L-035

ネットワークの分割を用いたコントロールプレーン分離型 OpenFlow モデルの提案

Proposal of an OpenFlow Model using Hierarchically Separated Control Plane and Network Partitioning

宮本　翔平 †
Shohei Miyamoto

今泉　貴史 ‡†
Takashi Imaizumi

1. はじめに

近年、サーバ群の柔軟な構成変更を可能とする、サーバの仮想化が広く用いられるようになってきている。これに伴い、従来のネットワークアーキテクチャが見直されており、その時々の需要に応えるために柔軟かつ迅速な構成変更が可能なネットワークが求められている [1]。

そこで注目されているのが、必要なトポロジーをソフトウェア的に創り出すネットワークの仮想化、SDN (Software Defined Network) である。SDN では、経路制御を行うコントロールプレーンがデータの転送を行うデータプレーンから分離されている。今まで個々のデバイスで行われていた制御を一元管理することでネットワークのトポロジーを必要に応じて変更でき、新たなアプリケーションやサービスをデプロイするのにかかる時間も短縮できる。現在 SDN の標準プロトコルとして注目されているのが、スタンフォード大学が開発した OpenFlow である [2]。多くの IT 企業が OpenFlow の開発・発展に力を入れている。一方で、既存のネットワークから SDN への移行は進んでいない。その原因の 1 つとして、OpenFlow の実装モデルにスケーラビリティの問題があることが挙げられる。

本論文では、OpenFlow が抱えるスケーラビリティの問題の原因となっている、スイッチが保持する情報量の削減と、一元管理で生じる処理のボトルネックを解消することを目的とする。

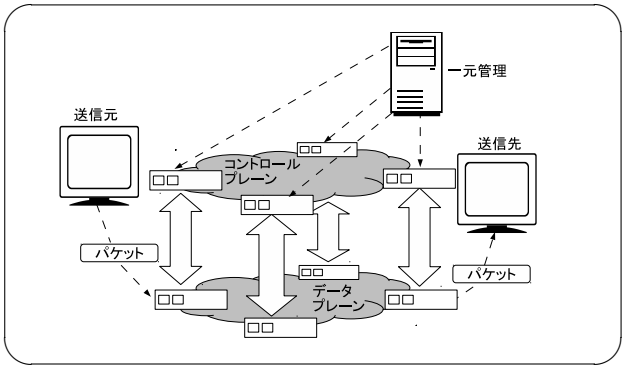

図 1: SDN の概要

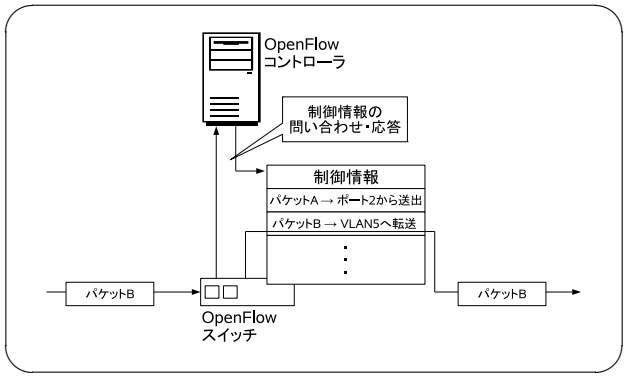

図 2: OpenFlow のコントローラとスイッチの関係図

2.OpenFlow

2.1.OpenFlow の概要

OpenFlow は、米スタンフォード大学で行われた研究であり、現在標準化が進められている SDN の実現方式の 1 つである。パケットの経路制御を行うコントロールプレーンは OpenFlow コントローラが担い、パケットの転送を行うデータプレーンは OpenFlow スイッチが担当する。各スイッチはコントローラに問い合わせを行いフローの転送先を決定するが、複数のスイッチを 1 台のコントローラが制御する形となっている。フローは OpenFlow での転送単位であり、従来レイヤごとに独自に管理されていた情報が統合されている。そのため、多様なレイヤの情報を基にした柔軟なネットワークの制御が可能となっている。

OpenFlow スイッチはコントローラから受信した制御情報をフローテーブルと呼ばれる表にまとめて保持している。フローテーブルにより各フローにどのような処理を実行するかが決定される。フローテーブルの項目をフローエントリと呼ぶ。各フローエントリにはフローが条件にマッチするかを判断する図 3 のようなマッチフィールド [3] と、条件にマッチしたパケットをどう処理するかを定義するインストラクションが含まれる。

2.2. 実装方式

現在、SDN/OpenFlow の実装方式として 2 つのモデルが考察されている。

†千葉大学大学院融合科学研究科
‡千葉大学統合情報センター

Copyright © 2013 by Information Processing Society of Japan and
The Institute of Electronics, Information and Communication Engineers
All rights reserved.

