OpenFlow アーキテクチャの自律分散化の実現に向けて

金海　好彦[†*]　　齋藤　修一[†]　　河合　栄治[†]

Yoshihiko KANAUMI[†*], Shuichi SAITO[†], and Eiji KAWAI[†]

1. まえがき

次世代のインターネットアーキテクチャについての議論が GENI [1] や、AKARI [2] で行われている。GENI 内の研究開発テーマの 1 つに、フロー (TCP/UDP ポートと宛先・送信元アドレスの組合せ) に注目したプログラマブルネットワークの OpenFlow [3] がある。

我々は、OpenFlow を JGN2plus [4] への構築を開始しており、その際に問題が明らかになった。そこで、OpenFlow のアーキテクチャの改善が必要であると認識した。OpenFlow が提案するアーキテクチャは、OpenFlow コントローラが OpenFlow スイッチを制御するが、OpenFlow コントローラと OpenFlow スイッチ間の接続が失われた場合の解決方法が、現在の OpenFlow には記述がない。

そこで、本稿では OpenFlow の問題点を解決するための改善方法案を提案する。

2. JGN2plus への展開計画とその準備から得た知見

図 1 のように、現在 JGN2plus 上に OpenFlow ネットワークの構築を計画している。構築にあたって、大手町リサーチセンタにて、Ethernet ベースの JGN2plus との接続検証、パフォーマンス検証を行った。OpenFlow ネットワーク構築には、OpenFlow スイッチとして NEC 社製の IP8800/S3640 シリーズをベースにした試作機を、OpenFlow コントローラとしては NOX [5] を使用した。

接続性検証とは、OpenFlow ネットワークの特徴を生かすため、OpenFlow スイッチを L2 ループで構成する必要があり、L2 ループを行うにあたって、JGN2plus と接続に必要な条件を明確にするための検証である。検証の結果、OpenFlow スイッチに、IEEE802.1q トンネルと、VLAN タグ変換を行う必要があることが確認できた。

パフォーマンス検証では、クラウドコンピューティングの技術の中から、MySQL Cluster [6] と Hadoop [7] を OpenFlow ネットワークの仮想利用者としてフロー発生を行った。結果、MySQL Cluster の初期状態では、約 200 セッションを MySQL Cluster ノードで確認でき、OpenFlow ネットワークには問題はなかった。

既存ネットワークとの接続は、OpenFlow スイッチの設定で対応することができた。また、パフォーマンスにも課題はあるが、約 200 セッションのフローを処理できたため、JGN2plus への展開可能になった。しかし、この検証を通して、OpenFlow コントローラと OpenFlow スイッチ間の接続が失われた場合、OpenFlow スイッチのパケットフォワーディングが止まってしまう現象があった。この問題の解決手法を、第 3. 章で提案する。

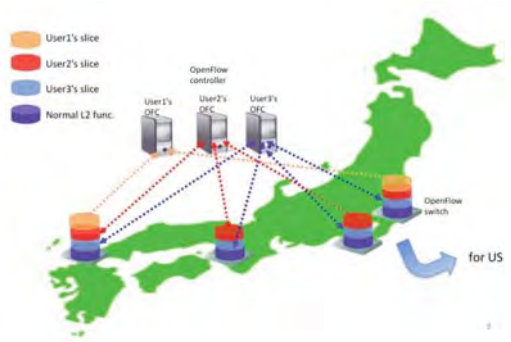

図 1　JGN2plus への展開 (予定)

3. OpenFlow のアーキテクチャとその課題

OpenFlow アーキテクチャは、OpenFlow スイッチと OpenFlow コントローラによって構成される (図 2)。(1)OpenFlow スイッチは、OpenFlow コントローラによって定義される処理方法を登録・参照するフローテーブルを持つ。(2)OpenFlow スイッチと OpenFlow コントローラ間は OpenFlow スイッチと OpenFlow コントローラの通信のためコントローラ用のオープンでかつ標準的な方法を許可する OpenFlow プロトコルで制御される。(3)OpenFlow プロトコルは Secure Channel と呼ばれる TCP もしくは SSL のセッション上で通信が行われる。

しかし、第 1. 章で述べたように現在の OpenFlow のアーキテクチャ(バージョン 0.8.9 [8]) には課題がある。OpenFlow スイッチは OpenFlow コントローラにより制御されるため、コントローラと接続が失われた場合の解決方法が必要である。解決方法には 2 つある。まず、OpenFlow スイッチ内でフローテーブルを消去しない実装が現状の OpenFlow のスペックである。しかし、フローテーブル保持を継続する場合、コントローラとの接続性が失われた時点でのフローテーブルが OpenFlow スイッチ内に残るため、フローの変化に応じたフローテーブルの更新が行われず、古

