

# OpenFlow を用いたネットワーク学習教材の開発

石川有彩<sup>\*1</sup>, 吉原和明<sup>\*2</sup>, 井口信和<sup>\*3</sup>, 渡辺健次<sup>\*4</sup>

<sup>\*1</sup> 広島大学大学院教育学研究科, <sup>\*2</sup> 福山大学工学部,

<sup>\*3</sup> 近畿大学理工学部, <sup>\*4</sup> 広島大学大学院人間社会科学研究科

## Development of Learning materials for network by using OpenFlow

Arisa Ishikawa<sup>\*1</sup>, Kazuaki Yoshihara<sup>\*2</sup>, Nobukazu Iguchi<sup>\*3</sup>, Kenzi Watanabe<sup>\*4</sup>

<sup>\*1</sup> Graduate School of Education, Hiroshima University

<sup>\*2</sup> Faculty of Engineering, Fukuyama University

<sup>\*3</sup> Faculty of Science and Engineering, Kindai University

<sup>\*4</sup> Graduate School of Humanities and Social Sciences, Hiroshima University

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, and firewalls. Hence it is difficult to learn experientially. Therefore, in this research, we implemented functions such as routers. on Raspberry Pi using OpenFlow and developed learning materials that students can learn the components of network experientially.

キーワード: 情報教育, 教材開発, ネットワーク, 体験的学習, Raspberry Pi, OpenFlow

### 1. はじめに

高等学校学習指導要領解説情報編<sup>1</sup>には, 学習内容として「情報通信ネットワークの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, 実世界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 などの教材はなく, 体験的な学習を行うことが難しい。

そこで本研究では, OpenFlow を用いて Raspberry Pi にルータ等を実装し, 生徒が情報通信ネットワークの構成要素を体験的に学習できる教材の開発を行った。

### 2. OpenFlow

#### 2.1 OpenFlow とは

OpenFlow は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技術であ

る。OpenFlow は「OpenFlow スイッチ (以下, スイッチ) と呼ばれる転送部と, 「OpenFlow コントローラ (以下, コントローラ) と呼ばれる制御部に分かれています。コントローラで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。

つまり, コントローラからスイッチに指示を与えることでスイッチにルータの機能を実装したり, L2 スイッチの機能を実装したりすることができる。

#### 2.2 OpenFlow の仕組み

スイッチは, 届いたパケットの転送先を, スイッチ内の「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<sup>2</sup>。フローテーブルには「このようなパケットが届いたら, ポート x 番に転送する」というルールがいくつか記録されており, このルールは「フローエントリ」と呼ばれる。フローエントリは「マッチングルール」, 「アクション」, 「統計情報」の 3 つの要素から成る。

スイッチがフローテーブルを参照した際、届いたパケットにマッチするフローエントリが登録されていない場合もある。その際、スイッチはどのようにパケットを処理すれば良いか、コントローラに指示を仰ぐ。この時用いられるのが、「Packet In メッセージ」というメッセージである。

コントローラでは実装する機能ごとにアプリケーションが用意されており、Packet In メッセージを受け取った場合どのような指示をスイッチに返すかといったことも、アプリケーションのプログラム中に記されている。

### 3. 教材の開発

#### 3.1 システム構成

本研究では Raspberry Pi 3 Model B V1.2 を4つ用いた。それぞれに OpenFlow のソフトウェアをインストールしており、各 Raspberry Pi はコントローラやスイッチ、ホストとして振舞うことができる。

Raspberry Pi はバージョン3以降、無線 WiFi モジュールを搭載しており、本研究では4つの Raspberry Pi すべてで無線通信が行えるようにした。コントローラとスイッチの間の通信を無線で行う。

システム構成図を図1に示す。

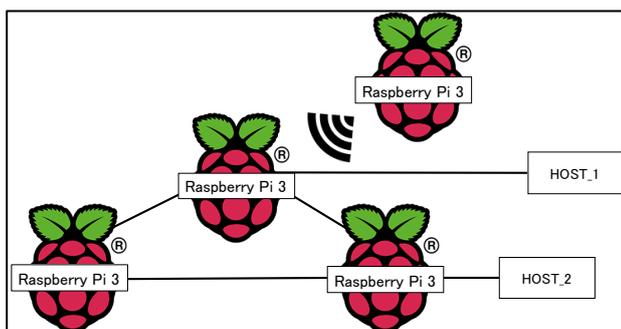


図 1 システム構成図

3つの Raspberry Pi と2つのホストは有線ネットワークで構成している。有線で接続されている箇所にはフルカラーシリアル LED テープを設置しており、有線ケーブルを流れるパケットを LED テープの光で可視化する。

