

広域SDNテストベッドのためのRISE Orchestrator

伊藤 暢彦^{1,a)} 石井 秀治^{1,2} 齋藤 修一^{1,2} 金海 好彦^{1,2} 河合 栄治¹ 下條 真司^{1,3}

概要: インターネットは情報社会を支える最も重要なインフラとなった。社会の継続的な発展には、研究者が利用可能なテストベッドが必要不可欠である。そこで、我々は広域SDN(Software Defined Networking) テストベッドである RISE(Research Infrastructure for large-Scale network Experiments) を研究開発し、運用を開始した。本論文では、運用を通して直面した課題を示し、その課題を解決する手法を示す。さらに、我々は広域SDN テストベッドの統合運用システムを開発した。本論文では、広域SDN テストベッドの新制御システムと統合運用システムで構成される RISE Orchestrator を提案する。

キーワード: SDN, オープンフロー, テストベッド, オーケストレータ

RISE Orchestrator for Wide-Area SDN Testbed

Abstract: The Internet succeeded in acquiring the position of the most important infrastructure. It's requisite to prepare the Testbed for researchers if we hope consecutive social growth. Therefore, we have been developing and operating the RISE (Research Infrastructure for large-Scale network Experiments) which is a wide-area SDN (Software Defined Networking) Testbed. In this paper, we show some issue in terms of the actual SDN operation and propose the method to resolve some issue. Moreover, We developed the integrated operation system for wide-area SDN Testbed. We propose RISE Orchestrator which has advanced control system and integrated operation system for wide-area SDN Testbed.

Keywords: SDN, OpenFlow, Testbed, Orchestrator

1. はじめに

インターネットは今日の情報社会を支える最も重要なインフラとなった。社会が継続的に発展するには、インフラが抱える課題を克服しなければならず、それらの課題を解決するために、研究者は次世代のインターネットアーキテクチャを研究している。

次世代のネットワークの1つとして OpenFlow Network (OFN)[1] が誕生した。近年では、Software Defined Networking (SDN) の概念が広まり、OFN は SDN を実現するためのプロトコルの1つとなった。

OFN は OpenFlow Controller (OFC) と OpenFlow

Switch (OFS) で構成される。OFN では、制御権は OFC にあり、OFC が OpenFlow Protocol を通して OFS を制御する。

情報社会が継続的に発展するには、新たな方式やアーキテクチャを検証することが可能なテストベッド環境が必要不可欠である。インターネットは情報社会を支える重要な基盤となった今、実インターネット網上で方式の検証を行うことは困難である。企業、個人問わず、セキュリティは極めて重要であり、多くの研究開発が行われるキャンパスネットワークでさえ、制限が多く、大規模な実験環境を準備することは非常に難しい。

また、新方式の大規模ネットワーク適用に関するノウハウは蓄積されておらず、小規模ネットワークでは顕在化していない問題が、大規模ネットワークでは発生することがある。そのため、方式の検証が可能な大規模テストベッド環境を構築することは重要である。

さらに、研究成果を含んだ装置を開発するには、企業と連携して製品化する必要があり、また、それらを既存のイ

¹ 情報通信研究機構
NICT, KDDI Building, 1-8-1 Otemachi, Chiyoda-ku, Tokyo
100-0004, Japan

² 日本電気株式会社
NEC Corporation, Tokyo 108-8001, Japan

³ 大阪大学
Osaka University, Osaka 567-0047, Japan

a) n-itoh@nict.go.jp

インターネットに配備して実験することは高コストであり、実現が難しい。

そこで、我々は研究者や開発者が試験可能な SDN テストベッドである RISE(Research Infrastructure for large-Scale network Experiments) を Layer-2 ネットワークである JGN-X[2] 上に構築した。

RISE の研究開発は 2009 年より行っており、初期の RISE(RISE1.0 と呼ぶ) では、広域 SDN テストベッドの構築を実現した。2012 年には、複数のユーザが同時に利用可能なマルチユーザ化を実現した RISE(RISE2.0 と呼ぶ) を開発した [3]。

しかしながら、RISE2.0 には 2 つの課題が存在する。1 つ目は Capacity 問題である。テストベッドの規模はポート数に制約される。また、ユーザの収容効率が最も低い場合は、同時に利用可能なユーザ数は 16 となる。

2 つ目は、Topology 問題である。RISE2.0 では、ユーザが望むトポロジを提供できない場合がある。

さらに、我々は、RISE2.0 の実運用を通して、広域 SDN テストベッドのような大規模ネットワークの運用には、運用コストを削減するための統合運用システムが必要であることがわかった。

本論文では、技術的課題と運用課題を解決する RISE Orchestrator を提案する。

2. 広域 SDN テストベッド

2.1 広域 SDN テストベッドのモチベーション

広域テストベッド環境を構築することは重要である。大規模ネットワークで試験することはサービスの継続性を検証する上で重要である。

したがって、我々は 2009 年から広域 SDN テストベッドである RISE を JGN-X 上に実装し、運用を開始した。RISE は多くの研究者に利用され、研究開発のインフラとして貢献した。また、運用による知見は R&E ネットワークのコミュニティにおいて共有を図った [6], [7], [8], [9], [10]。

