

# 情報ネットワーク学演習 2

## 課題: パッチパネルの拡張

33E16022 村上 遼

平成 28 年 10 月 25 日

### 1 課題内容

パッチパネルに機能を追加する。授業で説明のあったパッチの追加と削除以外に、以下の機能を追加する。

1. ポートのミラーリング
2. パッチとポートミラーリングの一覧

### 2 課題に対する回答

#### 2.1 ミラーリングの追加

辻健太さんの回答を元にソースコードの変更を考案した。ミラーリングの作成のために、`create_mirror` メソッドと `add_mirror_entries` メソッドを追加した。リスト 1 に `create_mirror` メソッドと `add_mirror_entries` メソッドを示す。

リスト 1: ミラーリング作成用のメソッド

```
1 def create_mirror(dpid, port_monitor, port_mirror)
2   add_mirror_entries dpid, port_monitor, port_mirror
3   @mirror[dpid] += [[port_monitor, port_mirror]]
4 end
5
6 def add_mirror_entries(dpid, port_monitor, port_mirror)
7   send_flow_mod_delete(dpid, match: Match.new(in_port: port_mirror))
8   for patch_tmp in @patch[dpid].each do
9     port_in = patch_tmp[0]
10    port_out = patch_tmp[1]
11    if port_in == port_monitor then
12      send_flow_mod_delete(dpid, match: Match.new(in_port: port_in))
13      send_flow_mod_add(dpid,
14                        match: Match.new(in_port: port_monitor),
15                        actions: [
16                          SendOutPort.new(port_out),
17                          SendOutPort.new(port_mirror)
18                        ])
19    end
20  end
21 end
```

`create_mirror` メソッドでは、入力された情報を元に変数 `@mirror` にミラーリングの情報を追加している。モニタ用のポートとミラーリング用のポートを区別するために、変数 `@mirror` に追加するときにソートを行っている。

`add_mirror_entries` では、一度スイッチ側の FlowTable 規則を削除してから、ミラーリング生成に必要な FlowTable の情報追加を行っている。

## 2.2 ミラーリングの削除

ミラーリングの削除のために、`delete_mirror` メソッドと `delete_mirror_entries` メソッドを追加した。リスト 1 に `create_mirror` メソッドと `delete_mirror_entries` メソッドを示す。

リスト 2: ミラーリング削除用のメソッド

```
1 def delete_mirror(dpid, port_monitor, port_mirror)
2   if @mirror[dpid].include?([port_monitor, port_mirror]) then
3     delete_mirror_entry dpid, port_monitor, port_mirror
4     @mirror[dpid] -= [[port_monitor, port_mirror]]
5   end
6 end
7
8 def delete_mirror_entry(dpid, port_monitor, port_mirror)
9   send_flow_mod_delete(dpid, match: Match.new(in_port: port_monitor))
10  for patch_tmp in @patch[dpid].each do
11    port_in = patch_tmp[0]
12    port_out = patch_tmp[1]
13    if port_in == port_monitor then
14      send_flow_mod_add(dpid,
15                        match: Match.new(in_port: port_in),
16                        actions: SendOutPort.new(port_out))
17    end
18  end
19 end
```

`delete_mirror` メソッドでは、変数 `@mirror` にミラーリングが登録されていた場合、変数 `@mirror` からミラーリングの情報を削除している。

`delete_mirror_entry` メソッドでは、一度スイッチ側の `FlowTable` を削除してから、ミラーリング削除後に必要な `FlowTable` の情報を追加している。

## 2.3 出力

情報の出力のために、`dump` メソッドを追加した。リスト 3 に `dump` メソッドを示す。

リスト 3: `dump`

```
1 def dump(dpid)
2   str = "Patches:\n"
3   for patch_tmp in @patch[dpid].each do
4     port_in = patch_tmp[0]
5     port_out = patch_tmp[1]
6     str += "\t"
7     str += port_in.to_s
8     str += "<->"
9     str += port_out.to_s
10    str += "\n"
11  end
12
13  str += "Mirrors:\n"
14  for mirror_tmp in @mirror[dpid].each do
15    port_monitor = mirror_tmp[0]
16    port_mirror = mirror_tmp[1]
17    str += "\t"
18    str += port_monitor.to_s
19    str += "->"
20    str += port_mirror.to_s
21    str += "\n"
22  end
23  str
24 end
```

`dump` メソッドでは、ミラーリングの情報だけでなく、接続情報も出力している。変数 `str` に情報を格納して、3 の 28 行目で出力している。

## 3 動作確認

動作確認のため、`trema run ./lib/patch_panel.rb -c patch_panel.conf` を実行した。しかしながら、エラーが発生して実効することができなかった。リスト 4 にエラーメッセージを示す。