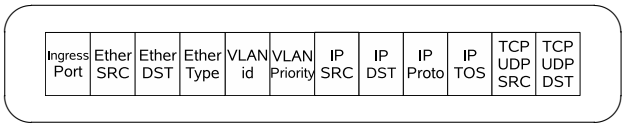

図 3: マッチフィールドの条件

ホップ・バイ・ホップモデルは、コントローラがネットワーク内の全ての OpenFlow スイッチを制御するモデルである。ホップ・バイ・ホップモデルでは、ホップ単位での細かな経路制御を行うことで OpenFlow の目的である柔軟なネットワークを実現できる。しかし、細かい制御が可能な反面、スケーラビリティの問題がある。

一方、オーバーレイモデルはエンドノードが接続されているスイッチのみを制御する。制御対象のスイッチ間はトンネリング技術を用いながら通常の経路制御により転送する。オーバーレイモデルではコントローラの制御対象となるのは両端のスイッチのみなので、それ以外のスイッチは OpenFlow 対応である必要がない。そのため、当面はオーバーレイモデルを用いた導入が有力視されている。しかし、オーバーレイモデルでは、ホップ・バイ・ホップモデルほどの柔軟な制御はできない。

SDN に求められている柔軟なネットワークを十分に活用できるのはホップ・バイ・ホップモデルであるが、2.3 に述べるスケーラビリティの問題のため実現が困難となっている。

2.3. ホップ・バイ・ホップモデルのスケーラビリティ問題

ホップ・バイ・ホップモデルでは、コントローラが全ての OpenFlow スイッチを制御する。更に、制御対象のネットワーク全体に関するトポロジー情報も保持しなければならない。スイッチに関しては、トポロジー情報を元に作られた経路情報を保持しなければならない。そのため、OpenFlow のホップ・バイ・ホップモデルには、次のようなスケーラビリティに関する問題がある。

- 処理の集中に起因するコントローラの負荷上昇
- 対象ネットワークの拡大によるスイッチのメモリ圧迫

各 OpenFlow スイッチからコントローラに対して処理が要求されるのは、主に以下の場合である。

1. タイムアウトしたフローテーブルの更新
2. フローに対応したエントリが存在しない場合の問い合わせ
3. インストラクションの指示によるフローの転送

コントローラが制御するスイッチの台数が増加するとこれらの処理による負荷も増大するため、大規模なネットワークではコントローラがボトルネックとなってしまう可能性がある [4]。

経路制御の処理は全てコントローラへ集中するが、経路情報に関しては各 OpenFlow スイッチがフローテーブルとして保持する。この際ネットワークの規模と、制御の細かさに比例してフローテーブルの情報量は増大する。膨大なフローテーブルに対応できるように設計されていない OpenFlow スイッチの場合、巨大なテーブルを保持できずに正しく動作しなくなる可能性もある [5]。

3.HARP モデル

OpenFlow のスケーラビリティ問題に対処するために、本論文では HARP (Hierarchical ARea Partitioned) モデルを提案する。HARP モデルでは、コントローラに対する処理量とスイッチが保持しなければならない情報量の両方を低減する。HARP モデルの概要を図 4 に示す。

3.1. ネットワークの分割

大規模なネットワーク全体を一元管理するのは困難である。そこで HARP モデルでは、ネットワークをいくつかのエリアに分割して管理する。

ネットワークの分割と共に、ネットワークのトラフィックも以下のように分類する。

- Intra-area Traffic: エリア内に収まるトラフィック
- Inter-area Traffic: エリアをまたぐエンドノード間でのトラフィック

エリアの外周に設置されており、隣接するエリアが存在するスイッチを Area Edge Switch (AES) と呼び、その他のエリアと接続しているポートを Area Edge Port (AEP) と呼ぶ。