[†] 情報通信研究機構, 東京都
東京都千代田区大手町 1-8-1 KDDI 大手町ビル 21F
[*] 東京大学大学院 情報理工学系研究科, 東京都

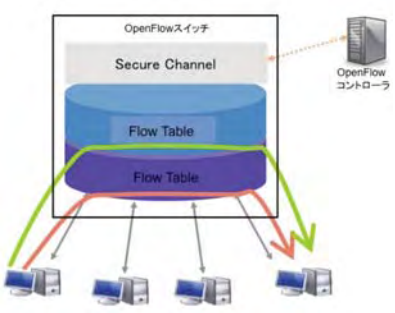

図 2　OpenFlow 全体図

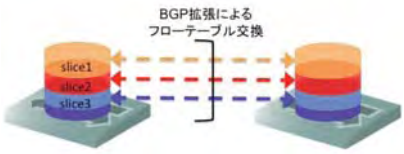

図 3　BGP 拡張によるフローテーブル交換

いフローテーブルでパケットフォワーディングする。次に、OpenFlow コントローラの冗長化が議論されている [9]。しかし、コントローラを冗長化するだけで、コントローラまでの到達性の冗長化は議論されていない。

4. 提　　案

OpenFlow コントローラとの接続性がなくなった場合に、OpenFlow スイッチが自律動作する手法を提案する。

各 OpenFlow スイッチが保持しているフローテーブルは、OpenFlow コントローラから与えられ、そのフローテーブルに従って、OpenFlow スイッチはパケットフォワーディングを行う (図 2)。

フローテーブルにはフロー検出条件として、レイヤ 1(物理ポート)、レイヤ 2(MAC アドレス、VLAN ID など)、レイヤ 3(IP アドレス、プロトコル番号)、レイヤ 4(TCP/UDP ポート番号) による任意のアドレス／識別子の組み合わせとフロー条件にマッチした場合に、OpenFlow スイッチがおこす Action(出力ポート、ヘッダ情報書き換えなど)、フローの統計情報が記載されている。

OpenFlow スイッチは、OpenFlow コントローラからの制御がない場合、OpenFlow スイッチが保持しているフローテーブルは更新されない。その場合、新たなフローの発生やフローの消去の変化に、追従できないため、OpenFlow スイッチがパケットフォワーディングできない。

そこで、OpenFlow コントローラとの接続性がない場合、OpenFlow スイッチが自律的にフローテーブル交換を BGP 拡張で実施する手法を提案する。我々は、BGP/MPLS VPNs [10] での、MPLS のタグ情報を BGP を用いて交換する手法を参考にした。

つまり、各スライス毎に保持しているフローテーブルを拡張した BGP を用いて、OpenFlow スイッチ間で交換する (図 3)。スライスとは、事前に OpenFlow ネットワーク内で定義されたもので、各スライスはそれぞれのフローテーブルを保持している。

5. 関 連 技 術

フローステートルーティング技術で、ユーザーの通信をまとめて処理する Angran 社の FR-1000 [11] や、IP ネットワーク上でも ATM のような細かなトラフィック制御機能を実装した Caspian Networks 社の Apeiro がある。

この 2 つ製品は、OpenFlow のようにフローテーブルをユーザ毎に定義する技術ではない。

6. 今後の課題

BGP 拡張の実施を OpenFlow のスイッチとコントローラに実装する必要があるが、まだ実施されていない。我々は、スイッチには、openflow-0.9 [8] や quagga [12] を利用予定である。フローテーブルは OpenFlow スイッチ側も保持されることになり、コントローラが保持しているフローテーブルと不整合が発生する可能性がある。そのため、フローテーブルの整合性をとるためのアプリケーション開発が必要になる。また、現在計画されている JGN2plus 上での OpenFlow 展開に併せて実証実験を行う。

なお、展開計画は、NTT/KDDI 大手町を中心とした JGN2plus の拠点から開始する。展開時に必要な検証も継続し、MySQL Cluster を使ったアプリケーションとして、今後 Mediawiki 等を用いての検証や、iperf や広帯域映像伝送によるパフォーマンス検証も実施することで、提案した実装の有効性を確認する。

文　　献

[1] “GENI,” http://www.geni.net/.
[2] “AKARI,” http://akari-project.nict.go.jp/.
[3] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “Openflow: enabling innovation in campus networks,” SIGCOMM Comput. Commun. Rev., vol.38, no.2, pp.69–74, 2008.
[4] “JGN2plus,” http://www.jgn.nict.go.jp/.
[5] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casadao, N. McKeown, and S. Shenker, “NOX: Towards an Operating System for Networks,” In submission. Also: http://nicira.com/docs/nox-nodis.pdf.
[6] “MySQL Cluster,” http://www-jp.mysql.com/products/database/cluster/.
[7] “Hadoop,” http://hadoop.apache.org/.
[8] “OpenFlow Reference System,” http://www.openflowswitch.org/.
[9] “Openflow 1.X Discussion,” http://www.openflowswitch.org/wk/.
[10] E. Rosen and Y. Rekhter, “BGP/MPLS IP Virtual Private Networks (VPNs),” RFC 4364 (Proposed Standard), Feb. 2006. Updated by RFCs 4577, 4684, 5462. http://www.ietf.org/rfc/rfc4364.txt
[11] “Angran, Inc.,” http://anagran.com/.
[12] “Quagga,” http://www.quagga.net/.