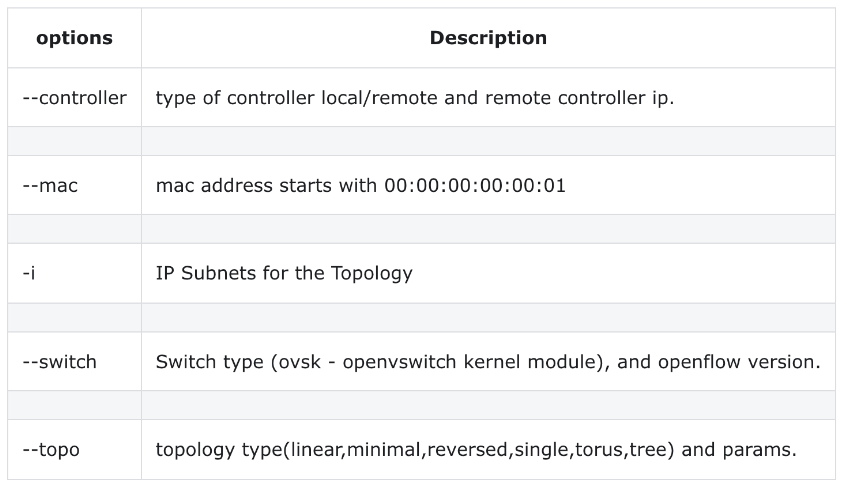
**Build basic topo (mininet):**

*sudo mn --controller=remote,ip=127.0.0.1 --mac --switch=ovsk,protocols=OpenFlow13 --topo=single,4*

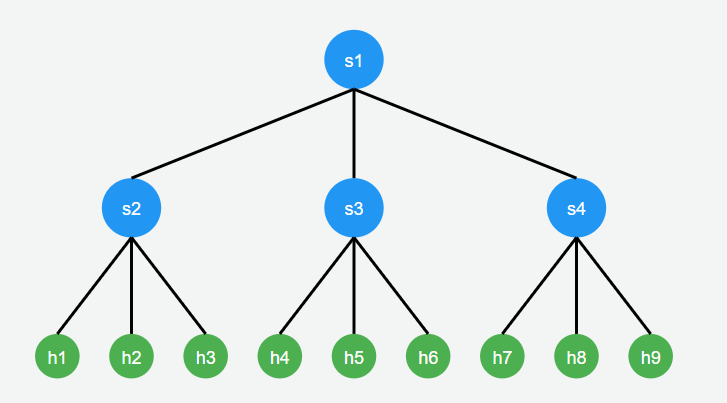
--mac -i (กำหนด ip) 10.1.1.0/24 ….. --topo=linear,4 // Switch 1 ตัว ต่อ Host 1 เครื่อง

-x หลังกำหนด Topo(mininet) = start xterm in all hosts

*sudo mn --controller=remote,ip=127.0.0.1:XXXX --mac … // กำหนด port*



--topo=tree,depth=2,fanout=3



**Start controller:**

*ryu-manager ryu.app.simple\_switch\_13*

*ryu-manager --ofp-tcp-listen-port 6634 ryu.app.simple\_switch\_13 //กำหนด port*

*ryu-manager ryu.app.simple\_switch\_13* ***ryu.app.ofctl\_rest*** *//* *REST API สำหรับการควบคุมและจัดการสวิตช์*

**Mode การทำงานเมื่อ Controller ล่ม**

* Standalone: Switch ทำงานเป็น Traditional Switch แม้ว่า controller down ก็จะยังงานได้ (Flow rules ยังอยู่)
* Secure: Switch พยายามเชื่อมต่อ Controller ตลอด ไม่ Forward packet (default) (ในกรณี flow timeout set)

1. ในโหมด Secure:

* Flow rules เดิมที่มีอยู่ใน table จะยังทำงานได้ (ถ้าไม่ได้กำหนด timeout)
* แต่ไม่สามารถสร้าง Flow rules ใหม่ได้เมื่อ Controller down
* ดังนั้นถ้ามีการสื่อสารระหว่าง hosts คู่ใหม่ที่ไม่เคยมี flow rules มาก่อน จะไม่สามารถสื่อสารกันได้

1. ในโหมด Standalone:

