

オーバーレイ型 OpenFlow ネットワークへの 移行支援システムの検討

野村圭太^{†1} 谷口義明^{†2} 井口信和^{†2} 渡辺健次^{†3}

クラウド環境やサーバ仮想化の普及により、ネットワーク設定を柔軟に変更可能な OpenFlow ネットワークが注目を集めている。今後、企業のシステム等において OpenFlow ネットワークへの移行が進められると予測される。しかし、従来型のネットワークから OpenFlow ネットワークへの移行には時間や人的コストがかかる。また、OpenFlow ネットワークに移行する場合には、全てのネットワーク機器を OpenFlow スイッチに一度に置き換えるのではなく、まずは、サーバ等のネットワークのエッジにある一部の機器のみに OpenFlow 機器を導入することにより、オーバーレイ型 OpenFlow ネットワークを構築することが考えられる。そこで本研究では、従来型ネットワークからオーバーレイ型 OpenFlow ネットワークへの移行を支援するシステムの検討を行う。動作検証の結果、本システムを用いることにより、仮想 OpenFlow スイッチの作成および仮想 OpenFlow スイッチ間のオーバーレイネットワークの構築を行えることを確認した。

Consideration of a System for Supporting Migration to Overlay OpenFlow Network

KEITA NOMURA^{†1} YOSHIAKI TANIGUCHI^{†2} NOBUKAZU IGUCHI^{†2}
KENZI WATANABE^{†3}

OpenFlow networks have attracted attention along with the popularization of cloud environment and server virtualization since it enables flexible network configuration. It is expected that traditional network systems of various organization will migrate to OpenFlow network in the future. However, it takes time and costs to completely migrate from traditional network to OpenFlow network since all network devices must be replaced to OpenFlow switches at once. In OpenFlow migration process, it is expected that the traditional network firstly migrates to an overlay OpenFlow network before it migrates to complete OpenFlow network. The overlay OpenFlow network is constructed among servers so that traditional network devices can be used. In this report, we study a system for supporting migration from traditional network to overlay OpenFlow network. Through experimental verification, we demonstrate that our system can support for building an overlay OpenFlow network.

1. はじめに

クラウド環境やサーバ仮想化の普及に伴い、ネットワークに対する要件が変化してきている。例えばサーバ仮想化により、サーバの追加や移動が起りやすくなっている。これに伴い、ネットワークの構成や機器の設定を変更する必要がある。しかし、従来型のネットワークでは、各ネットワーク機器に対して手動で設定する必要があるため、ネットワークの設定に時間を要し、ミスも発生しやすくなる。そこで、ネットワークの運用を自動化し、管理者の負担を軽減するための仕組みが求められている。

そうした背景から、SDN (Software-Defined Networking) [1]というコンセプトが注目を集めている。従来型のネットワークでは各ネットワーク機器上に、経路制御といった複雑な計算を行うコントロールプレーンと、フレームやパケ

ットの転送といった簡易な処理を行うデータプレーンの両方を持っている。これに対し SDN では、コントロールプレーンとデータプレーンが分離されている。従来型のネットワークでは、コントロールプレーンの実装はネットワーク機器のベンダに依存しているため、ベンダが提供する機能しか利用することができない。しかし SDN では、ネットワーク機器からコントロールプレーンが分離されるため、管理者が自由にコントローラを開発できる。これにより、ベンダ依存が解消され、ニーズに応じた柔軟なネットワークを構築できる。

SDN を実現するための技術として OpenFlow[2]があり、企業のシステム等において、今後、OpenFlow ネットワークへの移行が進められると予測される。しかし現状、すぐに利用可能なコントローラが存在しない場合、コントローラを開発する必要がある。また、OpenFlow ネットワークは従来型のネットワークとは異なるアーキテクチャであるため、移行に際して、OpenFlow の学習が必要となる。こうした理由から、OpenFlow ネットワークへの移行には時間や人的なコストがかかると考えられる。そのため、このようなコス

^{†1} 近畿大学大学院 総合理工学研究科
Graduate School of Science and Technology, Kindai University

