

RISE OpenFlow テストベッド

河合 栄治¹ 金海 好彦² 齋藤 修一^{1,2} 石井 秀治¹ 高田 智明¹ 小林 和真^{1,3} 下條 真司^{1,4}

概要: 従来のインターネットインフラでは一体化して設計されることが多かった制御プレーンとデータプレーンを分離し、より柔軟な通信制御を実現する OpenFlow 技術が注目を集めている。我々のグループでは、この OpenFlow 技術の広域インフラ展開に取り組み、現在では OpenFlow ネットワークテストベッド RISE (Research Infrastructure for large-Scale network Experiments) としてサービスを開始している。本論文では、その取り組みの中で行った技術的検討について述べる。

キーワード: 新世代ネットワーク、広域ネットワーク、運用管理、仮想ネットワーク

RISE OpenFlow Testbed

EIJI KAWAI¹ YOSHIHIKO KANAUMI² SHU-ICHI SAITO^{1,2} SHUJI ISHII¹ TOMOAKI TAKATA¹
KAZUMASA KOBAYASHI^{1,3} SHINJI SHIMOJO^{1,4}

Abstract: In the traditional design of the Internet infrastructure, the control plane and the data plane have been often tightly-coupled. The OpenFlow technology recently attracts much attention of researchers and developers because it decouples the two planes and achieves flexible control over network communications. We have been deploying a wide-area OpenFlow network to demonstrate its potential in wide-area environments. We are now operating the OpenFlow network as an OpenFlow testbed RISE (Research Infrastructure for large-Scale network Experiments). In this paper, we describe the technical consideration took in the development of our OpenFlow network.

Keywords: New Generation Networks, wide-area networks, operation and management, virtual networks

1. はじめに

我々のグループでは、OpenFlow 技術の広域インフラ展開に 2009 年より取り組み [1]、現在では OpenFlow ネットワークテストベッド RISE (Research Infrastructure for large-Scale network Experiments) としてサービスを開始している [2]。開発当初の OpenFlow 技術は、キャンパスネットワークなどの中規模ネットワークを対象にしていた [3] という背景のなかで、我々のグループでは広域ネットワークへの適応性について重点的に検討を行ってきた。これは、将来インターネット技術/新世代ネットワーク技術の研究開発において、その広域性は重要な要因の 1 つであるためである。

近年では、OpenFlow ネットワークを広域に展開する試みも世界的に増えてきており [4], [5]、我々もそれらとの連携を進めているところである。本論文では、我々がこれまでに取り組んできた広域 OpenFlow ネットワーク構築の中で行った技術検討について述べる。

2. OpenFlow について

2.1 OpenFlow の仕組み

OpenFlow 技術は、イーサネット通信機器において、そのパケット転送を外部のコントローラから制御することを可能にする技術である。具体的には、OpenFlow スイッチは、Flow Entry と呼ばれるデータに記述されたマッチングルール (パケットヘッダ情報などに対するルール) に従ってパケットを識別し、マッチしたパケットに対しては同じく Flow Entry に記述されている内容に従って処理 (パケットの特定ポートへの転送や廃棄など) を行う。一方で、OpenFlow コントローラは、OpenFlow スイッチの外部から、この Flow Entry の操作を行う。

¹ 独立行政法人情報通信研究機構
National Institute of Information and Communications Technology
² 日本電気株式会社
NEC Corporation
³ 倉敷芸術科学大学
Kurashiki University of Science and the Arts
⁴ 大阪大学
Osaka University

この OpenFlow の仕組みを用いることにより、従来のインターネットインフラでは一体化して設計されることが多かった制御プレーンとデータプレーンを分離することにより、より柔軟な通信制御が実現可能となる。すなわち、ユーザの要求を制御プレーンに実装することで、その要求にあったパケット転送をしてくれるネットワークインフラを構築することが可能となる。

