

情報ネットワーク学演習 2

33E16010 佐竹幸大

課題内容 (マルチプルテーブルを読む)

OpenFlow1.3 版スイッチの動作を説明しよう。

スイッチ動作の各ステップについて、`trema dump_flows` の出力 (マルチプルテーブルの内容) を混じえながら動作を説明すること。

実行のしかた

以下のように `--openflow13` オプションが必要です。

```
$ bundle exec trema run lib/learning_switch13.rb --openflow13  
-c trema.conf
```

解答内容 (マルチプルテーブルを読む)

動作説明

以下では、スイッチ接続時とパケット送信時に分けて動作説明をする。

スイッチの接続

```
ensyuu2@ensyuu2-VirtualBox:~/learning-switch-  
ksatake$ ./bin/trema dump_flows lsw  
cookie=0x0, duration=301.023s, table=0, n_packets=0,  
n_bytes=0,
```

```
priority=2,dl_dst=01:00:00:00:00:00/ff:00:00:00:00:00
0 actions=drop
cookie=0x0, duration=301.023s, table=0, n_packets=0,
n_bytes=0,
priority=2,dl_dst=33:33:00:00:00:00/ff:ff:00:00:00:00
0 actions=drop
cookie=0x0, duration=301.023s, table=0, n_packets=0,
n_bytes=0, priority=1 actions=goto_table:1
cookie=0x0, duration=301.023s, table=1, n_packets=0,
n_bytes=0, priority=3,dl_dst=ff:ff:ff:ff:ff:ff
actions=FL00D
cookie=0x0, duration=301.023s, table=1, n_packets=0,
n_bytes=0, priority=1 actions=CONTROLLER:65535
```

ここでは、フローテーブルの初期化が行われている。
フィルタリングテーブルのフローエントリ 1~3 及び転送テーブルのフローエントリ 4,5 がある。

パケットの送信 (host1 から host2 への送信)

マルチプルテーブルの出力を以下に示す。

```
ensyuu2@ensyuu2-VirtualBox:~/learning-switch-
ksatake$ ./bin/trema send_packets --source host1 --
dest host2
ensyuu2@ensyuu2-VirtualBox:~/learning-switch-
ksatake$ ./bin/trema dump_flows lsw
cookie=0x0, duration=362.112s, table=0, n_packets=0,
n_bytes=0,
priority=2,dl_dst=01:00:00:00:00:00/ff:00:00:00:00:00
0 actions=drop
```

```
cookie=0x0, duration=362.112s, table=0, n_packets=0,
n_bytes=0,
priority=2,dl_dst=33:33:00:00:00:00/ff:ff:00:00:00:0
0 actions=drop
cookie=0x0, duration=362.112s, table=0, n_packets=1,
n_bytes=42, priority=1 actions=goto_table:1
cookie=0x0, duration=362.112s, table=1, n_packets=0,
n_bytes=0, priority=3,dl_dst=ff:ff:ff:ff:ff:ff
actions=FL00D
cookie=0x0, duration=362.112s, table=1, n_packets=1,
n_bytes=42, priority=1 actions=CONTROLLER:65535
```

ここでは以下のような手順で処理が行われている。

- ・ host2 宛のパケットがフィルタリングにかからずに、フローエントリ 3 によって転送テーブルへ転送される
- ・ 転送テーブルにおいては、フラッディングは発生せず、フローエントリ 5 で、Packet In が生じる。

これらのことはフローエントリ 3,5 の n_packets が 1 になっていることから、フローエントリの 3,5 が用いられたことがわかる。

パケットの送信 (host2 から host1 への送信)

マルチプルテーブルの出力を以下に示す。

```
ensyuu2@ensyuu2-VirtualBox:~/learning-switch-
ksatake$ ./bin/trema send_packets --source host2 --
dest host1
ensyuu2@ensyuu2-VirtualBox:~/learning-switch-
ksatake$ ./bin/trema dump_flows lsw
```

```
cookie=0x0, duration=386.349s, table=0, n_packets=0,
n_bytes=0,
priority=2,dl_dst=01:00:00:00:00:00/ff:00:00:00:00:0
0 actions=drop
cookie=0x0, duration=386.349s, table=0, n_packets=0,
n_bytes=0,
priority=2,dl_dst=33:33:00:00:00:00/ff:ff:00:00:00:0
0 actions=drop
cookie=0x0, duration=386.349s, table=0, n_packets=2,
n_bytes=84, priority=1 actions=goto_table:1
cookie=0x0, duration=386.349s, table=1, n_packets=0,
n_bytes=0, priority=3,dl_dst=ff:ff:ff:ff:ff:ff
actions=FL00D
cookie=0x0, duration=3.526s, table=1, n_packets=0,
n_bytes=0, idle_timeout=180,
priority=2,in_port=2,dl_src=4f:a5:1a:ce:gh:ed,dl_dst
=d9:2c:2r:63:22:10 actions=output:1
cookie=0x0, duration= 386.349s, table=1, n_packets=2,
n_bytes=84, priority=1 actions=CONTROLLER:65535
```

ここでは以下のような手順で処理が行われている。

- ・・host1 宛のパケットがフィルタリングにかからずに、フローエントリ 3 によって転送テーブルへ転送される
- ・転送テーブルにおいては、フラッディングは発生せず、フローエントリ 5 で、Packet In が生じる。

また、前述の host1 から host2 へのパケット送信の際に発生した PacketIn において、host1 のポート番号を記録しているため、フローテーブルが更新されている。

参考文献

デビッド・トーマス+アンドリュー・ハント(2001)「プログラミング Ruby」ピアソン・エデュケーション．テキスト:8 章 “OpenFlow1.3 版ラーニングスイッチ”