^{†2} 近畿大学 理工学部 情報学科
School of Science and Engineering, Kindai University

^{†3} 広島大学大学院 教育学研究科
Graduate School of Education, Hiroshima University

トを削減するために、OpenFlow ネットワークへの移行を支援するシステムが求められる。

ここで、OpenFlow ネットワークの実現方式は、ホップバイホップ型とオーバーレイ型に分けられる。ホップバイホップ型 OpenFlow ネットワークでは、すべてのネットワーク機器を OpenFlow スイッチにより構成する。そのため柔軟な制御が可能であるが、移行のための初期コストが大きい。一方、オーバーレイ型 OpenFlow ネットワークでは、サーバ等のネットワークのエッジにある一部の機器間でトンネルを張り、OpenFlow による制御を行う。本ネットワークでは、柔軟な制御は行えないが、従来のネットワークで利用していた機器を活用できるため、移行のための初期コストが小さい。そのため、従来型のネットワークから OpenFlow ネットワークへの移行プロセスでは、ホップバイホップ型 OpenFlow ネットワークへ移行するプロセス(図1、移行プロセス①)、オーバーレイ型 OpenFlow ネットワークに移行するプロセス(図1、移行プロセス②)、オーバーレイ型 OpenFlow ネットワークからホップバイホップ型 OpenFlow ネットワークに移行するプロセス(図1、移行プロセス③)の3通りが考えられる。我々はこれまでに、移行プロセスのうち、従来型のネットワークからホップバイホップ型 OpenFlow ネットワークへの移行を支援するシステム(図1、移行プロセス①に相当)を開発してきた[3]。

本稿では、オーバーレイ型 OpenFlow ネットワークへの移行プロセス(図1、移行プロセス②に相当)に対応するため、従来型のネットワークからオーバーレイ型 OpenFlow ネットワークへの移行を支援するシステムを検討、開発する。本システムは、我々がこれまでに開発してきたシステム[3]を拡張したものである。本システム上の GUI を用いることにより、オーバーレイ型 OpenFlow ネットワークを一括して設定、構築することが可能である。そのため、従来型のネットワークからオーバーレイ型 OpenFlow ネットワークへの移行が容易に実現できる。本稿では、ネットワークシミュレータである GNS3[4]を用いて本システムの検証を行う。

本項の構成は以下の通りである。まず2章で本研究に関連する技術について述べる。3章でこれまでに開発したシステムについて述べ、4章で現在開発しているシステムの実装、検討について述べる。5章で動作検証を行った結果について述べる。6章でまとめと今後の課題について述べる。

2. 関連技術

2.1 OpenFlow

OpenFlow は、Stanford 大学において研究開発されたネットワークアーキテクチャである。OpenFlow の登場により、

SDN が注目を集めるようになった。そのため、現在 OpenFlow に関する研究が盛んに行われている[5][6][7]。

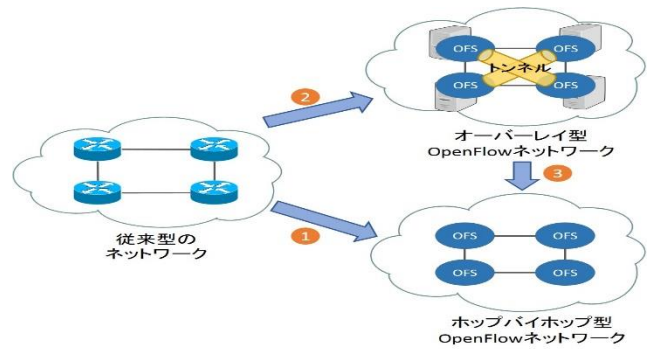


図1 OpenFlow ネットワークへの移行プロセス
Figure 1 Migration process to the OpenFlow network

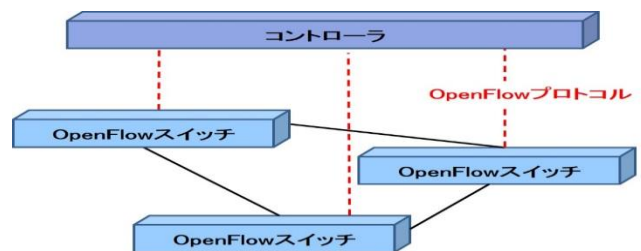


図2 OpenFlow のアーキテクチャ
Figure 2 Architecture of OpenFlow

OpenFlow のアーキテクチャを図2に示す。OpenFlow を用いたネットワークは、コントロールプレーンに相当するコントローラと、データプレーンに相当する OpenFlow スイッチから構成される。