また、スイッチが実装しているネットワークの機能を判別するために、各スイッチには小型キャラクター LCD を搭載した。

開発環境を表1に示す。OpenFlow スイッチとして

用いた Open vSwitch では、「ブリッジ」という仮想スイッチを作成して用いた。

表 1 開発環境

種類	詳細
コンピュータ	Raspberry Pi 3 Model B V1.2
OS	Raspbian Buster
プログラミング言語	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
スイッチに実装したネットワーク機能を表示する装置	I2C 小型キャラクター LCD
OpenFlow コントローラ	Ryu SDN framework
OpenFlow スイッチ	Open vSwitch

#### 3.2 設定項目

コントローラとして用いた Ryu SDN framework には Web サーバの機能が搭載されており、外部から HTTP メソッドを用いてネットワークの設定を呼び出したり、設定を変更したりすることができる。

設定できる項目を表2に示す。

表 2 設定項目

実装する機能	設定項目
ルータ	ルータの IP アドレス ルータのゲートウェイ
L2 スイッチ	ポート番号・MAC アドレス
Firewall	ルール
動的経路制御	ルータの IP アドレス

#### 3.3 LED テープの制御

Raspberry Pi 3 には 40 本の汎用入出力ポートが搭載されており LED テープや LCD などのハードウェアをソフトウェアで制御することができる。

また、LED テープを光らせる際に用いる PWM 信号は 2 チャンネルで出力できるため、1 台の Raspberry Pi で 2 本の LED テープを制御できる。

本教材ではパケットが移動する様子を、LED テープの光の流れで表現している。その際、制御している

Raspberry Pi から光が出ていくようにするか、逆に光が向かってくるようにするか、という光らせ方を決定する必要がある。

また、複数の LED テープを光らせる場合、光らせる順番を決定し、タイミングの制御をする必要がある。

### 3.4 パケットキャプチャ

有線ケーブルを流れるパケットを LED テープの光で可視化するには、スイッチ間を移動するパケットをキャプチャする必要がある。本教材では、OpenFlow を用いてパケットキャプチャを実装した。

まず、コントローラは起動時に、「ICMP パケットが届いたら、Packet In する」というフローエントリをスイッチに追加させる。

次に、Packet In してきた ICMP パケットについて、送信元 IP アドレスや宛先 IP アドレス、またどのポートから届いたか、といった情報を解析する。

次に、その情報を LED テープを制御しているスイッチに UDP メッセージを用いて送信する。

メッセージを受け取ったスイッチは、その情報をもとに光らせる LED テープと、光らせ方、光らせるタイミングを決定する。

### 3.5 ルータ

ルータの学習では、図 2 のようにコントローラを 1 つ、スイッチを 3 つ、ホストを 2 つ用いる。

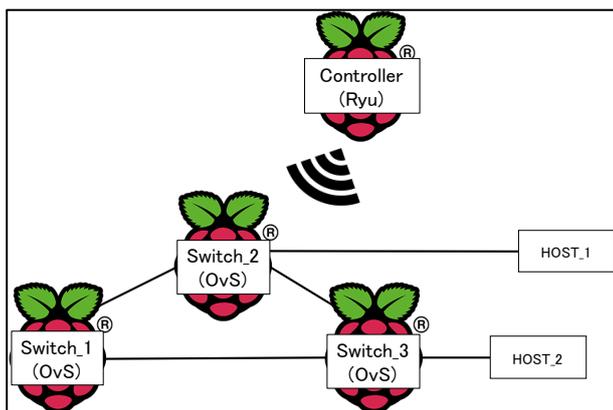


図 2 ルータの学習の構成

コントローラでは、Ryu に元々用意されている、「rest\_router.py」というアプリケーションにパケットキャプチャ用の修正を加えて実行する。

3 つあるスイッチの各インターフェイスに IP アドレスを割り振り、ゲートウェイの設定をするとネット

ワークが構築される。

### 3.6 L2 スイッチ

L2 スイッチの学習では、図 3 のようにコントローラを 1 つ、スイッチを 1 つ、ホストを 3 つ用いる。

コントローラでは、Ryu に元々用意されている、「simple\_switch\_rest\_13.py」というアプリケーションにパケットキャプチャ用の修正を加えて実行する。

この学習ではスイッチのポート番号と、ポートの先に接続されているホストの MAC アドレスを対応づけて設定することで、L2 スイッチの仕組みが学習できる。

また、ポート番号と MAC アドレスの設定をしない場合、自動で MAC アドレステーブルを作成する。その際に flooding という全ポートへパケットを転送する現象が起きるが、本教材ではその様子も LED テープの光で可視化をした。

