Layer 2 with Ryu 🐉

In order to create a Layer 2 switch in Ryu, we must get familiar of how to use its API. Ryu requires few generate packages for all its applications.

These are the general and useful packages:

```
from ryu.base import app_manager

from ryu.controller import ofp_event
from ryu.controller.handler import CONFIG_DISPATCHER, MAIN_DISPATCHER
from ryu.controller.handler import set_ev_cls

from ryu.ofproto import ofproto_v1_3

from ryu.lib.packet import packet
from ryu.lib.packet import ethernet
from ryu.lib.packet import ether_types
```

The packages list can be overwhelming, however the most importants ones are listed below with its description.

Package	Description
app_manager	main entry point for the application
set_ev_cls **ofp_event** **Dispatcher**	capture openflow event when openflow packets are received
ofproto_v1_3	OpenFlow version
packet **ethernet** **ether_types**	packet processing library

Create Class

When creating an ryu application class, always pass the `app_manager.RyuApp`.

```
class Layer2Switch(app_manager.RyuApp):
```

Set the OpenFlow version

Set the OpenFlow version that the application will use.

```
OFP_VERSIONS = [ofproto_v1_3.OFP_VERSION]
```

Define class constructor

For the constructor of the class, you must user `super` to inheritance Ryu properties.

```
def __init__(self, *args, **kwargs):
super(SimpleSwitch13, self).__init__(*args, **kwargs)
```

Add Events

In a Ryu controller, you could add events that you would want to listen too. With the use of decocrators in python we can add functionality to the controller: `@set_ev_cls`. For instance, the code below will be trigger for any event at the OpenFlow switch.

```
 @set_ev_cls(ofp_event.EventOFPSwitchFeatures, CONFIG_DISPATCHER)
def switch_features_handler(self, ev):
```

Next, we will add another event to listen to the packets that are being received. With the same functionality of the previous event, this `_packet_in_handler` function will be trigger once packets are being captured.

```
@set_ev_cls(ofp_event.EventOFPPacketIn, MAIN_DISPATCHER)
def _packet_in_handler(self, ev):
```

Run Ryu Application

This section will take in consideration that you know how to use, and run *Ryu* as well as *Mininet*. I will provide installation and getting started instructions.

- Installation
- Getting Started with Ryu

First, you must run the ryu application that hold the layer 2 switch.

```
ryu-manager layer2.py
```

However, we will still need a topology for the controller to have functionality. We would be using `Mininet` to create the different topologies to test.

```
sudo mn mininet-topo/layer2.sh
```

Author:

- **Jesus Minjares**
    - Master of Science in Computer Engineering
      Microsoft Outlook LinkedIn GitHub