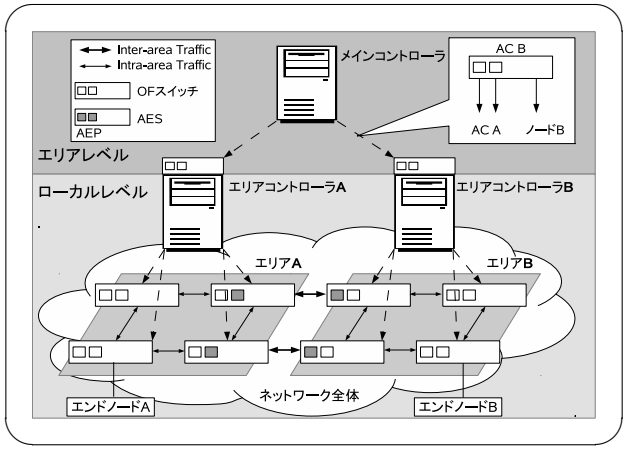

図 4: HARP モデルの概要

Copyright © 2013 by Information Processing Society of Japan and
The Institute of Electronics, Information and Communication Engineers
All rights reserved.

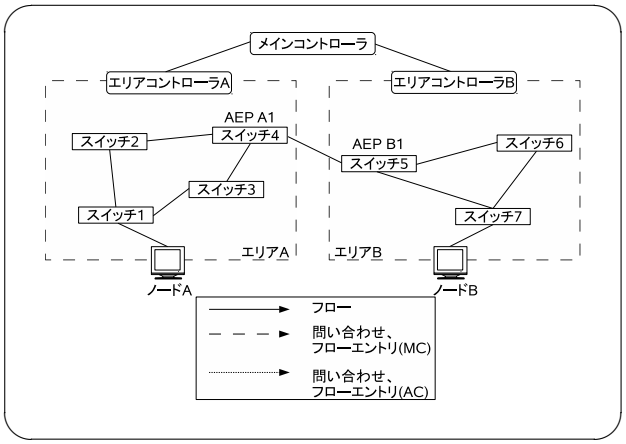

図 5: エリア間通信の例:ネットワーク構造

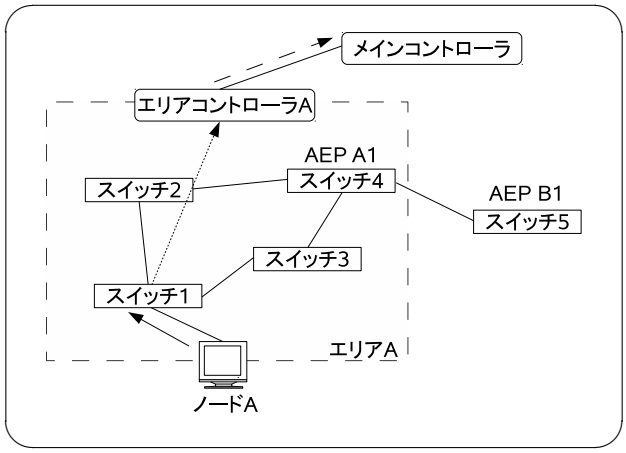

図 6: エリア間通信の例:動作 1

3.2. エリアコントローラ

各エリアにはコントローラを一台割り当て、エリアコントローラと呼ぶ。エリアコントローラは、管轄エリアを 1 つのネットワークとして認識し、エリア内のトポロジーのみ把握する。スイッチは振り分けられたエリアのエリアコントローラに対して問い合わせを行い、その命令に従う。ゆえに、スイッチが保持するフローテーブルの情報はエリア内トポロジーのみとなる。

各エリアコントローラは intra-area traffic の経路制御と、スイッチからの問い合わせに対応する。Inter-area traffic の問い合わせを受信した場合は、後述のメインコントローラへ問い合わせメッセージを送信し、その指示に従う。エリアコントローラは、エリアをネットワーク全体として認識している基本的な OpenFlow コントローラのように動作する。

3.3. メインコントローラ

メインコントローラは、エリアコントローラをそのエリアコントローラが管理するエリアにある全ての AEP をポートとして持つようなスイッチとみなし、通常の OpenFlow プロトコルに従って管理する。メインコントローラの主な役割は次のようになっている。