2.2 RISE の設計思想

RISE の設計では、以下の 4 点について考慮している。それは、性能、マルチユーザ、オペレーション、柔軟性である。

2.2.1 性能

RISE では、広域 SDN テストベッドをハードウェアスイッチで構築することを念頭に置いた。近年では、Open Virtual Switch などのソフトスイッチが存在する。しかしながら、ソフトスイッチの packets 転送能力はハードウェアスイッチの packets 転送能力と比較すると低いため、性能面を鑑みてハードウェアスイッチによる広域 SDN テストベッドを構築する。

2.2.2 マルチユーザ

複数のユーザが同時に利用できるテストベッドを設計することは運用コストの削減につながる。ただし、各ユーザの環境は仮想的に独立する必要がある。OFN では、複数の OFC がネットワークを制御することが可能である。FlowVisor[4], OpenVirteX[5] を用いれば、各ユーザを分離でき、同一の物理インフラ上に論理的に複数のネットワークを構築することが可能である。

2.2.3 オペレーション

大規模ネットワークを運用する場合、オペレーションの容易性を考慮することは重要である。我々はオペレーションを容易にするために、以下の 2 点を考慮した。

1 つ目は JGN-X に与える影響を小さくすることである。JGN-X は、RISE 以外のサービスも提供している。そのため、RISE ネットワークは JGN-X の変更を伴わずに変更可能なアーキテクチャを目指す。

2 つ目はトラブルシューティングである。JGN-X は複数のサービスを提供している。RISE ネットワークにおいて障害が発生した際に、他のサービスへの影響を小さくするためにも障害から早急に回復することはサービス継続性の観点からも重要である。

2.2.4 柔軟性

ユーザが望むテスト環境はユーザ毎に異なる。我々は以下の 2 点を満足するテストベッドを目指している。

1 つ目はユーザ所望のトポロジを提供できることである。これまでのテストベッド運用を通して、ユーザが望むトポロジはユーザ毎に異なり、ループトポロジ、ダンベルトポロジなど様々なトポロジのリクエストを受けた。

2 つ目は拡張性である。近年では、NFV (Network Functions Virtualization) におけるサービスチェイニングが着目されている。我々は、NFV を鑑みて、ユーザに提供可能なサーバを各拠点に配置している。各ユーザはコンピュータ資源とネットワークを自由に使用できる。

2.2.5 OpenFlow の段階的導入

我々は、まず、OpenFlow の段階的な導入と移行を見据えて、既存のネットワーク上に仮想的な広域 SDN テストベッドを構築するアプローチをとる。

RISE ネットワークは JGN-X にオーバレイしたネットワークである (図 1)。図 1 では、Layer-2 ネットワークとそのネットワークを構成するスイッチは図の下側に示され、SDN とそれを構成するオープンフロースイッチは図の上側に示される。SDN は JGN-X 上に仮想的に構築されたネットワークである。そのため、図 1 のオープンフロースイッチ間のリンクは仮想リンクである。

RISE ネットワークは国内と海外の拠点で構成されているため、これらの拠点をういた広域ネットワークを実験ネットワークとしてユーザに提供できる。海外拠点は、ロサンゼルス、シンガポール、バンコクである。