2.2 Software Defined Network の概念

OpenFlow の柔軟なネットワーク制御の概念をさらに一般化し、ネットワークリソースのオペレーティングシステムを実現する、Software Defined Network (SDN) と呼ばれる概念も登場してきている [6]。SDN では、OpenFlow をネットワーク機器の Hardware Abstraction Layer (HAL) として位置づけており、ネットワークリソースの汎用的な管理フレームワークをネットワーク機器の外部にソフトウェアにより構築することを目指している。言い換えれば、SDN の概念においては、OpenFlow は Flow Entry の形で抽象化されたネットワーク機器のリソースのインタフェースということになる。現在、OpenFlow は特定のサービスに特化したネットワーク制御を実装する手段として扱われることが多いが、将来的には SDN が具体化され、ネットワーク資源管理の汎用的なアーキテクチャが実現されることが期待されている。

2.3 OpenFlow の応用例

OpenFlow の応用例としては、クラウド環境において、仮想マシンやネットワークストレージ、ユーザ VLAN 等の仮想化されたリソース群を柔軟に連結したり [7]、データセンター間の広域ネットワークにおいて、サービス要求と連携してネットワークを制御したりする [8] ことなどが挙げられる。

一方で、この OpenFlow 技術を新世代ネットワーク技術の実装、展開のためのプラットフォームとして活用することも可能である。OpenFlow はイーサネット通信機器の拡張として設計されているため、通信に用いるパケットのフォーマットは、イーサネット、IP、TCP/UDP といった従来のプロトコルに従う必要がある。しかし、各 OpenFlow スイッチにおける転送アルゴリズムを自由に実装することが可能であり、OpenFlow はプログラマブルなネットワークを提供する。我々は、OpenFlow 技術こうした OpenFlow の機能に着目し、新世代ネットワーク技術のためのテストベッドとして機能させることを目指している。

3. 広域 OpenFlow ネットワーク

先に述べたように、OpenFlow 技術は、キャンパスネットワークなどの中規模ネットワークを対象として当初は開発されていた。OpenFlow は、コントローラによる集中管理のアーキテクチャを採用していることから、広域ネットワークへの展開が困難である面があるのは否定できない。しかし、OpenFlow の実用化の観点から、その広域ネットワークへの展開は不可欠なものであり、そうした試みの中

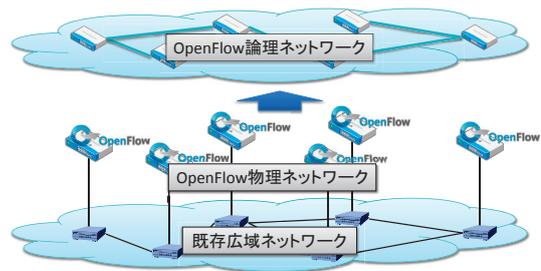


図1 OpenFlow 機器の既存広域ネットワークへの接続と論理 OpenFlow ネットワーク (データプレーン)。

から広域性を実現する新しい技術が生まれる可能性も期待できる。さらには、新世代ネットワーク技術の研究開発プラットフォームとしても広域 OpenFlow ネットワークが必要である。

そこで、我々は 2009 年より OpenFlow ネットワークの広域展開の取り組みを行っている。OpenFlow を広域ネットワークへ展開するにあたり、検討が必要な必要点として機能性と運用性の 2 つが挙げられる。

3.1 機能性

3.1.1 既存広域ネットワークとの融合

OpenFlow を広域展開する上で、機能性においてまず考えなければならないのが既存広域ネットワークとの融合である。OpenFlow ネットワークを広域に導入するために広域の専用回線を用意するのは、試行的ユーザが大半を占めている時点では費用対効果の観点から非現実的である。OpenFlow では、現行のインターネット通信で広く使用されるイーサネットのパケットフォーマットが使用されているため、既存広域ネットワークで OpenFlow パケットを伝送することを考えるべきである。

図1は、OpenFlow スイッチを既存広域ネットワークに接続し、その上に構築された論理的な OpenFlow ネットワークの概念を示したものである。これにより、OpenFlow コントローラに対しては、専用の OpenFlow ネットワークが構築されているように見え、実際の OpenFlow パケットの広域伝送は、既存広域ネットワークインフラにより行われる仕組みが実現される。

