

End-to-End SDN アーキテクチャの提案

山中 広明[†] 河合 栄治[†] 石井 秀治[†] 下條 真司^{†,††}

[†] 情報通信研究機構 ^{††} 大阪大学

Proposal of the End-to-End SDN Architecture

Hiroaki Yamanaka[†] Eiji Kawai[†] Shuji Ishii[†] Shinji Shimojo^{†,††}

[†] National Institute of Information and Communications Technology ^{††} Osaka University

概要

アプリケーションの特性をネットワーク制御に反映させるには、エンドホストアプリケーションからネットワークを直接制御することが有効である。そこで、エンドホストが OpenFlow を用いて、ネットワークを直接制御可能なアーキテクチャを提案する。提案アーキテクチャでは、アプリケーションプログラムに、制御可能な専用の仮想 OpenFlow ネットワークを提供する。提供する仮想 OpenFlow ネットワークは、物理 OpenFlow ネットワークに対して、抽象的なトポロジをもつ。これにより、インフラ所有者が、実際の物理 OpenFlow ネットワーク上での経路を動的に決定可能になり、エンドホストによるインフラ資源の乱用を回避できる。

1 はじめに

インターネットを介して通信を行うアプリケーションは多様化してきており、トラフィックの特性をネットワーク制御に反映させ、所望の通信品質を得るための技術が必要である。特に、トラフィックの特性(必要帯域など)の頻繁な変化を反映させるには、エンドホスト上のアプリケーションプログラムより、ネットワークを制御することが有効である。

代表的な SDN (Software Defined Networking) 技術である OpenFlow [1] では、ネットワークを制御するための標準化されたインタフェースがある。外部のコントローラより、ネットワーク上の全スイッチのフローエントリを管理でき、動的なフロー制御を実現できる。現状の OpenFlow は、トラフィックエンジニアリングや、動的な論理ネットワーク構築などの目的で、ネットワーク管理者により用いられている。

そこで、エンドホスト上のアプリケーションプログラムから直接、OpenFlow ネットワークを制御させることを考える。これを実現するため、物理 OpenFlow ネットワーク上で、それぞれのアプリケーション専用の仮想 OpenFlow ネットワークを構築する。一般にエンドホスト間の通信経路上には、多数のスイッチがあるため、大規模なネットワーク上で仮想 OpenFlow ネットワークを構築して、制御可能にする必要がある。

物理 OpenFlow ネットワークを提供する InP (Infrastructure Provider) としては、エンドホストによる所有インフラの乱用を避ける必要がある。また、仮想 OpenFlow ネットワーク上では、複数の InP の物理 OpenFlow ネットワークに接続しているエンドホスト同士が通信する。そこで、他の InP との連携で、仮想 OpenFlow ネットワークを構築する必要がある。特に、他の InP との事前調整は、可能な限り少ない

方が望ましい。

本稿では、エンドホスト上のアプリケーションプログラムが制御可能な仮想 OpenFlow ネットワークを提供する、「End-to-End SDN アーキテクチャ」を提案する。提案アーキテクチャでは、フロースペース(フロー定義のためのパケットヘッダ空間)を仮想化する。これにより、仮想 OpenFlow ネットワーク上では、自由なフロー定義が可能になる一方で、物理 OpenFlow ネットワーク上のフローは、仮想 OpenFlow ネットワークごとに隔離される。また、InP ごとに仮想化システムの独立な管理を可能にすることで、InP 同士の連携を容易にする。さらに、物理 OpenFlow ネットワークのトポロジを抽象化(物理 OpenFlow スイッチを隠蔽、統合)して、仮想 OpenFlow ネットワークのトポロジを構成する。これにより、物理 OpenFlow ネットワーク上にスイッチが多数あっても、小規模な仮想 OpenFlow ネットワークを構成できる。また、仮想 OpenFlow ネットワーク上で設定されたフローの経路に対して、実際の物理 OpenFlow ネットワーク上で対応する経路は、InP が動的に決定できるので、エンドホストによるインフラ資源の乱用は回避できる [2]。

2 End-to-End SDN アーキテクチャの提案

2.1 仮想/物理 OpenFlow ネットワーク対応モデル

物理 OpenFlow ネットワークは、複数のエッジネットワークと、それらを接続するコアネットワークにより構成される。エッジネットワーク、コアネットワークはそれぞれ、異なる InP が所有している。次に述べる、フロースペースとトポロジの対応関係に基づき、仮想 OpenFlow ネットワークを構成する。

フロースペース: 仮想 OpenFlow ネットワーク上で利用されるフロースペースに対して、物理 OpenFlow

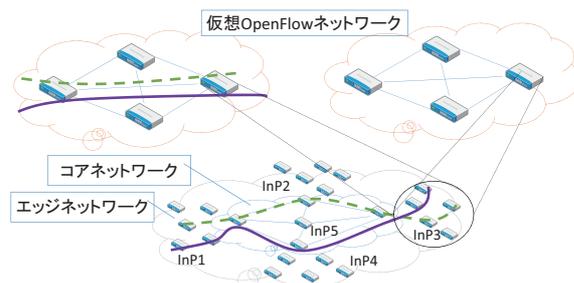


図1: トポロジ対応関係の例

ネットワーク上では、ユニークなフロースペースが割当てられる。具体的には、エンドホストが定義したフローエントリのマッチングルールは、仮に他の仮想 OpenFlow ネットワーク上で同一のルールが用いられていても、物理 OpenFlow ネットワーク上では異なるルールになる。アプリケーションプログラムが送受信するデータパケットヘッダについても同様に、異なる仮想 OpenFlow ネットワーク上のもは、必ず異なるヘッダに書換えられた上で、物理 OpenFlow ネットワーク上では扱われる。これにより、仮想 OpenFlow ネットワークの隔離と、その中での自由なフロー定義を両立する。

