

# OpenFlow を用いたネットワーク学習教材の開発

石川 有彩<sup>1,a)</sup> 吉原 和明<sup>2</sup> 井口 信和<sup>3</sup> 渡辺 健次<sup>1</sup>

**概要:** 高等学校学習指導要領解説情報編には、「情報通信ネットワークの構成要素を理解する」ことなどが学習内容として示されている。しかし、実際の情報通信ネットワークの構成要素である、ルータや L2 スイッチ、Firewall といったネットワーク機器を教材として用意することは難しく、体験的な学習はあまり行われていない。そこで本研究では OpenFlow を用いて、Raspberry Pi にルータと L2 スイッチ、Firewall の機能を実装した。また、OpenFlow で動的経路制御を行うためのアプリケーションを開発して実装した。本教材では実装したネットワーク機能の設定を学習者が行うことで、ネットワークの構築を体験することができる。また、学習者が構築したネットワーク上を流れるパケットは LED テープの光で可視化した。これにより、学習者にとってわかりやすい教材となるよう工夫した。

**キーワード:** ネットワーク学習教材, OpenFlow, 可視化, Raspberry Pi

## Development of Network Learning Materials Using OpenFlow

**Abstract:** In the course of study for information of high schools in Japan, the contents include "understand the structure and components of the network". However, there are few learning materials such as routers, L2 switches, firewalls and so on. Hence it is difficult to learn experientially. Therefore, in this research, we implemented these functions on Raspberry Pi using OpenFlow. We also developed an application for dynamic routing with OpenFlow. In this learning materials, the learner can configure the network implemented with OpenFlow. In addition, we visualized the packet flow using LED tape in the network configured by learner.

**Keywords:** Learning Materials of Network, OpenFlow, Visualization, Raspberry Pi

### 1. はじめに

平成 30 年告示の高等学校学習指導要領解説情報編 [1] には、学習内容として「情報通信ネットワークの仕組みと構成要素を理解する」ことや、「目的や状況に応じて情報通信ネットワークの構成要素を選択するとともに、情報セキュリティを確保する方法について考える」ことが示されている。しかし、実際の情報通信ネットワークの構成要素である、ルータや L2 スイッチ、Firewall といったネットワーク機器は教材として用意することが難しく、体験的な学習はあまり行われていない。

情報通信ネットワークを体験的に学習するための教材開発を行っている研究として、村松らは、情報通信ネットワークの物理的構造の理解を促すアナロジーモデルとして電話網を位置付けて教材化し、その効果を検証している [2]。この研究において電話網をアナロジーモデルとして用いるには、電話網が学習者にとって馴染みの深いものである必要がある。しかし、現在はスマートフォンなどの携帯電話が普及しているため、学習者にとって電話網は実感の湧きやすいものでなくなっているという問題がある。

学習者に実感を湧かせる方法として、ネットワークを流れるデータを可視化する方法がある。荒井らは実際のパケットをモニタリングし、パケットのヘッダ情報や通信手順を可視化するツールを開発し、評価を行った [3]。この研究により可視化が情報通信ネットワークの概念の理解に有効であることが確認されたが、学習するデータの情報が文字表示で、直感的な理解が難しいという問題がある。

立岩ら是一台のコンピュータ上に仮想的にネットワーク

<sup>1</sup> 広島大学大学院  
Hiroshima University

<sup>2</sup> 福山大学  
Fukuyama University

<sup>3</sup> 近畿大学  
Kindai University

a) m196554@hiroshima-u.ac.jp

を構築し、通信シミュレーションが行えるシステムを開発した [4]。このシステムではパケットの通った経路を、パケットを表すアイコンのアニメーション表示によって示すことで、直感的でわかりやすくしている。しかし、一台のコンピュータ上に仮想ネットワークを構築するため、物理的な配線が行われないことや、可視化の手法がソフトウェアであり、実感が湧きづらいという問題がある。