* Flow rules เดิมยังทำงานได้เหมือนกัน
* แต่สามารถสร้าง Flow rules ใหม่ได้เอง โดยทำงานเป็น traditional switch
* hosts คู่ใหม่สามารถสื่อสารกันได้ปกติ

sudo ovs-vsctl set-fail-mode <switch\_name> standalone | secure

**คำสั่งตรวจสอบ:**

*sudo ovs-vsctl show*

*sudo ovs-ofctl -O OpenFlow13 dump-flows s1*

*sudo ovs-ofctl -O OpenFlow13 dump-ports s1*

*sudo ovs-ofctl -O OpenFlow13 show s1*

*sudo ovs-appctl fdb/show s1 // Forwarding Database (FDB) หรือตาราง MAC Address*

**คําสั่งพื้นฐานเกี่ยวกับ arp ประกอบไปด้วย**   
*arp -a. แสดงรายละเอียดข้อมูลใน ARP Table   
arp -d 10.1.1.2 ลบข้อมูลใน ARP Table*

**Clear config mininet:**

*sudo mn -c*

**mininet commands:**

*links, pingall, h1 ping h2, h1 arp -a, net, xterm h1, h1 ifconfig, h1 ip route, link s1 s2 down*

*h1 python -m http.server (http server on host 1, use curl to fetch data: curl [ip]:[port] หรือใช้ ApacheBench),*

nodes, dump

**iperf:**

*h4 iperf -s & : -s หมายถึง การกําหนดให้เป็น Server Mode ,& : Run in background*

*h1 iperf -c h4 -i 10 -P 10 -t 10:*

-c หมายถึง การกําหนดให้เป็น Client Mode   
-i หมายถึง ช่วงเวลาในการรายงานผล เช่น -i 10 คือ รายงานผลทุก 10 วินาที

-t หมายถึง ช่วงเวลาในการทดสอบในหน่วยวินาที -t 30 คือ ช่วงเวลาทั้งหมดในการทดสอบเท่ากับ 30 วินาที   
-b หมายถึง การกําหนดแบนด์วิดท์ เช่น -b 10m คือ แบนด์วิดท์ขนาด 10 Mbps   
-P หมายถึง การเชื่อมต่อแบบขนาน (Parallel) คือการสร้าง Connection เชื่อมต่อระหว่าง Client และ Server ขึ้นมาพร้อมๆ กัน เช่น -P 10 คือ สร้าง Connection ขึ้นมา 10 Connections แล้วส่งข้อมูลออกไปพร้อมๆ กัน

**Ipetf UDP: (ใช้ keyword optional ได้เหมือน tcp)**

*h1 iperf -u -s &*

*h4 iperf -u -c h1*

**ApacheBench(http test traffic)**

*ab -n 500 -c 50* [*http://10.1.1.4:8000/*](http://10.1.1.4:8000/)

*- c 50 หมายถึง จํานวนของ Client ที่ร้องขอไปยัง Server เท่ากับ 50 Clients*

*- n 500 หมายถึง จํานวนการร้องขอจาก Client 1 ตัว ไปยัง Server 500 ครั้ง*

**Curl Methods & REST API FOR Controller (before start controller add this *ryu.app.ofctl\_rest)***

**GET Methos:**

* curl http://localhost:8080/stats/switches | jq //ดู switches ทั้งหมด
* curl http://localhost:8080/stats/desc/1 | jq //ดู Details switch 1
* curl http://localhost:8080/stats/flow/1 | jq //ดู flows switch 1
* curl http://localhost:8080/stats/port/1 | jq //ดู port switch 1

