A Packet-In Message Filter in OpenFlow Switches for Reducing Control Messages

Daisuke Kotani, Yasuo Okabe (Kyoto University)
OpenFlow:
A South Bound Interface in Software Defined Networks

- We can define how switches process packets via Flow Tables
- Centralized Management of Flow Tables in a Controller
- We can implement any controller we want
Packet Forwarding in OpenFlow

Matched With Flow Entry: Follow Actions in Flow Entry

Missing Flow Entry, or Action is to Send to Controller: Forward to Controller as Packet-In message
A Problem occurred by many Packet-In messages

- Some Packet-In messages are necessary
  - E.g. MAC address learning, Multicast Routing
- Hosts can overload switches, controllers, and control network by transmitting many packets until flow entries are set.
  - Suddenly transmit a large number of packets without any negotiation in advance
  - Video streaming, ARP flooding, etc…

```
Host A
```
```
Host B
```
```
Switches would be overloaded
```
```
Traffic in Control Network increases
```
```
A controller would be overloaded.
```

```
OpenFlow Switch
```
```
OpenFlow Controller
```

```
Packets
```
```
Packets
```
```
Packet-In messages
```

13/07/31
A Problem occurred by many Packet-In messages

- Some Packet-In messages are necessary
  - E.g. MAC address learning, Multicast Routing
- Hosts can overload switches, controllers, and control network by transmitting many packets until flow entries are set.
  - Suddenly transmit a large number of packets without any negotiation in advance
  - Video streaming, ARP flooding, etc…
A Problem occurred by many Packet-In messages

- Rate Limit of all Packet-Ins is harmful
  - This approach may also drop important Packet-In messages
    E.g. ones that trigger to update Flow tables
    ones that update data in the controller

OpenFlow Switches need to have a mechanism to pass through only important Packet-In messages
Overview of Our Work

- A Problem by many Packet-In messages
  - Many Packet-Ins are sent to a controller before a flow entry is set (Consume resources for the control plane)
  - Rate Limit of all Packet-Ins is harmful
    - Such method may drop many important Packet-Ins

- Categorize Packet-In messages from controller’s view
  - State Change, Flow Setup, and Forward
  - State Change and Flow Setup are more important than Forward

- Propose a method to filter out Packet-Ins in “Forward”
  - Switches record flows whose packets are sent to the controller
  - Wildcards can be controlled by specifying from the controller in advance
Related Work

- Improve performance and stability of controllers
  - ONIX [Koponen 2010], HyperFlow [Tootoonchian 2010]
  - Loads of switches and management networks would not be reduced

- Enable more complex operations in switches
  - DIFANE [Yu 2010], DevoFlow [Curtis 2011]
  - Packet-In messages would be reduced

- Meter (introduced in OpenFlow 1.3)
  - Can be used in our work as a part of rate limit mechanism

- Our previous work [Kotani 2012]
  - No consideration for the use of wildcards in flow entries
Categorize Packet-In messages from controller’s view

① **State Change**: Important for control
   - Change data in controllers
   - Ex: Obtain host locations using ARP
   - Set Flow Entries manually to send to controllers in advance

② **Flow Setup**: Important for control
   - Trigger to insert new flow entry
   - The first packet that will be matched with new flow entries

③ **Forward**: Less Important for control
   - Forward packets to other switches
   - Second or later packets that will be matched with flow entries set by ②

Note: A message would be categorized to two or more categories.

Need a mechanism to distinguish ② and ③
Pending Flow Rules and Tables

- Pending Flow Rules
  - Rules show which headers are used in flow entries that controllers will generate
  - Each header has fixed value, or wildcard with a clone flag
- Pending Flow Table in each switch
  - Entries show header values of packets that have already been sent to the controller
  - Entries are generated from Rules and packets
- Generation of Pending Flow Table Entry from Rule and Packet

<table>
<thead>
<tr>
<th>Rule</th>
<th>Dest MAC</th>
<th>Src IP</th>
<th>Dest IP</th>
<th>Proto</th>
<th>Timeout</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>clone true</td>
<td>*</td>
<td>clone true</td>
<td>*</td>
<td>clone false</td>
</tr>
</tbody>
</table>

- Copy values from the packet if wildcarded and clone flag is true
- Keep wildcard if wildcarded and clone flag is false
- Copy fixed values from Rule
Procedure of handling Flow Table Miss

- Controllers set Pending Flow Rules to switches in advance
- When a switch received packets for controllers,
  - Follows the flow chart below

Packet for controllers received

- Matched with Pending Flow Table
  - YES: Forward to the controller through some rate limit mechanism (Meter, etc..)
  - NO: Forward category (less important)

- Matched with Pending Flow Rules
  - YES: Generate Pending Flow Table Entry, Forward to the controller without any limitation
  - NO: Drop

Flow Setup category (important)
Concluding Remarks and Future Work

- Packet-In messages are necessary, but many Packet-Ins are harmful
- Propose a mechanism to pass through only important Packet-Ins
  - One Packet-In message per new flow entry
- Set fields used in flow entries as “Pending Flow Rules” in advance
  - Each field in a rule have fixed value, or wildcard with clone flag
- Record flows in “Pending Flow Table” in switches
  - Entries of table are generated from Rules and Packets
  - Matched packets with Table are filtered out

Future Work
- Design Protocol Extension for Pending Flow Rules/Table
- Implement Prototype Switch, and Evaluate overheads