

# 情報ネットワーク学演習 2

## 課題：複数スイッチ対応

秋下 耀介

情報ネットワーク学専攻

学籍番号；33E16001

E-mail: y-akishita@ist.osaka-u.ac.jp

2016 年 10 月 18 日 ( 火 )

### 1 問題：複数スイッチ対応版 ラーニングスイッチ

複数スイッチに対応したラーニングスイッチ (multi\_learning\_switch.rb) の動作を説明しよう。

- 複数スイッチの FDB をどのように実現しているか，コードと動作を解説する
- 動作の様子やフローテーブルの内容もステップごとに確認すること
- 必要に応じて図解すること

### 2 解答：複数スイッチ対応版 ラーニングスイッチ

#### 2.1 各メソッドの動作

まず各メソッドの動作を説明する。

##### 2.1.1 start(\_argv)

```
@fdbbs = {}
```

```
logger.info "#{name} started."
```

このメソッドは，プログラムを開始した際に呼び出される．まず，スイッチに対応する Forwarding data base (FDB) をインスタンス変数 fdbbs という連想配列として初期化する．このような形にすることで，スイッチを複数扱えるようにしている．そして，クラス名を表示する．

##### 2.1.2 switch\_ready(datapath\_id)

```
@fdbbs[datapath_id] = FDB.new
```

このメソッドは，コントローラに対してスイッチが接続した際に呼び出される．連想配列 fdbbs の 1 要素に対して，新たに接続されたスイッチの datapath\_id をキーとして，FDB のオブジェクトを追加している．

### 2.1.3 packet\_in(datapath\_id, packet\_in)

```
return if packet_in.destination_mac.reserved?  
  @fdb.fetch(datapath_id).learn(packet_in.source_mac, packet_in.in_port)  
  flow_mod_and_packet_out packet_in
```

このメソッドは、Flow Table に記載のないパケットがスイッチに到達した際に呼び出される。FDB に記載がなかった (packet\_in.destination\_mac.reserved ではなかった) 際には、FDB を更新する。ただし、switch\_ready メソッドで作成した、各スイッチに対応する FDB を用いる必要があるため、datapath\_id をキーとした fetch 操作により、対応した FDB のオブジェクトを更新する。更新する内容として、MAC アドレス (packet\_in.source\_mac) と、ポート番号 (packet\_in.in\_port) を学習させる。その後、flow\_mod\_and\_packet\_out packet\_in メソッドを呼び出す。

### 2.1.4 flow\_mod\_and\_packet\_out(packet\_in)

```
port_no = @fdb.fetch(packet_in.dpid).lookup(packet_in.destination_mac)  
  flow_mod(packet_in, port_no) if port_no  
  packet_out(packet_in, port_no || :flood)
```

このメソッドは、packet\_in がコントローラに対して送られた際に呼び出される。まずはスイッチで送り出すポート番号を指定する。このとき、パケットの datapath\_id をキーとして FDB を参照するが、ポート番号がわかった場合は、packet\_in および port\_no を引数として flow\_mod メソッドを呼び出す。そして、packet\_out メソッドを呼び出す。ただし、ポート番号がわからない場合は引数で flood を指定する。

### 2.1.5 flow\_mod(packet\_in, port\_no)

```
send_flow_mod_add(  
  packet_in.datapath_id,  
  match: ExactMatch.new(packet_in),  
  actions: SendOutPort.new(port_no)  
)
```

このメソッドは、flow\_mod\_and\_packet\_out メソッドにより呼び出される。このメソッドによってスイッチの Flow Table の更新を行う。この際、オプションとしてマッチフィールド (ここではマッチフィールドの中でもすべての条件を指定したものである ExactMatch)、アクションを指定する (ここでは出力先のポート番号を指定)。

### 2.1.6 packet\_out(packet\_in, port\_no)

```
send_packet_out(  
  packet_in.datapath_id,  
  packet_in: packet_in,  
  actions: SendOutPort.new(port_no)  
)
```