ネットワーク機能をソフトウェアではなく LED で可視化する方法としては、吉原らの研究がある。吉原らは物理的可視化と物理的直接操作をコンセプトに、本物のネットワークの構築を通して IP アドレスの仕組みを学習するための教材を開発し、評価を行った [5]。吉原らの教材ではパケットが機器に届いた時に LED を点灯させることでネットワーク技術の可視化を行っている。

本研究では吉原らの研究を発展させ、IP アドレスの仕組みだけではなくネットワーク機器の仕組みについて学習することができる教材の開発を行った。そのために、OpenFlow を用いて Raspberry Pi にルータや L2 スイッチ、Firewall といったネットワーク機能を実装し、それらを用いて小規模ネットワークを構築することで体験的に学習が行えるような教材の開発を行った。

また、ネットワーク機器の仕組みを学習するにあたり、パケットがネットワーク機器を介して移動する様子を可視化するため、LED ではなく LED テープを用いた。

## 2. OpenFlow とは

### 2.1 OpenFlow スイッチと OpenFlow コントローラ

OpenFlow は、SDN (Software Defined Network) を実装するための技術のうちの一つで、ソフトウェアによりネットワークの制御機能を自由に設計・実装することができる [6]。OpenFlow は OpenFlow スイッチと呼ばれる転送部と、OpenFlow コントローラと呼ばれる制御部に分かれており、OpenFlow コントローラで OpenFlow スイッチの転送機能を自由に設計・実装することができる。

本研究では OpenFlow スイッチとして Open vSwitch [7] を、OpenFlow コントローラとして Ryu [8] を用いた。

OpenFlow スイッチは Datapath ID と呼ばれる 64bit の識別子を持っており、OpenFlow コントローラはこの識別子により OpenFlow スイッチを識別する [6]。そのため、Datapath ID は OpenFlow コントローラが制御するネットワーク中で一意である必要がある。

OpenFlow スイッチは、届いたパケットの転送先を「フローテーブル」と呼ばれるデータベースを参照して決める [9]。フローテーブルにはパケットを OpenFlow コントローラに転送するアクションも指定することができる。OpenFlow スイッチが OpenFlow コントローラにパケットを転送することを PacketIn と呼ぶ。

### 2.2 OpenFlow を用いるメリット

OpenFlow はネットワーク機能をソフトウェアで実装できる技術であり、ネットワークの機能に応じた特別なハードウェアを必要としない。そのため、教材として専用のハードウェアを複数用意する必要はなく、OpenFlow に対応した一つのコンピュータで様々なネットワーク機能を実装することができる。

実際のネットワーク機器を用いた学習では、教員が多数のネットワーク機器を学習者が操作できる状態に準備する必要があったり、学習者は個々のネットワーク機器の操作方法や技術の習得に多くの時間を割いてしまい、情報通信ネットワークの基礎となる原理の学習に十分時間をとれない傾向がある。

一方、OpenFlow は OpenFlow コントローラと呼ばれる制御部分と OpenFlow スイッチと呼ばれる転送部分に分離していることから、複数の OpenFlow スイッチに実装するネットワーク機能の切替えや設定などを OpenFlow コントローラで一元管理することができる。そのため、教員の準備時間の短縮にもなりうる。また、OpenFlow ではネットワークの設定方法もソフトウェアで定義することができるため、学習者がわかりやすい方法で設定を行えるように開発することができる。

これらの理由から、本研究では OpenFlow を用いてネットワークの学習教材の開発を行った。

## 3. 教材の開発

### 3.1 教材の概要

本教材では、OpenFlow コントローラで OpenFlow スイッチの転送機能を制御することで、ルータ、L2 スイッチ、Firewall、動的経路制御といったネットワーク機能を Raspberry Pi に実装した。

本教材では実装したネットワーク機能を小型キャラクター LCD を用いて表示し、ネットワーク上を流れるパケットを LED テープの光で可視化した。これにより、学習者にとって理解しやすい教材となるように工夫した。

