



Software Defined Networking (SDN) を支える OSS 技術

基
専

■ 藤田智成 (NTT ソフトウェアイノベーションセンタ)

従来ネットワークの課題

従来のネットワーク装置はデータプレーン（ハードウェア）とコントロールプレーン（装置内ソフトウェア）の2つの機能が1つに集約されている。装置の1台ずつが独立したOS、制御ソフトウェアやデータ転送機能を有しており、機能は固定されている。ユーザがネットワーク装置に新たな機能を追加するためには、IETF（Internet Engineering Task Force）などの団体を通じて標準化し、ネットワーク装置ベンダがその機能を実装するのを待つ必要があった。

しかし、数年単位で時間を要する従来の標準化方法では、現在のビジネスで求められる要件に迅速に対応できない。たとえば、米国 Google は、アプリケーションの要件に合わせてネットワークを制御するために、ネットワーク装置に独自の機能を実装しているが、その機能は標準化していない¹⁾。

現在、クラウドコンピューティングなど、ネットワークに関連する新しい技術が数多く生まれている。そのため、ネットワーク装置を買い換えることなくこれらの新しいネットワーク技術に対応できる、拡張性や柔軟性があるネットワーク機能が強く求められている。この要求に迅速に対応するため、従来課題であったネットワーク装置ベンダによる機能の実装を待つのではなく、独自に機能が実装できる、SDN（Software Defined Networking）という新しいコンセプトの普及が進んでいる。

SDN の概要

SDN は特定の技術や製品を指すのではなく、ネッ

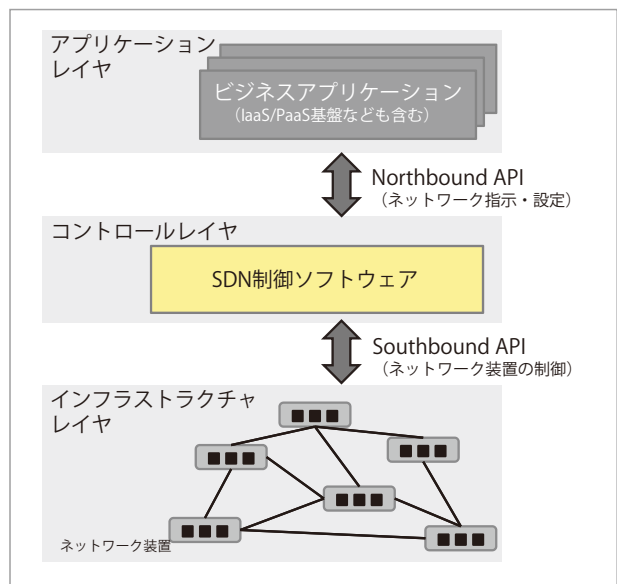


図-1 SDN の論理的な構成

トワーキングの設計・構築・運用に関する新しい概念である。SDN は、文字通り、「ネットワークをソフトウェアで定義する」というコンセプトを意味する。また、このコンセプトに基づいたネットワークのアーキテクチャの意味でも使われている。

従来のネットワーク装置も、データプレーンと、コントロールプレーンの2つを1つにまとめた形で構成されており、その意味では SDN と同様に、ソフトウェアで定義されているということもできる。しかし、ネットワーク装置ベンダは、ユーザがデータプレーンにアクセスすることを許していないため、ユーザが独自のネットワークプロトコルを装置に実装するようなことは不可能であった。

SDN では、これまでネットワーク装置内部で密に結合されていたデータプレーンとコントロールプレーンを別レイヤに分離する。

SDN の論理的な構成を図-1に示す。ビジネスロジックなど、それぞれのユースケースの要件を実現するアプリケーションレイヤ、アプリケーションレイヤからの指示をルーティングやフォワーディングなどのネットワーク装置の動作に解釈し、ネットワーク装置を制御するコントロールレイヤ、1台以上のネットワーク装置で構成されるインフラストラクチャレイヤからなる。