- ネットワーク全体のトポロジーの把握
- エリアコントローラからの要求への対応

フローテーブルのエントリとなる制御ルールを定義するアプリケーションは、メインコントローラで開発され、エリアコントローラへデプロイされる。

3.4.HARP モデルの動作

Intra-area traffic の制御は、従来のホップ・バイ・ホップモデルに近い形となっている。一方、inter-area traffic はエリアをあたかも 1 つのスイッチであるかのように扱うので、ネットワーク全体として見た場合にはオーバーレイモデルに近い形となっている。エリアレベルで見た場合にはエリアごとのホップ・バイ・ホップモデルと言える。

Intra-area Traffic　エリア内のルーティングのみで対応できる intra-area traffic のルーティングは、従来のホップ・バイ・ホップモデル同様にエリアコントローラが行う。各エリアの intra-area traffic の制御をエリアコントローラが自律して行うのが本手法の特徴となっている。これにより、複数のエリアコントローラにネットワーク全体のトラフィックが分散される。AEP はエリア外からのトラフィックは全て新規のフローと認識し、エリア外へ送出したトラフィックは目的地へ到達したと認識する。エリアコントローラからはエリア内のノードがネットワークの全ノードのように認識されている。これにより、エリア内のスイッチが保持するフローテーブルの情報量も削減可能である。

Inter-area Traffic　エリア間の通信が必要とされる inter-area traffic のルーティングは、エリアコントローラのみでは対応できない。エリアコントローラは受信した問い合わせメッセージを、あたかも自身がスイッチであるかのようにメインコントローラへ転送する。メインコントローラは全エリアコントローラからの情報を用いて目的地エリアを検索し、エリア間のルーティングを問い合わせを行ったエリアコントローラに設定する。エリアコントローラはメインコントローラの指示に従い、AEP までのエリア内スイッチのルーティングを設定する。そして、エリアコントローラは AEP 間でのルーティングが行われた時点で目的地と判断する。フローを受け取るエリアのエリアコントローラには、メインコントローラからプロアクティブにルールが設定される。このルールに基づき、エリアコントローラが目的地までの intra-area traffic を制御する。

図 5 のネットワーク構造を例として動作の流れを述べる。スイッチ 4 のポート A1 とスイッチ 5 のポート B1 はそれぞれのエリアの AEP である。

Copyright © 2013 by Information Processing Society of Japan and
The Institute of Electronics, Information and Communication Engineers
All rights reserved.

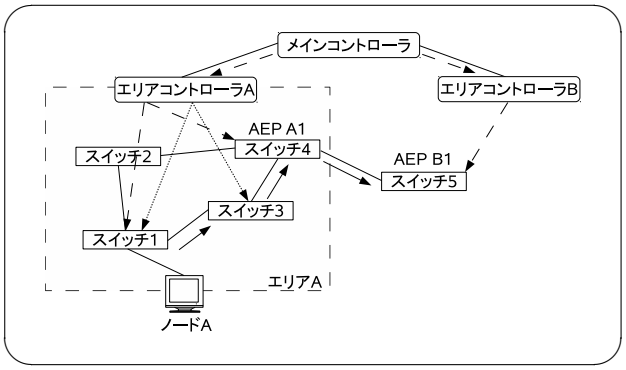

図 7: エリア間通信の例:動作 4

1. ノード A からノード B を目的地としたフローがスイッチ 1 に到達する。初期状態ではフローテーブルにマッチするエントリが存在しないので、エリアコントローラへ問い合わせメッセージを送信する。
2. メッセージを受信したエリアコントローラは、宛先から自身の管理するエリア内部のトラフィックではないことがわかり、メインコントローラへ問い合わせメッセージを転送する。
3. メインコントローラはエリアコントローラ B のトポロジー情報からノード B が設置されているエリアを判別する。
4. メインコントローラはエリアコントローラ A には、
   - 送信先がノード B の場合:送信先をスイッチ 5 に変更。MPLS ラベル 100 を付与。

   のようなフローエントリを返答する。スイッチ 5 までの経路はエリアコントローラ A が決める。
5. エリアコントローラ B には、
   - MPLS ラベル=100 の場合:MPLS ラベルを除去、送信先をノード B に変更。再度フローテーブルを検索。

   のようなフローエントリをプロアクティブに投入する。スイッチ 1 からスイッチ 4 までの intra-area traffic は、エリアコントローラ A が自身の持つポリシーに従って決定する。
6. スイッチ 5 からノード B までは、エリアコントローラ B が intra-area traffic として制御する。