**POST Methos: // ใน windows ใช้ curl.exe และ Double-quotes**

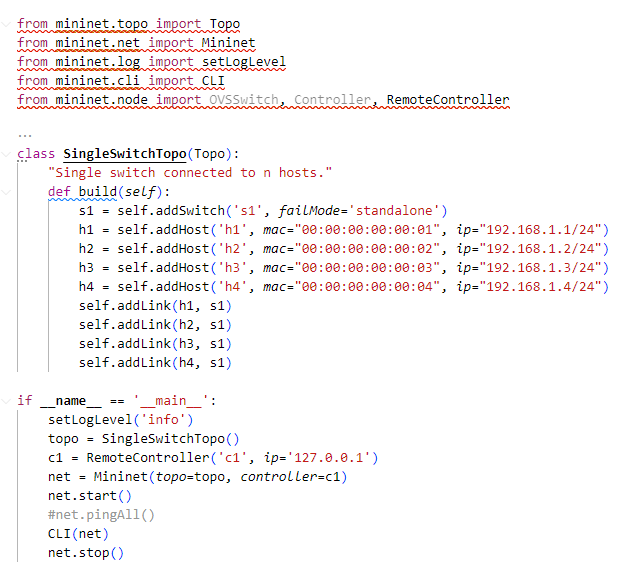
* curl -X POST http://localhost:8080/stats/flowentry/add -d '@hub\_flow.json'//ส่ง flowrule ให้ controller

**DELETE Methods:**

* curl -X DELETE <http://localhost:8080/stats/flowentry/clear/1> //ลบ flow rules เดิมทั้งหมด

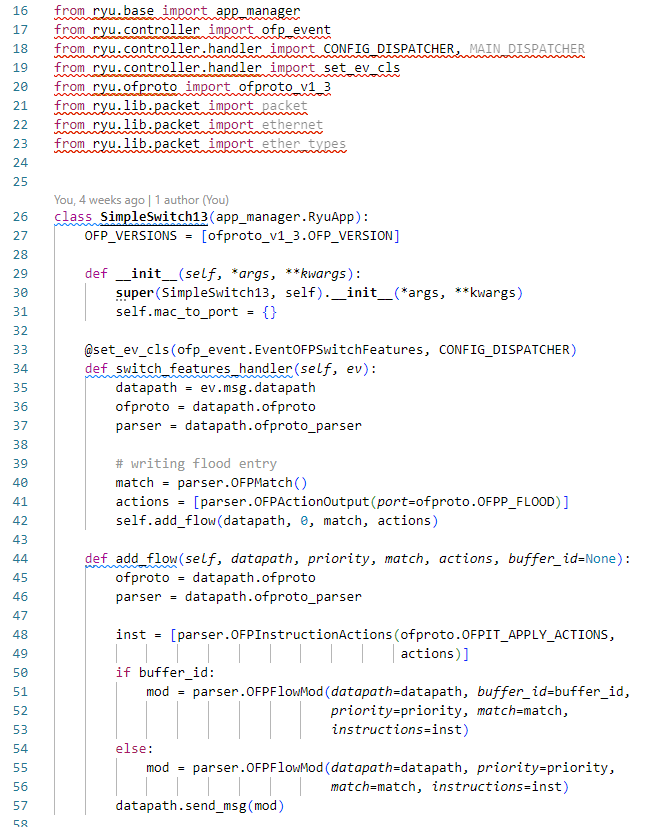
**Code python build topo (Run miniter topo by “sudo python [filename.py])**