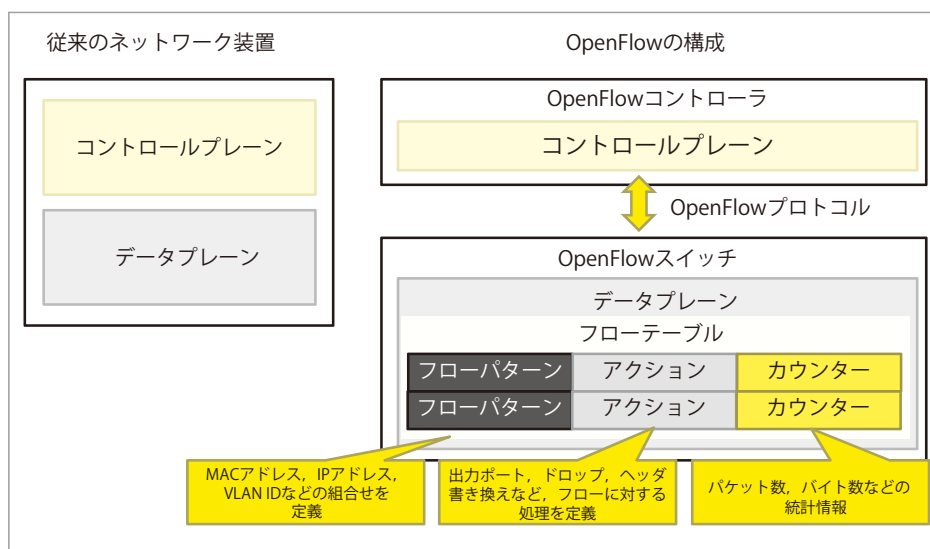


図-2 従来のネットワーク装置と OpenFlow との違い

SDN と OSS プロジェクト

OpenFlow

OpenFlow は、SDN のデータプレーンとコントロールプレーン間で使われるデータプレーン制御プロトコルであり、図-1のコントロールレイヤとインフラストラクチャレイヤの間をつなぐものである。OpenFlow 以外にもネットワーク装置の制御プロトコルは存在するため SDN の実現に必須ではないが、後述するように、従来のプロトコルよりも柔軟な制御が可能でさまざまなユースケースに対応できることもあり、現在注目を浴びている。

OpenFlow プロトコルは、2008 年頃より、米国 Stanford 大学を中心とした OpenFlow スイッチコンソーシアムで検討が進められてきた。現在は、SDN の普及を促進する ONF (Open Networking Foundation)²⁾が、OpenFlow の標準化を進めている。

OpenFlow は、プロトコルヘッダ情報の組合せに基づき、データプレーンを流れるパケットの転送処理を制御する。これらプロトコルヘッダ情報の組合せはフローパターンと呼ばれ、たとえば、「ある特定の送信元 MAC アドレスから送信する、ある特定の宛先 IP アドレス」のように、フローパターンを

定義することができる。OpenFlow は、フローパターンに一致するパケットに対して、廃棄、特定のポートへの出力、パケットの書き換えなど、さまざまなアクションを組み合わせて実行でき、従来のネットワーク装置よりも柔軟な制御ができる。

OpenFlow プロトコルで制御可能なネットワーク装置を OpenFlow スイッチ、OpenFlow プロトコルを使ってそれらスイッチを制御するコンポーネントを OpenFlow コントローラと呼ぶ。図-2に、従来のネットワーク装置と OpenFlow との違いを示す。

OpenFlow スイッチの実現方式には、ハードウェアとソフトウェアの2種類がある。前者は、従来のネットワーク装置同様に、パケット転送などの中核となる機能を実現する集積回路(スイッチチップ)を搭載している。後者は、汎用のサーバで動作し、CPU でパケット転送を実行する。この OpenFlow ソフトウェアスイッチは、一般に、複数の仮想マシンが接続される仮想的なスイッチとして利用される。

OpenFlow はパケット転送を制御するためのプロトコル処理であり、装置の設定、たとえば、物理ポートを有効・無効にするなどの操作はサポートしない。ONF では、装置の設定プロトコルとして、OF-Config (OpenFlow Configuration Protocol) の標準化を進めているが、OF-Config はまだ普及が進んでおらず、多くの OpenFlow スイッチは、独自