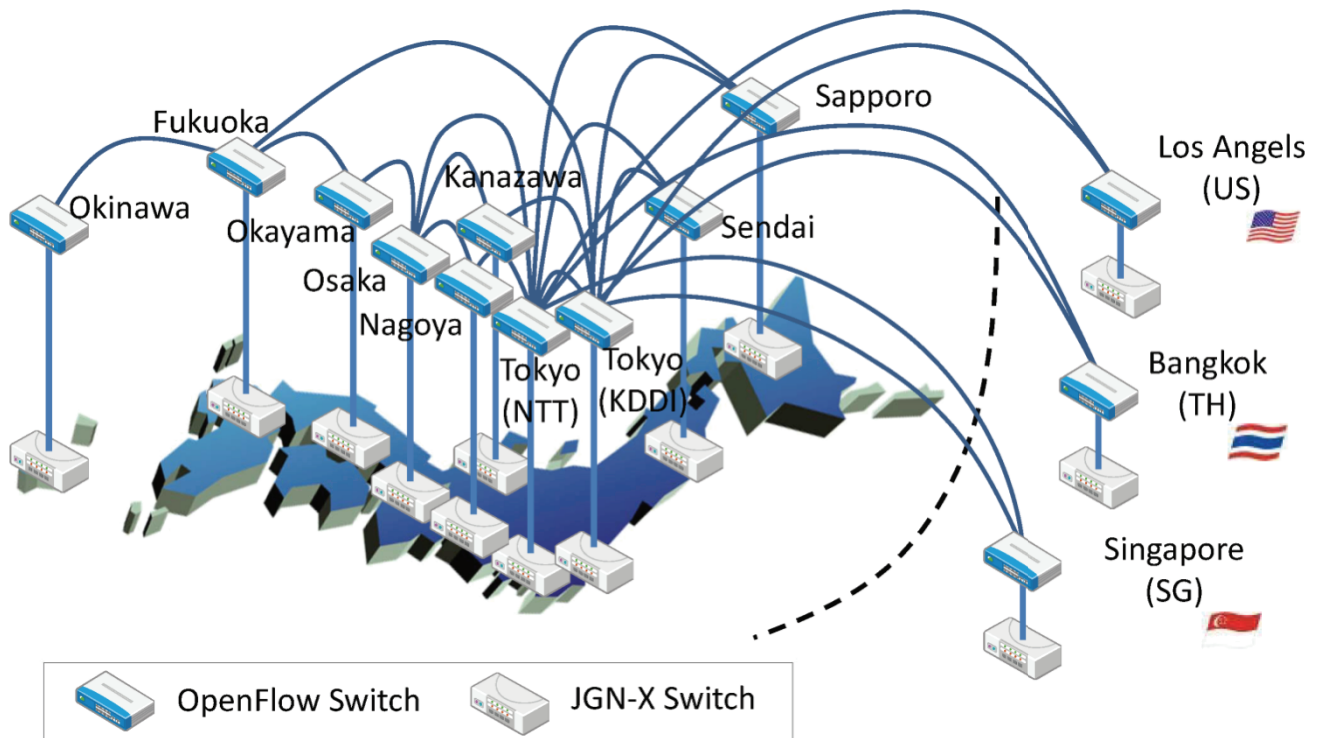


図 1 RISE トポロジ
Fig. 1 RISE topolog

3. RISE2.0

2.2 の設計思想に基づいたシステムを構築し、これまで運用を行ってきた [3]。本論文では、このシステムアーキテクチャを RISE2.0 と呼ぶ。

3.1 ユーザ識別子

我々のテストベッドは複数のユーザに同時に使用されることを想定している。そのため、RISE2.0 では、各ユーザを識別する必要がある。そこで、我々は、ユーザ毎に異なる VLAN ID を割り当てた。RISE2.0 では、VLAN ID を用いてユーザを識別する。

3.2 アーキテクチャ

RISE2.0 のアーキテクチャを図 2 に記載する。RISE2.0 のアーキテクチャは 2 つのオープンフロースイッチと 1 つのスイッチで構成される。2 つのオープンフロースイッチのうち 1 つはエッジオープンフロースイッチ (E-OFS)、残りの 1 つはディストリビューションオープンフロースイッチ (D-OFS) と呼ぶ。E-OFS と D-OFS はユーザによって制御されるスイッチであり、複数のユーザで共有されるスイッチである。E-OFS と D-OFS 間の物理リンク数は 1 である。また、D-OFS と JGN-X 間の物理リンク数も 1 である。

E-OFS はユーザホストからパケットを受信すると、ユー

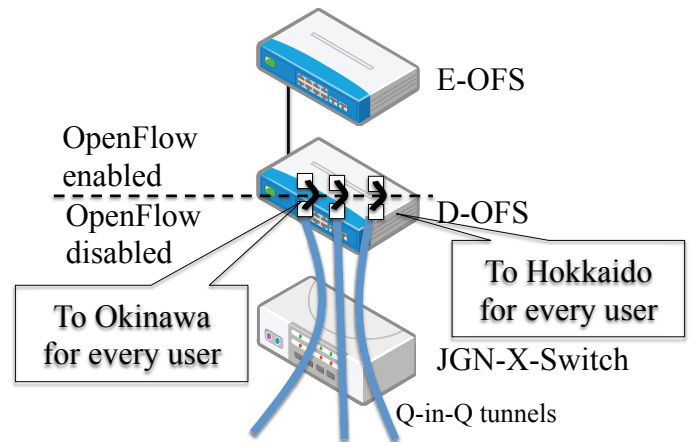


図 2 RISE2.0 のアーキテクチャ
Fig. 2 RISE2.0 Architecture

ザを一意に示す VLAN ID を付与し、ユーザホストに向かうパケットを受信すると、VLAN ID を削除する。ユーザには物理ポート単位でネットワークを割り当てているため、E-OFS は受信したポート情報よりユーザを一意に識別することが可能である。

次に、D-OFS について述べる。D-OFS はオープンフローネットワークと JGN-X を接続するスイッチである。そのため、D-OFS にはオープンフロースイッチとして動作するポート (このポートを OpenFlow-enabled ポートと呼ぶ) と従来スイッチとして動作するポート (このポートを OpenFlow-disabled ポートと呼ぶ) を設定し、それらの

ポートは物理リンクで結線している。本論文では、この結線をハードループと呼ぶ。

ハードループは対向拠点数分結線され、ハードループと対向拠点のマッピングは1対1対応として設計している。なお、対向拠点数とは、拠点数から自拠点を除いた、拠点数-1を意味している。各ハードループの宛先は固定的であり、例えば、図2の左のハードループは沖縄、右のハードループは北海道のように設定する。つまり、各ハードループの宛先は全てのユーザで同一である。ユーザは宛先拠点に応じたハードループにパケットを送信するように制御することで、パケットを正しく転送することができる。そして、ハードループにより、OpenFlow-disabledポートにパケットが届くと、Q-in-Q[11]を用いて宛先拠点を示すVLAN IDを付与する。JGN-Xスイッチはパケット内のVLAN IDから宛先拠点を判断し、パケットを転送する。

RISE2.0のアーキテクチャは以下の点で優れている。メリットは、JGN-XスイッチとD-OFS間の必要リンクは1本であることである。JGN-Xは複数のサービスを提供しているため、JGN-Xスイッチは複数のサービスで共有されるスイッチである。そのため、JGN-XとD-OFS間のリンク数を抑制出来ることはメリットである。

4. RISE2.0の課題

RISE2.0には2つの課題が存在する。それは、多ユーザ多拠点を収容できない問題（Capacity問題と呼ぶ）、ユーザトポロジを柔軟に作成できない問題（Topology問題と呼ぶ）である。