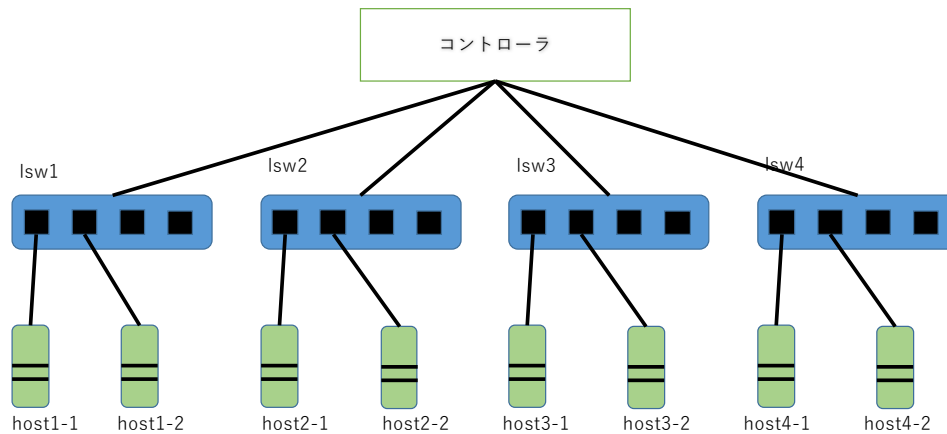


図 1 trema.multi.conf

このメソッドは、`flow_mod_and_packet_out` メソッドにより呼び出される。このメソッドによって、スイッチからパケットを出力する。この際、オプションとしてアクションを指定する（ここでは出力先のポート番号を指定）。

## 2.2 動作の確認と解説

`trema.multi.conf` で指定されている設定は、スイッチ 4 台（`lsw1`、`lsw2`、`lsw3`、`lsw4`）と 8 台のホスト（`host1-1`、`host1-2`、`host2-1`、...、`host4-2`）を構成する。具体的には、図 1 のような形となる。図からもわかるように、`lsw1` に `host1-1` および `host1-2` が、`lsw2` に `host2-1` および `host2-2` が、`lsw3` に `host3-1` および `host3-2` が、`lsw4` に `host4-1` および `host4-2` が接続している。上記のような構造となっていることを確かめるために、`host1-1` と `host1-2`、`host2-1` と `host2-2`、`host1-1` と `host2-1` の接続をステップごとに確認した。

### 2.2.1 `host1-1` と `host1-2` の接続

`show_stats` と `dump_flows` によってホスト間のパケットの送受信および、フローテーブルについて確認を行なった。

```

ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work2/learning-switch-yosuke-akishita$
./bin/trema show_stats host1-1
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work2/learning-switch-yosuke-akishita$
./bin/trema show_stats host1-2
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work2/learning-switch-yosuke-akishita$
./bin/trema send_packets --source host1-1 --dest host1-2 --npackets 5
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work2/learning-switch-yosuke-akishita$
./bin/trema show_stats host1-1
Packets sent:
  192.168.0.1 -> 192.168.0.2 = 5 packets
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work2/learning-switch-yosuke-akishita$
./bin/trema show_stats host1-2
Packets received:
  192.168.0.1 -> 192.168.0.2 = 5 packets
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work2/learning-switch-yosuke-akishita$
./bin/trema send_packets --source host1-2 --dest host1-1 --npackets 3

```

```

ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work2/learning-switch-yosuke-akishita$
./bin/trema show_stats host1-1
Packets sent:
  192.168.0.1 -> 192.168.0.2 = 5 packets
Packets received:
  192.168.0.2 -> 192.168.0.1 = 3 packets
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work2/learning-switch-yosuke-akishita$
./bin/trema show_stats host1-2
Packets sent:
  192.168.0.2 -> 192.168.0.1 = 3 packets
Packets received:
  192.168.0.1 -> 192.168.0.2 = 5 packets
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work2/learning-switch-yosuke-akishita$
./bin/trema dump_flows lsw1

ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work2/learning-switch-yosuke-akishita$
./bin/trema send_packets --source host1-1 --dest host1-2 --npackets 4
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work2/learning-switch-yosuke-akishita$
./bin/trema show_stats host1-1
Packets sent:
  192.168.0.1 -> 192.168.0.2 = 9 packets
Packets received:
  192.168.0.2 -> 192.168.0.1 = 3 packets
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work2/learning-switch-yosuke-akishita$
./bin/trema show_stats host1-2
Packets sent:
  192.168.0.2 -> 192.168.0.1 = 3 packets
Packets received:
  192.168.0.1 -> 192.168.0.2 = 9 packets
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work2/learning-switch-yosuke-akishita$
./bin/trema dump_flows lsw1
cookie=0x0, duration=37.371s, table=0, n_packets=1, n_bytes=42, idle_age=37,
priority=65535,udp,in_port=1,vlan_tci=0x0000,d_l_src=de:df:91:c6:ee:23,d_l_dst=f9:3c:
27:bd:cb:dc,nw_src=192.168.0.1,nw_dst=192.168.0.2,nw_tos=0,tp_src=0,tp_dst=0 actions=output:2