の設定プロトコルをサポートしている。

■ SDN のインフラストラクチャレイヤの OSS

OpenFlow スイッチコンソーシアムは、仕様の策定と連携して、OSS のリファレンス実装を公開している。ONF もこの OSS プロジェクトを支援するなど、OpenFlow はその誕生当初から OSS との強いつながりを持っている。

Open vSwitch (OVS) は、OpenFlow の策定に携わっていたメンバが起業した米国 Nicira が、中心となって実装したソフトウェアで、Nicira が米国 VMware に買収された後も活発に開発が続けられている。

OVS の開発者が、ONF で OpenFlow の仕様の策定にかかわっているため、OVS は、OpenFlow の将来的な機能の実験の場として、OpenFlow に大きな影響を与えている。OVS に新たな機能として実装され、その有効性などが確認されてから、ONF で OpenFlow として標準化された機能が多く存在する。たとえば、さまざまなフローパターンの定義が可能な type-length-value のデータ構造は、OVS から持ち込まれたものである。

また、ONF が標準化を進めるスイッチ管理プロトコル OF-Config の普及が進まない反面、OVS のスイッチ管理プロトコル OVSDB (Open vSwitch Database Management Protocol) が他の OpenFlow スイッチの管理プロトコルとして採用されるなど普及が進んでおり、OSS の実装が標準化よりも先行し、大きな影響を与えている。

OVS の他の代表的な OpenFlow ソフトウェアスイッチとしては、NTT 研究所が中心に開発を進めている Lagopus、米国 Big Switch Networks が中心に開発を進めている Indigo などがある。Lagopus は、米国 Intel の DPDK (Data Plane Development Kit) と呼ばれる開発キットを活用し、汎用サーバで高速なデータプレーンを実現する。Indigo は、ソフトウェアスイッチとして動作するだけでなく、ハードウェアスイッチに搭載されているスイッチチップも制御できるという特徴を持つ。

■ SDN のコントロールレイヤの OSS

スイッチを制御するコントローラは、前述の OpenFlow コントローラのほかに、OpenFlow 以外のスイッチ制御プロトコルもサポートすることで SDN までを視野に入れたソフトウェア (SDN コントローラ) がある。OpenFlow コントローラは、OpenFlow スイッチだけのネットワークをサポートするが、SDN コントローラは、たとえば、NETCONF など既存の装置管理プロトコルや BGP (Border Gateway Protocol) など既存のルーティングプロトコルをサポートし、OpenFlow スイッチと既存ネットワーク装置の混在したネットワークをサポートすることができる。

OpenFlow コントローラと SDN コントローラはどちらも、スイッチ制御プロトコルの詳細、たとえば、プロトコルのバイナリフォーマットを隠蔽することで、ユーザのアプリケーションレイヤのソフトウェアの実装の負荷を軽減している。

OpenFlow コントローラとして、最初の OSS は、初期の OpenFlow の策定にかかわった研究者が中心となって開発した NOX である。NOX は C++ で実装されており、ユーザは、C++ または、Python でアプリケーションを実装できる。NOX、および、その後継として開発された OpenFlow コントローラの POX は、主に研究目的で開発されており、OpenFlow が実用時期となった現在は開発が停止している。

現在、最も普及している OSS の OpenFlow コントローラは、Big Switch Networks が中心となり開発を行っている Floodlight である。Floodlight は、Java で実装されており、ユーザアプリケーションも Java で実装可能である。その他の活発に開発されている OSS の OpenFlow コントローラとしては、NEC が中心となって開発を進めている Trema があげられる。Trema は C 言語で実装されており、C または Ruby で、ユーザアプリケーションを実装できる。