本教材でパケットを可視化した様子を図 1 に示す。

### 3.2 システムの構成

本研究の開発環境を表 1 に示す。

OpenFlow コントローラとして用いた Ryu には、ネットワーク機能を実装するためのアプリケーションがいくつか用意されており、本教材ではルータと L2 スイッチ、Firewall の実装を、提供されたアプリケーションを修正して行っている。

本教材の機器の配置図を図 2 に示す。

本研究では Raspberry Pi3 Model B V1.2 を 4 機用いて教材の開発を行った。各 Raspberry Pi は無線で通信を行う。また、3 機の Raspberry Pi と 2 機のホストは有線 LAN

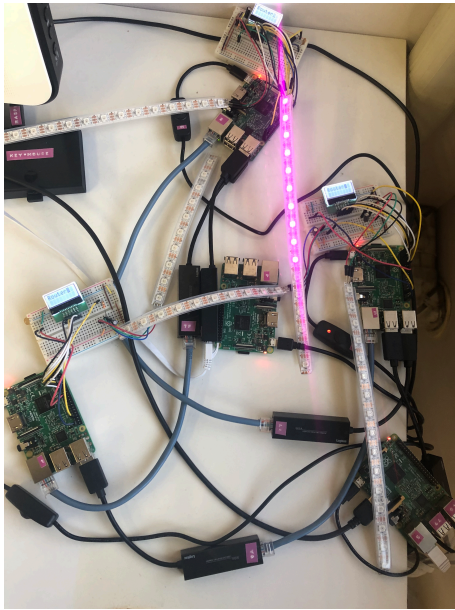


図 1 パケットを可視化した様子  
Fig. 1 Visualization of packet flow

表 1 開発環境

Table 1 Development Environment

種類	詳細
コンピュータ	Raspberry Pi 3 Model B V1.2
OS	Raspbian Buster
開発言語	Python3.7
パケットを可視化する装置	フルカラーシリアル LED テープ
機能を表示する装置	I2C 小型キャラクタ LCD
OpenFlow コントローラ	Ryu
OpenFlow スイッチ	Open vSwitch

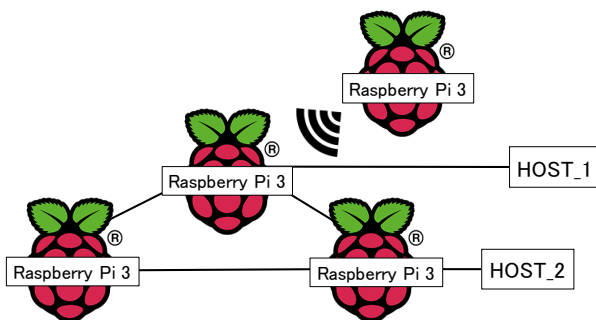


図 2 機器の配置図  
Fig. 2 Equipment Layout

ケーブルで接続されている。本研究では、無線の通信機能を OpenFlow による制御などのシステムの管理に用い、有線 LAN で教材のネットワークを構築する。本教材では、各 LAN ケーブルに対応する LED テープを用いてパケットを LED の点灯により可視化する。

Raspberry Pi はデフォルトで 1 つの LAN ポートを搭載している。本研究では有線接続できるポートを増設するために、1 つの Raspberry Pi につき最大 2 つの USB LAN アダプタを接続した。これにより、1 つの Raspberry Pi が持

つ NIC は eth0, eth1, eth2, wlan0 (無線用) となる。

### 3.3 本教材で設定できる項目

OpenFlow コントローラとして用いた Ryu には Web サーバの機能が搭載されており、HTTP リクエストを送信することでネットワークの設定を追加したり、設定した情報を取得したりすることができる。本教材では Web のアーキテクチャスタイルである REST API を作成することで、設定機能を実装した [10]。本教材で HTTP リクエストを送信する方法は、curl コマンドを実行する方法と設定プログラムを編集して実行する方法がある。