3.1.2 複数の OpenFlow ネットワークの相互接続

広域 OpenFlow ネットワークを実現する上で、機能面において考えなければならないもう 1 つの点は、複数の OpenFlow ネットワークの相互接続の仕組みである。先に述べた論理 OpenFlow ネットワークにより、物理的な制約を越えて広域 OpenFlow ネットワークが展開できるようになるが、世界規模で OpenFlow ネットワークを利用できるようにするためには、各地域に存在する OpenFlow ネットワークを相互接続し、より大規模な OpenFlow ネットワークを構築する手法が必要となる。

図2に、2つの OpenFlow ネットワーク A および B がある時に、A のユーザ^{*1}が A に加えて B を使用したい場合に可能な 3 つの方法を示している。いずれの場合も、データ

^{*1} ここで OpenFlow ネットワークのユーザとは、その OpenFlow ネットワークのコントローラに対する操作が可能な者のことである。

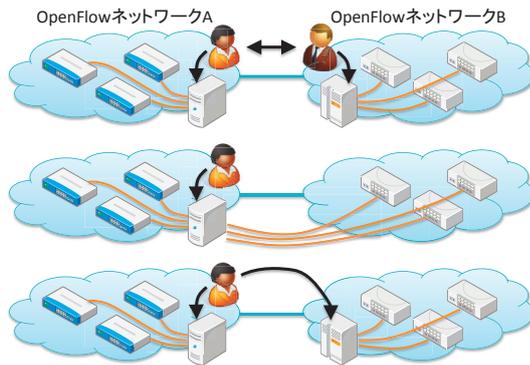


図2 データプレーンが接続された2つのOpenFlowネットワークの制御方法(AのユーザがBを使いたい場合で、上段の図はAのユーザがBのユーザに依頼して制御してもらう方法、中段の図はBのコントローラとしてAのコントローラを使うように設定する方法、下段の図はBのコントローラをAが操作する方法をそれぞれ示している)。

プレーンは何らかの方法により相互接続されているものと仮定する(既存広域ネットワークを介したOpenFlowネットワークのデータプレーン接続方法については後述する)。

まず、AのユーザがBのユーザに対してBにおいて制御して欲しい内容を伝え、代理で制御してもらう方法がある(上段の図)。OpenFlowにおける具体的な操作(Flow Entryの操作)は非常に低レベルなものであることから、ユーザ間のコミュニケーションコストが高く、複雑な制御を行うには本手法は不向きである。例えば、従来のVLAN設定のように特定のフローのパス設定を依頼するといった単純な制御にとどめるのが現実的である。

次に考えられるのは、BのOpenFlowスイッチのコントローラとしてAのコントローラを設定し、AのコントローラでBを制御する方法である(中段の図)。これは、2つのOpenFlowネットワークAおよびBを合併し、単一のOpenFlowネットワークを構築することに相当する。OpenFlowスイッチとして、ソフトウェアスイッチ(例えば、Open vSwitch)や、仮想化技術を実装したハードウェアスイッチ内の仮想スイッチ(例えば、NEC IX8800やPF5240等で設定可能なVirtual Switch Instance)等を用いる場合には、動的にOpenFlowスイッチを生成し、ユーザに貸し出すことが容易であることから、本手法も選択肢の1つとなる。一方で、広域OpenFlowの実現という観点からは、コントローラの負荷が問題となったり、AとBが物理的に離れている場合にはコントローラとBのスイッチの間の通信オーバーヘッドが大きくなるといった欠点があり、OpenFlowの集中管理のアーキテクチャに起因する制約を解決しているとは言えない。

最後のものは、AのユーザがAのコントローラとBのコントローラの両方を操作する方法である(下段の図)。本手法は、OpenFlowネットワークの規模拡張性の課題を解決するアプローチとしては自然なものであるが、確立された手法は存在しておらず、現行のOpenFlowの仕様でも定義されていない。

