VM 移動におけるネットワークの設定変更は 0.08〜0.1 秒で完了した．VM 移動時間の短縮はこの論文の範囲外だが，それが実現され，今回は実験ネットワーク上で実現された設定時間短縮が実際に WAN 上でも実現されれば，リアルタイム・アプリケーションの広域ライブ・マイグレーションも実現可能になると考えられる．なお，初期設定に関しては，IM から NM への定義の配布は 1 秒未満の時間で完了するが，各 NM から L3 スイッチへの設定配布には 30〜60 秒の時間がかかった．

仮想ネットワークのモデル化とその汎用化に関する今後のおもな課題は次のとおりである．

- NETCONF などのプロトコルを使用することによって，スイッチ 1 台あたりの設定時間を短縮するとともに，標準に準拠した設定を実現すること．
- 各スイッチへの設定を並列化して設定時間の短縮をはかること．
- 実装依存のパラメータなどを隠蔽もしくはスライス定義から分離し，実装方式を変更したときの変更を最小限にすること．
- ネットワーク・パーティションやアドレス・マッピングだけでなく他の実装方式への展開をこころみること．

VM の広域ライブ・マイグレーションに固有な課題としては次のものがある．

- VM 移動終了の検出機構と IM，NM による設定変更機構とを高速に連動させ，スケーラブルな切替え機構を実現すること．

謝辞　この論文に記述した研究結果の一部は総務省委託による 2009 年度のエコインターネット・プロジェクト（ネットワークとアプリケーションシステムの協調による省電力化技術の研究開発）の成果である．アラクサラの木谷誠，黒崎芳行，日立の高瀬晶彦の各氏には議論に参加していただき，有益なコメントをいただいたので感謝する．また，ていねいに査読していただいた査読者の方々に感謝する．

参　考　文　献

1) アラクサラ：AX シリーズ　ネットワーク・パーティション　ソリューションガイド［基本編］，第 2 版 (2010). http://www.alaxala.com/jp/techinfo/archive/guide/pdf/N08R065_NP_Guide_basic_V2R0.pdf
2) 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).
3) Bradford, R., Kotsovinos, E., Feldmann, A. and Schiöberg, H.: Live Wide-Area Migration of Virtual Machines Including Local Persistent State, *3rd Int'l Conference on Virtual Execution Environments*, pp.169–179 (2007).
4) Dijkstra, F., Andree, B., Koymans, K. and van der Ham, J.: Introduction to ITU-T Recommendation G.805, SNE Technical Report SNE-UVA-2007-01 (Dec. 2007). http://www.science.uva.nl/research/sne/reports/
5) Enns, R. (Ed.): NETCONF Configuration Protocol, RFC 4741, IETF (Dec. 2006).
6) Farinacci, D., Li, T., Hanks, S., Meyer, D. and Traina, P.: Generic Routing Encapsulation (GRE), RFC 2784 (Mar. 2000).
7) The GENI Project: Lifecycle of a GENI Experiment, GENI-SE-SY-TS-UC-LC-01.2 (Apr. 2009). http://groups.geni.net/geni/attachment/wiki/ExperimentLifecycleDocument/ExperimentLifeCycle-v01.2.pdf?format=raw
8) ITU-T Recommendation G.805: Generic Functional Architecture of Transport Networks, Technical Report, International Telecommunication Union (Mar. 2000). http://www.itu.int/rec/T-REC-G.805
9) Kohler, E., Morris, R., Chen, B., Jannotti, J. and Frans Kaashoek, M.: The Click Modular Router, *ACM Trans. Computer Systems* (*TOCS*), Vol.18, No.3, pp.263–297 (2000).
10) Kounavis, M., Campbell, A., Chou, S., Modoux, F., Vicente, J. and Zhuang, H.: The Genesis Kernel: A Programming System for Spawning Network Architectures, *IEEE Journal on Selected Areas in Communications*, Vol.19, No.3, pp.511–526 (2001).
11) Li, Q., Huai, J., Li, J., Wo, T. and Wen, M.: HyperMIP: Hypervisor Controlled Mobile IP for Virtual Machine Live Migration across Networks, *11th IEEE High Assurance Systems Engineering Symposium*, pp.80–88 (2008).
12) Nakao, A.: Network Virtualization as Foundation for Enabling New Network Architectures and Applications, *IEICE Trans. Commun.*, Vol.E93-B, No.3, pp.454–457 (2010).
13) 中尾彰浩：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).
14) Ohta, M. and Fujikawa, K.: IP−−: A Reduced Internet Protocol for Optical Packet Networking, *IEICE Trans. Communications*, Vol.E93-B, No.3, pp.466–469 (2010).
15) 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 (2003).
16) Ramakrishnan, K.K., Shenoy, P. and van der Merwe, J.: Live Data Center Migration Across WANs: A Robust Cooperative Context Aware Approach, *2007 SIGCOMM Workshop on Internet Network Management*, pp.262–267 (2007).
17) Ricci, R. (Ed.): RSpec, GENI: Global Environment for Network Innovations, GDD-08-xx (Mar. 2008).
18) Ricci, R.: ProtoGENI Experimenter Tools, GEC 5 (July 2009). http://groups.geni.net/geni/attachment/wiki/Gec5ServicesAgenda/ProtoGENI-Exptools.pdf
19) 竹房あつ子，中田秀基，工藤知宏，田中良夫，関口智嗣：計算資源とネットワーク資源を同時確保する予約ベースグリッドスケジューリングシステム，*SACSIS'06*. http://www.g-lambda.net/gridars/dataDir/sacsis06takefusa_slide.pdf
20) Tanenbaum, A.S.: *Modern Operating Systems*, *3rd Edition*, Pearson Prentice Hall (2008).
21) Travostinoa, F., Daspitb, P., Gommansc, L., Joga, C., de Laatc, C., Mambrettid, J., Mongaa, I., van Oudenaardec, B., Raghunatha, S. and Wang, P.Y.: Seamless Live Migration of Virtual Machines over the MAN/WAN, *Future Generation Computer Systems*, Vol.22, No.8, pp.901–907 (2006).
22) 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 (2007).
23) NICT 仮想化ノードプロジェクト：仮想化ネットワーク技術仕様，情報通信研究機構 (2009).
24) XML-RPC Home Page. http://www.xmlrpc.com/
25) 吉澤政洋，垂井俊明，沖田英樹：サーバ仮想化環境における管理コストを低減するネットワーク管理システムの実装および評価，電子情報通信学会 CQ/NS/ICM 研究会，NS2009-116 (2009-11).

(平成 22 年 6 月 7 日受付)
(平成 22 年 12 月 1 日採録)

金田　泰（正会員）

1956 年生．1979 年東京大学工学部計数工学科卒業．1981 年同大学大学院情報工学専門課程修了．同年（株）日立製作所中央研究所入所後，Fortran コンパイラ，ベクトル・プロセッサによる記号処理と論理型言語処理，創発的計算のモデル，百科事典等からの情報抽出/検索/組織化，ネットワークとポリシ，仮想環境型コミュニケーション，ネットワーク仮想化等の研究開発に従事．工学博士．ACM，IEEE，ソフトウェア科学会各会員．

垂井　俊明（正会員）

1962 年生．1985 年東京大学工学部電子工学科卒業．1987 年同大学大学院工学系研究科情報工学専門課程修士課程修了．同年（株）日立製作所入社．入社以来中央研究所に勤務．現在，主任研究員．並列計算機，サーバアーキテクチャ，自律管理システム，仮想化の研究開発に従事．IEEE/Computer 会員．