例ではフローの識別に MPLS ラベルを利用したが、マッチフィールドは環境に応じて自由に変更できる。

3.5.HARP モデルのプロトコル

HARP モデルでの OpenFlow スイッチとエリアコントローラのやりとりは、従来の OpenFlow プロトコルの仕様に合致している。そのため従来のプロトコルをそのまま利用することが可能である。同様に、メインコントローラがエリアコントローラを制御する動作でも、従来の OpenFlow プロトコルを使用する。HARP モデルではスイッチの動作を、従来と同様にパケットの転送のみとしているので、ハードウェアも既存のものを利用できる。

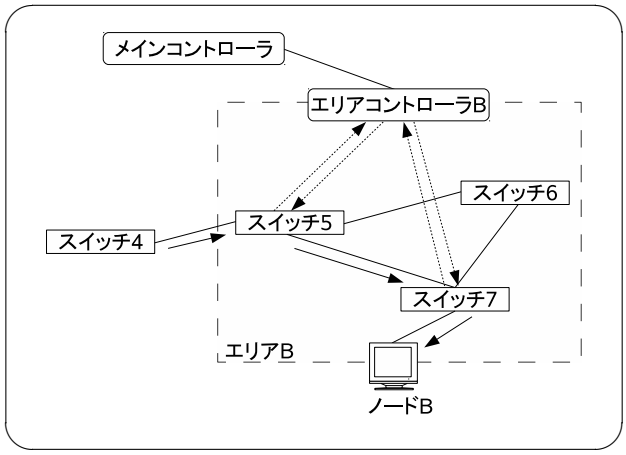

図 8: エリア間通信の例:動作 6

4. 関連研究との比較

HyperFlow　HyperFlow[4] は、物理的には複数台のコントローラを用いるが、論理的には集中型のコントロールプレーンとなっている。これによりネットワークの一元管理を実現しつつ、スケーラビリティをそなえている。全ての HyperFlow コントローラは同一のネットワークトポロジー情報を保持しており、あたかもネットワーク全体を制御しているかのように動作する。更に、各コントローラは同じアプリケーションのセットを使用する。

HyperFlow は HARP モデルと異なり、スイッチの情報量を考慮していない。本モデルでは、スイッチはネットワーク全体のトポロジーではなく、所属エリアのトポロジーを元にフローテーブルが生成される。そのため、使用するメモリを抑えることができる。しかし、本モデルではエリアとメインコントローラ間の問い合わせがあるため、HyperFlow よりレイテインシが悪くなることが予想できる。

DIFANE　DIFANE[6] では、コントローラとスイッチの役割を見直している。コントローラは制御ルールの生成はするが、リアルタイムなパケットの転送には直接関わらない。フローの問い合わせはコントローラへは届かず、データプレーン内で処理される。スイッチに到達したフローの最初のパケットはオーソリティスイッチにリダイレクトされる。オーソリティスイッチはパケットをネクストホップのスイッチに転送すると

Copyright © 2013 by Information Processing Society of Japan and
The Institute of Electronics, Information and Communication Engineers
All rights reserved.

同時に、必要な制御ルールをパケットの送信元スイッチにキャッシュする。

プレーン内で処理の分割を行っている点では本モデルは DIFANE と同様である。DIFANE でのオーソリティスイッチとスイッチの関係が、本モデルでのエリアコントローラとスイッチの関係に類似している。しかし、本モデルでは既存の OpenFlow スイッチを利用できるが、DIFANE では専用のスイッチが必要となる。ただし、DIFANE ではデータプレーン内でフローの最初のパケットが問い合わせではなくリダイレクトという形となるため効率が良い。スイッチの性能が問題とならない場合、DIFANE は本モデルよりパケット転送の能力が高いと言える。

5. 考察

HARP モデルはコントローラでの集中処理と OpenFlow スイッチが保持する情報量の両点を意識したモデルとなっている。ネットワークをエリアに分割することで、スイッチが保持するフローテーブルの基となるトポロジー情報がエリアネットワークのものに限定される。コントローラに集中してしまう処理は複数台のエリアコントローラで分散する。処理を水平に分散しただけではなく、エリア内のスイッチを制御するエリアコントローラと、エリアコントローラを上からスイッチとして扱い制御するメインコントローラがあることにより垂直にも処理が分散できている。また、メインコントローラを設置することによりネットワークの一元管理ポイントができ、エリアコントローラは管轄のエリアのみ把握していればよくなる。