OpenFlow スイッチはフローテーブルと呼ばれるテーブルを持っている。フローテーブルにはフローエントリが格納されている。フローエントリはパケットのヘッダ情報とマッチングする値が格納されているマッチフィールド、マッチングする順序を決定するのに用いるプライオリティ、統計情報が格納されているカウンタ、パケットに対する処理を表すインスタレーション、フローエントリの存在期間を表すタイムアウト、任意の用途に使用できるクッキーから構成される。

OpenFlow スイッチがパケットを受信すると自身のフローテーブルを参照し、パケットのヘッダ情報とマッチするフローエントリが存在するかどうかを調べる。存在する場合、その内容に従ってパケットを処理する。存在しない場合、コントローラへ Packet-In メッセージと呼ばれるパケットの処理を問い合わせるメッセージを送信する。コントローラは Packet-In メッセージから得られた情報を基に該当パケットをどう処理するかを決定し、その結果を Flow-Mod メッセージを用いて OpenFlow スイッチに通知する。

このように OpenFlow では、コントローラの実装内容によりネットワーク全体の挙動が決定される。そのため、OpenFlow ではコントローラを作成することによりニーズに応じたネットワークの構築が可能となる。

2.2 VXLAN

VXLAN[8]は、オーバーレイネットワークを構築するためのプロトコルである。同様の目的のプロトコルとして、ほかにNVGRE[9]やSTT[10]などがある。VXLANの目的は、複数のL2ネットワークを集約して、L3ネットワークにトンネリングすることである。基本的な用途としては、複数のL3ネットワークのドメインを接続して共通のL2ドメインに見せることである。VXLANでは、中継ノードにおいてUDP/IPを用いたカプセル化を行うことにより、L3ネットワーク越えを実現する。これにより、異なるネットワーク上にある仮想マシン同士が、同一L2ネットワーク内にあるように通信できる。

3. ホップバイホップ型 OpenFlow ネットワークへの移行支援システム

本章では、我々がこれまでに開発してきた、従来型のネットワークからホップバイホップ型 OpenFlow ネットワークへの移行を支援するシステム[3]の概要を述べる。

本システムの構成を図3に、GUIを図4に示す。本システムは、ルータから設定情報を取得するルータ設定情報取得機能、設定情報を OpenFlow スイッチに反映するルータ設定情報反映機能、移行する OpenFlow ネットワークが正しく動作するかの確認を事前に行う動作検証機能、を持つ。以下、本システムを用いたホップバイホップ型 OpenFlow ネットワークへの移行支援の利用手順について述べる。

まず、利用者（ネットワーク管理者）は、本システムのルータ情報設定取得機能を用いて、従来型のネットワークの設定情報を取得する。ルータから設定情報を取得するために、本システムは NETCONF[11]を使用する。このことにより、複数のベンダ製のネットワーク機器が混在する環境においても、ベンダごとの違いに影響されずルータの設定情報を取得できる。現在のシステムでは、ルーティングの設定情報と ACL の設定情報を取得できる。本システムは、取得したそれらの設定情報を XML ファイルとして保存する。

