

# 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

小出 俊夫 芦田 優太 下西 英之

## 要旨

SDNの出現によって、固定的な制御プロトコルの束縛から解放され、ネットワーク運用者のニーズを満たす制御ソフトウェアをSDNコントローラとして柔軟に記述可能となりました。今後はさまざまなSDNコントローラを短時間で数多く作成するための、コントローラ作成のシンプル化が課題となります。本稿では、ネットワーク機器の制御プロトコルの差異を吸収し、ネットワーク資源の仮想化や複雑な経路制御アルゴリズムの実装をシンプルに記述可能とする、ネットワーク抽象化モデルを提案します。提案するモデルに基づき設計されたSDNコントローラ構築基盤のプロトタイプによって、SDNコントローラを短時間でシンプルに作成できることを示します。



SDN/SDNコントローラ/OpenFlow/OpenFlowコントローラ/ネットワーク抽象化

## 1. はじめに

一般的なプログラミング言語で実装されたソフトウェアにより、柔軟な情報通信ネットワーク制御を実現するSDN (Software-Defined Networking) が注目を集めています。ONF (Open Networking Foundation)<sup>1)</sup>で標準化が進むOpenFlow<sup>2)</sup>を用いたネットワーク制御はその具体例の1つです。従来、情報通信ネットワークの構築と運用は、制御機能とデータ処理機能が一体となった専用のネットワーク装置を組み合わせることで構築し、設定することで運用していました。そのため運用者が制御機能を柔軟に変更できず、これまで想定していなかった高度なネットワーク仮想化や可視化といった新たな要求が生まれても素早く対応できないという問題がありました。近年はOpenFlowなどの技術の出現により、ネットワーク装置から制御機能を分離して、これをSDNコントローラとして一般的なプログラミング言語で記述可能な環境が整い始めており、その状況は改善されつつあります(図1)。

一方で、各ネットワークの運用ポリシーを反映したSDNコントローラを用意しようとする、柔軟な制御が可能になる代わりに、プログラミングという新しいスキルが要求されることとなります。SDNへの移行の敷居を下げるため、プ

ログラミングスキルに偏重しすぎないような、SDNコントローラ作成のシンプル化が望まれます。また、SDNのコンセプトはGMPLSネットワーク、IP-VPNなど、OpenFlow以外のさまざまなネットワーク種別でも実現可能ですが、ネットワーク種別ごとの個別の再実装や、複数のネットワーク種別にまたがるネットワーク制御の複雑化が起らないよう、ネットワーク種別の差異にとらわれない抽象化手法が望まれます。

そこで本稿では、SDNコントローラ作成の容易化と、ネットワーク種別の差異の吸収を実現するための、ネットワーク抽象化モデルを提案します。提案するモデルにより、SDNコントローラの作成に必要とされる機能群を汎用的なソフトウェア部品としてあらかじめ用意してつなぎ合わせることや、下位のネットワーク種別の差異を隠すことが容易になり、部

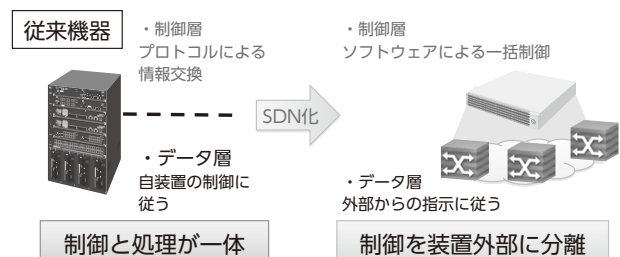


図1 制御層とデータ層の分離により実現するSDN

品を組み合わせるというシンプルな作業でSDNコントローラが作成できるようになります。本稿では、その実現可能性を確認するために実装したプロトタイプ的设计についても述べます。

## 2. ネットワーク抽象化モデル

本章では、シンプルかつ普遍的なネットワーク抽象化モデルについて述べます。

SDNコントローラの開発においては、プロトコルによる各スイッチの設定よりも、複数のスイッチにまたがるネットワーク全体の制御、すなわちネットワークトポロジーの把握、複数スイッチにまたがったパケットの転送経路の制御などが重要となります。加えて、ネットワークの状態変化に対しての動的な最適化も必要です。そこで、我々はネットワーク全体のビューを持った制御と、そのビューにおける状態変化に対する動的な制御を可能とするための、必要最小限のネットワーク抽象化モデルを設計しました。具体的には、ネットワークの状態をトポロジー、フロー群、パケット群の3つのデータで表現し(図2)、ネットワークの変化をイベントで表現するよう設計しました。「フロー」はOpenFlowのフローエントリとは異なり、ネットワーク全体にわたる転送経路を含む概念です。更に、ネットワーク全体の状態を包括的に表現す

るデータモデルを定義し、そのデータに対する操作でネットワークの制御を表現することにより、物理層のプロトコルとネットワーク制御の分離を実現しました。図3は、これらを標準的なハッシュオブジェクトとして表現した例です。