本モデルは従来の基本的なホップ・バイ・ホップモデルに比べ、ネットワークを分割して扱っていることとコントローラを複数台使用していることが異なる。ここで、

- n: エリア数
- m: エリア内の平均スイッチ台数

としたときの各コントローラの負荷について考察する。

トラフィックの大半がエリア内に閉じており、エリアごとのトラフィックが均等に近い状況では、従来の単一コントローラによるホップ・バイ・ホップモデルと比較して、HARP モデルではコントローラ 1 台に対しての負荷が約 $\frac{1}{n}$ となる。ただし、メインコントローラの負荷とトポロジーの変化で生じる負荷は考慮していない。また、エリアコントローラがスイッチとして振舞う負荷も考慮していないが、影響を及ぼすほどの負荷が増すとは考えにくい。

更に、従来のモデルでは各スイッチにネットワーク全体のトポロジー情報を基に生成されたフローテーブルが保持されていたのに対して、HARP モデルではエリアのトポロジー情報が基となっている。従来のモデルでは全体の nm 台の情報量を扱うのに対して、エリア内のトポロジーでは m 台の情報量に抑えられる。隣接するエリアの AEP は inter-area traffic の目的地として認識されているので、スイッチに接続されているノードと同様に扱う。そのため隣接する AEP のトポロジー情報を用いたフローエントリは無い。同様に、同じ AEP へ送信される inter-area traffic は、異なるエンドノードからの情報を用いたフローエントリでも 1 つのフローエントリに集約することができる。

各エリアコントローラも、把握する必要があるのはエリアのトポロジーのみなので、保持するエンドノードとスイッチのトポロジーの情報量が従来モデルより少ない。メインコントローラはネットワーク全体のトポロジーを把握する必要がある。しかし、メインコントローラが全ての情報を保持するのではなく、必要に応じてエリアコントローラから情報を入手すればよい。従って、メインコントローラが保持する必要があるトポロジーの情報はエリアコントローラとエンドノードだけであり、従来モデルより少ないと言える。

本モデルではネットワークをエリアに分割し、各エリアコントローラはエリア外にフローが到達した段階で目的地へ到達したと判断している。更に、メインコントローラからはエリアが 1 つのスイッチのように見えている。このような階層型にすることにより、同じエリアを複数回通るようなルーティングが困難となっている。従来のモデルではエリアという概念が無いので、特に問題とはなっていない。前章の例で用いた MPLS のように自由に使えるフィールドを用いれば不可能ではないが、注意して制御を行わないと AEP 同士の間でパケットが往復するようなループが発生してしまう恐れがある。

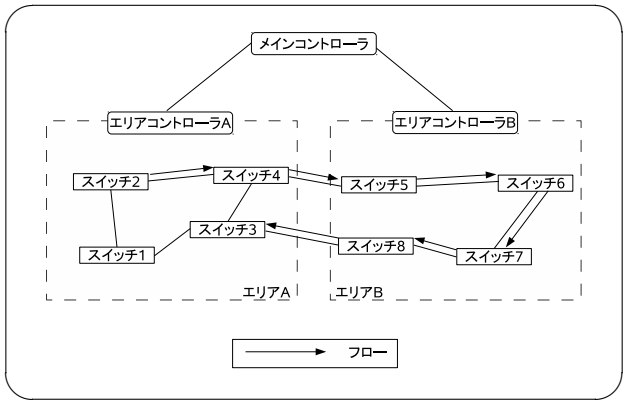

図 9: エリアを複数回通るルーティング

また、本論文のように inter-area traffic の制御に MPLS を用いる場合、エリア内とエリア間で使用するラベルを、

- 16 から 100 以下のラベル値は inter-area traffic 制御
- 101 以上のラベル値は intra-area traffic 制御

のように区別する必要がある。Intra-area traffic でのラベル値の利用状況に応じて構築できるエリア数が制限されてしまうと、スケーラビリティの問題に繋がる恐れがある。

Copyright © 2013 by Information Processing Society of Japan and
The Institute of Electronics, Information and Communication Engineers
All rights reserved.