また、本手法では、Bにおいてマルチユーザ環境を実現する必要があり、それには2つの方法がある。1つは、Bに

おいて各ユーザが設定するフローを識別する方法であり、例えばFlowVisor[9]のようなシステムを利用することで実現が可能である。FlowVisorでは、あらかじめ各ユーザに利用可能なFlow空間(Flow Entryとして設定可能なヘッダ情報により構成されるベクトル空間)を割り当て、お互いの制御が干渉しないようにマルチユーザ環境を実現することができる。もう1つは、占有して使用することのできるスイッチを各ユーザに割り当てる方法である。物理的にOpenFlow機器をユーザの数だけ用意するのは非現実的であるが、先に述べたソフトウェアスイッチや仮想スイッチを活用すれば、本手法を用いることは比較的容易である。

なお、相互接続された各OpenFlowネットワークにおいて、別々のコントローラを用いる場合、コントローラ間の通信方式についても考える必要がある。ここでも、複数のコントローラの上位により抽象化されたコントローラを設置する方法[10]や、コントローラ間の分散ネットワークを構築しデータを交換し合う方法[11]などが考えられる。

3.2 運用性

3.2.1 ネットワーク構造の単純化、明確化

今のインターネットは、自律分散型のアーキテクチャになっているため、ネットワークの複雑さに対する許容度が高いと言える。さらには、階層構造の導入により、リンク層とIP層が管理機能の観点から役割分担をしており、大規模ネットワークを現実的なコストで運用可能にしている。

一方で、OpenFlowネットワークは、いわば非自律的に動作し、ユーザは非常に低レベルな制御まで自由に行うことができる。見方を変えれば、コントローラの操作誤りやコントローラに実装した制御プログラムのバグにより、ネットワークを機能不全にしてしまう可能性が高いと言える。

そのようなOpenFlowネットワークを運用していくためには、日頃からノウハウの蓄積が必要であるが、少なくとも、ネットワーク構造を単純に保ち、運用者がネットワーク内で起きていることを理解可能にしておくことが、重要である。

3.2.2 遠隔サイトにおける物理操作の削減

先に述べたように、OpenFlowネットワークは、機能不全に陥る可能性が高いネットワークである。また、コントローラに誤りがなくても、スイッチが受信したパケットによっては高負荷となり操作不能となったりすることもある。そのような場合には、最悪の場合スイッチをリセットする必要等があるが、広域ネットワークの場合、そうした現地での物理的な操作は、非常にコストが高い。そのため、例えば遠隔からの電源操作を可能にする、スイッチ操作のためのネットワークを制御プレーンやデータプレーンが輻輳を起こしても影響がないように別回線にするなど、工夫をすることが重要である。

3.2.3 既存ネットワークの運用との連携

先に述べたように、広域OpenFlowネットワークを構成するには、既存広域ネットワークと融合する必要がある。そのため、既存広域ネットワーク側の運用者との連携を考える必要がある。

表 1 仮想 L2 ネットワーク技術の例 .

7 層	OpenVPN
4 層	Ethernet over GRE, L2TP/IPsec
3 層	Ethernet over IP (IPsec)
2.5 層	Ethernet over MPLS, VPLS
2 層	VLAN, Q-in-Q, MAC-in-MAC (PBB)

一般的に、現行のインターネット通信機器は、頻繁な設定変更に対応できないことが多い。一方で、OpenFlow ネットワークは、ソフトウェアで実装されるコントローラにより自由に設定変更が可能である。極端な例では、パケット単位で転送制御を変更することが可能である。

そのため、既存広域ネットワークの役割と、OpenFlow ネットワークの役割を考慮し、既存広域ネットワーク上への OpenFlow ネットワークの収容手法を設計する必要がある。OpenFlow によって何を実現したいのかに依存するが、一般的には既存広域ネットワーク側の操作を必要最小に抑制することが重要である。

