

SDN を用いたクラシカルなネットワーク機器の 自動制御フレームワークに関する一検討

王 冠男[†]神崎 映光[‡]島根大学自然科学研究科[†]島根大学学術研究院理工学系[‡]

1. はじめに

SDN (Software Defined Network) はネットワーク機器におけるデータ転送等の制御をソフトウェアで実現可能とする技術である。SDN により、コントローラと呼ばれるソフトウェアを用いて、各機器におけるデータ転送を制御できるため、柔軟なネットワーク制御が可能となる。しかし、SDN によるネットワーク制御を行うためには、OpenFlow をはじめとした専用のプロトコルを用いる必要があり、このプロトコルに対応していない既存のクラシカルなネットワーク機器はそのまま利用できない。

本研究では、SDN に対応していないクラシカルなネットワーク機器を、SDN を用いて自動的に制御可能とするフレームワークの設計および実装を行う。提案フレームワークでは、SDN コントローラとクラシカルな機器とを仲介するソフトウェアアダプタを導入し、SDN コントローラによる制御情報を、クラシカルな機器の設定情報に変換する機能を提供する。これにより、クラシカルな機器の柔軟な制御を可能とする。

2. 関連研究

Csikor らは文献[1]において、サーバを中継機器として使用し、クラシカルな機器を SDN に対応させるアーキテクチャを提案している。この研究は、本研究と目的が一致しているが、このアーキテクチャでは、ネットワーク機器に流入した通信の大半がサーバに転送されるため、サーバの処理能力がボトルネックとなってしまう。一方、本研究では、クラシカルな機器において可能な限り通信を処理するため、サーバに相当するソフトウェアアダプタの負荷を軽減できる。

Shukla らは文献[2]において、既存のルーティングテーブルを SDN 機器用のルールに変換する機能を提案している。しかし、変換する情報がルーティングテーブルに限定されており、またクラシカルなプロトコルの情報を SDN 機器向

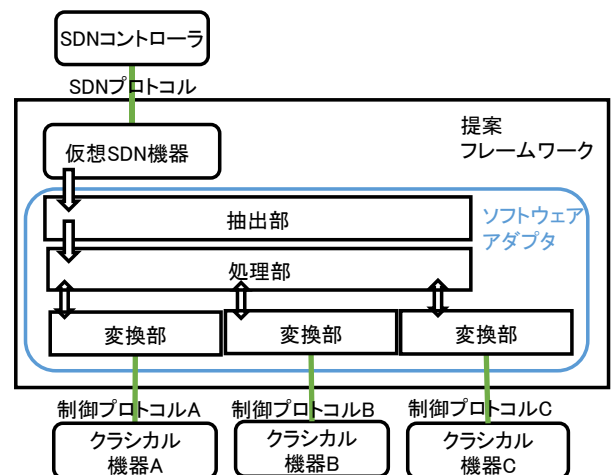


図1 ソフトウェア構成

けに変換する点が本研究と異なる。

3. 設計

提案フレームワークの構成を図1に示す。以下では、提案フレームワークにおける各機能について述べる。

3.1 仮想SDN機器

仮想SDN機器は、SDNに対応した機器を仮想的に再現したものであり、SDNコントローラとの間でSDNに基づく通信を行い、SDNにおける転送ルールであるフローテーブルを生成する。

3.2 フローテーブル抽出部

フローテーブル抽出部は、仮想SDN機器内において生成されたフローテーブルに関する情報を定期的に読み取り、フローテーブル処理部に渡す。

3.3 フローテーブル処理部

フローテーブル処理部は、フローテーブル抽出部から受け取った情報を解析し、変換部と連携しながら、クラシカルなネットワーク機器（以降、クラシカル機器と表記する）が対応可能な設定を変換部に渡す。また、クラシカル機器が対応不可能な設定については、アラートとして管理者に提示する。

3.4 変換部

変換部はクラシカル機器ごとに生成され、フ

A Study of Automatic Control Framework for Classical Network Devices using SDN

[†]WANG Guannan · Graduate School of Natural Science and Technology, Shimane University

[‡]KANZAKI Akimitsu · Institute of Science and Engineering, Academic Assembly, Shimane University

```

root@ovs-test4p:~# ovs-ofctl dump-flows brvlan
DPST_FLOW reply (OF1.3) (xid=0x2):
 cookie=0x0, duration=4.129s, table=0, n_packets=0, n_bytes=0, priority=100,in_port=enp0s3 actions=p
ush_vlan:0x8100,set_field:4098->vlan_vid,output:enp0s8

