

運用省力化を実現するIP-VPN向け OpenFlowコントローラ

鈴木 一哉 金子 紘也

要旨

MPLSを用いたIP-VPNサービスでは、顧客経路の伝搬のためにBGPが用いられています。BGPを用いたIP-VPNでは、新規顧客収容時、BGPで伝搬される経路数が増えるため、ネットワーク内の各ルータの処理能力を超えないよう運用しなければならないという課題がありました。本稿では、この課題を解決するためのIP-VPN対応OpenFlowコントローラを提案します。提案手法ではBGPの処理を全てコントローラ側で行うため、ルータごとに必要だった制御リソースの管理が不要となります。



OpenFlow/BGP/IP-VPN/SDN/転送・制御分離

1. はじめに

本稿では、運用省力化を実現するIP-VPN向けOpenFlowコントローラを紹介します。

キャリアが提供するIP-VPNサービスは、顧客ごとのトラフィック分離のために、MPLS (Multi Protocol Label Switching) を用いて運用されています。MPLS網を構成する各ルータ同士は、サービス実現に必要なさまざまな情報の交換を、制御プロトコルを用いて自律的に行っています。例えば、顧客からの経路情報受け取り、及びネットワーク中の各ルータ間で経路情報を共有するために、BGP (Border Gateway Protocol) が用いられています。

このように自律分散型で制御されているIP-VPN網の運用では、これらの制御プロトコルの処理に用いられるCPUやメモリなど制御リソースについて考慮が必要です。新規顧客の収容時には、MPLS網側で扱う経路情報がその顧客分だけ増加します。増加分の処理のため、各ルータの制御リソースに余裕があるかを確認する必要があります。しかし、サーバ系の機器と比較すると、ルータなどのネットワーク機器に搭載されているCPUの処理性能は低く、搭載メモリの量は少ないのが一般的です。また、処理性能不足のために、ルータの制御部のみ高性能なものに置き換えることは、容易ではありません。

一方で、制御・転送を分離したアーキテクチャを採用するOpenFlow技術が登場し、実環境への導入が始まっています。OpenFlowに対応したスイッチは、フローエントリと呼ばれる規則に従ってパケットの転送処理を行います。一般的なルータでは、パケットの転送規則は、制御プロトコルにより収集された情報を基に自律的に行います。しかし、OpenFlowスイッチは、制御プロトコル処理のための制御部を持ちません。その代わりに、スイッチとは別に用意されるコントローラがフローエントリと呼ばれる転送規則を生成し、OpenFlowプロトコルを通して、スイッチへと設定します。

本稿では、OpenFlowを用いてIP-VPNを実現するためのコントローラの実現方法について提案します。提案手法では、IP-VPNサービスを実現するために必要となる制御プロトコル処理を全てコントローラ側で行います。また、提案手法では、スケールアウトによって制御プロトコル処理の負荷分散が可能です。これにより、ルータごとに行う必要があった制御リソースの管理が不要になるため、運用の省力化が実現できます。

2. IP-VPNにおけるBGPの役割

IP-VPNは、顧客の複数IPネットワークを接続し、それ

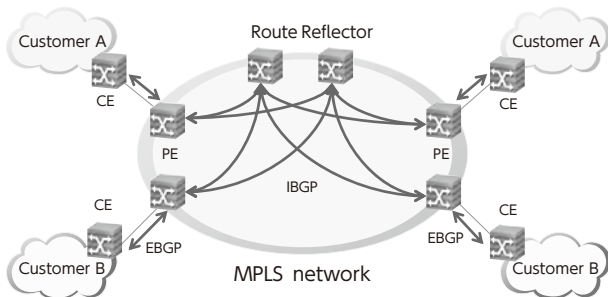


図1 IP-VPN網におけるBGPを用いた経路交換

それぞれのネットワークに割り当てられたIPアドレスを基に、パケットの転送を行います。そのため、顧客側の各拠点においてどのようなIPアドレスを使っているかを網側は知る必要があります。顧客のネットワークと接続するIP-VPN網側のルータ (Provider Edge: PE) は、顧客のネットワーク側ルータ (Customer Edge: CE) との間で、BGPを用いて経路情報の交換を行います。