これに対し、現在広く使われている OSS の SDN コントローラとしては、ODL (OpenDaylight) と

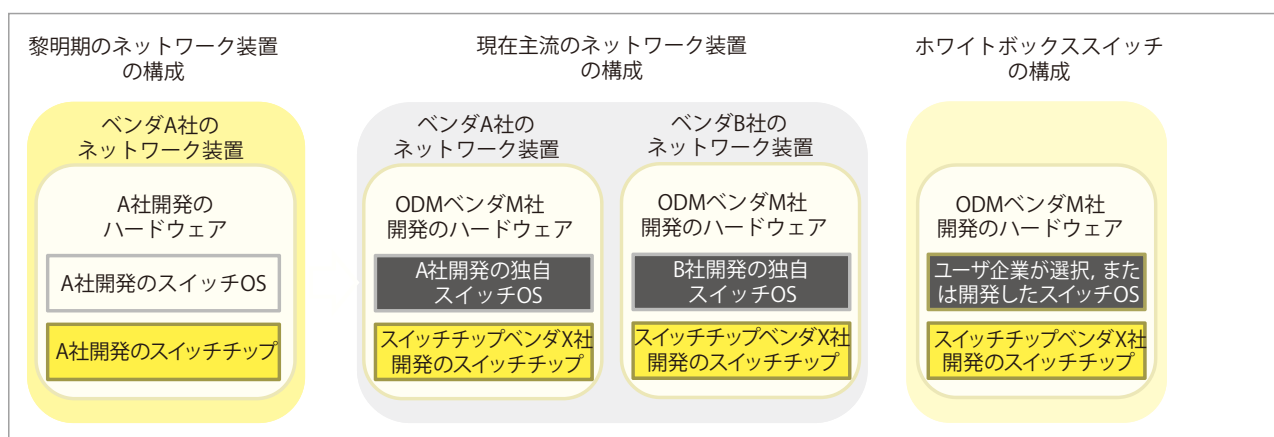


図-3 ネットワーク装置の構成の変化

Ryu がある。

ODL は、米国 Cisco が立ち上げ、現在は、Linux Foundation 傘下の OSS プロジェクトとして、主にネットワーク装置ベンダによって開発が進められている。ODL は Java の OSGi Framework を利用して実装されており、バンドルされている多くの機能を動的に有効にできるという特徴を持つ。ユーザは Java を用いてアプリケーションを実装することができる。

Ryu は、NTT 研究所を中心に開発が進められている。Ryu は、Python で実装されており、ユーザは Python を用いてアプリケーションを実装することができる。スイッチを制御するコントローラとしてだけでなく、クラウドにおける IaaS 基盤を構築するための OSS である OpenStack の、BGP 機能や BGP Monitoring Protocol サーバなど SDN を実現するコンポーネントとして利用されている。

装置のコモディティ化

SDN の一環として、ネットワーク装置において、ホワイトボックススイッチと呼ばれる新しいトレンドが生まれている。ホワイトボックスとは、汎用の部品を組み合わせて設計・製造し、スイッチ OS を搭載しない状態で販売する、ネットワーク装置のことである。大手ネットワークベンダのハードウェアを設計・製造していた ODM (Original Design Manufacturer) ベンダ、たとえば台湾 Quanta Comput-

ing などがこのトレンドを牽引している。Google、米国 Facebook など、大規模なデータセンタを持つ事業者がホワイトボックススイッチを使い始め、ほかの大手サービス事業者やデータセンタ事業者も導入を始めている。ホワイトボックススイッチは、その多くが 1U サイズのもので、ラック内の最上位に配置され、ラック内のサーバを束ねるトップ・オブ・ラックスイッチとして利用されることが多い。

ネットワーク装置の黎明期には、各ネットワーク装置ベンダが、自社製品のために独自のスイッチチップを開発していた。しかし現在は、米国 Broadcom や Intel などの、低価格で量産できる汎用のスイッチチップが十分高性能となり、ネットワークチップを開発できない小規模なベンダも安価なネットワーク装置を開発できる。このため、汎用スイッチチップを用いて ODM ベンダが設計・製造したネットワーク装置を、各装置ベンダが自社のスイッチ OS と組み合わせ、自社製品として販売する形態が主流になっている。この変化を図-3 に示す。

ホワイトボックススイッチを利用するためには、ユーザは、ホワイトボックススイッチ用のオペレーティングシステム (OS) を開発するか、別途入手する必要がある。米国 Cumulus Networks のように、自社のネットワーク装置は販売せずに、スイッチ OS のみを提供するベンダも存在する。