- トポロジー (図3 (a))**  
 トポロジーは、ネットワークの識別子と、ノードの一覧と、リンクの一覧からなります。ノードは、ノード識別子と、通信ポートの一覧からなります。リンクは、リンク識別子と、送信先のノードと通信ポートの識別子、宛先のノードと通信ポートの識別子からなります。
- フロー群 (図3 (b))**  
 フロー群は、複数のフロー情報と、フローの優先順位からなります。フロー情報は、フロー識別子と、フロー

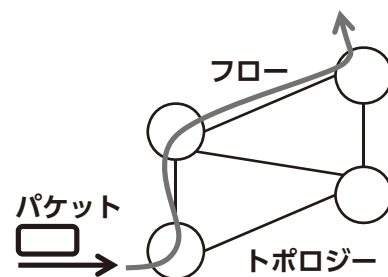


図2 トポロジー、フロー、パケットの概念図

<pre> {   "network": "network1",   "nodes": {     "1": {       "id": "1",       "ports": {         "1": { "id": "1", "out_link": "1", "in_link": "3" },         "2": { "id": "2", "out_link": null, "in_link": null },       },     },     "2": {       "id": "2",       "ports": {         "1": { "id": "1", "out_link": null, "in_link": null },         "2": { "id": "2", "out_link": "2", "in_link": "4" },       },     },   },   "links": {     "1": { "id": "1", "src_node": "1", "src_port": "1", "dst_node": "2", "dst_port": "3" },     "2": { "id": "2", "src_node": "2", "src_port": "2", "dst_node": "1", "dst_port": "4" },   }, }             </pre> <p>(a)</p>	<pre> {   "1": {     type: "BasicFlow",     id: "1",     match: {       in_node: "1",       in_port: "1",       dl_dst: "66:55:44:33:22:11"     },     path: ["1", "2", "3"],     edge_actions: {       node1: [{"output": "2"}, {"output": "3"}],     },   },   "2": {     ...   },   ...   "priority": ["2", "5", "1", "3"] }             </pre> <p>(b)</p>
<pre> {   "data": &lt;packet_data&gt;,   "node": "1",   "port": "5",   "time": 1353397038 }             </pre> <p>(c)</p>	<pre> {   "event_type": "PortChanged",   "node": "1",   "port": "2",   "action": "update",   "old": {"id": "2", "out_link": null, "in_link": null},   "new": {"id": "2", "out_link": "5", "in_link": "6"} }             </pre> <p>(d)</p>

図3 ネットワーク抽象化モデルに基づく表現例

の処理規則について書かれたオブジェクトからなります。処理規則は、処理するパケットの条件 (match)、パケットの送信経路 (path)、パケットの処理方法 (actions) からなります。pathはリンク識別子のリストで表現され、フローの優先順位はフロー識別子のリストの順序として表現されます。

• **パケット群 (図3 (c))**

パケット群は、入力パケットと出力パケットのそれぞれのリストからなります。パケット情報はノードの識別子、通信ポートの識別子、パケットヘッダ、ペイロードからなります。入力パケットには、これらに受信時刻も加わります。

• **イベント (図3 (d))**

トポロジー、フロー群、パケット群の情報の変化は、イベントによってコンポーネントに通知されます。各コンポーネントは、このイベントを用いてネットワークの変化をリアルタイムに察知し、即座に対応をすることが可能です。

### 3. SDNコントローラ構築基盤の設計

提案するネットワーク抽象化モデルに基づいて、SDNコントローラ構築基盤のプロトタイプを設計・実装しました。本プロトタイプにより、SDNコントローラ作成のシンプル化を検証します。本プロトタイプでは、ネットワーク抽象化モデルに基づいて、SDNコントローラを構成する各コンポーネントを定義しました。

コンポーネントは、「ネットワークコンポーネント」と「ロジックコンポーネント」の2つに分けて考えることができます。ネットワークコンポーネントは、前述のネットワーク抽象化モデルに基づいてネットワーク状態を保持するデータベースからなるコンポーネントです。ロジックコンポーネントは、1つ以上のネットワークコンポーネントと接続して、ネットワークコンポーネント内のデータを操作するコンポーネントです。ロジックコンポーネントの目的はさまざまですが、ネットワークの抽象化、変形、制御などを行うコンポーネントに大別できます。

このようにして定義されたネットワークコンポーネントとロジックコンポーネントを、あたかもブロックをつなぎ合わせるように接続するだけで、基本的なSDNコントローラが作成できるようになります。したがって、制御対象であるネット