PEルータは、顧客側ネットワークから受け取った経路情報を他のPEルータに通知する必要があります。通知のための方法はいくつかありますが、例えば図1のような構成をとることで、ルートリフレクタを介したPEルータ間で顧客経路の共有を実現できます。ルートリフレクタは、あるPEから受け取った経路情報を、他の全てのPEへ送ります。

ルートリフレクタとPEの間での経路情報交換は、どの顧客の経路情報なのかを区別して行う必要があります。そのために、BGPマルチプロトコル拡張を用います。PEは受け取った経路情報に、顧客を識別するための識別子を付け、VPN経路としてルートリフレクタに送ります。

このように、BGPにより運用されているIP-VPN網において新規顧客を収容すると、網内に伝搬する経路情報の数とその顧客から通知する分だけ増加します。ネットワーク中の各ルータでは、増加分を処理するためのCPUリソースやメモリが消費されます。そのため、新規顧客の収容は、各ルータが搭載するCPUやメモリなどの制御リソースに余裕があるかを確認しながら行う必要があります。

3. IP-VPN対応 OpenFlowコントローラ

この章では、今回提案するIP-VPN対応OpenFlowコントローラについて説明します。提案コントローラの構成を、図2に示します。

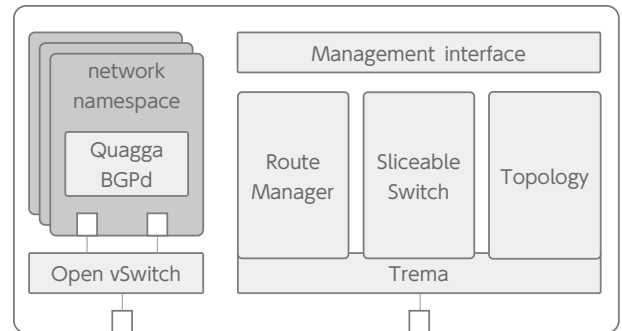


図2 IP-VPN対応OpenFlowコントローラ

3.1 BGPによる経路情報の交換

図2に示すように、BGPプロトコルを用いた顧客拠点側ルータとの間で経路情報の交換を行うためのBGPデーモン (BGPd) を、提案コントローラ中に組み込みました。このBGPdは、顧客ごとに1つずつ起動し、ある顧客ネットワークから受け取った経路情報を、同じ顧客の他のネットワークへと通知します。更に、顧客から受け取った経路情報は、OpenFlowネットワーク上でのパケット転送に反映させるために、「Route Manager」に渡されます。BGPdの実装には、オープンソースの経路制御ソフトウェアである「Quagga」を用いました。

BGPプロトコル処理を行うBGPdはコントローラ内に配置したため、CEルータから送られてきたBGPのパケットをコントローラまで送り届ける必要があります。そのために、OpenFlowネットワーク上に、BGPコネクション用のパスを構成します (図3)。BGPコネクション用パスを構成するために、OpenFlowネットワーク上に複数の独立したL2ネットワークを作る OpenFlowコントローラアプリケーションである「Sliceable Switch」¹⁾を利用しました。今回提案のコントローラには、Trema Appsの一部として公開されているSliceable Switchを組み込みました。

BGPdは顧客ごとに動作させる必要があるため、顧客数が増えた場合は多数のBGPdが動作することになり、これらを動作させるマシンの負荷が高くなります。その場合、これらのBGPdを、複数の異なるマシン上に分散させて動作させます。ある顧客向けのBGPdが動作しているマシンはSliceable Switchに通知され、そのマシンを端点とするBGPコネクション用のパスが構成されます。このように、顧客ごとのBGP処理をスケールアウトさせることで、負荷分散が可能です。