#### リスト 4: 動作結果

```

1 ensyuu2@ensyuu2-VirtualBox:~/week3/patch-panel-Ryo-Murakami$ trema run ./lib/patch_panel.rb
2 -c patch_panel.conf
3 /home/ensyuu2/.rvm/rubies/ruby-2.2.5/lib/ruby/2.2.0/drb/unix.rb:41:in 'initialize': Address already
4 in use - connect(2) for /tmp/trema.PatchPanel.ctl (Errno::EADDRINUSE)
5 from /home/ensyuu2/.rvm/rubies/ruby-2.2.5/lib/ruby/2.2.0/drb/unix.rb:41:in 'open'
6 from /home/ensyuu2/.rvm/rubies/ruby-2.2.5/lib/ruby/2.2.0/drb/unix.rb:41:in 'open_server'
7 from /home/ensyuu2/.rvm/rubies/ruby-2.2.5/lib/ruby/2.2.0/drb/drb.rb:767:in 'block in open_server'
8 from /home/ensyuu2/.rvm/rubies/ruby-2.2.5/lib/ruby/2.2.0/drb/drb.rb:765:in 'each'
9 from /home/ensyuu2/.rvm/rubies/ruby-2.2.5/lib/ruby/2.2.0/drb/drb.rb:765:in 'open_server'
10 from /home/ensyuu2/.rvm/rubies/ruby-2.2.5/lib/ruby/2.2.0/drb/drb.rb:773:in 'open_server'
11 from /home/ensyuu2/.rvm/rubies/ruby-2.2.5/lib/ruby/2.2.0/drb/drb.rb:1405:in 'initialize'
12 from /home/ensyuu2/.rvm/rubies/ruby-2.2.5/lib/ruby/2.2.0/drb/drb.rb:1695:in 'new'
13 from /home/ensyuu2/.rvm/rubies/ruby-2.2.5/lib/ruby/2.2.0/drb/drb.rb:1695:in 'start_service'
14 from /home/ensyuu2/.rvm/gems/ruby-2.2.5/gems/trema-0.9.0/lib/trema/command.rb:124:
15   in 'start_controller_and_drb_threads'
16 from /home/ensyuu2/.rvm/gems/ruby-2.2.5/gems/trema-0.9.0/lib/trema/command.rb:51:in 'run'
17 from /home/ensyuu2/.rvm/gems/ruby-2.2.5/gems/trema-0.9.0/bin/trema:54:in 'block (2 levels) in <module:App>'
18 from /home/ensyuu2/.rvm/gems/ruby-2.2.5/gems/gli-2.13.4/lib/gli/command_support.rb:126:in 'call'
19 from /home/ensyuu2/.rvm/gems/ruby-2.2.5/gems/gli-2.13.4/lib/gli/command_support.rb:126:in 'execute'
20 from /home/ensyuu2/.rvm/gems/ruby-2.2.5/gems/gli-2.13.4/lib/gli/app_support.rb:296:in 'block in call_command'
21 from /home/ensyuu2/.rvm/gems/ruby-2.2.5/gems/gli-2.13.4/lib/gli/app_support.rb:309:in 'call'
22 from /home/ensyuu2/.rvm/gems/ruby-2.2.5/gems/gli-2.13.4/lib/gli/app_support.rb:309:
23   in 'call_command'
24 from /home/ensyuu2/.rvm/gems/ruby-2.2.5/gems/gli-2.13.4/lib/gli/app_support.rb:83:in 'run'
25 from /home/ensyuu2/.rvm/gems/ruby-2.2.5/gems/trema-0.9.0/bin/trema:268:in '<module:App>'
26 from /home/ensyuu2/.rvm/gems/ruby-2.2.5/gems/trema-0.9.0/bin/trema:14:in '<module:Trema>'
27 from /home/ensyuu2/.rvm/gems/ruby-2.2.5/gems/trema-0.9.0/bin/trema:12:in '<top (required)>'
28 from /home/ensyuu2/.rvm/gems/ruby-2.2.5/bin/trema:23:in 'load'
29 from /home/ensyuu2/.rvm/gems/ruby-2.2.5/bin/trema:23:in '<main>'
30 from /home/ensyuu2/.rvm/gems/ruby-2.2.5/bin/ruby_executable_hooks:15:in 'eval'
31 from /home/ensyuu2/.rvm/gems/ruby-2.2.5/bin/ruby_executable_hooks:15:in '<main>'

```

エラーメッセージの原因を特定することができなかった。書き換えた ruby ファイルではない場所からエラーが発生しているため、環境構築が上手く出来ていない可能性がある。