設定できる項目を表 2 に示す。設定は HTTP メソッドを用いて、ディクショナリ型で送信する。ディクショナリの値には設定するデータが格納される。

ルータの演習では 3 機の OpenFlow スイッチの各インターフェイスの IP アドレスと、ゲートウェイの設定を行う。L2 スイッチの演習では 1 機の OpenFlow スイッチの MAC アドレステーブルの作成を行う。Firewall の演習では 1 機の OpenFlow スイッチの Firewall の有効化と、ルールの設定を行う。Ryu で提供されている Firewall のアプリケーションでは、ルールとして宛先 IP アドレスやプロトコルを指定することができる。しかし、本教材では外部ネットワークと内部ネットワークの間に Firewall を実装し、「どんなパケットでも通過を許可する」場合と、「内部ネットワークから送信されたパケットのみ通過を許可する」場合を、演習内容として想定している。動的経路制御の演習では 3 機の OpenFlow スイッチの IP アドレスとポート番号の設定を行う。

### 3.4 LED テープの制御

#### 3.4.1 LED テープを制御する機器

本教材では図 2 のホストから送信された ICMP パケットを、フルカラーシリアル LED テープの光で可視化する。可視化する ICMP パケットは有線 LAN ケーブルによって転送されるものである。そこで、本研究では 1 本の LAN ケーブルにつき 1 本の LED テープを用意した。

フルカラーシリアル LED テープは、Raspberry Pi から出力される PWM 信号を用いて制御する。Raspberry Pi が出力できる PWM 信号のチャンネルは 2 チャンネルあり、BCM18 と BCM12 が 0 チャンネル、BCM13 と BCM19 が 1 チャンネルに対応している。そのため、一機の Raspberry Pi が制御できる LED テープは 2 本までとなる。

第 3.2 節で示したように、本研究で一機の Raspberry Pi が持つイーサネットのインターフェイスは最大 3 つである。つまり、1 つの Raspberry Pi に接続される LAN ケーブルは 3 本あるが、1 つの Raspberry Pi で制御できる LED テープは 2 本まで、という問題がある。

この問題を解決するために、Raspberry Pi の接続方法を

表 2 ネットワーク設定の項目  
Table 2 Network Setting Items

機能	設定項目	キー	メソッド
ルータ	IP アドレス	"address"	POST
	ゲートウェイ	"gateway"	
L2 スイッチ	MAC アドレス・ポート番号	"mac"・"port"	PUT
Firewall	有効化	—	PUT
	ルール	"nw_src"	POST
動的経路制御	IP アドレス・ポート番号	"RouterAddr"・"port"	POST

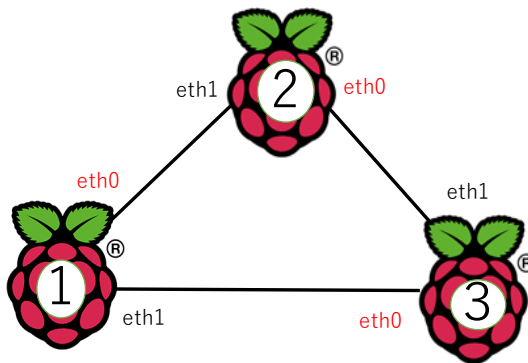


図 3 Raspberry Pi の接続方法  
Fig. 3 Connection of Raspberry Pis

図 3 のようにした．eth0 と接続する先は eth1 になるようにし，eth0 のインターフェイスを持った Raspberry Pi が，その接続の LED テープを制御する．図 3 であれば，①と②の間を移動するパケットは①が制御する LED テープで可視化し，①と③の間を移動するパケットは③が制御する LED テープで可視化する．

なお，eth2 に接続された LAN ケーブルを移動するパケットは，自機で制御する．

### 3.4.2 メッセージによる LED テープの制御

フルカラーシリアル LED テープは，光の色や光るパターンなどをプログラムで自由に制御することができる．本教材では，ネットワーク上をパケットが流れているような表現にするため，LED テープの端から端へ，光が移動するような動きにした．また，光の色はネットワークアドレスに基づいて決定する．これにより，例えばパケットがルータを介して別ネットワークへ移動すると LED テープの光の色は変化するが，L2 スイッチを介した場合は色が変わらない，といった様子も確認することができる．