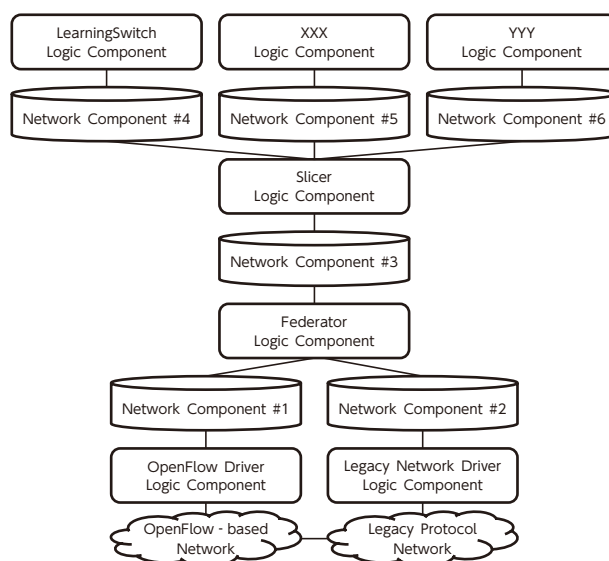


図4 SDNコントローラ構成例

ワークを構成する物理装置やその制御プロトコルに関する知識を持たないユーザーであっても、SDNコントローラを作成し、ネットワークを運用することができます。

図4は、それぞれ異なった制御方式を持つネットワークドメインを抽象化するDriver、それらを統合して1つのネットワークとして構成するFederator、更に複数のユーザー（制御ロジック）にスライスするSlicerといったロジックコンポーネントを組み合わせる例を示しています。標準的なコンポーネントを組み合わせるだけで、物理ネットワークの制御プロトコルが異なる複数ドメインの統合、仮想ネットワークの構築、仮想ネットワークごとに異なった経路制御アルゴリズムの適用を実現する高機能なSDNコントローラを作成することも可能です。

更に、今回開発したプロトタイプでは、SDNコントローラを作成をよりシンプルにするため、Web経由でコンポーネントの操作が可能なGUIを備えています。ブラウザ上でこれらのコンポーネントを自由に組み立てることで、SDNコントローラを作成できることが確かめられました。各ネットワークコンポーネントで把握されているトポロジー情報などもGUI上で表示することができ、作成だけではなく運用サポートツールとしての発展も考えられます (図5)。

### 4. むすび

本稿では、シンプルかつ普遍的なネットワーク抽象化モデ

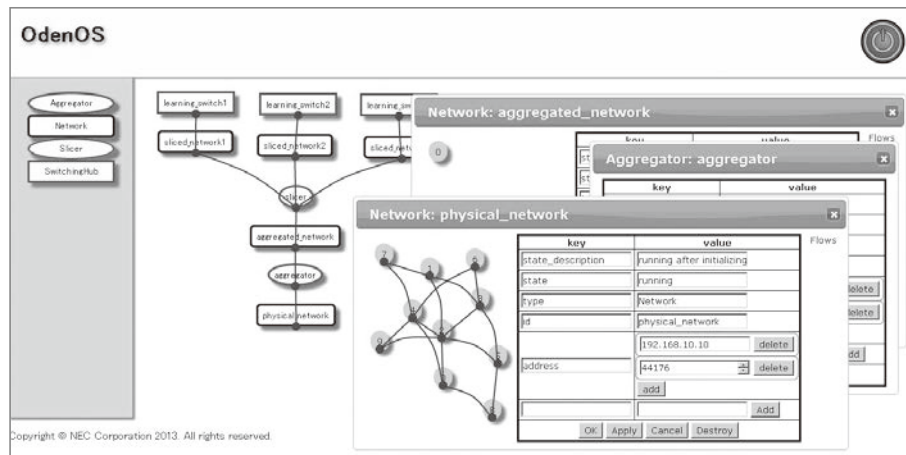


図5 GUI上でのSDNコントローラ作成とネットワーク可視化

ルを提案し、これに基づくSDN コントローラ構築基盤のプロトタイプを設計・実装しました。プロトタイプ実装を通して、Webブラウザ上のマウスクリックといった操作によってシンプルにSDNコントローラを作成できることを確認しました。

今後は、今回実装したプロトタイプをベースに実際のユースケースに適用可能な高度なコンポーネントを実装して実環境への投入を行うことで、コンポーネント群の充実化とネットワーク抽象化モデルやインタフェース仕様を洗練する予定です。

## 5. 謝辞

本研究の一部は、平成25年度総務省委託研究「変動する通信状況に適應する省エネなネットワーク制御基盤技術の研究開発」の成果です。

\* OpenFlowはOpen Networking Foundationの商標または登録商標です。

### 参考文献

- 1) Open Networking Foundation, <http://www.opennetworking.org/>
- 2) N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus," SIGCOMM Comput. Commun. Rev., vol.38, no.2. 2008.

### 執筆者プロフィール

小出 俊夫

情報・ナレッジ研究所  
主任

下西 英之

情報・ナレッジ研究所  
主任研究員

芦田 優太

情報・ナレッジ研究所

# 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](#)