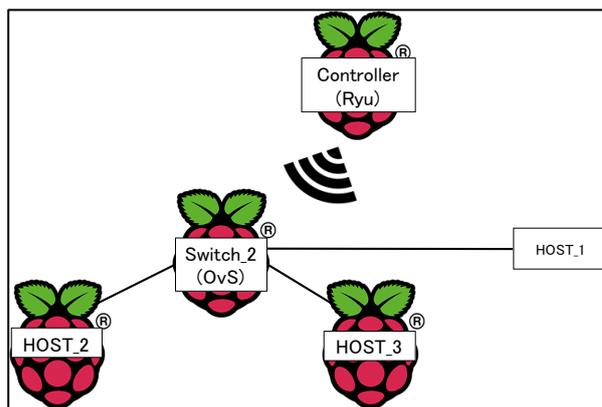


図 3 L2 スイッチの学習の構成

### 3.7 Firewall

Firewall の学習では、図 4 のようにコントローラを 2 つ、スイッチを 2 つ、ホストを 2 つ用いる。

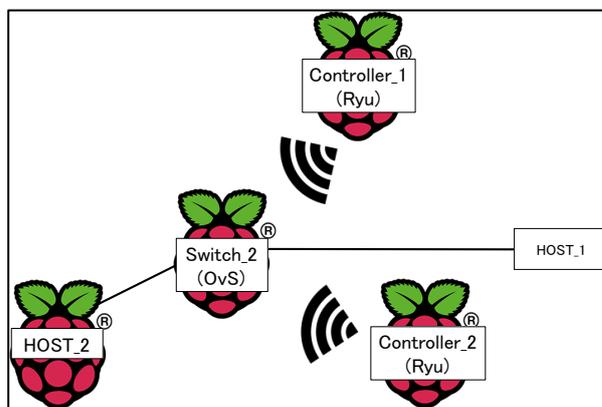


図 4 Firewall の学習の構成

図 4 では物理的なスイッチが 1 つになっているが、

Open vSwitch の設定でブリッジを2つ用意した。

一つ目のコントローラでは、Ryu に元々用意されている「rest\_firewall.py」というアプリケーションにパケットキャプチャ用の修正を加えて実行する。

また、この「rest\_firewall.py」というアプリケーションだけでは、ルータの機能を実装することができない。そこで、二つ目のコントローラで「rest\_router.py」を実行し、ルータと Firewall の両方の機能を実装した。

また、Firewall の学習では、外部ネットワークと内部ネットワークを用意する。本教材では、図 4 の HOST\_1 側が内部ネットワーク、HOST\_2 側が外部ネットワークとした。そして、外部ネットワークからのパケットがルールによって通過したり、遮断されたりする様子を LED テープの光で可視化した。

### 3.8 動的経路制御

動的経路制御の学習では、コントローラを1つ、スイッチを3つ、ホストを2つ用いた。構成は図2と同様である。

#### 3.8.1 動的経路制御のプロトコル

動的経路制御のプロトコルには、大きく分けて IGP と EGP という2種類のプロトコルがある。IGP は小規模なネットワークで用いられ、EGP は大規模なネットワークで用いられる。

Ryu には元々 BGP という、EGP のプロトコルを実装するためのアプリケーションが用意されているが、本研究の実験環境は非常に小規模であるため、このプロトコルは適していない。また、オープンソースで公開されているものも、本研究の実験環境に合うものは無い。よって、本研究では実験環境にあった動的経路制御のアプリケーションを、Ryu で開発して使用する。

#### 3.8.2 開発した動的経路制御のプロトコル

開発した動的経路制御のプロトコルは RIP を参考にした。

このプロトコルでは、経由するルータの数をメトリックとして管理し、メトリックの少ない経路を選択してパケットを転送する。

また、「自分のリンクに直接繋がっているネットワーク」のネットワークアドレスとメトリックの情報を含んだメッセージを UDP で隣接するルータに送信する。このメッセージを受け取ったルータは、そのネットワ

ークアドレスと UDP の送信元 IP アドレス、メトリックを元に経路表を更新する。

次に、更新したネットワークの情報を、メトリックを増やして隣接するルータに送信する。これにより、経路表を動的に作成することができる。

また、ネットワーク上のケーブルが抜かれた場合は、障害があったインターフェイスへパケットを転送しないように、経路表から経路を削除する。

#### 3.8.3 OpenFlow で ARP 処理

OpenFlow スイッチとして制御している Raspberry Pi では、ARP の処理を OS に任せることができない。そのため、コントローラでパケットを生成し、ARP リクエストや ARP リプライをスイッチに送信させるための命令を出す必要がある。この時用いるのが「Packet Out メッセージ」である。