```

まず、host1-1 と host2-1 の状態を確認している。ここでは何もやりとりしていないため、どちらも何も表示されない。そしてその後 host1-1 から host2-1 に対してパケットを 5 つ送信している。その後の状態を確認すると、host1-1 は 5 つパケットを送信し、host1-2 はパケットを 5 つ受信したことが読み取れる。このとき、フローテーブルには情報がいないため、Packet In が発生し、仮想スイッチ lsw1 の FDB が更新される (図 2)。この後、host1-2 から host1-1 にパケットを 3 つ送信する。このとき、同様にフローテーブルには情報がいないため、Packet In が発生し、lsw1 の FDB が更新される (図 3)。このとき ./bin/trema dump\_flows lsw1 によってフローテーブルを確認すると、何も表示されないことがわかる。この後 host1-1 から host2-1 に対してパケットを 4 つ送信し (図 4)、フローテーブルを確認すると、host1-1 から host1-2 へのフローエントリが追加されていることがわかる。よって期待通りの動作をしていることがわかる。

## 2.2.2 host2-1 と host2-2 の接続

host2-1 と host2-2 の接続を行なった際の動作を以下に示す。

```

ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work2/learning-switch-yosuke-akishita$
./bin/trema show_stats host2-1

```

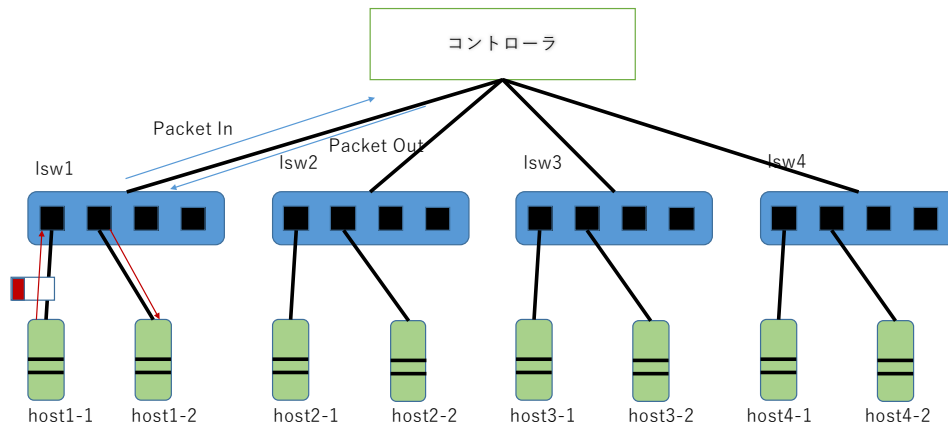


図2 host1-1 と host1-2 の通信 (1/3)

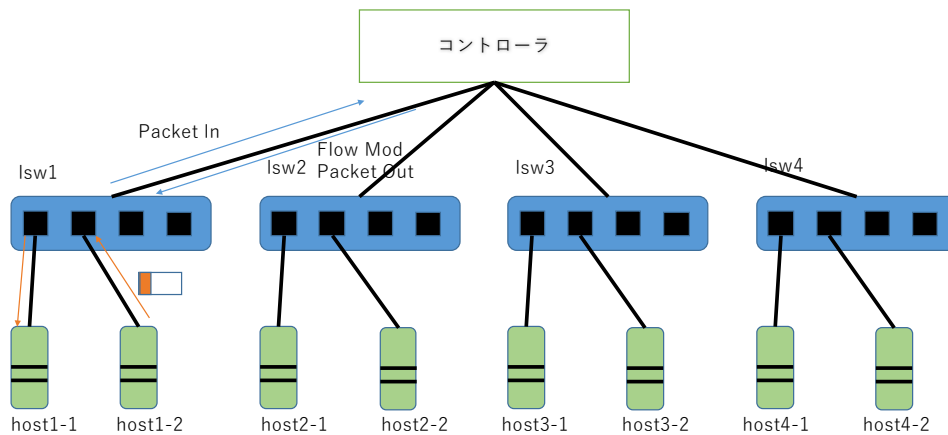


図3 host1-1 と host1-2 の通信 (2/3)

```

ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work2/learning-switch-yosuke-akishita$
./bin/trema show_stats host2-2
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work2/learning-switch-yosuke-akishita$
./bin/trema send_packets --source host2-1 --dest host2-2 --npackets 2
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work2/learning-switch-yosuke-akishita$
./bin/trema show_stats host2-1
Packets sent:
  192.168.0.3 -> 192.168.0.4 = 2 packets
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work2/learning-switch-yosuke-akishita$
./bin/trema show_stats host2-2
Packets received:
  192.168.0.3 -> 192.168.0.4 = 2 packets
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work2/learning-switch-yosuke-akishita$
./bin/trema send_packets --source host2-2 --dest host2-1 --npackets 2
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work2/learning-switch-yosuke-akishita$
./bin/trema show_stats host2-1
Packets sent:
  192.168.0.3 -> 192.168.0.4 = 2 packets
Packets received:
  192.168.0.4 -> 192.168.0.3 = 2 packets