本研究では ICMP パケットのキャプチャを OpenFlow コントローラで行っている．OpenFlow コントローラはキャプチャしたパケットの解析結果に基づいてメッセージを生成し，UDP 通信で LED テープを制御する Raspberry Pi に送信する．

LED テープを制御する Raspberry Pi は，そのメッセージの情報に基づいて LED テープの光の色，光の向き，光るタイミングを決定して光らせる．

### 3.5 パケットキャプチャ

本研究では OpenFlow スイッチに届いた ICMP パケットを OpenFlow コントローラへ PacketIn し，OpenFlow コントローラがそのパケットのヘッダ情報などを解析することでパケットキャプチャを行う．また，OpenFlow コントローラは解析した結果をメッセージにし，LED テープを制御している Raspberry Pi に UDP で送信する．

パケットキャプチャは次の手順で実装した．

- (1) ICMP パケットを OpenFlow コントローラに PacketIn させるフローエントリの追加
- (2) OpenFlow コントローラに PacketIn した ICMP パケットのヘッダ情報の解析と，LED テープを制御する Raspberry Pi へメッセージを送信

OpenFlow コントローラに PacketIn したパケットは，Ryu で提供されるモジュールを用いてヘッダ情報の解析を行うことができる．それにより，次のような情報を得ることができる．

- PacketIn を送った OpenFlow スイッチ（つまり，ICMP パケットが届いた OpenFlow スイッチ）の Datapath ID
- 送信元 MAC アドレス
- 宛先 MAC アドレス
- 送信元 IP アドレス
- 宛先 IP アドレス
- ICMP タイプ

これらの情報から，どの OpenFlow スイッチに，どこからどこへ向かう ICMP エコー要求（エコー応答）パケットが届いたか，という情報を得ることができ，その情報を LED テープを制御する Raspberry Pi へ送信することで，ICMP パケットが移動する様子を可視化することができる．

### 3.6 動的経路制御のアプリケーション開発

Ryu には小規模ネットワークにおいて動的経路制御を行うアプリケーションが提供されていない．そこで，本研究では小規模ネットワークにおける代表的なルーティングプロトコルの一つである RIP[11] に倣って，動的経路制御を行うアプリケーションを Ryu で開発した．

動的経路制御の学習は，OpenFlow コントローラを 1 つ，

OpenFlow スイッチを3つ用意して行う。

### 3.6.1 実装した機能

動的経路制御のアプリケーションでは、以下のような機能を Ryu で実装した。

- 学習者が IP アドレスとポート番号を設定する機能
- 設定された情報から経路表を自動的に作成する機能
- LAN ケーブルが引き抜かれるなど、物理的な障害が OpenFlow スイッチに発生した時、該当する経路を経路表から削除する機能

### 3.6.2 イベントハンドラ

Ryu でアプリケーション開発を行う際に重要になるのが、イベントハンドラの用意である。Ryu は OpenFlow スイッチから受け取ったメッセージをイベントとして処理する。OpenFlow コントローラのアプリケーションでイベントハンドラを実装することで、OpenFlow スイッチからメッセージを受け取った時の処理を記述することができる。

本研究で使用したイベントハンドラと、行った処理を以下に示す。switch\_features\_handler にある Table-miss フローエントリとは、OpenFlow スイッチに届いたパケットがマッチするフローエントリが他に存在しない場合にマッチするフローエントリである。