```

図2 仮想SDN機器におけるフローテーブル

ローテーブル処理部から受け取った設定に関する情報を、対応する機器が処理可能な設定情報に変換する。また、機器が対応可能な設定に関する情報を保持しており、対応不可能な設定についてはフローテーブル処理部に報告する。さらに、変換した設定情報を、機器に対応した仕様を用いてクラシカル機器に送信する。

4. 実装

3章で述べた設計に基づき、プロトタイプシステムの実装を行った。ここでは、クラシカルなネットワーク機器として、MikroTik社製のイーサネットルータ RB760iGS を用いた。この機器はOSとしてRouterOSが動作している。ここでは、このOSにおける転送ポート制御およびVLANの制御をSDNを用いて実現する機能を実装した。

SDNプロトコルはOpenFlow 1.3を用い、SDNコントローラとしてRyu[3]、仮想SDN機器としてOpen vSwitch[4]を用いた。また、フローテーブル抽出部には、Open vSwitchにて提供されているovsdb-toolを用いた。フローテーブル処理部および変換部はPython 3.7を用いて記述し、変換部ではRouteOS-apiパッケージを利用して変換処理を記述した。

以上のプロトタイプシステムにおいて、クラシカル機器のポート1で受信したフレームに対し、以下の2つの制御を行うルールをSDNで記述し、クラシカル機器の設定に反映させる動作検証を行った。

- ・VLANタグを付与
- ・ポート2に転送

上記のルールをRyuにて記述し、仮想SDN機器であるOpen vSwitchにおけるフローテーブルを生成した結果を図2に示す。図中のポートenp0s3およびenp0s8は、それぞれクラシカル機器のポート1および2に対応している。このフローテーブル内の情報をソフトウェアアダプタにおいて処理し、クラシカル機器に反映させた結果を図3に示す。この図は、RouterOS上で、転送およびVLANに関する設定を出力した結果であり、図中のether1およびether2が、それぞれポート1および2を表している。結果より、Open vSwitch上で規定された設定が、クラシカル機器においても正しく設定されていることが

```

[admin@MikroTik] /interface vian> pr
Flags: X - disabled, R - running
# NAME
[admin@MikroTik] /interface vian> /interface ethernet switch pr
Flags: I - invalid
# NAME      TYPE      MIRROR-SOURCE      MIRROR-TARGET
0 switch1   -         MediaTek-MT7621     none

```

(a) 設定反映前

```

[admin@MikroTik] /interface vian> /interface ethernet switch
[admin@MikroTik] /interface ethernet switch pr
Flags: I - invalid
# NAME      TYPE      MIRROR-SOURCE      MIRROR-TARGET
0 switch1   -         MediaTek-MT7621     ether1
[admin@MikroTik] /interface ethernet switch> /interface vian
[admin@MikroTik] /interface vian> pr
Flags: X - disabled, R - running
# NAME      MTU ARP      VLAN-ID INTERFACE
0 vian1     1500 enabled  1500 enabled  2 ether1
1 vian2     1500 enabled  1500 enabled  2 ether2

```

(b) 設定反映後

図3: クラシカル機器における設定確認結果

わかる。

5. おわりに

本研究では、クラシカルなネットワーク機器をSDNを用いて制御可能とするフレームワークを提案した。また、実際の機器を対象としたプロトコルを実装し、提案フレームワークが正常に動作することを確認した。

今後は、今回実装した機能以外の各種機能を変換できるよう、フローテーブル処理部の拡張を行う予定である。また、さまざまなクラシカル機器に対応した変換部を実装する予定である。

参考文献

- [1] Csikor,L., Szalay,M., Rétvári,G., Pongrácz,G., Pezaros,D. and Toka,L.: Transition to SDN is HARMLESS: Hybrid architecture for migrating legacy ethernet switches to SDN, IEEE/ACM Trans. Networking, vol.28, pp.275-288 (2020).
- [2] Open vSwitch: <https://www.openvswitch.org/>.
- [3] Ryu SDN Framework: <https://ryu-sdn.org/>.
- [4] Shukla,A., Shi,M. and Feldmann,A.: Automatic custom generation of topologies and configuration of routing protocols in SDN, SIGCOMM 2017 Posters and Demos, pp.3-5 (2017).