ARP パケットを生成する際には、ethernet ヘッダの情報と arp ヘッダの情報が必要である。それらの中で、特に必要な情報を表3に示す<sup>3</sup>。

表 3 ARP パケットの生成に主に必要な情報

Attribute	説明
opcode	リクエストなら 1, リプライなら 2
src_mac	送信元 MAC アドレス
src_ip	送信元 IP アドレス
dst_mac	宛先 (ターゲット) MAC アドレス
dst_ip	宛先 IP アドレス

コントローラの起動と同時に ARP リクエストが送信されるようにした。

動的経路制御では予めゲートウェイが設定されていないため、コントローラ起動時に宛先 IP アドレスをゲートウェイに指定して ARP リクエストを送ることができない。よって、表3の dst\_ip にはスイッチのリンクのブロードキャストアドレスを指定する。

また、ARP リクエストを受け取ったスイッチは、Packet In メッセージでコントローラに指示を仰ぎ、コントローラは ARP リプライパケットを生成して Packet Out メッセージをスイッチへ送信する。

#### 3.8.4 UDP で経路情報を交換

開発した動的経路制御のプロトコルでは、UDP メッセージを用いて経路情報の交換を行う。

ARP パケットと同様に、UDP パケットもコントロ

ーラ上で生成する必要がある。UDP パケットの生成には ethernet ヘッダと ipv4 ヘッダと udp ヘッダの情報が必要となる。udp ヘッダの要素を表 4 に示す。

表 4 UDP ヘッダの要素

Attribute	説明
src_port	送信元ポート番号
dst_port	宛先ポート番号
total_length	0 にしておく
csum	0 にしておく

表 4 の src\_port と dst\_port にはプライベートポート番号を割り当てた。total\_length と csum には 0 を指定しておく、自動計算された値が入る。

UDP メッセージを送るタイミングは、コントローラの起動時と、他のスイッチから送信された UDP メッセージがコントローラに Packet In した時である。

起動時に送るメッセージには、各スイッチのリンク上のネットワークの情報が含まれる。この時のメトリックは 1 である。

新たに UDP メッセージがコントローラに Packet In した時には、まず経路表を更新する。その後、メッセージ中のメトリックに 1 を足して、届いたネットワーク情報とメトリックを含んだ UDP メッセージを生成する。次に、生成したパケットをスイッチのリンク上へ送信させるための、Packet Out メッセージをスイッチへ送信する。

### 3.8.5 障害発生時の処理

動的経路制御では、経路に障害が発生した時、迂回路を選択してパケットを宛先に届ける。本研究では、有線 LAN ケーブルがスイッチから引き抜かれたことを具体的な障害とする。

スイッチからケーブルが引き抜かれると、スイッチはコントローラに「Port Status」というメッセージをコントローラへ送信する。このメッセージには障害があったスイッチの id と障害の理由、そして障害のあったポートの番号の情報が含まれている。

コントローラはこのメッセージを受け取った際に、障害があったスイッチのフローテーブルを参照し、「障害があったポートに転送する」というフローエントリを削除する。

障害によって一つの経路が削除されても、UDP メッ

セージの交換により、スイッチは複数の経路情報を保持しているため、別経路を選択して迂回することができる。

### 3.8.6 REST API の組み込み

次に、動的経路制御を実装するスイッチの IP アドレスをコントローラの外部から設定できるようにするため、REST API の組み込みを行う。これにより、外部の Web アプリケーションから、Web サーバとして起動したコントローラに、HTTP リクエストを送れるようになる。

HTTP リクエストの種類としては「GET」と「POST」があり、GET では既に設定した IP アドレスを確認することができ、POST では新たに IP アドレスを設定することができる。

