

情報ネットワーク学演習 2

課題:複数スイッチ対応版 ラーニングスイッチ

氏名: 佐竹 幸大

学籍番号: 33E16010

課題内容(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

解答内容

プログラム説明

start メソッド

```
def start(_argv)
  @fdb = {}
  logger.info 'MultiLearningSwitch started.'
end
```

本メソッドでは、`@fdb` を、連想配列として初期化する。
連想配列とは、オブジェクトの集合のことであり、`@fdb` には、各スイッチと対応した FDB(フォワーディングデータベース)を格納する。その後、本クラスの処理が開始した旨を出力する。

switch_ready メソッド

```
def switch_ready(datapath_id)
  @fdb[@fids[datapath_id]] = FDB.new
end
```

本メソッドでは、`start` メソッドで初期化された連想配列`@fdb` に対して、引数 `datapath_id` によって指定されたスイッチ(新たに接続したいスイッチ)に対応する FDB オブジェクトを新たに追加する。

packet_in メソッド

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

本メソッドは、新たに到着したパケットの送信先がフローテーブルに記述されていない場合に、呼び出される。

本メソッドでは、FDB に記述されていない場合、新たに到着したパケットの MAC アドレス、ポート番号を、FDB に記憶する。

次に、`flow_mod_and_packet_out` メソッドを呼び出す。

flow_mod_and_packet_out メソッド

```
def flow_mod_and_packet_out(message)
  port_no = @fdb.fetch(message.dpid).lookup(message.destination_mac)
  flow_mod(message, port_no) if port_no
  packet_out(message, port_no || :flood)
end
```

本メソッドではまず、対象パケットの送信先ポート番号についての記述を、FDB から探し出し、それに関する記述があった場合には、`flow_mod` により、フローテーブルの更新を行う。

次に、ポート番号の記述があった場合には、そのポート番号に向けて、

なかった場合には、フラッディングを用いて、パケットアウトによりパケットを届ける。

flow_mod メソッド

```
def flow_mod(message, port_no)
  send_flow_mod_add(
    message.datapath_id,
    match: ExactMatch.new(message),
    actions: SendOutPort.new(port_no)
  )
end
```

本メソッドでは、フローテーブルの更新を行う。

flow_mod_and_packet_out で、ポート番号が引数として与えられ、呼び出され、そのポート番号へ送出するよう登録される。

packet_out メソッド

```
def packet_out(message, port_no)
  send_packet_out(
    message.datapath_id,
    packet_in: message,
    actions: SendOutPort.new(port_no)
  )
end
```

本メソッドでは、パケットアウトの処理を行う。

パケットアウトでは、スイッチからパケットを送出する。

動作確認

trema.multi.conf の設定では、図 1 のように、スイッチ 4 台とホスト 8 台から構成されている。4 つのスイッチ(lsw1,lsw2,lsw3,lsw4)に対してホストが 2 つずつ接続されている。

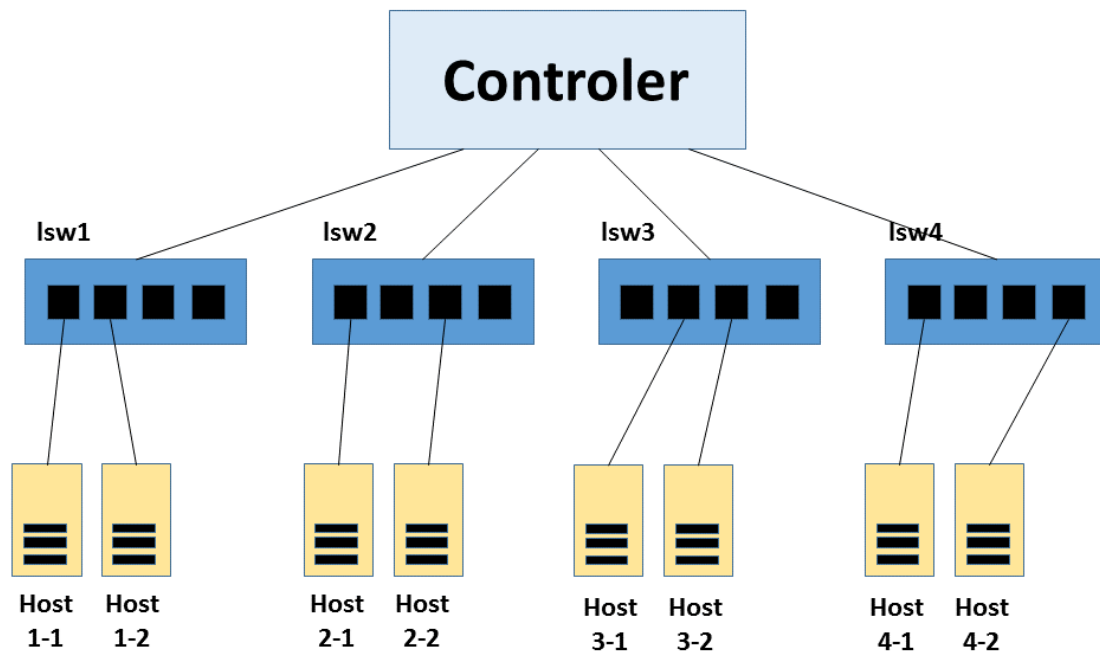


図 1 : tremamulti.conf の設定における SDN 構造

ここで、以下の作業を行い、動作確認を行った。以下で行った作業と確認できた結果について説明する。

・ host1-1 と host1-2 の接続

host1-1 と host1-2 の接続を確認するに当たり、以下の作業を行った。

step1. host1-1 から host1-2 へパケットを送出

step2. host1-2 から host1-1 へパケットを送出

step3. host1-1 から host1-2 へパケットを送出

その結果を以下に示す。

- **step1.**

Packet In が発生し、lsw1 の FDB に host1-1 の情報が記述される。

次に、Packet Out が発生し、host1-2 へパケットが送出される。

- **step2.**

Packet In が発生し、lsw1 の FDB に host1-2 の情報が記述される。

この時点ですでに lsw1 の FDB に記述されている host1-1 の情報に基づき、host1-1 へパケットが送出される。

この際、host1-1 へのパケットの送出先ポートがフローテーブルに記述される。

- **step3.**

host1-1 から host1-2 へのフローエントリが追加され、host1-2 へとパケットは送出される。

- **host2-1 と host2-2 の接続**

Host2-1 と host2-2 の接続を確認するに当たり、前節と同様に以下の作業を行った。

step1. host2-1 から host2-2 へパケットを送出

step2. host2-2 から host2-1 へパケットを送出

step3. host2-1 から host2-2 へパケットを送出

その結果、すべての step において、host1-1,host1-2 間と同様の結果が得られた。

- **host1-1 と host2-1 の接続**

host1-1 から host2-1 へパケットを送出

host2-1 から host1-1 へパケットを送出

host1-1 から host2-1 へパケットを送出

この場合、送信側のホストでは送信が確認されるが、受信側のホストでは受信が確認できなかった。これは異なるスイッチに接続しているためである、フローテーブルにも変化は見られなかった。