4. 広域 OpenFlow ネットワーク展開における技術的検討の詳細

次に、我々が広域 OpenFlow ネットワークを実際に展開する際に検討した具体的な技術詳細について述べる。OpenFlow は、低レベルな制御を実現する技術であり、それを用いてどのような高レベル機能を実現したいかによって、そのインフラ要件は変化する。我々の目的は、新世代ネットワーク技術の実装プラットフォームとして OpenFlow を位置づけ、できるだけ汎用的な広域テストベッドを実現することである。そのため、ここで述べる内容が広域 OpenFlow ネットワーク実現のために必須というわけではない点については注意が必要である。

4.1 既存広域ネットワークへの収容方法

3.1 節で述べたように、広域 OpenFlow ネットワークを実現するためには、既存広域ネットワーク上に OpenFlow ネットワークを収容するアプローチをとる必要がある。図 1 に示した、OpenFlow スイッチ間の仮想的な回線を実現するには、OpenFlow はイーサネットパケットを使用しているため、いわゆる仮想 L2 (イーサネット) ネットワークを OpenFlow スイッチ間に設定すればよい。

これまでにアプリケーション層からリンク層まで、様々な仮想 L2 ネットワーク技術が提案され、実際に使用されている (表 1)。原理的には、これらの仮想 L2 ネットワーク技術を OpenFlow スイッチ間接続に使用することができ。我々は、これらの仮想 L2 ネットワーク技術の中で、2 層の技術を用いることにした。

これは、我々が構築したい OpenFlow ネットワークがテストベッドとしての目的を持っているため、できるだけ OpenFlow スイッチ間の論理接続と、実際の物理ネットワークを静的に関連付けることにより、OpenFlow 上でのパケット転送制御が、実際の物理ネットワーク上での転送制御に 1 対 1 に反映される方がよいと考えたためである。また、OpenFlow ネットワークの運用ノウハウの蓄積等がない導

入段階では、静的な管理が可能で 2 層の技術を用いることが望ましいという理由もある。3 層以上の仮想ネットワーク技術を用いると、経路がインターネットプロトコルにより自律的に制御されてしまうため、実際の物理ネットワーク経路の管理が困難になるとともに、テストベッド上で実施される実証実験にとってもそのことが外乱要因となる可能性がある。

なお、2.5 層の技術を用いなかったのは、OpenFlow ネットワークの展開を開始した当時の JGN2plus のインフラでは、MPLS のサポートが十分でなかったためである*2。

一方で、現行のインターネットインフラでは、途中経路上のスイッチでの対応が必要となる 2 層の仮想 L2 ネットワーク技術が利用できない環境も多く存在する。3 層以上の技術では、エンド側のみの対応で仮想 L2 ネットワークを利用することができることから、3 層以上の技術の方がより手軽に利用できると言える。そのため、広域 OpenFlow ネットワークの構築方法の検討の観点からは、3 層以上の接続技術を用いた場合の OpenFlow による制御の実効性の検証等も行う必要が将来的にはあると考えている。

4.2 既存広域ネットワーク上での技術課題

既存広域ネットワークで OpenFlow トラフィックを伝送する L2 ネットワークを作成する際に問題となるのが MAC アドレス学習である。イーサネットは物理的/論理的に木構造のパケット伝送経路を構築するが、OpenFlow は非木構造の経路を構築することが可能である。

図 3 の例では、クライアントからサーバへのパケットを IDS のようなノードにも転送したい場合を示している。ここでは、パケットを右側の OpenFlow スイッチまで転送し、そこでパケットを複製し、1 つは IDS へ、1 つはそれまでの経路を逆行して中央の OpenFlow スイッチに転送し、最終的にサーバへ転送している。一般的に、仮想 L2 ネットワークは、図 3(a) に示したようにイーサネットスイッチとして動作し、パケットの MAC アドレスを学習することから、複数のポートに同一 MAC アドレスが出現することとなり、機能しない場合がある。