- switch\_features\_handler  
OpenFlow スイッチと OpenFlow コントローラが接続した時に一度だけ発生する、「SwitchFeatures イベント」が発生したときに実行される。Table-miss フローエントリの追加に用いた。
- state\_change\_handler  
SwitchFeatures イベントが発生した後、OpenFlow スイッチのポートの MAC アドレス情報などを登録する、「StateChange イベント」が発生したときに実行される。各ルータのポート情報をアプリケーションのデータに保存するために用いた。
- packet\_in\_handler  
OpenFlow スイッチに届いたパケットが OpenFlow コントローラに転送された時に発生する、「PacketIn イベント」が発生したときに実行される。PacketIn したパケットが ARP リクエストであれば ARP リプライの返送を行い、UDP パケットであれば経路表の更新とネットワーク情報の再配布、ICMP パケットであれば LED テープを制御する Raspberry Pi へメッセージを送信する。
- port\_state\_change\_handler  
OpenFlow スイッチのポートから LAN ケーブルが物理的に引き抜かれた時に発生する、「StateChange イベント」が発生したときに実行される。経路表から該当する経路を削除し、迂回路を選択させる処理を行う。

### 3.6.3 REST 実装

本研究で開発した動的経路制御のアプリケーションで

```
routing_tbl = {1:{"192.168.1.0":{"0.0.0.0":["1", 1]},  
                "192.168.3.0":{"0.0.0.0":["2", 1]}},  
              2:{"192.168.2.0":{"0.0.0.0":["1", 1]},  
                "192.168.1.0":{"0.0.0.0":["2", 1]},  
                "192.168.4.0":{"0.0.0.0":["3", 1]}},  
              3:{"192.168.3.0":{"0.0.0.0":["1", 1]},  
                "192.168.2.0":{"0.0.0.0":["2", 1]},  
                "192.168.5.0":{"0.0.0.0":["3", 1]}}
```

図 4 IP アドレス設定後の経路表

Fig. 4 Routing Table after IP address setting

は、学習者が OpenFlow スイッチの IP アドレスとポート番号を設定できるように、REST API の作成を行なった。

これにより、指定した URL に POST メソッドで HTTP リクエストが送信された時に、IP アドレスとポートのデータを受け取ってアプリケーションの経路表の作成を行うことができる。また、GET メソッドで HTTP リクエストが送信された場合は、アプリケーションが保持している経路表を学習者に返送することができる。

### 3.6.4 経路表の作成

このアプリケーションでは、1つの OpenFlow コントローラで全ての OpenFlow スイッチの経路表を管理する。

用意した経路表は、以下の要素からなる。転送先のインターフェイスは NIC 名ではなく、OpenFlow スイッチに登録した番号で管理する。OpenFlow スイッチには eth0 に 1, eth1 に 2, eth2 に 3 というポート番号を割り当てて設定している。

- (1) Datapath ID
- (2) 宛先ネットワーク
- (3) ゲートウェイ
- (4) インターフェイス (ポート番号)
- (5) メトリック

本研究ではこれらを以下のような構造で管理している。

```
routing_tbl = { (1) : { (2) : { (3) : [ (4), (5) ] } } }
```

本研究では割り当てる IP アドレスのプレフィクス値を 24 に固定しているため、学習者が全てのスイッチの IP アドレスとポートを設定し終わると、リンク上のネットワークアドレスを算出することができる。それにより、図 4 のような経路表を生成することができる。

次に、自機の経路情報をメッセージに格納し、UDP で隣接する OpenFlow スイッチに配布する。その情報を受け取った OpenFlow スイッチは、受け取った経路情報が自機の経路表に存在するか確認し、存在しなければ受け取った経路情報を、メトリックを増やして自機の経路表に追加する。また、受け取った経路情報が自機の持つ経路表にあった場合でも、そのゲートウェイがメッセージの送信元 IP アドレスと一致しない場合は経路表を更新する。

この UDP メッセージは宛先ネットワークとメトリックで構成される。例えば、スイッチ①が自機のリンク上のネットワークである「192.168.1.0」のネットワーク情報を

UDP で送信する際、ペイロードは、ネットワークを表す「192.168.1.0」とメトリックの「1」を合わせて「192.168.1.01」となる。また、そのメッセージをスイッチ②が受け取った場合、スイッチ②の経路表には「192.168.1.0」ネットワークが存在するが、そのゲートウェイは「0.0.0.0」のみである。その場合は受け取ったメトリックに2を足して3にし、UDP メッセージの送信元 IP アドレスである「192.168.1.5」を「192.168.1.0」ネットワークのゲートウェイとして新たに登録する。