スイッチ OS を開発する上での障壁の 1 つに、スイッチチップにアクセスするソフトウェアの実装に必要な情報やライブラリを得るためにチップベンダ

との秘密保持契約が必要なことが挙げられる。実装したソフトウェアのソースコードも公開できないため、現在はまだ、コミュニティ開発という意味でのOSSのスイッチOSは存在しない。

しかしながら、Broadcomが、ホワイトボックススイッチをOpenFlowスイッチとして動作させるためのスイッチOS、OF-DPA (OpenFlow Data Plane Abstraction) をOSSとして公開するなど、オープン化の動きが出てきている。Facebookは、自社で設計したネットワーク装置の仕様およびスイッチOSを、Open Compute Project (OCP)³⁾を通じて、公開するとアナウンスしている。OCPは、データセンターで使われるサーバやネットワーク装置などのハードウェア仕様のオープンソース化を推進する団体である。

今後の展望

SDNにおける、データプレーンとコントロールプレーン間の標準制御プロトコルがOpenFlowになるのかは、まだ不透明である。OpenFlowプロトコルをベースとして独自に拡張したソリューションを販売するベンダや、ホワイトボックススイッチを利用して、独自のデータプレーン制御プロトコルを実装する大規模事業者も存在しており、現在のOpenFlowがユーザのニーズを完全に満たしているとは言えない状況である。今後もOpenFlowが主要なデータプレーンの制御プロトコルであるかどうかは、ONFが、OSSの動向やユーザのニーズを、素早く標準化できるかどうかにかかっている。

OpenFlowプロトコルの動向としては、ONFにおいてフローパターンをプロトコル非依存にする方向が議論されている⁴⁾。現在のOpenFlowプロトコルは、既存のプロトコル、たとえば、イーサネットヘッダの特定のフィールドの値というようにフローパターンを定義する。これを、パケットのバイト列の特定の位置の値というようにフローパターンを定義できるようにすることで、さらにデータプレーンの制御の自由度が高まる。しかしその反面、この

ような処理ができるスイッチチップはコストが高いため、今後、OpenFlowプロトコルがこの方向に拡張されるかどうかは不明である。

今回は解説しなかったが、SDNにおける、アプリケーションレイヤがコントロールレイヤにアクセスするためのAPI (Northbound API) の標準化についてもONFが進められている。ただし、典型的なSDNのユースケースやデザインパターンなどが確立していないため、まだ先行きは不透明な状況である。Northbound APIが標準化されるとしても、まだ時間がかかるだろう。

また、ホワイトボックススイッチが大規模な事業者以外にも広がるかどうかは、Facebookが中心となって進めているOCPなどのホワイトボックススイッチ仕様のコモディティ化や、スイッチチップへのアクセスのオープン化の動きにかかっている。ネットワーク装置がコモディティ化し、ハードウェアへのアクセスが自由にできる状況になれば、OSS、および商用スイッチOSの選択肢が増える。

SDNの普及はまだ始まったばかりで、さまざまな課題があるが、OSSの力を借りて急速に進歩していくに違いない。今後、サーバ分野のように、ネットワーク分野でも、専用のハードウェアがコモディティ化し、オープンソースを含めたソフトウェアがビジネスの主流となっていくであろう。

参考文献

- 1) Jain, S., et al. : B4 : Experience with a Globally-Deployed Software Defined WAN, ACM SIGCOMM Computer Communication Review, Vol.43, No.4, pp.3-14 (2013).
- 2) Open Networking Foundation, [https:// www. opennetworking.org/](https://www.opennetworking.org/)
- 3) Open Compute Project, <http://www.opencompute.org/>
- 4) Bosshart, P. et al. : P4 : Programming Protocol-Independent Packet Processors, ACM SIGCOMM Computer Communication Review, Vol.44, No.3, pp.87-95 (2014).

(2014年10月1日受付)

■ 藤田智成 (正会員) fujita.tomonori@lab.ntt.co.jp

2000年早稲田大学大学院理工学研究科修士課程修了。同年日本電信電話(株)に入社。オペレーティングシステムに関する研究に従事。