この問題を回避するためには、MAC アドレス学習を停止し、巨大なブロードキャストドメインを構成するか、MAC アドレス学習のドメインを OpenFlow スイッチ間の個々の接続に限定し、細分化する手法がある。しかし、前者の手法は、帯域を無駄に消費するブロードキャストドメインを、高価な広域ネットワーク上で構築することになり、非現実的である。さらには、OpenFlow でスイッチ間をマルチポイント接続する手法は原理上可能であるものの、その場合はマルチポイント接続された全てのスイッチに Flow Entry を設定する必要があり、規模拡張性の観点からも問題である。以上の検討により、図 3(b) に示す通り、OpenFlow スイッチ間はポイントツーポイント接続とし、そのための仮想 L2 ネットワーク、すなわち L2 トンネルを個別に用意するのが現実解となる。

L2 トンネルの場合、MAC アドレス学習が原理上問題と

*2 現在は、2 層の技術に加え、2.5 層の技術も使用している。

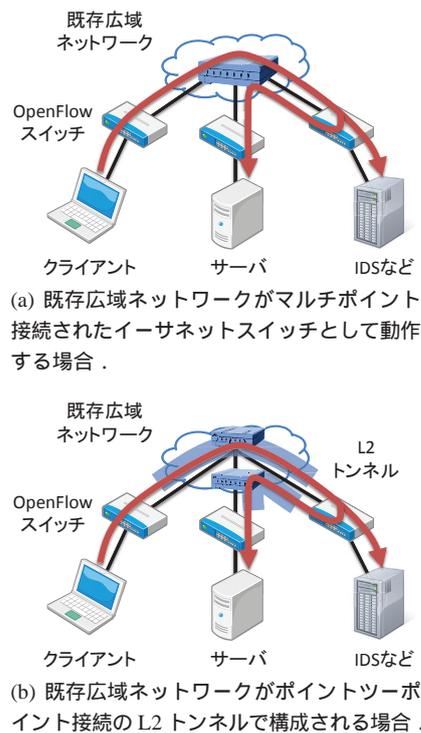


図3 非木構造の経路の例.

なることはなくなるが、物理的には既存広域ネットワークを構成するスイッチの MAC テーブルリソースを、そこを通る全 L2 トンネルにより共有することとなる。そのため、L2 トンネルの数が多くなったり、接続されるエンドホストの数が多くなったりした場合は L2 トンネルを使用するとともに、MAC アドレス学習を停止するといった設定も必要になってくるだろう。

4.3 L2 トンネルと OpenFlow スイッチの接続方法

L2 トンネルと OpenFlow スイッチを接続する方法には、OpenFlow スイッチが L2 トンネルを識別する方法が必要となる (図 4)。L2 トンネルを識別するには、物理的に異なる接続を用いる方法 (図 4(a)) と、物理的接続を共有しつつ論理的に異なる接続を用いる方法がある。後者は、さらに OpenFlow 側と L2 トンネル側で異なる論理的識別方法を用いる場合 (図 4(b)) と、同じ論理的識別方法を用いる場合 (図 4(c)) に分類することができる。

4.3.1 物理的接続により識別する方法

本手法は、L2 トンネルの数だけ、OpenFlow スイッチとの物理接続を用意する方法である。この場合、OpenFlow としては L2 トンネルを意識することなく、物理的に OpenFlow スイッチ間が接続されているかのように扱うことができる。問題点としては、L2 トンネルの数だけ OpenFlow スイッチおよび既存広域ネットワーク側のスイッチの物理ポートを消費するため、規模拡張性に欠ける点が挙げられる。

4.3.2 論理的接続により識別する方法その 1 (OpenFlow 側と L2 トンネル側で異なる識別方法)

本手法は、OpenFlow スイッチと L2 トンネルを論理的に接続する方法であり、かつ OpenFlow スイッチが論理接続を識別するために用いる情報と、L2 トンネルに用いる情

報が異なる場合である。そのため、既存広域ネットワーク側のスイッチは、OpenFlow スイッチで識別に使用している情報と、L2 トンネルの識別で使用している情報の関係を管理しなければならない。

4.3.3 論理的接続により識別する方法その 2 (OpenFlow 側と L2 トンネル側で同じ識別方法)