このように経路情報を交換することで、自動で経路表を完成させることができる。

### 3.6.5 障害発生時の処理

LAN ケーブルが OpenFlow スイッチから引き抜かれると、PortStateChange イベントが発生する。PortStateChange イベントでは、障害が発生した OpenFlow スイッチの Datapath ID とケーブルが引き抜かれたポートの番号が渡される。それにより、該当する Datapath ID の経路表を参照し、障害が発生したポートの番号が、インターフェイスのポート番号となっている経路を削除することができる。

また、OpenFlow コントローラから、OpenFlow スイッチのフローテーブルを参照し、out\_port（転送先のポート）が障害が発生したポートになっているフローエントリを削除することができる。それにより、パケットは迂回路を選択して移動することができる。

## 4. 本教材で学習できる内容

### 4.1 学習範囲の拡張

本教材は吉原ら [5] が開発した教材を、ネットワーク機器の学習を行えるように発展させたものである。吉原らはボタンやダイヤルを手で直接操作することで自機の IP アドレスの設定を行ったり、ping の送信を行ったりする教材を開発した。吉原らの教材を用いて学習できる内容は、IP アドレスの構造とその一意性についてであったが、本教材と組み合わせることにより、ネットワークを構成する要素についてより広く学習することができる。次節から具体的に述べる。

### 4.2 ルータの演習

ルータの演習では、ルータの IP アドレスやゲートウェイを設定して小規模ネットワークの構築を行うことで、情報通信ネットワークにおける IP アドレスの役割や、経路制御の仕組みなどを学習することができる。

### 4.3 L2 スイッチの演習

L2 スイッチの演習では、MAC アドレステーブルの作成や、Flooding といった現象を LED テープの光で確認することで、情報通信ネットワークにおける MAC アドレスの役割や、L2 スイッチが MAC アドレステーブルを参照して

パケットを転送する仕組みについて学習することができる。

### 4.4 Firewall の演習

Firewall の演習では、内部ネットワークと外部ネットワークに見立てたネットワークの間に位置する Firewall のパケットフィルタリングのルールを設定を行うことで、内部ネットワークと外部ネットワークの安全性の違いについて扱ったり、外部ネットワークからの悪意ある通信を Firewall のルールの設定で防いだりする学習活動を行うことができる。

### 4.5 動的経路制御の演習

動的経路制御の演習では、ルータの IP アドレスとポートの情報を設定するだけでルーティングが正常に行われる様子を LED テープの光で確認したり、ルータを接続する LAN ケーブルを学習者自身が引き抜いて、パケットが迂回路を選択する様子を確認する学習活動を行うことができる。これらにより、経路制御をプログラムで行う技術や、それにより障害が発生した時に素早く迂回路を選択する機能、効率的な通信を行う機能について学習することができる。この演習ではルータの演習で行う、ゲートウェイを手動で設定する演習と比較することで、2つの経路制御の方法の違いやメリット・デメリットを考えさせ、経路制御についての理解を深めることができる。

## 5. おわりに

### 5.1 結論

本研究では、OpenFlow を用いて Raspberry Pi にルータや L2 スイッチ、Firewall といったネットワークの機能を実装した。また、動的経路制御を行うアプリケーションを Ryu で開発し、OpenFlow スイッチの経路表の作成を自動で行ったり、障害が発生した時に迂回路を選択する機能を実装した。

OpenFlow で実装したネットワーク機能の IP アドレスやゲートウェイ、MAC アドレステーブル、パケットフィルタリングのルールなどの設定は学習者自身で行うことができる。これらの小規模ネットワークの構築を行う演習を通して、情報通信ネットワークの仕組みや構成要素、また情報セキュリティを確保する方法について体験的に学習することができる。

加えて、学習者が構築したネットワーク上を流れる ICMP パケットを、OpenFlow コントローラでパケットキャプチャし、その結果を LED テープの光の流れで可視化することにより、学習者にとって実装したネットワーク機能の原理が理解しやすいような教材となるよう工夫した。