4.1 Capacity問題

RISE2.0では、3つのCapacity問題を抱えている。1つ目は、D-OFSの必要ポート数である。D-OFSの必要ポート数は、E-OFSとの接続ポート数(1ポート)、JGN-Xとの接続ポート数(1ポート)、ハードループのポート数の合計値である。具体的には、D-OFSの必要ポート数 $P_{2.0}(A)$ は

$$P_{2.0}(A) = 1 + 1 + 2(A - 1) = 2A \quad (1)$$

となる。Aは拠点数である。つまり、RISE2.0のポート数は拠点数に対して線形増加となる。ハードウェアスイッチのポート数は有限のため、収容可能なユーザ数と拠点数はD-OFSのポート数により制約される。これは多拠点で構成される大規模テストベッドを構築する時の制約条件となる。

2つ目はVSI(Virtual Switch Instance)リソースの枯渇である。我々はOFSとしてNEC PF5240を使用している。PF5240には、1つの物理スイッチ上に複数の論理スイッチを構成する機能(VSIと呼ぶ)がある。論理スイッチの上限値は、ハードウェアの制約上、1スイッチあたり16

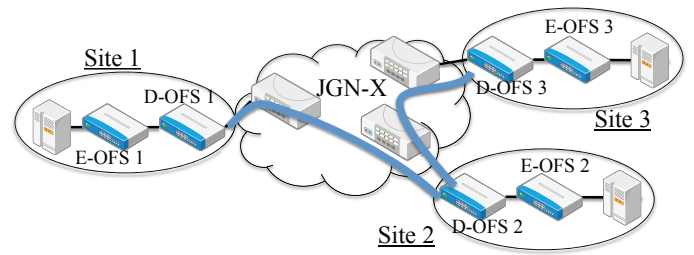


図3 RISE2.0のトポロジ例
Fig. 3 Example of RISE2.0 topology

である。RISE2.0では、E-OFSとD-OFS共にVSIで動作している。

VSIを利用した点は、テストベッドをハードウェアスイッチで構成することと、単一障害点を回避するためである。OpenVirteXやFlowVisorを用いて複数ユーザを同時に収容する場合、OFCとOFSの間にそれらを配備するアーキテクチャをとる。OpenVirteXやFlowVisorに障害が発生すると、全てのユーザのサービスが停止する。そこで、単一障害点問題を回避するために、OpenVirteXやFlowVisorを介すアーキテクチャとせず、VSIを用いたアーキテクチャとした。3つ目はハードループの宛先が固定的であることである。各拠点への送受信トラフィック量は、拠点毎に異なり、人気のある拠点には負荷が集中する。また、ハードループが故障した際に、故障していないハードループにトラフィックを振り分けることができず、復旧には新たにハードループを設定する必要がある。ケーブル断が発生した場合は、交換作業が伴い、障害復旧に時間を要する。

4.2 Topology問題

図3を用いてTopology問題を述べる。1拠点は2台のOFS(E-OFS, D-OFS)で構成され、3拠間で構成される例である。3拠点間はJGN-Xにより接続されている。D-OFS間のリンクはVLANによる仮想リンクを示している。

Topology問題は2つある。1つ目はユーザ所望のトポロジを提供できない点である。ユーザに提供するトポロジには必ずD-OFSとE-OFSの2台のOFSが各拠点毎に含まれる。これは、D-OFSおよびE-OFSの両方を経由しなければならない、ユーザがこの状況を望んでいない場合、要望に応えられていないことを意味している。さらに、オペレーションの観点では、管理・設定が必要なOFSが増えるため、オペレーションコストも増加する。

2つ目は拠点間をつなぐOFSにおいてはVSIリソースの枯渇が発生する点である。これは、エッジ拠点のVSIのリソースが空いている状況でも、中継拠点においてVSIの枯渇が発生すると、ユーザに提供するトポロジを作成できないことを意味する。4.1で述べたように、E-OFSとD-OFSはVSIモードで動作しており、その論理スイッチの上限値

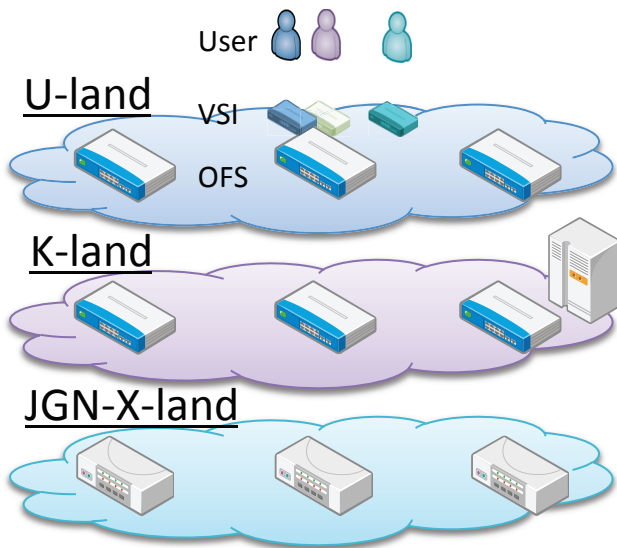


図 4 RISE Orchestrator のアーキテクチャ
Fig. 4 The architecture of RISE Orchestrator

は 16 であるため、枯渇する可能性がある。

フルメッシュなネットワークを構築すれば、中継拠点を介さずにトポロジを構築可能であるが、各拠点間をフルメッシュで構築するには、以下の VLAN を消費するため、大規模ネットワークでは、フルメッシュを構築することはコストの観点から難しい。

$$VLAN_{num} = (N(N - 1))/2 \quad (2)$$

5. RISE Orchestrator

本章では、広域 SDN テストベッドの新制御システムと統合運用システムで構成される RISE Orchestrator について説明する。

5.1 アーキテクチャ

RISE Orchestrator はユーザに提供するオープンフローネットワークの下位層にネットワークオペレーション用のオープンフローネットワークを配備した階層構造をとる。RISE Orchestrator のアーキテクチャを図 4 に示す。