1. Simple = single,4

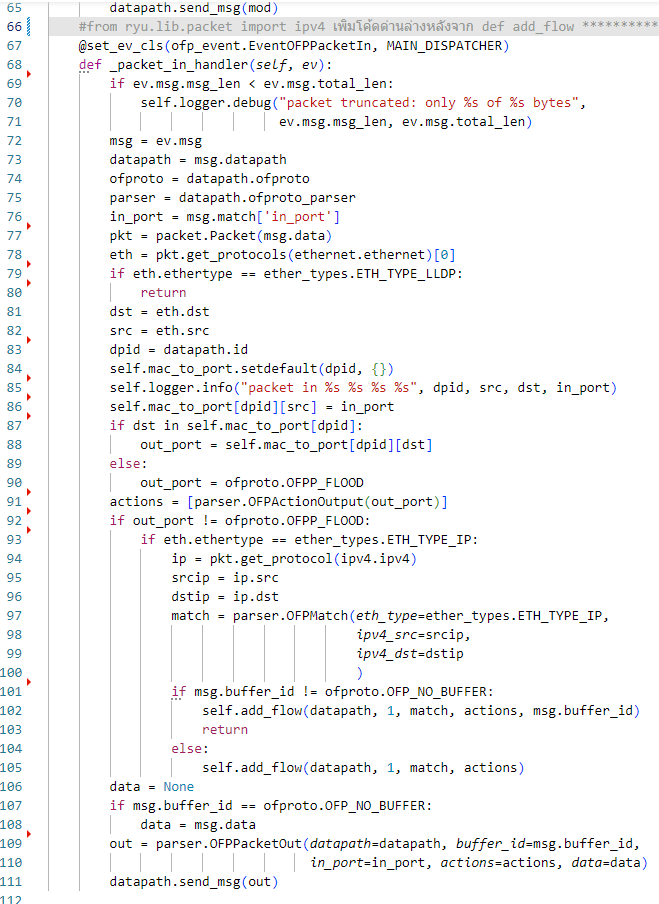


**Code python build controller (Run controller by “ryu-manager [filename.py])**

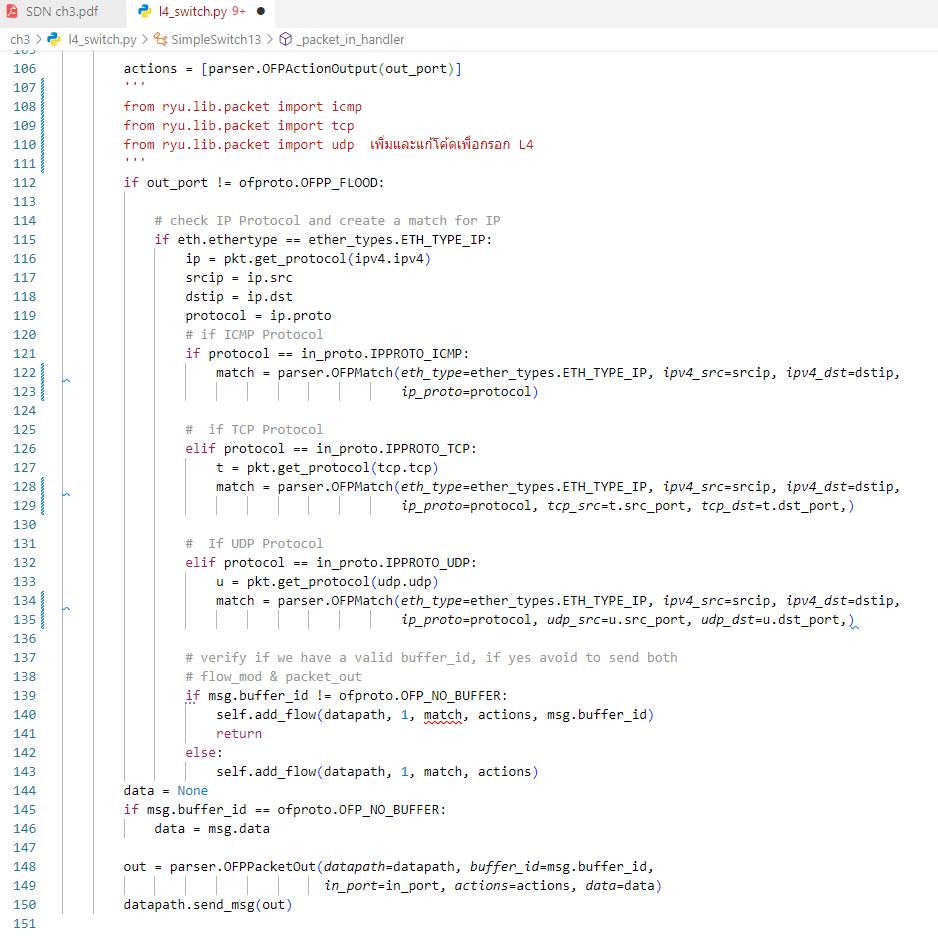
1. Hub controller (For ryu-manager: ryu-manager hub.py)