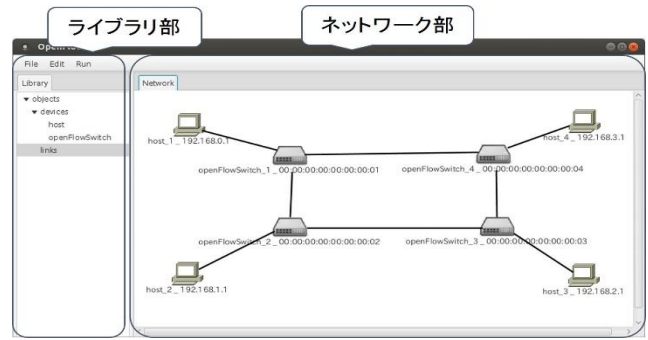


図 4 システム GUI
Figure 4 System GUI

次に、利用者は、本システムのルータ設定情報反映機能を用いて、移行後の OpenFlow ネットワーク中の各 OpenFlow スイッチの挙動の設定を行う。そのために、利用者は、まず、システム GUI(図4)上に移行予定の OpenFlow ネットワークトポロジを仮想的に作成する。その後、利用者は、各 OpenFlow スイッチが従来型のネットワークのどのルータに対応するかを設定する。以上により、OpenFlow ネットワーク全体の挙動を設定できる。なお、現在のシステムでは、移行後の OpenFlow ネットワークのトポロジは従来型のネットワークと同じトポロジであり、また、ルータの設定情報のみを OpenFlow ネットワークに反映することを想定している。しかし、今後、スイッチの設定情報を取得し、OpenFlow ネットワークに反映することを検討している。そのため手動で対応づけることにより、トポロジの構成次第では、1 台の OpenFlow スイッチにルータとスイッチの機能を持たせることができ、機器の台数の削減が見込める。

利用者は、本システムの動作検証機能を用いることにより、設定した OpenFlow ネットワークの動作検証を行うことができる。本システムは、Mininet[12]により、GUI 上に構築されたネットワークと同一構成の OpenFlow ネットワークを仮想的に構築する。これを使用し、利用者は、実際の OpenFlow ネットワークを用いた動作検証を行う前に、移行後の OpenFlow ネットワークを仮想的に検証できる。

最後に、利用者は、実際の OpenFlow ネットワークを用いて動作検証を行う。本システムは、OpenFlow ネットワークにおけるコントローラとして動作する。そのため、各 OpenFlow スイッチに対して、本システムをコントローラとして設定することにより、検証を行える。

以上のように、本システムを用いることで、従来型のネットワークからの設定情報の収集、GUI を用いた OpenFlow スイッチの設定、仮想 OpenFlow ネットワークを用いた本システムの動作の検証、実 OpenFlow ネットワークを用いた検証等を行うことができ、従来型のネットワークからホップバイホップ型 OpenFlow ネットワークへの移行を支援できる。

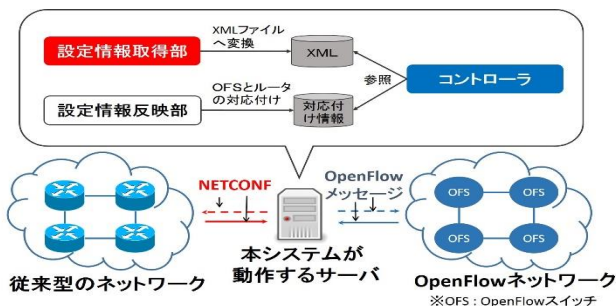


図 3 ホップバイホップ型 OpenFlow ネットワークへの移行支援システムの構成

Figure 3 System architecture for supporting migration to the hop-by-hop OpenFlow network

4. オーバーレイ型 OpenFlow ネットワークへの移行支援システム

3章で述べたこれまでに開発してきたシステムでは、ホップバイホップ型 OpenFlow ネットワークへの移行を想定しているため、移行の初期コストが大きい。組織によっては、冒頭で述べたように、従来型のネットワーク機器をそのまま使用できるため、移行コストが小さいオーバーレイ型 OpenFlow ネットワークに移行することが考えられる。そこで、本章では、オーバーレイ型 OpenFlow ネットワークへの移行支援システムについて検討する。なお、本章で検討するシステムは、3章で述べた我々がこれまでに開発してきたシステムを拡張したものである。