RISE Orchestrator は 3 層構造をとる。上位層と中間層は OFS により構成され、上位 2 層はオープンフローにより制御される。

上位層はユーザに提供するスライスであり、本論文では、この層を U-land と呼び、U-land の OFS を U-land-OFS と呼ぶ。なお、U-land-OFS は VSI モードで動作する。U-land では、ユーザが作成したコントローラを用いて、U-land-OFS を自由に制御することが可能である。

中間層はオペレータのためのスライスであり、本論文では、この層を K-land と呼び、K-land の OFS を K-land-OFS と呼ぶ。なお、K-land-OFS は U-land-OFS とは異なり、非

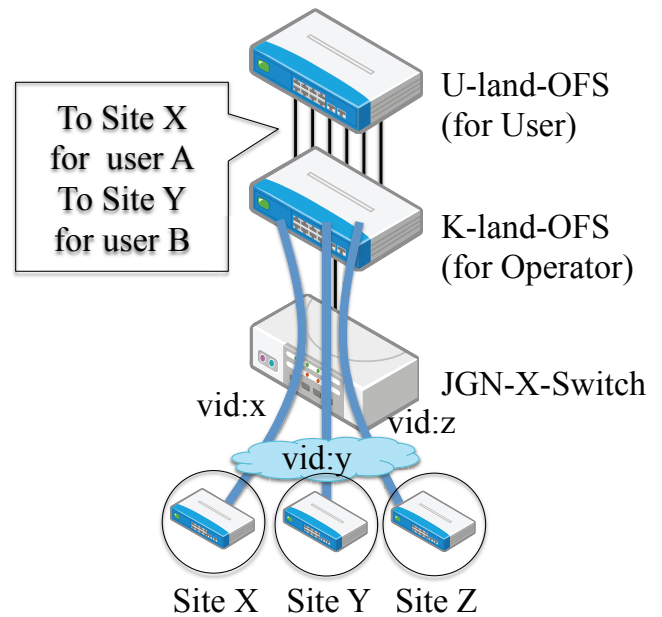


図 5 RISE Orchestrator の基本構成
Fig. 5 A basic set of RISE Orchestrator

VSI モードで動作している。つまり、K-land では、VSI リソースの枯渇発生しない。

次に、RISE Orchestrator のアーキテクチャについて図 5 を用いて詳細に述べる。具体的には、各層のスイッチ間の接続関係を明確にする。

まず、U-land-OFS と K-land-OFS 間について述べる。図 5 の両 OFS 間の必要リンク数は最大で対向拠点数 (拠点数-1) である。具体的には、必要リンク数は、次のとおりである。 H_n はユーザ n が必要とする対向拠点数である。 $Link_{num}$ はある拠点における必要リンク数を示している。

$$Link_{num} = \max(H_a, H_b, \dots, H_n) \leq (A - 1) \quad (3)$$

必要リンク数が式 (3) になるのは U-land-OFS と K-land-OFS 間の U-land-OFS のポートの対向が固定的に決まっているわけではなく、その対向はユーザ毎に異なるからである。この割当により、図 5 の U-land-OFS と K-land-OFS 間の左のリンクの U-land-OFS のポートは、例えば、ユーザ A には拠点 X の U-land-OFS と仮想的に繋がっているポート、ユーザ B には拠点 Y の U-land-OFS と仮想的につながるポート、とデザインすることができる。

また、このデザインは、4.1 で述べたポートの負荷分散と障害復旧に役立つ。例えば、図 5 の左のリンクが故障した時には、ユーザ A 用の拠点 X 行きとユーザ B 用の拠点 Y 行きの設定を、トラヒックの負荷分散も鑑みて、トラヒック量の少ないリンクを選択し、その選択したリンクに再割り当てすることが可能である。このデザインが成立する理由は、ユーザ毎に異なる VLAN ID を割り当てているため、K-land-OFS は VLAN ID によりユーザを識別できるからである。

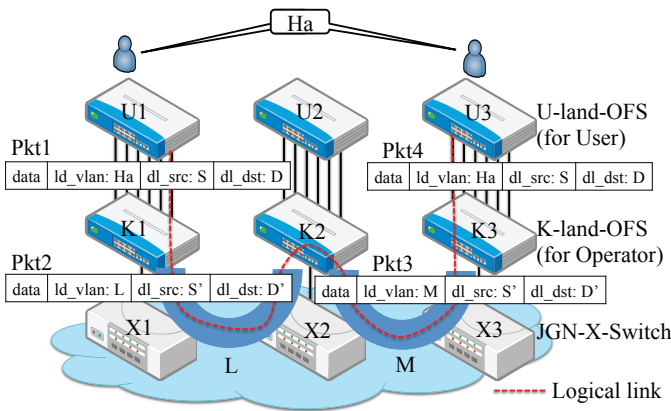


図 6 RISE Orchestrator の動作

Fig. 6 The operation of RISE Orchestrator

次に、K-land-OFSに必要なポート数を以下に示す。K-land-OFSの必要ポート数 $P_{proposed}(n, A)$ は、式(3)で示したリンクを収容するためのポート数とJGN-Xと接続するためのポート数の合計値である。Aは拠点数を示している。

