

情報ネットワーク学演習 2

課題：パッチパネルの拡張

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 25 日 (火)

1 課題：パッチパネルの機能拡張

パッチパネルに機能を追加しよう。授業で説明したパッチの追加と削除以外に，以下の機能をパッチパネルに追加してください。

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

それぞれ `patch_panel` のサブコマンドとして実装してください。なお 1 と 2 以外にも機能を追加した人には，ボーナス点を加えます。

2 解答：パッチパネルの機能拡張

2.1 ポートのミラーリング

今回実装する機能であるポートのミラーリングとは，接続されているポート同士だけではなく，その他のポートから，ポートを流れるパケットを確認できるようにすることである。本機能は，ネットワークの管理者が流れているトラフィックを監視する際に役立つ。具体例として，ポート 1 とポート 2 が接続されていたとして，ポート 1 に送受信されるパケットを，送受信元ではないポート 3 から確認できるようにする，などが考えられる。

今回の課題では，ポートのミラーリング機能を実装するために，`lib/patch_panel.rb` および `bin/patch_panel` の変更を行った。

2.1.1 `lib/patch_panel.rb`

まず，`lib/patch_panel.rb` に，`create_mirror` というメソッドを追加した。メソッドの内容は以下である。

```
add_mirror_entries dpid, port_a, mirror_port
  @mirror[dpid] << [port_a, mirror_port].sort
```

ここでは、元から実装されている create_patch メソッドと同様に、private として add_mirror_entries メソッドを作成し、それを呼び出す形とした。その後、ミラーリングの設定ポートをインスタンス変数 @mirror に保存している（ミラーリングの一覧表示に使用）。add_mirror_entries の内容は以下である。

```
@patch[dpid].each do |port_a, port_b|
  port_t=nil
  port_t=port_b if port_a==port_source
  port_t=port_a if port_b==port_source
  if port_t!=nil then
    logger.info "port#{mirror_port}:Mirror of port#{port_source} is added."
    send_flow_mod_delete(dpid, match: Match.new(in_port: port_source))
    send_flow_mod_delete(dpid, match: Match.new(in_port: port_t))
    send_flow_mod_add(dpid,
                      match: Match.new(in_port: port_source),
                      actions:[
                        SendOutPort.new(port_t),
                        SendOutPort.new(mirror_port)
                      ])
    send_flow_mod_add(dpid,
                      match: Match.new(in_port: port_t),
                      actions:[
                        SendOutPort.new(port_source),
                        SendOutPort.new(mirror_port)
                      ])
  end
end
```

上記によって、期待する動作を行うメソッドの実装は完了したが、配布されたプログラムでは、2次元 Hash で保存されないため、正常に設定できないという不具合が発生した。そこで start ハンドラにおいて

```
@patch = Hash.new { |p1,p2| p1[p2]=[] }
```

といった形で初期化を行うように変更を行った。

2.1.2 bin/patch_panel

あらかじめ記述されていた create コマンドに関する記述を参考にして、mirrorCreate コマンドを作成した。記述内容は以下である。

```
desc 'Create a mirror'
arg_name 'dpid port#1 mirrorPort'
command :mirrorCreate 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
    mirror_port = args[2].to_i
    Trema.trema_process('PatchPanel', options[:socket_dir]).controller.
      create_mirror(dpid, port1, mirror_port)
```

```
end
end
```

引数としては、監視するポート番号およびミラーリング先のポート番号を入力させるようにした。このコマンドによって、lib/patch_panel.rb の create_patch メソッドが呼び出される。

2.1.3 動作確認

まず動作確認を行う前に、patch_panel.conf において設定の変更を行った。今回はポート監視を行うホストを host3 であるとしたので、

```
vhost ('host3') { ip '192.168.0.3'
promisc true
}
```

のように変更を行った。ホストは通常、宛先が自分になっているパケットのみを受信するよう設定されているため、ポートから別の宛先のパケットが届いても読み込まずに破棄するようになっている。そのため promisc true の記述によって、自分宛ではないパケットであっても読み込むように設定を行った。