4.1 前提

まず、本章で移行対象とする従来型のネットワークと移行後のオーバーレイ型 OpenFlow ネットワークについて述べる。移行対象とするネットワークを図5に示す。ネットワーク中には複数台のサーバがあり、各サーバ内では複数台の仮想マシンと1台の仮想スイッチが動作するものとする。仮想スイッチには同じサーバ内にある各仮想マシンが接続されているものとする。仮想スイッチ間の通信には従来型のネットワークを用いる。なお、本稿では、ネットワーク管理者を利用者として想定しており、利用者は各サーバの仮想スイッチおよび仮想サーバの構成情報をあらかじめ知っているものとする。

次に、移行後のオーバーレイ型 OpenFlow ネットワークの構成を図6, 7に示す。本稿では移行の対象として2種類のオーバーレイ型 OpenFlow ネットワークを考えており、仮想 OpenFlow スイッチは Open vSwitch [13] で実現される。図6のネットワーク構成は、図5に示す移行対象ネットワーク中の仮想スイッチを仮想 OpenFlow スイッチに置き換えたものである。本ネットワークでは、仮想 OpenFlow スイッチ間で、VLAN 毎にトンネルを張ることにより、仮想 OpenFlow スイッチ間でオーバーレイネットワークを構成する。トンネリングに使用するプロトコルは、複数のベンダのネットワーク機器において対応している、VXLAN を使用する。以降、本ネットワークを OpenFlow ネットワーク A と呼ぶ。

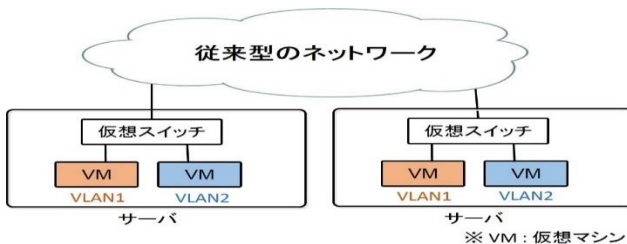


図5 移行対象のネットワーク
Figure 5 Migration target of network

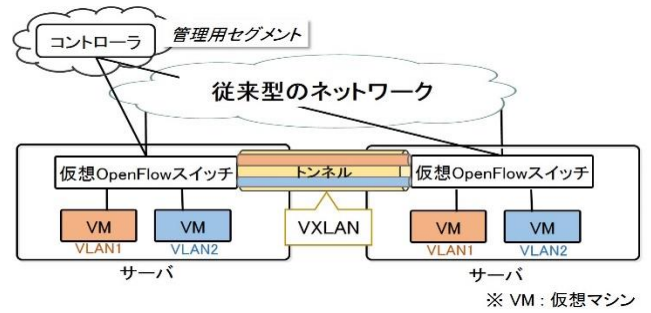


図6 移行後の OpenFlow ネットワーク A
Figure 6 OpenFlow network A

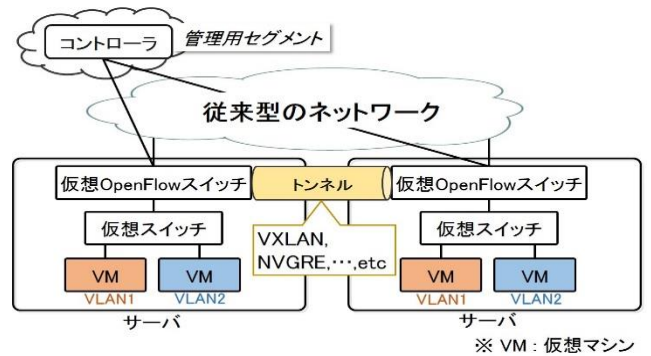


図7 移行後の OpenFlow ネットワーク B
Figure 7 OpenFlow network B

一方、図7のネットワーク構成は、仮想マシンのゲートウェイに仮想 OpenFlow スイッチを配置し、仮想スイッチと仮想 OpenFlow スイッチが接続したものである。本ネットワークでは、仮想スイッチからフレームを受け取った仮想 OpenFlow スイッチは、受信したフレームが L3 のネットワークを跨いで通信できるようコントローラにより制御される。以降、本ネットワークを OpenFlow ネットワーク B と呼ぶ。