$$P_{proposed}(n, A) = \max(H_a, H_b, \dots, H_n) + 1 \quad (4)$$

式(1), (3), (4)より、以下の式が成り立つ。RISE Orchestratorの必要ポート数の最大値はRISE2.0の半分であり、RISE2.0よりも多拠点収容可能なことがわかる。

$$P_{proposed}(n, A) \leq P_{2.0}(A)/2 \quad (5)$$

次に、VLANを用いた拠点間の接続について述べる。RISE OrchestratorではK-land-OFS間毎に異なるVLAN ID(図5中ではvidと表記)を割り当てている(例えば、拠点XへはVLAN ID: xを割り当てる)。JGN-X-SwitchはVLAN IDより、転送先を判断することができるため、K-land-OFSとJGN-X-Switch間のリンクは1本となる。

なお、本論文では、K-land-OFS間のリンクを拠点間リンクと呼ぶ。

5.2 RISE Orchestrator の動作

本節では、RISE Orchestratorの動作について述べる。RISE Orchestratorでは、U-land-OFS間が仮想的に直接繋がっているものと見なし、そのリンクを論理リンクと呼ぶ。図6はRISE Orchestratorの説明図である。まず、最初に図中の記号について説明する。U1~U3はU-land-OFS, K1~K3はK-land-OFS, X1~X3はJGN-X-Switchを示している。LとMは5.1にて定義した拠点間リンクである。最後に、Pkt1~Pkt3は各区間のパケットである。Pkt1はU1がK1に対して送信したパケット、Pkt2はK1がX1に対して送信したパケット、Pkt3はK2がX2に対して送信したパケット、Pkt4はK3がU3に対して送信したパケット、である。Logical linkは論理リンクを示している。

U1はユーザからパケットを受信すると、ユーザを識別

するIDをVLAN IDフィールドに埋め込み、送信する(Pkt1)。U1におけるユーザ端末側のポートとユーザを事前に紐づけた情報をOFCから受信しているため、U1はパケットを受信した際に、入力ポート情報よりユーザを一意に特定できる。

次に、K1がU1からパケットを受信すると、K1はまず始めに論理リンクを示すIDを発行する。本論文では、このIDを論理リンクIDと呼ぶ。なお、K-land-OFSとU-land-OFS間のK-land-OFS側のポートはユーザ毎の対向拠点と紐づけられている。ユーザの識別はLAN IDにより可能である。このIDはパケットのヘッダ情報と入力ポート情報をキーとして一意に特定可能なIDとして払い出される。

K-land-OFSでは、このIDをキーとしてパケットを処理するため、このIDをK1において埋め込む必要がある。本論文では、その埋め込み先をMACアドレスフィールドとした。その理由は、MACアドレスの書き換えはOpenFlowの標準スペックでサポートしているため、プロトコルを拡張すること無く、実現できるからである。

さらに、K1はU3への中継ノードであるK2にパケットを届けるために、VLAN IDに拠点間リンク情報を埋め込み、パケットを転送する(Pkt2)。ユーザを示すVLAN IDは上書きされるが、論理リンクIDにユーザを特定するIDを含めておくことで、上書きされても識別が可能である。

K2がパケットを受信すると、K2は論理リンクID、すなわち、MACアドレスフィールドをマッチ条件としたフローエントリに従い、パケットを転送する(Pkt3)。そのエントリには、K2-K3間の拠点間リンクを示すVLAN IDをVLAN IDフィールドに埋め込むよう定義されている。K2はそのフローエントリに従ってパケットを処理する。

K3はX3よりパケットを受信すると、K2は論理リンクID、すなわち、MACアドレスフィールドをマッチ条件としたフローエントリに従い、パケットを処理する。具体的には、K3はVLANとMACアドレスの書き換えを行う。VLAN IDフィールドはユーザを示すID、つまり、Haに書き換えられ、MACアドレスフィールドは元の値、つまり、送信元MACアドレスはSに、宛先MACアドレスはDに書き換えられる。K3はこれらの書き換えを行った後、U3に向けてパケットを送信する(Pkt4)。

5.3 RISE Orchestrator の効果

図7はRISE Orchestratorの効果を説明するための図である。Congestion OFSはVSIのリソースが枯渇しているU-land-OFSをしている。Not-congestion OFSはVSIのリソースが枯渇していないE-land-OFSをしている。K-land-OFS/JGN-X-SwitchはK-land-OFSとJGN-X-Switchを示している。K-land-OFSは非VSIモードで動作しているため、この階層においてはVSIのリソース不足は発生し