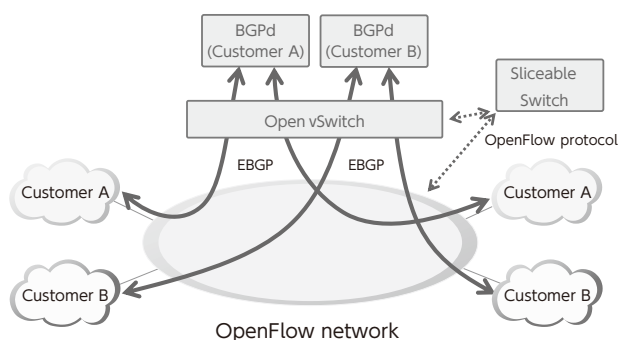


図3 BGPコネクション用パスの作成

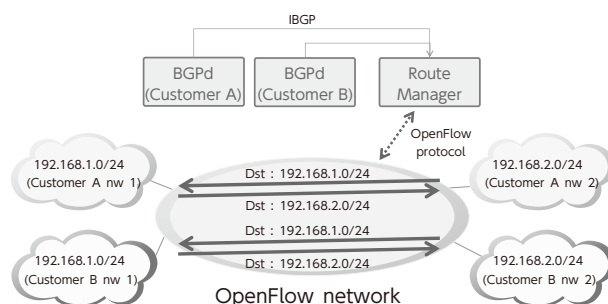


図4 トラフィック転送用パスの作成

3.2 トラフィック転送用パスの設定

Route Managerは、顧客側ネットワークで使用しているアドレスに基づいたトラフィック転送用のパスを構成します。顧客側の使用アドレスは、BGPプロトコルを用いて、経路情報としてRoute Managerに通知されます。Route Managerは通知されたアドレスを基に、フローエントリのMatch条件を作成します。また、同一顧客の他のネットワークから、アドレスが通知されたネットワークまで、OpenFlowネットワーク上のパスを計算し、パスに沿ってパケットが転送されるよう各スイッチにフローエントリを設定します。例えば、図4に示すように、顧客Aのネットワーク1には192.168.1.0/24というアドレスが割り当てられています。この時Route Managerは、192.168.1.0/24宛てのトラフィックを、同じ顧客Aのネットワーク2からネットワーク1まで転送するためのパスを作ります。

図4の例では、顧客のネットワーク数が2の場合について説明しましたが、3拠点以上の場合でも同様に動作します。図5のようにネットワーク数が3である場合、それぞれのネットワーク間でパスを構成します。

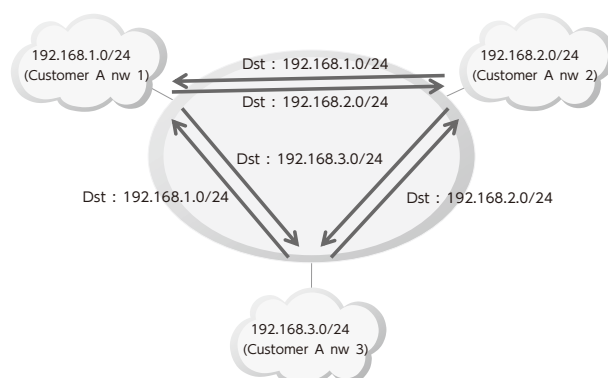


図5 3つのネットワークを収容する場合

3.3 顧客ごとのトラフィック分離

IP-VPNを実現するためには、1つのOpenFlowネットワーク上に複数の顧客を収容する必要があります。例えば、図4に示すように、異なる顧客が同じIPアドレス空間を用いるケースに対応しなければなりません。Route Managerは、顧客ごとのトラフィック分離にVLANタグを用いています。顧客拠点側からOpenFlowネットワークへと入ってきたパケットに対してVLANタグを付与し、OpenFlowネットワークから顧客拠点へパケットを送る際に付与したVLANタグを除去します。このようにすることで、OpenFlowネットワーク内で転送されるパケットには、顧客ごとに異なるVLANタグ