なお、OpenFlow ネットワーク A, B ともに、仮想 OpenFlow スイッチと接続するコントローラは、仮想 OpenFlow スイッチとの接続性が保たれる管理用セグメントに配置される。

4.2 移行支援機能

従来型のネットワークからオーバーレイ型 OpenFlow ネットワークに移行するためには、仮想 OpenFlow スイッチの作成、設定、仮想 OpenFlow スイッチ間のオーバーレイネットワークの構築およびコントローラの開発が必要となる。ネットワーク内に多くのサーバがある場合、本作業には時間や人的コストがかかる。本章で検討するシステムでは、GUI を用いて本作業を一括して設定することにより、オーバーレイ型 OpenFlow ネットワークへの移行を効率化する。以下に、OpenFlow ネットワーク A および OpenFlow ネットワーク B への移行支援の利用手順について述べる。

4.2.1 OpenFlow ネットワーク A

利用者は初めに、システム GUI (図4) に、オーバーレイ

イ型 OpenFlow ネットワークを導入する対象のサーバ、サーバ内の仮想スイッチ、仮想マシンおよびそれらの接続情報を図 6 のように仮想的に配置する。これは、オーバーレイネットワークのトポロジの確認や設定情報の反映などを GUI を用いて行えるようにするためである。なお、現在のシステムでは、本作業は手動で行うことを想定している。しかしながら、サーバ数、仮想サーバ数が増えると、操作数が増える。そのため、3 章で述べた手法のように、サーバや仮想スイッチ等から自動的に設定情報を取得することが今後の課題である。

次に、利用者は、サーバ内に仮想 OpenFlow スイッチである Open vSwitch を作成するため、本システムに対してサーバの IP アドレス、管理者パスワード、コントローラ（本システムが動作するサーバ）の IP アドレスとポート番号、Open vSwitch 名を与える（図 8）。本システムはこれらの情報を基に、サーバと SSH を用いて接続し Open vSwitch を自動的に作成する。この時、サーバ内の仮想スイッチは、作成した Open vSwitch におきかえられる。仮想スイッチと仮想マシンとの接続関係、VLAN 情報等が、Open vSwitch に引き継がれる。仮想スイッチにおいて VLAN が用いられていた場合、Open vSwitch では VXLAN に変更される。一方、仮想スイッチに VLAN の設定が施されていない場合、各仮想マシンが同一セグメントに属するものと判断する。

最後に、利用者は、仮想 OpenFlow スイッチ間のオーバーレイネットワークを作成するため、図 8 のようなウィンドウを用いて、各仮想 OpenFlow スイッチに対して、オーバーレイネットワークを構築する相手の仮想 OpenFlow スイッチに対応する相手サーバの IP アドレスを設定する。なお、現在のシステムでは、IP アドレスを指定することにより、オーバーレイネットワークを構築する相手のサーバを指定しているが、操作の視覚化のため、GUI 上でサーバ間を結線することによりオーバーレイネットワークを構築できる機能を導入予定である。本システムはこれらの情報を基に、仮想 OpenFlow スイッチ間のトンネルを自動的に設定する。ここで、トンネリングに用いる VXLAN の識別子である VNI は、サーバ間で共通して使用されているものを

用いる。例えば、図 6 の場合、VLAN1, 2 が共通して用いられているため、これらを用いる。

以上を行うことにより、GUI を用いる仮想 OpenFlow スイッチの作成および仮想 OpenFlow スイッチ間のオーバーレイネットワークの一括設定と構築が可能となる。