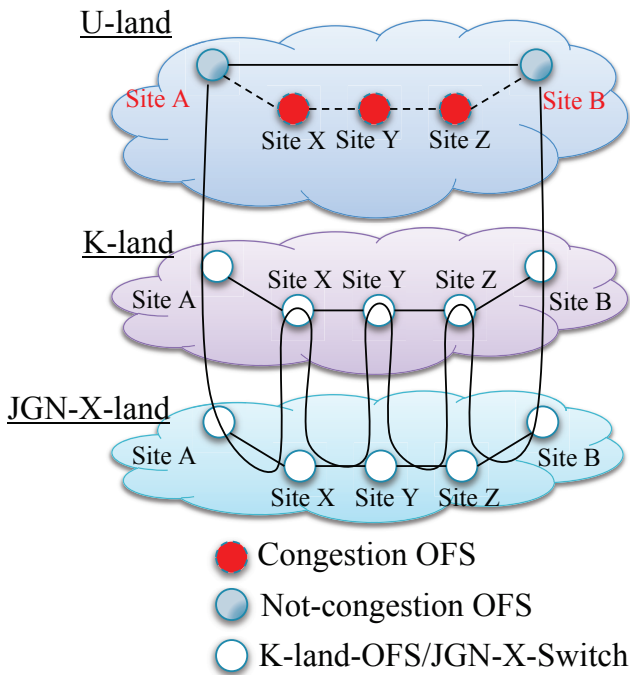


図 7 RISE Orchestrator の効果
Fig. 7 The effect of RISE Orchestrator

ない。

RISE2.0 の 1 つ目の問題は、ユーザが拠点 A と拠点 B の間に拠点 X, Y, Z が存在している場合、拠点 A と拠点 B が直接つながるトポロジを提供できない点である。2 つ目の問題は拠点間をつなぐ中継 OFS において VSI リソースが枯渇する問題である。

これらの問題が RISE Orchestrator では解決されることを示す。RISE Orchestrator では、オペレーションのための K-land を導入している。拠点 A と拠点 B の間に拠点 X, Y, Z が存在している場合でも、RISE Orchestrator は K-land-OFS において U-land にパケットを転送せずに、次の中継拠点にパケットを届けるために JGN-X-land に向けてパケットを送信する。これは、任意の 2 つの拠点間に中継拠点が存在している場合でも、任意の 2 つの拠点間が仮想的に直接つながるトポロジを提供できることを意味している。さらに、これは、U-land-OFS の VSI リソースの枯渇を軽減できることを意味している。

5.4 統合運用

近年、コンピュータ資源とネットワーク資源の両側面を考慮した研究が着目されており、運用者は、ユーザのコンピュータ資源とネットワーク資源に関する要望に応じたテスト環境を提供する必要がある。近年では、NFV が着目されており、その中のサービスチェイニングは、両側面を考慮した研究の 1 つである。

ネットワークの設定だけでなく、サーバ等のコンピュータ資源の設定も必要であるため、テストベッドを運用する

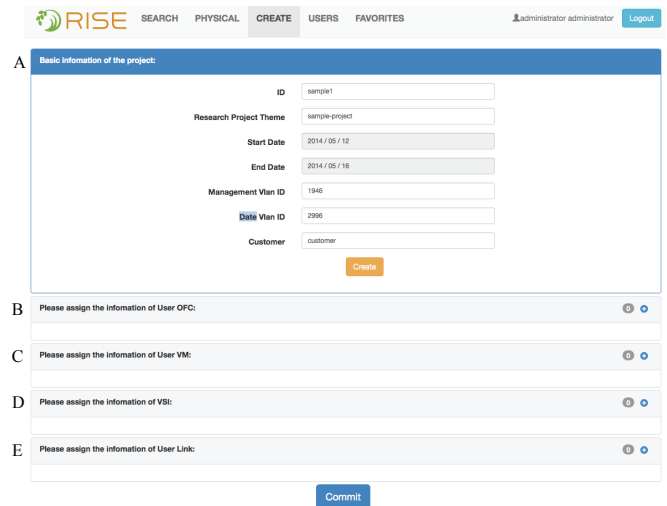


図 8 統合運用システム
Fig. 8 Integrated operational system

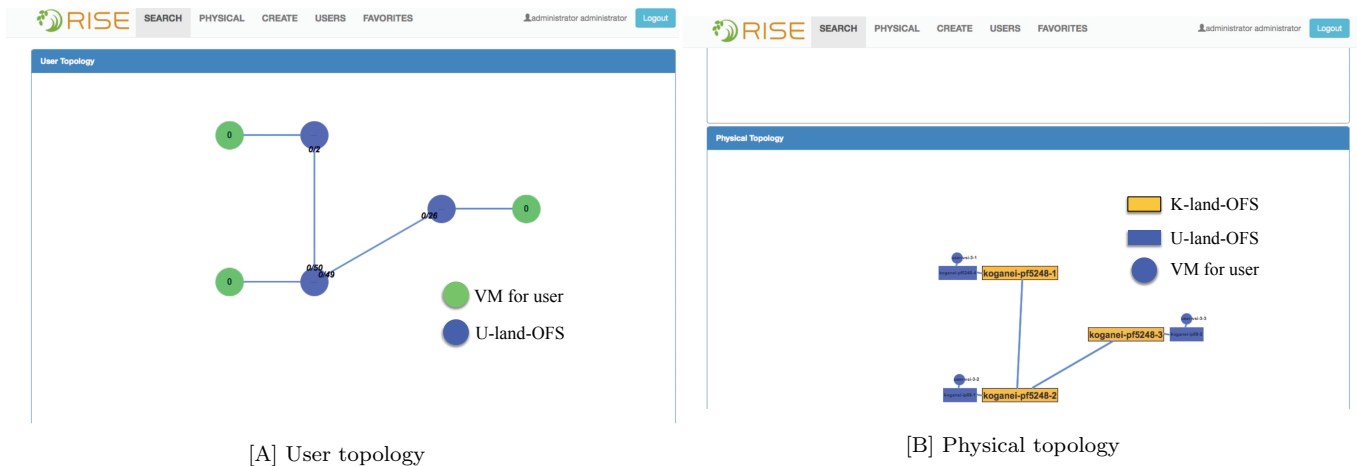
にあたってオペレーションを容易にすることは重要であり、ユーザ毎に異なる要望を満足したテストベッド環境を、可能な限り短時間で、構築できることが望ましい。さらに、我々は、遅延装置、Firewall, gateway などのネットワーク機能を、VM 上にインストールし、これらの機能をユーザに提供することを考えている。

