

OpenFlow のネットワーク管理と連携するルーティング制御手法

市江 穂乃香[†] 橋本 直樹[‡] 廣津 登志夫[†]法政大学情報科学部[†] 法政大学情報科学研究科[‡]

1. はじめに

近年, VLAN 技術や仮想サーバなどの登場により, データセンタなどの大規模ネットワークの構成は複雑化してきている. そのため, ネットワークをソフトウェアで動的に集中管理することにより管理コストを抑制する Software Defined Network(SDN)というネットワーク構築のアプローチが注目されている. しかし, SDN の代表技術である OpenFlow を, 組織内の LAN など現在 TCP/IP で構築されているネットワークに適用しようとしても, その制御手法が IP ルーティングとは異なっているためスムーズな移行が容易ではないという問題がある. 本研究では OpenFlow の管理情報として得られるネットワークのトポロジマップを元に, 既存 IP ルーティングアルゴリズムを処理する仮想ネットワークを制御し, OpenFlow ネットワークと既存の IP ネットワークとの接続性を実現する.

2. RouteFlow

現状の IP のルーティング制御をそのまま OpenFlow 上に実現する技術の一つに RouteFlow [1]があげられる. RouteFlow はルーティングソフトウェアである Quagga と, OpenFlow の技術を組み合わせて仮想 IP ルーティングサービスを提供する QuagFlow[2]を発展させたものである.

RouteFlow の大きな特徴の一つに, 実環境と仮想環境という 2 つの環境が存在していることがあげられる (図 1). 実環境は物理スイッチを接続して構成される環境で, 接続されたノードや他のネットワークの間でのデータ通信を担う. 仮想環境内には LXC という軽量仮想化技術を用いて生成された仮想マシンが稼働しており, 実環境の 1 台のスイッチに対して 1 台の仮想マシンが配置される. 各々の仮想マシンには Quagga がインストールされ, 経路制御プロトコルの処理を行うと共に, 経路情報の収集などを担う RF-Client(RouteFlow Client)が稼働している.

An IP Routing Control Scheme using the OpenFlow Network Management Information

[†]Faculty of Computer and Information Sciences, HOSEI University

[‡]Graduate School of Computer and Information Sciences, HOSEI University

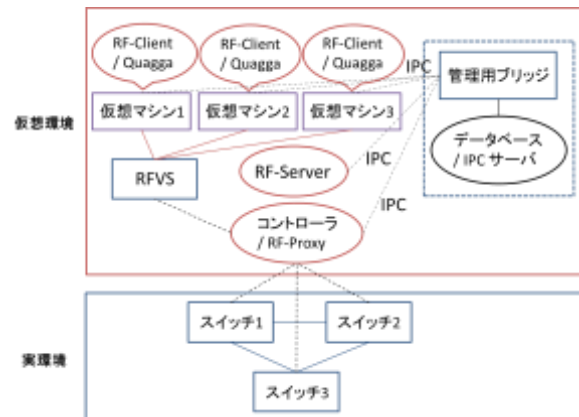


図 1 RouteFlow 構成図

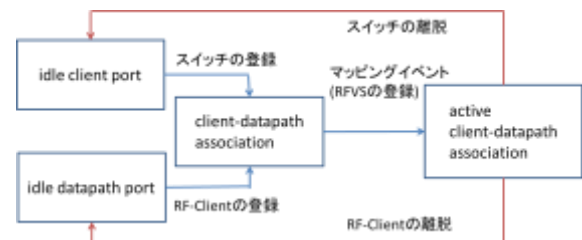


図 2 仮想マシンと実環境スイッチの対応管理

仮想マシン間の通信は OpenFlow スイッチ (RFVS) で処理される. RFVS や実環境の通信は OpenFlow コントローラにより管理されており, 仮想環境と実環境間での通信のやりとりを制御している. RF-Server(RouteFlow Server)は各 RF-Client の情報を収集し RF-Proxy(RouteFlow Proxy)を経由してコントローラに情報を提供する.

RouteFlow では, RF-Server に読み込まれる設定ファイルによって, 仮想マシンの ID, ポート番号と, スイッチの ID, ポート番号の対応付けをネットワーク管理者が設定する必要がある. この設定ファイルの情報のうち, どの情報が把握されているかによって状態が表 1 のように定義されている. また, 仮想マシンおよび実環境スイッチの状態における遷移を図 2 に示す.