本教材を用いることで、実際のネットワーク機器を教材として用意せずとも、学習者は情報通信ネットワークの構成要素であるルータや L2 スイッチ、Firewall、動的経路制

御などのネットワーク機能を、体験的に学習することができる。

## 5.2 今後の課題

今後の展望を4つ述べる。

1つ目は、教材のユーザインターフェイスの開発についてである。本教材はネットワークの設定を学習者が行うことができるが、その方法は curl コマンドの実行やプログラムの実行など、高校生にとっては少し難易度の高いものとなっている。そのため、今後は学習者が操作しやすいようなユーザインターフェイスを開発する必要がある。

2つ目は、LED テープを制御するプログラムの汎用性についてである。本研究では LED テープを制御するプログラムを演習ごとに別々で作成しているため、複数の機能を持ったネットワーク、例えばルータと L2 スイッチを組み合わせたネットワーク構成などを作って学習を行うことができない。しかし、実際のネットワークはそのような複数のネットワーク機能で構成されていることが一般的である。そのため、今後は LED テープを制御するプログラムを汎用的なものに改善し、複数のネットワーク機能を組み合わせた演習が行えるようにする必要がある。

3つ目は、機器の物理的な配線の複雑さである。本教材では、Raspberry Pi に接続されているケーブル及びジャンパーワイヤの数が多く、教材の本質である LAN ケーブルに注目させることが難しくなっている。そのため、LED テープや I2C 小型キャラクタ LCD をユニバーサル基盤に半田付けし、ジャンパーワイヤの数を減らす工夫を行う必要がある。また電源ケーブルや周辺機器を接続するケーブルなども、学習者の目に触れにくいような配置にする必要がある。

4つ目は、教材の有効性の検証である。本研究では教材の開発のみを行ったため、有効性を調査できていない。今後は本教材を用いて高等学校でネットワークに関する授業実践を行い、効果を検証する必要がある。

## 謝辞

本研究は JSPS 科研費 18K11570 の助成を受けたものです。

## 参考文献

- [1] 文部科学省 (2018): 「高等学校学習指導要領解説情報編」
- [2] 村松浩幸・本多満正・坂口謙一・鈴木善晴・川俣純・渡辺浩康 (2004): 「中学校技術科での情報通信ネットワークの学習における電話網の教材化」, 日本教育工学会論文誌, 28, pp.237-240
- [3] 荒井正之・田村尚也・渡辺博芳・小木曾千秋・武井恵雄 (2003): 「TCP/IP プロトコル学習ツールの開発と評価」, 情報処理学会論文誌, vol.44, no.12, pp.3242-3251
- [4] 立岩祐一郎・安田孝美・横井茂樹 (2005): 「仮想環境ソフトウェアに基づくネットワーク処理可視化教育システムの開発」, 社団法人情報処理学会研究報告コンピュータと

- 教育, 2005, 104, pp. 7- 14
- [5] 吉原和明・井口信和・渡辺健次 (2018): 「物理的可視化と物理的直接操作による IP アドレスの仕組みを学習するための教材の開発と評価」, 日本産業技術教育学会誌, 第 60 巻, 第 2 号, pp.73-80
- [6] JPNIC (2012): 「インターネット 10 分講座」, 『JPNIC ニュースレター』, No.52, pp.38-41
- [7] Linux Foundation: Open vSwitch  
<https://www.openvswitch.org> (参照 2021-01-18).
- [8] NTT: Ryubook 1.0 ドキュメント  
<https://osrg.github.io/ryu-book/ja/html/index.html> (参照 2021-01-18).
- [9] 高宮安仁・鈴木一哉 (2013): 「クラウド時代のネットワーク技術 OpenFlow 実践入門」
- [10] 山本陽平 (2010): Web を支える技術 HTTP, URI, HTML, そして REST
- [11] RFC 1058  
<https://tools.ietf.org/html/rfc1058> (参照 2021-01-25)