そこで、我々は、オペレーションコストを容易にするために、統合運用システムを開発した。統合運用システムを図 8 と図 9 に示す。

図 8 はユーザが要求するテスト環境に合わせてパラメータを入力する画面である。入力画面は A から E の 5 つのパラメータで構成される。A では、プロジェクトの基本情報を入力する。具体的には、ユーザ毎のプロジェクト ID、使用期間などである。B では、ユーザ用 OFC の設定を行う。我々のテストベッドでは、仮想マシン (以下、VM) を提供している。ユーザは研究成果を VM にインストールすることができる。その際、ユーザが望む VM はユーザ毎に異なる。そこで、我々は、OS、コア数、メモリ、Disk 容量等に関するユーザからの要求を満足できるように、これらを選択できるようにした。C では、エンドホストのための VM のスペックを定義する。B と同様、ユーザ所望の VM を作成することができる。D では、ユーザが必要とする台数分の U-land-OFS の設定を行う。最後に、U-land-OFS 間のリンク (論理リンク) を設定を行う。

オペレータは、A から E のパラメータを入力するだけで、ユーザ毎のテスト環境を構築できる。その結果、テスト環境を構築するための時間の短縮とオペレーションコストを削減できる。

オペレータは、ユーザのための U-land のトポロジとオペレータのための U-land と K-land を描写したトポロジを本ツールで確認することができる。図 9 の [A]User topology は前者を、[B]Physical topology は後者を示している。[A]



[A] User topology

[B] Physical topology

図 9 論理トポロジと物理トポロジ

Fig. 9 topology of both logical and physical

の VM for user はユーザ用 VM を，U-land-OFS はユーザに提供するスイッチを示している．オペレータはこの GUI を確認することで，ユーザに提供しているトポロジを確認することができる．また，オペレータは [B] の Physical topology を確認することで，K-land-OFS の接続関係を確認することができる．

6. おわりに

本論文で，我々は広域 SDN テストベッドのための RISE Orchestrator を提案した．RISE Orchestrator では，仮想リンクの導入により，スイッチ間の接続関係を柔軟に定義可能とし，ユーザが望むトポロジの提供とスイッチの VSI リソースの負荷分散を実現した．さらに，我々は NFV の潮流を鑑みて，オペレーションを容易にする統合運用システムを開発した．

現在，実運用化に向けて RISE Orchestrator を用いた広域 SDN テストベッドを試験的に運用している．

参考文献

- [1] N. McKeown et al, "OpenFlow: enabling innovation in campus networks," SIGCOMM Comput. Commun. Rev.38, no.2, pp. 69-74, 2008.
- [2] National Institute of Information and Communications Technology, "JGN-X," available from <http://www.jgn.nict.go.jp/english/index.html>
- [3] Y. Kanaumi et al, "RISE: A Wide-Area Hybrid OpenFlow Network Testbed," IEICE TRANS. COMMUN., VOL. E96-B, NO. 1, Jan., 2013.
- [4] R. Sherwood et al, "Can the production network be the testbed?," in Proc. OSDI '10, pp. 1-6, 2010.
- [5] A. Al-Shabibi et al, "OpenVirteX: make your virtual SDNs programmable," Proc. the third workshop on Hot topics in software defined networking (HotSDN'14), pp. 25-30, 2014.
- [6] Y. Kanaumi et al, "Deployment of a programmable network for a nation wide R&D networkk," Proc. 2nd IEEE/IFIP International Workshop on Management of

the Future Internet (ManFI 2010) , Osaka, Japan, April 2010.

- [7] Y. Kanaumi, "Openflow switch demonstration at GENI conference 3rd on JGN2plus/APAN," 27th APAN Meeting, March 2009. Proc. 2nd IEEE/IFIP International Workshop on Management of the Future Internet (ManFI 2010) , Osaka, Japan, April 2010.
- [8] Y. Kanaumi, "Large-scale OpenFlow testbed in Japan," 31th APAN Meeting, Feb. 2011.
- [9] Y. Kanaumi et al, "Toward large-scale programmable networks: Lessons learned through the operation and management of a wide-area openflow-based network," Proc. 6th International Conference on Network and Services Management (CNSM 2010), Niagara Falls, Canada, 2010.
- [10] N. Itoh, "RISE3.0: The Design and Implementation of SDN/OpenFlow Testbed Considering Node Capacity and Inflexible Topology," 38th APAN Meeting, Aug. 2014.
- [11] "IEEE standard for local and metropolitan area networks, virtual bridged local area networks, amendment 4: Provider birdges."