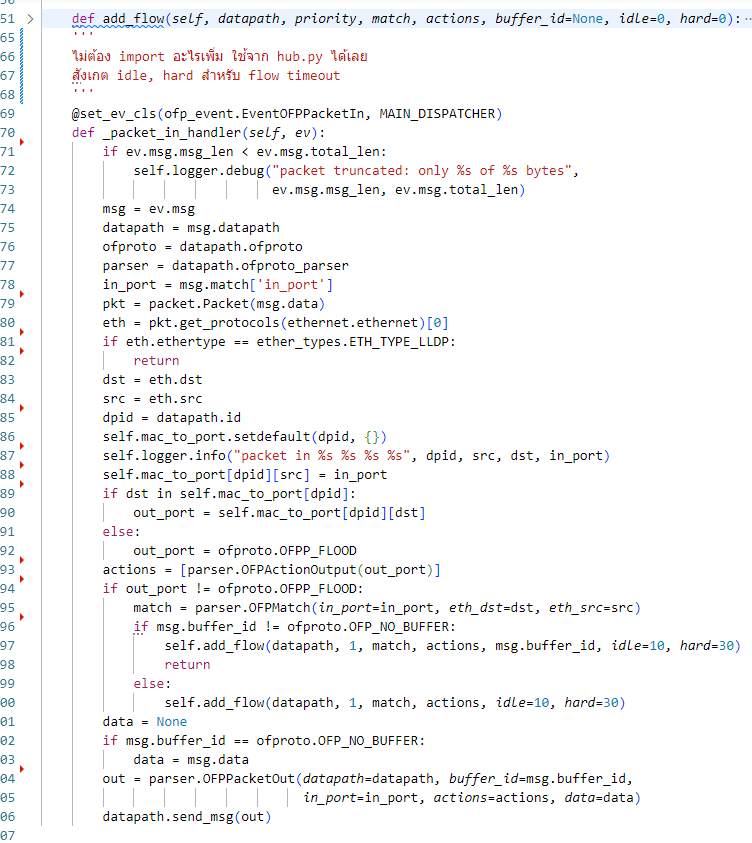


1. L3 switch controller (ryu-manager l3\_switch.py)



1. L4\_Switch (port) สามารถใช้โค้ดจาก L3 แต่แก้โค้ดและ import เพิ่ม



1. Flow timeout : 

**Code for REST API to push in flow of controller (ส่ง flow rule ให้ controller)**

**ความหมายค่า Port ตรง Action ใน Flow rule สามารถใช้คำแทนได้ "port": "CONTROLLER"**

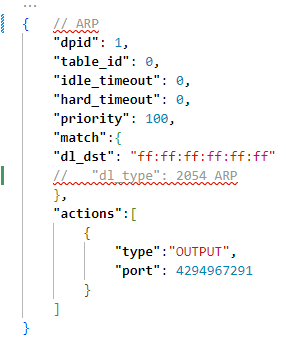
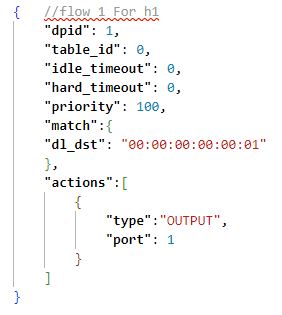
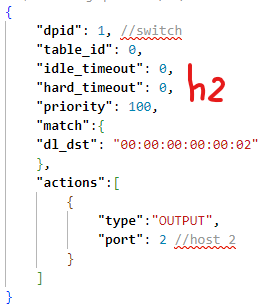
* OFPP\_FLOOD (0xFFFFFFFB) = 4294967291 //packet ออกทุก port ยกเว้น port ที่ packet เข้ามา
* OFPP\_ALL (0xFFFFFFFC) = 4294967292 ส่ง //packet ออกทุก port
* OFPP\_CONTROLLER (0xFFFFFFFD) = 4294967293 //packet ไปที่ controller
* OFPP\_TABLE (0xFFFFFFFA) = 4294967290 //packet เข้า flow table
* OFPP\_IN\_PORT (0xFFFFFFF8) = 4294967288 //packet กลับออกไปทาง port ที่เข้ามา
* OFPP\_NORMAL (0xFFFFFFFA) = 4294967290 //ให้ switch จัดการ packet ตามการทำงานปกติ

OFPP\_TABLE และ OFPP\_NORMAL มีค่าเท่ากัน (4294967290) เพราะทั้งสองค่านี้ใช้แทนพฤติกรรมเดียวกันคือการให้ switch จัดการ packet ตามการทำงานปกติผ่าน flow table

* 1. **Hub.js //วิธีส่งให้ controller ใช้ curl -X POST (stats/flowentry/add) ดูในสรุปด้านบน**

****

* 1. **L2**
  2. **ARP & Flow1 (ของ host1) // ARP ถ้าไม่ได้ให้แก้เป็น 2054 (dl\_type หรือ "eth\_type": 2054**

** ** ****

* 1. **L2**