トポロジ: 図1に示すように、物理 OpenFlow ネットワークのトポロジを抽象化して、仮想 OpenFlow ネットワークのトポロジを構成する。エッジネットワークについては、単一の仮想 OpenFlow スイッチに抽象化する。コアネットワークについては、物理 OpenFlow スイッチは隠蔽して、仮想リンクに抽象化する。仮想リンクにより、エッジネットワークを抽象化した仮想 OpenFlow スイッチを接続して、仮想 OpenFlow ネットワークのトポロジを構成する。この構成モデルは、エッジネットワークでは複雑なフロー制御を行う一方、コアネットワークではラベルを用いて、比較的単純なパケット転送を行う、広域ネットワークへの SDN 適用モデル [3] に基づくものである。図1に例示するように、物理ネットワークのトポロジを抽象化することで、多数の物理 OpenFlow スイッチに対して小規模な仮想 OpenFlow ネットワークを構成できる。また、物理 OpenFlow ネットワーク上での実際の経路を InP が動的に決定できるので、エンドホストによる物理資源の乱用を回避できる。

2.2 仮想 OpenFlow ネットワークの構成と制御

図2に示すように、物理 OpenFlow ネットワークと「階層仮想化システム」により、仮想 OpenFlow ネットワークを構成する。最上位コントローラは、仮想 OpenFlow ネットワークごとにあり、エンドホストが送る制御メッセージを階層仮想化システムに中継する。階層仮想化システムは、InP ごとにあり、所有する物理 OpenFlow ネットワーク上で、フロースペースの仮想化とトポロジの抽象化を行う。具体的には、上述したトポロジとフローの対応関係に基

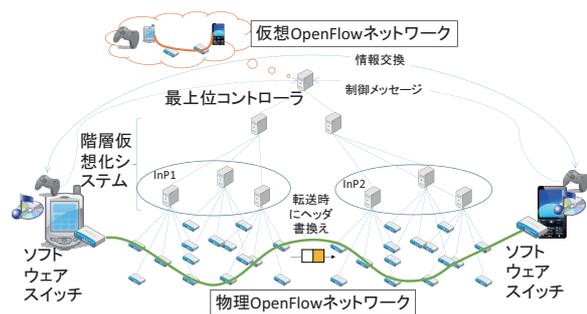


図2: End-to-End SDN アーキテクチャ

づき、OpenFlow 制御メッセージの内容を変換して中継する。階層的な分散システムにより、InP 内の膨大な数のスイッチに対応できる。そして、複数の階層仮想化システムが集まり、複数の InP 上で仮想 OpenFlow ネットワークを構成する。仮想 OpenFlow ネットワークに対する制御メッセージは、階層仮想化システムを通して、物理 OpenFlow ネットワーク上での制御内容に反映される。

InP の連携において、事前調整を少なくするため、InP ごとにフロースペースの割当てを管理している。このため、仮想 OpenFlow ネットワーク上では同一のデータパケットでも、InP ごとにヘッダが異なることがある。そこで、異なる InP の物理 OpenFlow スイッチ間でデータパケットが転送される際は、InP 相互に対応パケットヘッダを問い合わせ、ヘッダを書換えた上で転送する。

エンドホストは、必要に応じて情報を交換しながら、分散的に仮想 OpenFlow ネットワークを制御する。フロースペースを仮想化しているため、アプリケーションプログラムが認識しているデータパケットヘッダと、物理 OpenFlow ネットワーク上での実際のデータパケットヘッダは異なる。そこで、データプレーンについては、エンドホスト内のソフトウェアスイッチを用いて、ヘッダを書き換えた上でデータパケットを物理 OpenFlow ネットワークに送信する。

3 まとめと今後の課題

アプリケーションの特性を柔軟にネットワーク制御に反映させるため、End-to-End SDN アーキテクチャの提案を行った。今後の課題として、アプリケーションプログラムによるフロー制御モデルの考察、フロー制御のためのインタフェースの設計と実装がある。

参考文献

- [1] N. McKeown, et al. OpenFlow: Enabling innovation in campus networks. In *Proceedings of SIGCOMM 2008*, pp. 69–74, 2008.
- [2] 嶋村昌義, et al. 多様なサービスプロバイダの収容を可能とするミドルレイヤ型ネットワークアーキテクチャの検討. 電子情報通信学会技術報告 (NS2011-192), 第 111 巻, pp. 75–80, 2012.
- [3] The SPARC Consortium. SPARC deliverable d2.1: Initial definition of use cases and carrier requirements. <http://www.fp7-sparc.eu/home/deliverables/>, 2010.