4.2.2 OpenFlow ネットワーク B

OpenFlow ネットワーク A では、仮想スイッチを仮想 OpenFlow スイッチに置き換えるため、仮想マシンと仮想スイッチ間の接続情報が必要となる。そのため、サーバ内の仮想マシンの数が多い場合には操作数が増える。また、なるべく現在の設定を変更することなく、オーバーレイ型 OpenFlow ネットワークを導入する場合は、仮想スイッチを仮想 OpenFlow スイッチで置き換えるのではなく、仮想マシンのゲートウェイに仮想 OpenFlow スイッチを追加で導入することも考えられる。さらに、OpenFlow ネットワーク A では、従来型のネットワーク上の機器が特定のプロトコルに対応している必要があり、使用には一定の制約を課すものとなっている。そこで、現在検討している OpenFlow ネットワーク B の場合の本システムについて述べる。

利用者は初めに、前節と同様に図 6 に示すようなトポロジの作成を行う。この時、サーバおよびサーバ内の仮想スイッチのみを配置する。OpenFlow ネットワーク A の場合と異なり、サーバ中の仮想マシンおよび仮想マシンと仮想スイッチとの接続関係を設定する必要はない。これは、フレームを受信した仮想スイッチが、移行前と同じフレームの処理を行うためである。

次に、利用者は、前節と同様に仮想 OpenFlow スイッチの作成を行う。本システムは設定された情報を基に、サーバ内に Open vSwitch を作成し、仮想スイッチに接続する。

最後に利用者は、仮想 OpenFlow スイッチ間のオーバーレイネットワークの作成のために必要な設定を行う。この時、トンネリングに使用するプロトコルはユーザが指定する。例えば、VXLAN や NVGRE が挙げられる。フレームを受信した仮想 OpenFlow スイッチは、コントローラに処理を問い合わせる。その結果、受信したフレームをユーザが指定したプロトコルでカプセル化する。これにより、VLAN over VXLAN, NVGRE, ..., etc が実現でき、ユーザ任意のプロトコルで L3 ネットワークを跨いだ通信が可能となる。また、トンネリングに使用する識別子は利用者が設定する。カプセル化されたパケットを受信した仮想 OpenFlow スイッチは、コントローラの設定に基づき、パケットを非カプセル化し、仮想スイッチに送信する。フレームを受信した仮想スイッチは、自身の設定に従い、宛先ホストへと送信する。

以上を行うことで、GUI を用いる仮想 OpenFlow スイッチの作成および仮想 OpenFlow スイッチ間のオーバーレイネットワークの一括設定と構築が可能となる。



図 8 オーバーレイネットワーク構築ウィンドウ
Figure 8 Overlay network construction window

5. 動作検証

ここでは、実施した動作検証の結果について述べる。実験では、表 1 に示すスペックの PC を図 9 に示すトポロジで使用した。まず、本システムを用いて Open vSwitch 間の作成およびオーバーレイネットワークの構築が行えるかを確認した。検証の対象は、OpenFlow ネットワーク A とした。検証には、ネットワークシミュレータである GNS3 を用いた。図 9 において、本来、サーバ 1-1、サーバ 1-2 はサーバ 1 の中で動作する仮想マシンであるが、本稿では動作検証のため、別のサーバとし、サーバ 1-1、サーバ 1-2 には PC のシミュレータである VPCS を用いた。サーバ 2-1、サーバ 2-2 についても同様である。また、従来型のネットワークである R1、R2 には Cisco IOS を使用している。以降、動作検証の内容を述べる。

まず、システム GUI (図 4) 上に図 6 のように仮想的にサーバを配置した。その後、本システムに、サーバ 1 およびサーバ 2 の情報を与えた。このことにより、各サーバ上に Open vSwitch が作成されることを確認した。次に、本システムを用いて、オーバーレイネットワークの構築に必要な情報の設定を行った。

Open vSwitch 間でトンネルが確立されることを確認するために、Open vSwitch 間を流れる ICMP パケットをキャプチャし、解析した。解析では、まず、キャプチャしたパケットの VXLAN のヘッダにある VNI を参照した。この時、ICMP Request と、それに対する ICMP Reply の VNI が一致しているため、トンネルが確立できていると言える。