本手法は、OpenFlow スイッチと L2 トンネルを論理的に接続する方法であり、かつ OpenFlow スイッチが論理接続を識別するために用いる情報と、L2 トンネルに用いる情報が同じ場合である。そのため、L2 トンネルがそのまま OpenFlow スイッチまで延長しているイメージとなる。

4.4 L2 トンネル技術と接続方法

以上の考察を元に、具体的な L2 トンネル技術の使い方を検討することができるようになる。我々は、次の 3 つの L2 トンネル技術について考えた。

- タグ VLAN (802.1Q)
- Q-in-Q (802.1ad)
- MAC-in-MAC (802.1ah)

ここではスペースの関係で詳細を省略するが、検討の結果として Q-in-Q を図 4(a) の構成で使用の方針とした。検討に関する詳細については、[12] を参照のこと。具体的な接続構成については次章で述べる。

5. RISE の基本アーキテクチャ

3 章および 4 章での検討を基礎として、RISE の基本アーキテクチャを設計した。RISE アーキテクチャの基本的な要件は次の通りである。

- エンドユーザの packets を OpenFlow ネットワーク内で容易に識別できること。
- L2 トンネル技術としては Q-in-Q を使い、OpenFlow スイッチとの接続には物理接続をトンネルの数だけ用意すること (図 4(a) の方式)。
- 既存広域ネットワーク側の動的な設定変更はできる限り少なくし、かつ既存広域ネットワーク側のスイッチで使用する物理ポートを節約すること。

これらの要件を満たすアーキテクチャを検討し、RISE の各サイトにおいて図 5 に示すスイッチ構成とした。特徴としては、Edge OpenFlow Switch (E-OFS) と Distribution OpenFlow Switch (D-OFS) と呼ぶ 2 台の OpenFlow スイッチを用い、それぞれ役割を明確にしている点である。

E-OFS は、ユーザホストを収容するための OpenFlow スイッチである。この E-OFS では、各ユーザに対して使用する物理ポートを割り当て、その物理ポートに対して、仮想スイッチを構成している。具体的には、OpenFlow スイッチとして NEC の IP8800 を用いており、その Virtual Switch Instance (VSI) 機能を設定している。この仮想スイッチによるユーザホストの収容により、特定のユーザの通信に対する緊急の制御を行う (例えば、ループトラフィック等によりネットワークを過負荷状態にしているような場合に通信を停止する制御を実施する) 際に、他のユーザの通信に影響を与えなくて済む。