本論文で提案した HARP モデルは、コントロールプレーンがエリアコントローラとメインコントローラの 2 階層型となっているが、エリアコントローラは下に対してはコントローラ、上に対してはスイッチとして動作するので、多段階に階層化することも可能である。例えば、混雑しているエリアの 1 階層下にサブ・エリアコントローラを設置することにより処理を分散できる。しかしこの場合、inter-area traffic の制御に用いている MPLS の更なる区別が生じてしまう等の問題がある。また、HARP モデルでは同レベルの階層でのエリアの分割方法によってパフォーマンスが左右される。実際のトラフィックに含まれる intra-area traffic と inter-area traffic の割合により最適な分割方法は変わってくる。例えば、1 つのエリアにファイアウォールのみを接続したリソースプールとなるエリアネットワークを作れば、フローはとどまることなくエリアを通過するだけなのでフローテーブルを簡略化できる。更に、ホップ・バイ・ホップモデルではスイッチがフルメッシュ接続されているトポロジーが想定される。この場合、エリア間を結ぶリンクが複数存在することになり、inter-area traffic を分割する必要がある。エリアコントローラにスイッチが保持しているカウンタ情報を用いるロードバランシング機能をデプロイし、状況に応じてフローエントリを変更する方法等が考えられる。

また、HARP モデルではネットワークの全てのスイッチが OpenFlow スイッチであるホップ・バイ・ホップモデルが前提となっている。しかし、エリア単位でオーバーレイモデルを用いて運用することも可能である。この場合、オーバーレイモデルを用いるエリアでは OpenFlow に対応していないスイッチを用いることもできる。

6. おわりに

6.1. まとめ

本論文では、OpenFlow のスケーラビリティ問題を解消する HARP モデルを提案した。ネットワーク全体をエリアに分割して、各エリアを担当するコントローラを配置することにより、OpenFlow コントローラに集中してしまう処理を分散し、更に OpenFlow スイッチ内のフローテーブルに保持される情報量も削減している。ネットワークの分割と複数のコントローラを使用することにより OpenFlow の特徴である一元管理が失われないように、各エリアコントローラを制御するメインコントローラを設けた。エリアレベルの制御と全体を見渡すエリア間の制御を分離することにより、コントロールプレーンを水平と垂直の両方向に分散できている。

6.2. 今後の課題

現在、この HARP モデルのプロトタイプ実装を行っている段階である。従来のモデルと比較し本モデルの実装に必要な主な要素は、下の階層にはコントローラとして振舞い、上の階層にはスイッチとして振舞うエリアコントローラである。現在、エリアコントローラ部分は、Trema[§] を基にして実装中である。今後、実装を用いて各種動作の確認が必要である。また、考察で述べた多段階の階層化を行った場合の挙動についても確認する必要がある。

参考文献

[1] Open Networking Foundation. Software-defined networking: The new norm for networks. https://www.opennetworking.org /sdn-resources/sdn-library/whitepapers (2013/4/12 アクセス), 2012.

[2] Zdravko Bozakov and Volker Sander. OpenFlow: A perspective for building versatile networks. In *Network-Embedded Management and Applications*, chapter 11. Springer, 2012.

[3] Open Networking Foundation. OpenFlow switch specification. https://www.opennetworking.org/sdn-resources/onf-specifications (2013/4/12 アクセス), 2009.

[4] Amin Tootoonchian and Yashar Ganjali. “ HyperFlow: A distributed control plane for openflow ”. pp. 3–3. Proceedings of the 2010 internet network management conference on research on enterprise network, 2010.

[5] Andrew R. Curtis, Jeffrey C. Mogul, Jean Tourrilhes, Praveen Yalagandula, Puneet Sharma, and Sajuta Banerjee. “ DevoFlow: Scaling flow management for high-performance networks ”. pp. 254–265. Proceedings of the ACM SIGCOMM 2011 conference, 2011.

[6] Minlan Yu, Jennifer Rexford, Michael J. Freedman, and Jia Wang.“ scalable flow-based networking with DIFANE ”. pp. 351–362. Proceedings of the ACM SIGCOMM 2010 conference, 2010.

[7] 高宮安仁, 鈴木一哉. OpenFlow 実践入門. 株式会社技術評論社, 2013.

[§] OpenFlow コントローラの Ruby と C 言語による開発用フレームワーク [7]。

Copyright © 2013 by Information Processing Society of Japan and
The Institute of Electronics, Information and Communication Engineers
All rights reserved.