RouteFlow では実環境の個々の OpenFlow スイッチに対応するように仮想マシンを用意し, 経路制御ソフトウェアを稼働させているが, 対応する仮想マシンとスイッチ間で全ての通信をパススルーすることにより経路計算が実現されている (図 3). しかし通常の OpenFlow ネットワ

表 1 設定ファイル構成と状態

設定ファイルの構成	状態
vm id, vm port, -, -, -, -	idle client port
-, -, -, dp id, dp port, ct id	idle datapath port
vm id, vm port, dp id, dp port, -, -, ct id	client-datapath association
vm id, vm port, dp id, dp port, vs id, vs port, ct id	active association

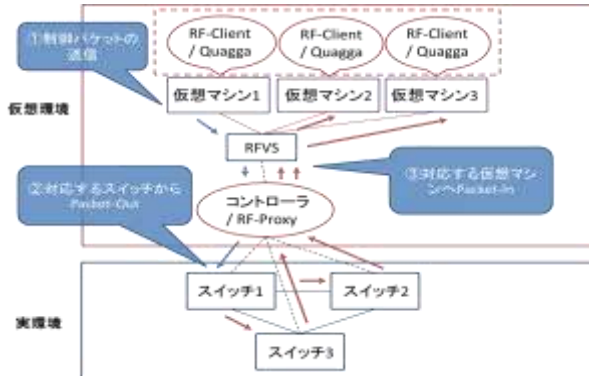


図 3 経路制御パケットのやりとり

ークではトポロジマップが把握されているため、これを用いることで実環境にパケット送出をすることなく経路制御が可能になると考えられる。

3. 提案手法

本研究では、仮想環境と実環境とを分割し、経路パケットを実環境へ流出させないように仮想環境上のパケットの配送を制御する。実際のOpenFlow ネットワークでは、LLDP などを用いてネットワークのトポロジ情報の取得を行っている。そこで、このトポロジ情報を用いてRFVS にフローエントリを設定すれば、仮想環境と実環境を独立に稼働させながら、実環境のトラフィックを制御することが可能になる。

4. 実装

提案手法を RouteFlow を改造することで実装した。具体的な処理の流れを図 4 に示す。まず、実環境上でスイッチが立ち上がった際に、コントローラ上でスイッチの追加を通知するイベントと、それらのスイッチの接続性の確立を確認するイベントが発生する(①②③)。そのイベントの発生時に、LLDP によって得られた接続性情報から得られた情報をフローエントリに変換し、RFVS に設定する(④)。仮想マシンから送出された経路制御パケットは、RFVS 上のフローエントリを参照することで、パススルーを行っていた際に送られていたスイッチに対応

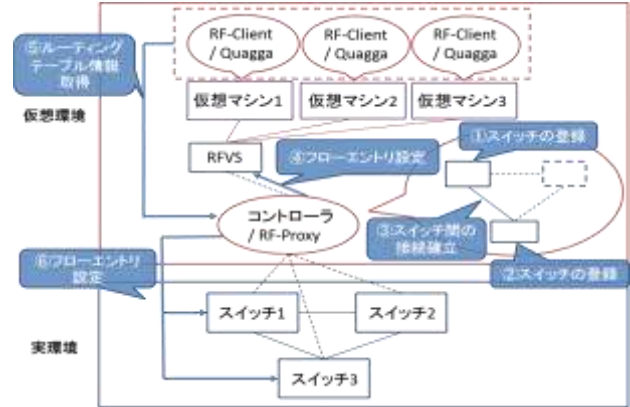


図 4 制御の流れ

する仮想マシンに送ることが可能になる。これにより、経路制御パケットの転送を仮想環境内のみに閉じることができ、仮想環境内で経路制御パケットが交換されることで、Quagga は通常のネットワーク上で稼働している場合と同様の経路情報を構築する。その後、各仮想マシンで稼働する RF-Client は Quagga が設定した経路情報を読み取り、それがコントローラに伝わるため(⑤)、それをフローエントリ変換した後スイッチにそのフローエントリを設定する(⑥)。

5. まとめ

本研究では OpenFlow ネットワーク上での IP ルーティングの実現において、経路計算を行う仮想ネットワークとデータ転送を行う実ネットワークのトラフィックを分離した構造を取っていることに特徴がある。今後、ネットワークの一部に OpenFlow スイッチが混在したような複雑なネットワーク構造についても処理可能なように改良を進めていく。

文 献

- [1] M. Ribeiro Nascimento, C. Esteve Rothenberg, M. R. Salvador, C. Corrêa, S. Lucena and M. F. Magalhães. "Virtual Routers as a Service: The RouteFlow Approach Leveraging Software-Defined Networks". In 6th International Conference on Future Internet Technologies 2011 (CFI 11), Seoul, Korea, June 2011.
- [2] M. Ribeiro Nascimento, C. Esteve Rothenberg, M. R. Salvador and M. F. Magalhães. "QuagFlow: Partnering Quagga with OpenFlow". In Proceedings of the ACM SIGCOMM 2010 Conference on SIGCOMM (New Delhi, India, August 30 - September 03, 2010). SIGCOMM '10. ACM, New York, NY, 441-442.
- [3] RouteFlow
<https://sites.google.com/site/routeflow/>