表 1 実験に使用した PC
Table 1 PCs used for experiment

	OS	CPU	MM
本システム	Ubuntu 13.04 64bit	Intel core i5-4460 3.20GHz * 2	16GB
仮想OpenFlow スイッチ (ovs01, ovs02)	Ubuntu 14.04 LTS 64bit	Intel core i5-4460 3.20GHz * 2	16GB

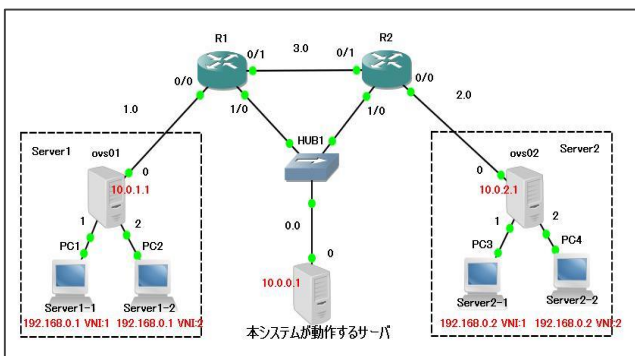


図 9 実験に使用した GNS3 上のトポロジ

Figure 9 Topology on GNS3 used in the experiments

以上より、本システムを使用することで、オーバーレイ型 OpenFlow ネットワークを構築できることを確認した。しかし、オーバーレイ型 OpenFlow ネットワーク B に対する移行支援の動作検証は行えておらず、今後、実施する予定である。

6. おわりに

本研究では、オーバーレイ型 OpenFlow ネットワークへの移行を支援するシステムを開発した。動作検証の結果、本システムを用いることにより、仮想 Open Flow スイッチの作成や、仮想 OpenFlow スイッチ間のオーバーレイネットワークを構築できることを示した。

今後の課題として、既存の仮想スイッチの設定情報を自動的に収集することにより、移行の負荷を低減する機能の導入や OpenFlow ネットワーク B への移行支援システムについて検討している。また、オーバーレイ型 OpenFlow ネットワークからホップバイホップ型 OpenFlow ネットワークへの移行を支援するシステムを検討する予定である。さらに、本稿では動作検証しか行っていないため、支援の有効性を定量的に示す実験を行うことを検討している。

参考文献

- 1) Software-Defined Networking, <https://www.opennetworking.org/>
- 2) McKeown, N. et al.: OpenFlow: Enabling Innovation in Campus Networks, ACM SIGCOMM Computer Communication Review V ol.38 No.2 pp.69-74 (2008)
- 3) 野村圭太, 堤啓彰, 谷口義明, 井口信和: ルータの設定を OpenFlow ネットワーク反映可能とするシステムの開発, 情報処理学会全国大会講演論文集, Vol.77, No3, pp.359-360 (2015)
- 4) GNS3, <http://www.gns3.com/>
- 5) 橋本直樹, 他: OpenFlow による認証基盤と連携したネットワークアクセス制御の実現, 研究報告インターネットと運用技術, Vol. 2014-IOT-24, No. 24, pp. 1-6 (2014)
- 6) 小谷大祐, 他: OpenFlow スイッチにおけるワイルドカードヘッダを考慮した Packet-In メッセージの制御手法, 信学技報, vol. 113, no. 443, IA2013-86, pp. 43-48 (2014)
- 7) 岡山聖彦, 他: DNS と OpenFlow スイッチとの連携による動的ファイアウォール, インターネットと運用技術シンポジウム 2013 論文集, Vol.2013, pp.95-98 (2013)
- 8) VXLAN: Virtual eXtensible Local Area Network, <http://tools.ietf.org/html/rfc7348>
- 9) NVGRE: Network Virtualization using Generic Routing Encapsulation, <https://tools.ietf.org/html/draft-sridharan-virtualization-nvgre-00>
- 10) STT: A Stateless Transport Tunneling Protocol for Network Virtualization, <https://tools.ietf.org/html/draft-davie-stt-01>
- 11) Network Configuration Protocol, <http://tools.ietf.org/html/rfc6241>
- 12) Mininet, <http://mininet.org/>
- 13) Open vSwitch, <http://openvswitch.org/>