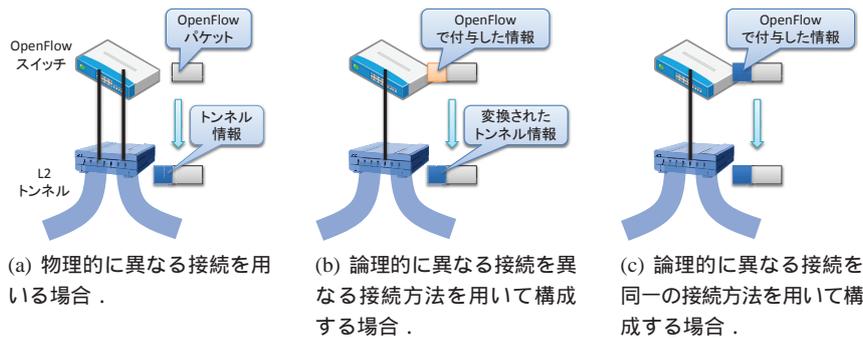


図4 L2 トンネルと OpenFlow スイッチの接続方法の分類。

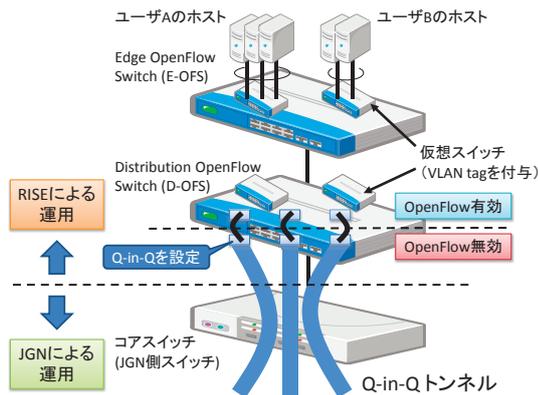


図5 RISE の各サイトにおけるスイッチ構成。

D-OFS では、一部の物理ポートの OpenFlow 機能を無効にし、通常のスイッチとして動作させ、Q-in-Q の L2 トンネルを設定している。図 4(a) の方式では、各 L2 トンネルに対して OpenFlow スイッチへの物理接続を用意する必要があるが、その物理接続を D-OFS 内で実現している。これにより、既存広域ネットワーク側のスイッチの物理ポートの消費を削減している。

6. おわりに

本稿では、我々が 2009 年より取り組んでいる広域 OpenFlow ネットワーク展開において行った技術的検討について整理した。今後の RISE のテストベッドとしての機能の高度化およびそれに付随する要素技術の研究開発方針としては、次のものを予定している。

一つは、ユーザコントローラの受け入れ体制の確立である。現行の RISE インフラでは、原理的には仮想スイッチによるユーザ環境の隔離が実現可能であるではある。しかし、緊急時の制御などの運用まで含めた体制の検討が必要である。

もう一つは、分散コントローラについての研究開発である。現在、世界規模の OpenFlow ネットワークを構築するべく、米国の NDDI/OS³E プロジェクト、欧州の OFELIA プロジェクトとインフラの相互接続に取り組んでいるところである。そうした規模になると、コントローラの分散化が不可欠となり、何かしらの形での自律分散制御の仕組みの導入が求められると考えている。今後は、そうした技術の研究開発も行っていく予定である。

参考文献

- [1] Yoshihiko Kanaumi, Shuichi Saito, and Eiji Kawai. Deployment of a Programmable Network for a Nation wide R&D Network. In *Proceedings of the 2nd IEEE/IFIP International Workshop on Management of the Future Internet (ManFI 2010)*, Osaka, Japan, April 2010.
- [2] 独立行政法人情報通信研究機構. Openflow テストベッド. <http://www.jgn.nict.go.jp/ja/info/technologies/openflow.html>.
- [3] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. OpenFlow: Enabling Innovation in Campus Networks. *ACM SIGCOMM Computer Communication Review*, Vol. 38, No. 2, pp. 69–74, April 2008.
- [4] NDDI and OS³E. <http://www.internet2.edu/network/ose/>.
- [5] OFELIA – OpenFlow in Europe. <http://www.fp7-ofelia.eu/>.
- [6] Scott Shenker. The Future of Networking, and the Past of Protocols. In *Open Networking Summit 2011*, October 2011.
- [7] It's Time to Virtualize the Network – Network Virtualization for Cloud Data Centers. http://nicira.com/sites/default/files/docs/mediaresources/Nicira_Overview.White.Paper.pdf, 2012.
- [8] Urs Hölzle. Openflow @ google. Open Networking Summit 2012, April 2012.
- [9] Rob Sherwood, Glen Gibb, Kok-Kiong Yap, Guido Appenzeller, Martin Casado, Nick McKeown, and Guru Parulkar. Can the Production Network Be the Test-bed? In *Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation*, Vancouver, BC, Canada, October 2010.
- [10] Eric Pouyoul. Dynamic creation of end-to-end virtual networks for science and cloud computing leveraging OpenFlow/Software Defined Networking. In *TERENA Networking Conference 2012*, May 2012.
- [11] 金海好彦, 齋藤修一, 河合栄治. Openflow アーキテクチャの自律分散化の実現に向けて. インターネットコンファレンス 2009, October 2009.
- [12] Yoshihiko Kanaumi, Shu-ichi Saito, Eiji Kawai, Shuji Ishii, Kazumasa Kobayashi, and Shinji Shimojo. Deployment and Operation of Wide-area Hybrid OpenFlow Networks. In *Proceedings of the Fourth IEEE/IFIP International Workshop on Management of the Future Internet (ManFI 2012)*, Maui, Hawaii, USA, April 2012.