コマンドとしては、

```
bundle exec ./bin/patch_panel createMirror [port_a] [mirror]
```

によって実行される。port_a は監視したいポート番号、mirror はミラーリングの出力先のポート番号となっている。実行結果は以下のようになっている。

```
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work3/patch-panel-yosuke-akishita$ bundle exec
./bin/trema show_stats host1
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work3/patch-panel-yosuke-akishita$ bundle exec
./bin/trema show_stats host2
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work3/patch-panel-yosuke-akishita$ bundle exec
./bin/trema show_stats host3
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work3/patch-panel-yosuke-akishita$ bundle exec
./bin/patch_panel create 0xabc 1 2
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work3/patch-panel-yosuke-akishita$ bundle exec
./bin/trema send_packet --source host1 --dest host2
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work3/patch-panel-yosuke-akishita$ bundle exec
./bin/trema show_stats host1
Packets sent:
  192.168.0.1 -> 192.168.0.2 = 1 packet
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work3/patch-panel-yosuke-akishita$ bundle exec
./bin/trema show_stats host2
Packets received:
  192.168.0.1 -> 192.168.0.2 = 1 packet
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work3/patch-panel-yosuke-akishita$ bundle exec
./bin/trema show_stats host3
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work3/patch-panel-yosuke-akishita$ bundle exec
./bin/patch_panel mirrorCreate 0xabc 1 3
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work3/patch-panel-yosuke-akishita$ bundle exec
./bin/trema send_packet --source host1 --dest host2
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work3/patch-panel-yosuke-akishita$ bundle exec
./bin/trema show_stats host1
Packets sent:
```

```

192.168.0.1 -> 192.168.0.2 = 2 packets
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work3/patch-panel-yosuke-akishita$ bundle exec
./bin/trema show_stats host2
Packets received:
192.168.0.1 -> 192.168.0.2 = 2 packets
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work3/patch-panel-yosuke-akishita$ bundle exec
./bin/trema show_stats host3
Packets received:
192.168.0.1 -> 192.168.0.2 = 1 packet
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work3/patch-panel-yosuke-akishita$ bundle exec
./bin/trema send_packet --source host2 --dest host1
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work3/patch-panel-yosuke-akishita$ bundle exec
./bin/trema show_stats host1
Packets sent:
192.168.0.1 -> 192.168.0.2 = 2 packets
Packets received:
192.168.0.2 -> 192.168.0.1 = 1 packet
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work3/patch-panel-yosuke-akishita$ bundle exec
./bin/trema show_stats host2
Packets sent:
192.168.0.2 -> 192.168.0.1 = 1 packet
Packets received:
192.168.0.1 -> 192.168.0.2 = 2 packets
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work3/patch-panel-yosuke-akishita$ bundle exec
./bin/trema show_stats host3
Packets received:
192.168.0.1 -> 192.168.0.2 = 1 packet
192.168.0.2 -> 192.168.0.1 = 1 packet

```

上記から、パッチ操作後のパケット送信によって、host1 と host2 が繋がったことがまず読み取れる。そして、ミラーリング設定を行った後、パケット送信を行うことで host3 に host1 を出入りするパケットがミラーリングされていることが読み取れる。以上をもって、期待する動作を行うようになったと言える。

2.2 パッチとポートミラーリングの一覧

今回の課題では、パッチとポートミラーリングの一覧表示機能を実装するために、lib/patch_panel.rb および bin/patch_panel の変更を行った。

2.2.1 lib/patch_panel.rb

list という新たなメソッドを実装した。実装内容は以下である。

```

list_temp = Array.new()
list_temp << @patch
list_temp << @mirror
return list_temp

```

上記では、@patch と@mirror を配列に入れて戻り値として返す。コマンドを入力した端末側で情報が見たいので、ここでは一覧を読み込むだけに操作をとどめている。

2.2.2 bin/patch_panel

mirrorCreate と同様にして list コマンドを実装した。実装した内容は以下のようになっている。

```
desc 'List 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
      list_temp=Trema.trema_process('PatchPanel', options[:socket_dir]).controller.
        list()
      @patch = list_temp[0]
      @mirror = list_temp[1]

      print("patch list:\n")
      @patch[dpid].each do |port_a, port_b|
        print("port", port_a, " << >> port", port_b, "\n")
      end
      print("mirror list:\n")
      @mirror[dpid].each do |port_a, mirror_port|
        print("port", port_a, " >> port", mirror_port, "\n")
      end
    end
  end
end
```

引数としては、表示させたいスイッチの dpid を入力するようにした。このコマンドによって、lib/patch_panel.rb の list メソッドが呼び出される。そして、dpid で指定されたパッチの一覧およびミラーリングの一覧を表示させる。

2.2.3 動作確認

コマンドとしては、

```
bundle exec ./bin/patch_panel list [dpid]
```

のようにして実行される。dpid には表示させるスイッチを指定する。実行結果は以下のようになっている。

```
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work3/patch-panel-yosuke-akishita$ bundle exec
./bin/patch_panel create 0xabc 1 2
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work3/patch-panel-yosuke-akishita$ bundle exec
./bin/patch_panel mirrorCreate 0xabc 1 3
ensyuu2@ensyuu2-VirtualBox:~/ensyuu2/work3/patch-panel-yosuke-akishita$ bundle exec
./bin/patch_panel list 0xabc
patch list:
port1 << >> port2
mirror list:
port1 >> port3
```

上記から、パッチ操作によって port1 と port2 が接続され、その後 port1 のミラーが port3 に作られてい

ることがわかる．その後 list コマンドによってそれらが一覧として表示されていることから，正しく動作していると言える．

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/#patch_panel
- [3] Ruby で多次元 hash の初期化をまとめてみた <http://qiita.com/shibukk/items/35c4859e7ca84a427e25>