

情報ネットワーク学演習 II

第3回レポート課題

所属：大阪大学 大学院情報科学研究科 情報ネットワーク専攻

提出者：33E16019 満越貴志

電子メールアドレス：t-mangoe@ist.osaka-u.ac.jp

提出年月日：平成 28 年 10 月 23 日

課題内容

パッチパネルに以下の機能を追加する。

1. ポートのミラーリング
2. パッチとポートのミラーリングの一覧の表示

1 ポートのミラーリング

ポートのミラーリングを実装するにあたり、lib 配下の patch_panel.rb に以下の public メソッドを追加した。

```
def mirror_patch(dpid, port_a, port_b)
  add_mirror_flow_entrie dpid, port_a, port_b
  @mirror[dpid] += [port_a, port_b]
end
```

このメソッドでは、ミラーリングの設定をする private メソッド (add_mirror_flow_entrie メソッド) の呼び出しと、ミラーリング情報をハッシュ (@mirror) へと登録する処理が行われる。add_mirror_flow_entrie メソッドの定義は、以下の通りである。

```
def add_mirror_flow_entrie(dpid, port_a, port_b)
  send_flow_mod_add(dpid,
                    match: Match.new(in_port: port_a),
                    actions: SendOutPort.new(port_b))
end
```

add_mirror_flow_entrie メソッドでは、dpid で指定されたスイッチに flow_mod を送り、port_a から送られてきたパケットを port_b に出力するように、スイッチのフローテーブルにエントリーを追加する。行っている処理は、フローテーブルにパッチ用のエントリーを作成する add_flow_entries と似ているが、add_mirror_flow_entrie メソッドでは flow_mod は 1 回しかスイッチに送らない。そのため、port_a からのパケットは port_b に出力されるが、その逆は起こらず、port_a が port_b にミラーリングされることとなる。

ミラーリングの情報を登録するハッシュは、start メソッドで以下のように定義されている。

```
@mirror = Hash.new { [] }
```

パッチの情報を登録するハッシュと同じ形式で定義されている。

追加した `mirror_patch` メソッドを呼び出せるように、`bin` 配下の `patch_panel` にコードを以下のよう追加した。

```
desc 'mirroring from port#1 to port#2'
arg_name 'dpid port#1 port#2'
command :mirror do |c|
  c.desc 'Location to find socket files'
  c.flag [:S, :socket_dir], default_value: Trema::DEFAULT_SOCKET_DIR

  c.action do |_global_options, options, args|
    dpid = args[0].hex
    port1 = args[1].to_i
    port2 = args[2].to_i
    Trema.trema_process('PatchPanel', options[:socket_dir]).controller.
      mirror_patch(dpid, port1, port2)
  end
end
```

このコードを追加することで、`mirror` というコマンドが実装される。スイッチの ID とミラー元のポート、ミラー先のポートを引数として指定して `mirror` コマンドを実行することで、ポートのミラーリングを行うことができる。

2 パッチとポートのミラーリングの一覧の表示

パッチとポートのミラーリングの一覧を表示するメソッドとして、`show_list` というメソッドを `lib` 配下の `patch_panel.rb` に以下のように追加した。

```
def show_list(dpid)
  @patch[dpid].each do |port_a, port_b|
    puts "port_#{port_a} and port_#{port_b} is patched."
  end
  @mirror[dpid].each do |port_a, port_b|
    puts "port_#{port_a} and port_#{port_b} is mirrored."
  end
end
```

このメソッドでは、パッチとミラーリングの情報を保持するハッシュの内容を出力させる。このメソッドをコマンドとして呼び出せるように、`bin` 配下の `patch_panel` に以下のコードを追加した。

```
desc 'show list of patch and mirror'
arg_name 'dpid'
command :list do |c|
  c.desc 'Location to find socket files'
  c.flag [:S, :socket_dir], default_value: Trema::DEFAULT_SOCKET_DIR

  c.action do |_global_options, options, args|
    dpid = args[0].hex
    Trema.trema_process('PatchPanel', options[:socket_dir]).controller.
      show_list(dpid)
  end
end
```

一覧の表示は、`list` コマンドとして実装した。スイッチの ID を引数として実行することで `show_list` のメソッドが呼び出され、一覧が表示される。