```

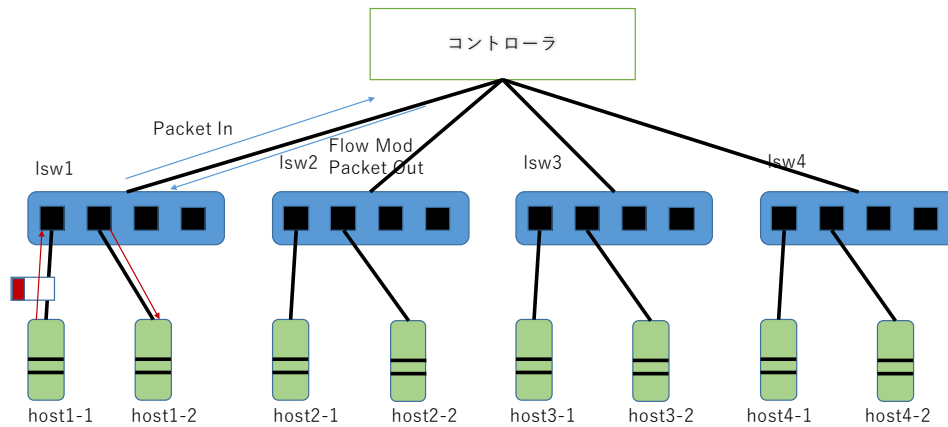


図 4 host1-1 と host1-2 の通信 (3/3)

```
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work2/learning-switch-yosuke-akishita$
./bin/trema send_packets --source host2-2 --dest host2-1 --npackets 2
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work2/learning-switch-yosuke-akishita$
./bin/trema show_stats host2-1
Packets sent:
  192.168.0.3 -> 192.168.0.4 = 2 packets
Packets received:
  192.168.0.4 -> 192.168.0.3 = 4 packets
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work2/learning-switch-yosuke-akishita$
./bin/trema show_stats host2-2
Packets sent:
  192.168.0.4 -> 192.168.0.3 = 4 packets
Packets received:
  192.168.0.3 -> 192.168.0.4 = 2 packets
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work2/learning-switch-yosuke-akishita$
./bin/trema dump_flows lsw2
cookie=0x0, duration=62.169s, table=0, n_packets=2, n_bytes=84, idle_age=52,
priority=65535,udp,in_port=2,vlan_tci=0x0000,dl_src=7f:98:43:02:5c:bf,dl_dst=6d:a9:dd:
5b:f2:7f,nw_src=192.168.0.4,nw_dst=192.168.0.3,nw_tos=0,tp_src=0,tp_dst=0 actions=output:1
```

上記からわかるように、lsw1 と同様の結果となった。

### 2.2.3 host1-1 と host2-1 の接続

lsw1 と lsw2 が接続されていないことを確認するために host2-1 から host2-1 に対してパケットを送信した。その際の結果を以下に示す。

```
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work2/learning-switch-yosuke-akishita$
./bin/trema show_stats host1-1
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work2/learning-switch-yosuke-akishita$
./bin/trema show_stats host2-1
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work2/learning-switch-yosuke-akishita$
./bin/trema send_packets --source host2-1 --dest host1-1 --npackets 3
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work2/learning-switch-yosuke-akishita$
./bin/trema show_stats host1-1
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work2/learning-switch-yosuke-akishita$
```

```
./bin/trema show_stats host2-1
Packets sent:
  192.168.0.3 -> 192.168.0.1 = 3 packets
```

上記からわかるように，host2-1 はパケットを 3 つ送信しているが，host1-1 はパケットを受信していないので，lsw1 と lsw2 は接続されていないことがわかる．他のスイッチ間も同様である，すなわち図 1 のような構成となっていることがわかった．

## 参考文献

- [1] 情報ネットワーク学演習 2 事前準備 <https://github.com/handai-trema/syllabus#事前準備>
- [2] Trema で OpenFlow プログラミング [http://yasuhito.github.io/trema-book/#learning\\_switch](http://yasuhito.github.io/trema-book/#learning_switch)