## 4. 動作確認

### 4.1 ルータ

ルータの学習では、まずコントローラを起動し、3 つのスイッチの IP アドレスを設定する。その後、3 つのスイッチのゲートウェイを設定する。

すべての設定が完了した後にホストから ping を送信すると、自分が設定した経路上をパケットが流れる様子を、LED テープの光で確認することができる。

図 5 は LED テープがパケットの流れを可視化する様子である。

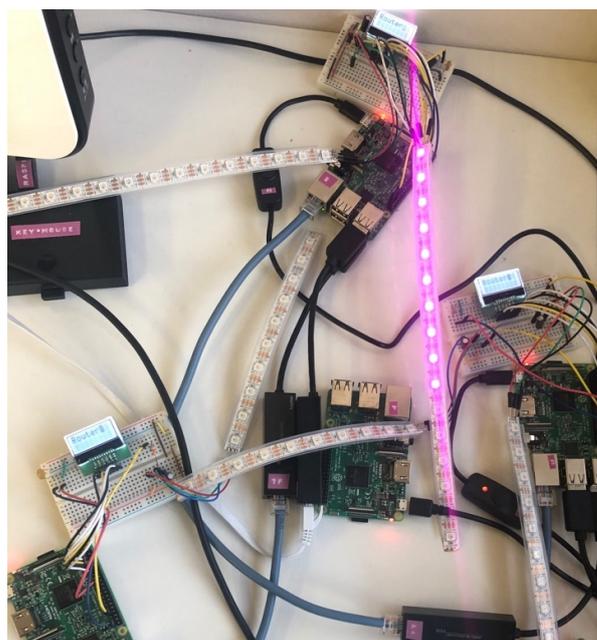


図 5 LED テープがパケットの流れを可視化する様子

また、各スイッチに実装した LCD には図 6 のように、「Router」という文字を表示させた。

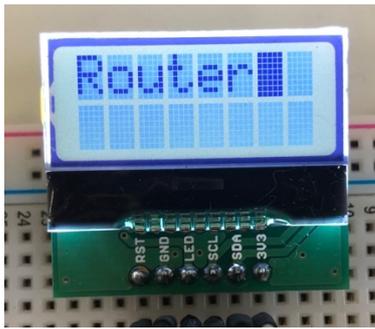


図 6 LCD によるモード表示

#### 4.2 L2 スイッチ

L2 スイッチの学習では、コントローラを起動した後、1つのスイッチのポート番号と、ポートの先に接続されているコンピュータの MAC アドレスを設定することができる。そして、ping を送信することで、自分が設定したポートへパケットが転送されている様子を LED テープの光で確認することができる。

L2 スイッチの学習では、ポート番号と MAC アドレスを正しく設定した場合に ping が成功する様子と、誤って設定した場合に ping の応答が返ってこない様子、また設定を行わずに flooding が起きる様子の3つの場合について、LED テープの光で可視化することができた。

#### 4.3 Firewall

Firewall の学習では、スイッチで「送信元 IP アドレスが外部ネットワークのものなら、通過を許可する」というルールと、「送信元 IP アドレスが内部ネットワークのものなら通過を許可する」というルールを設定する。これらの設定により、図 4 の HOST\_1 からのパケットも、HOST\_2 からのパケットも通過が許可されるので、ping は成功する。

一方、「送信元 IP アドレスが外部ネットワークのものなら、通信を許可する」というルールを削除した場合、HOST\_2 からのパケットは Firewall を実装しているスイッチで遮断され、ping は成功しない。

この様子を、LED テープの光で可視化することができた。

#### 4.4 動的経路制御

動的経路制御の学習では、コントローラを起動した後、3つのスイッチの IP アドレスを設定し、ping を実行すると、パケットは最短経路を選択してネットワーク上を移動する。

次に、最短経路のケーブルを引き抜くと、迂回路を選択して移動する。

その様子を LED テープの光で可視化することができた。

### 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、L2 スイッチ、Firewall、動的経路制御の機能を実装し、それらで構築されたネットワーク上を移動するパケットを、LED テープの光の流れで可視化した。

また、コントローラを起動して設定を行い、ping を送信することで、教材の動作確認を行なった。

これらにより、生徒が実世界の情報通信ネットワークの構成要素について体験的に学習できる教材の開発を行なった。

#### 5.2 今後の課題

本教材では生徒がネットワークの設定を行う際に、curl コマンドや設定ファイルの実行コマンドなどを実行する必要がある。

しかし、高校生にとって Linux のコマンドラインを操作することは難しく、学習意欲を低下させることにつながりかねない。

よって、今後は生徒がネットワークの設定を簡単に行えるように、Web アプリケーションの開発も行う必要がある。

#### 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: "高等学校学習指導要領解説情報編", p.35 (2018)

- (2) 高宮安仁, 鈴木一哉: "クラウド時代のネットワーク技術  
OpenFlow 実践入門"
- (3) NTT: "Ryu 4.34 documentation",  
<https://ryu.readthedocs.io/en/latest/index.html>(2020  
年 12 月 2 日確認)
- (4) Ryubook 1.0 ドキュメント,  
[https://osrg.github.io/ryu-  
book/ja/html/index.html](https://osrg.github.io/ryubook/ja/html/index.html)(2020 年 12 月 3 日確認)