が付与されていることとなります。そのため、異なる顧客間で同じIPアドレスが使われていたとしても、それらを区別して転送することができます。

4. 将来に対する取り組み

顧客ごとのトラフィック分離にVLANタグを用いましたが、将来はMPLSラベルを用いることを計画しています。今回提案したコントローラでは、現在広く使われているOpenFlowプロトコルバージョン1.0を使用しました。しかし、このバージョンはMPLSに対応していないため、MPLSラベルを用いるためにはOpenFlowプロトコルバージョン1.1以降が必要となります。

MPLSに対応することで、本コントローラのトランスポートSDNシステム²⁾への適用が可能となります。このトランスポート装置の持つ帯域制御やプロテクション機能と組み合わせることで、より高信頼なネットワークを実現することが可能となります。

5. むすび

本稿では、IP-VPNの運用省力化を実現する OpenFlow コントローラの提案と、今後の取り組みについての紹介を行いました。

提案手法では、IP-VPNを実現するための制御プロトコル処理全てを、OpenFlowコントローラ側のみで実現しています。そのため、新規顧客収容時、制御リソースの管理はコントローラ側でのみ行えばよいため、運用作業の負担軽減を実現できます。

6. 謝辞

本研究の一部は、総務省の委託研究「ネットワーク仮想化基盤技術の研究開発」プロジェクトの成果です。

参考文献

- 1) 鈴木一哉ほか：OpenFlow 技術とその応用，コンピュータソフトウェア Vol.30 No.2 pp.2-13, 日本ソフトウェア科学会, 2013
- 2) 三野勝幸ほか：基幹系ネットワークを支える要素技術とパケット光統合トランスポート装置, NEC 技報 Vol.66 No.1 pp. 22-25, 2013.

執筆者プロフィール

鈴木 一哉

情報・ナレッジ研究所
主任

金子 紘也

情報・ナレッジ研究所

関連 URL

Trema : Full-Stack OpenFlow Framework in Ruby and C

<http://trema.github.io/trema/>

Trema Apps

<https://github.com/trema/apps>

Quagga Routing Suite

<http://www.nongnu.org/quagga/>

NEC 技報のご案内

NEC 技報の論文をご覧くださいありがとうございます。
ご興味がありましたら、関連する他の論文もご一読ください。

NEC技報WEBサイトはこちら

[NEC技報\(日本語\)](#)

[NEC Technical Journal\(英語\)](#)

Vol.66 No.2 ICTシステムを高度化するSDN特集

ICTシステムを高度化するSDN 特集によせて
SDNがもたらすICTシステムの高度化とIT・ネットワーク市場の変化
NECのSDNへの取り組みとNEC SDN Solutions
SDN実用化に向けた標準化

◇ 特集論文

NEC Enterprise SDN Solutions

WANの利用、運用を効率化する拠点・データセンター接続最適化ソリューション
安全で柔軟なネットワークアクセスを提供する「アクセス認証ソリューション」

NEC Data Center SDN Solutions

仮想環境の効率化を実現するIaaS運用自動化ソリューション

NEC SDN Solutionsを支える最新技術

SDNコントローラ作成のシンプル化を実現するネットワーク抽象化モデル
Wi-Fiの利便性向上を実現するスマートデバイス通信制御技術
大規模SDNネットワークを実現するOpenFlowコントローラアーキテクチャ
ヘテロジニアス網統合制御基盤を実現するマルチレイヤ抽象化技術
運用省力化を実現するIP-VPN向けOpenFlowコントローラ

導入事例

乱立する部門LAN、移動する検査機器 医療現場のネットワークをOpenFlowで改革
事業拡大を見据えデータセンターにSDNを導入 サービスのスピード、信頼性、他社優位性を向上

◇ 普通論文

iPASOLINK All Outdoor Radio (AOR) 装置の開発
iPASOLINKシリーズ及び超多値変調技術の開発
10Gbps伝送を実現する超大容量無線伝送技術
メタマテリアルを用いた電磁ノイズ抑制技術とその実用化

◇ NEC Information

C&Cユーザーフォーラム&iEXPO2013

人と地球にやさしい情報社会へ ~インフラで、未来をささえる~

NEC講演
展示会報告

NEWS

2013年度C&C賞表彰式典開催



Vol.66 No.2
(2014年2月)

[特集TOP](#)