**ที่ต้องติดตั้งกรณีที่อาจารย์ไม่ลงมาให้**

**sudo apt update**

**sudo apt install python3 python3-pip xterm iperf hping3 net-tools wireshark apache2-utils curl sudo apt install mininet sudo pip3 install ryu**

**sudo pip3 install mininet**

**sudo cp /usr/bin/python3 /usr/bin/python ryu-manager --version sudo mn --version**

**ข้อ 1 สร้าง topology ให้เป็นโดยใช้ python**

ขั้นตอน 1 หาตำแหน่งไฟล์ที่จารย์เตรียมไว้ให้เจอ (ถ้าจารย์เตรียมให้น่าจะอยู่ path นี้)



-หากมี ให้เปิด Editor ตัวไหนก้ได้แล้วไปดู *ขั้นตอน 2*

-หากไม่มี ให้ไป Desktop แล้วสร้าง folder มาเก็บโค้ด

cd ~ เพื่อกลับมา home

cd Desktop/ เพื่อไป Desktop

mkdir midterm650 เพื่อสร้าง folder

cd midterm650 เพื่อไป midterm650

nano topo.py เพื่อเขียนไฟล์ .py (เมื่อจบ ctrl o, enter, ctrl x)

A close up of a folder

Description automatically generated

A black square with white text

Description automatically generated

File Explorer Terminal

A black background with white squares

Description automatically generatedA logo of a company

Description automatically generated

Editor All Apps (Search)

A screenshot of a computer

Description automatically generated

การเปิดไฟล์ด้วย editor แอพ

ไม่ก็ถ้าจารย์เขาเตรียมให้แค่เปิด

อันนี้ก็เจอเลย

A screenshot of a computer

Description automatically generated ขั้นตอน 2 การเขียนโค้ดไฟล์ topo.py

คำสั่งเชื่อมสาย

(อุปกรณ์1,อุปกรณ์2,portของอุปกรณ์1,portของอุปกรณ์2)

s1 port1 เชื่อม s2 port1 (เลข 1 ตรง topo ด้านบน)

s1 port2 เชื่อม s3 port1

ส่วนอื่นของโค้ดไม่ต้องแก้

หากแก้ใน editor แบบแอพ ก้ ctrl s เพื่อ save

หาก nano ก็ ctrl o, enter, ctrl x

คำสั่งเพิ่ม pc ลงใน topo

คำสั่งเพิ่ม switch

ลงใน topo

นี่คือหน้าตา topo ของโค้ดนี้

A screen shot of a computer program

Description automatically generated

**ข้อ 2 ใช้ topo ข้อแรกมาใส่ flow และ group**

ขั้นตอน 1 เปิด terminal 3 แท็บ (ปุ่มเพิ่มแท็บอยู่ซ้ายบน)

A screen shot of a computer

Description automatically generated

ขั้นตอน 2 tab1 ใช้งาน topo.py (ต้องเข้าไป path ที่ไฟล์อยู่) ผ่านคำสั่ง sudo python topo.py

A screenshot of a computer program

Description automatically generated

ข้อควรรู้: (ถ้าเป็น vm ตามนี้ user:test, password:test)

-หากต้องการปิด tab ของ topo ให้พิม mininet>quit

-หากต้องการปิด tab ของ ryu ให้ Ctrl c

-ขั้นตอน 2.5 คิดว่าให้ทำใส่ให้ครบรอบตัวถ้ามี host เพิ่มเช่น

h1 arp ใส่ h2 และ h3

h2 arp ใส่ h1 และ h3

h3 arp ใส่ h1 และ h2



ขั้นตอน 2.5

ขั้นตอน 3 tab2 ใช้งาน ryu ผ่านคำสั่ง ryu-manager ryu.app.ofctl\_rest

A screenshot of a computer code

Description automatically generated

ระวังตรงนี้ให้ดี

ขั้นตอน 4 tab3 ใส่ flow และ group ต่างๆที่จะทำให้ switch ใน topo มันรู้เรื่อง

Add flow ผ่านคำสั่ง curl -X POST http://localhost:8080/stats/flowentry/add -d ‘@ชื่อโฟลว์.json’

Add group ผ่านคำสั่ง curl -X POST http://localhost:8080/stats/groupentry/add -d ‘@ชื่อกรุ๊ป.json’

ไปหา path ที่ไฟล์ .json พวกนั้นอยู่ด้วย (tips: ถ้าสังเกตุดูดีๆ sw แต่ละตัวมี flow เท่าจำนวน port)

A screenshot of a computer

Description automatically generated

A screen shot of a computer code

Description automatically generatedเป็นอันเสร็จขั้นตอนที่นี้ pc ก็สามารถปิงกันได้ตามที่เราเขียนใน .json มาดูตัวอย่าง .json

dpid คือเลขอุปกรณ์switch ในที่นี้คือ sw1

tableid คือหมายเลข table

match คือ packet ตรงกับอะไรไหมถ้าตรงให้ใส่

priority เป็น 100 (หรือตามที่จารย์กำหนด)

match ในที่นี้แปลได้ว่า packet ที่วิ่งเข้า port 1

actions คือการกระทำ ในที่นี้แปลได้ว่า

ให้ packet นั้นๆวิ่งออก port 3

หากสงสัยว่าทำไมต้องเขียนโฟลว์แบบนี้

sw1 เข้าขา 1 ออกขา 3 ให้ย้อนไปดูหน้า topo

ว่าถ้าออกขา 1 หรือขา 2 ตัว packet มันก็จะวน

A screen shot of a computer code

Description automatically generatedA screen shot of a computer code

Description automatically generated

เมื่อเข้าขา 3 ตัว packet จะถือว่าเป็น group 50

ส่วนโค้ดขวาคือ จับตัว packet ที่เป็นgroup 50

ให้โอกาส 50% ไปออก port1 อีก 50% ออก port2

ส่วนโค้ดล่างลองดู dpid และการเข้าออกเป็น

A screen shot of a computer code

Description automatically generatedกรณีศึกษา เทียบกับ s1\_flow1.json

คำสั่งเช็ค flow

sudo ovs-ofctl -O OpenFlow13 dump-flows s1

คำสั่งเช็ค group

sudo ovs-ofctl -O OpenFlow13 dump-groups s1

อยากดูตัวอื่นก็ s2,s3,s4

คำสั่งนี้ไว้พิมใน tab อื่นๆที่ไม่ใช่ topo กับ ryu

**ข้อ 3 ทำ multitble และกำหนด policy(จริงๆแล้วแค่ block การปิงของ pc)**

ขั้นตอน 1 สร้าง topo ในที่นี้เป็น 1 switch, 4 pc

A screenshot of a computer

Description automatically generated

ลองปิงแบบ add policy

สร้าง topo

ลองปิงแบบยังไม่ add flow

ลองปิงแบบ add flow

ขั้นตอน 2 ใช้งาน ryu คำสั่งเดิม

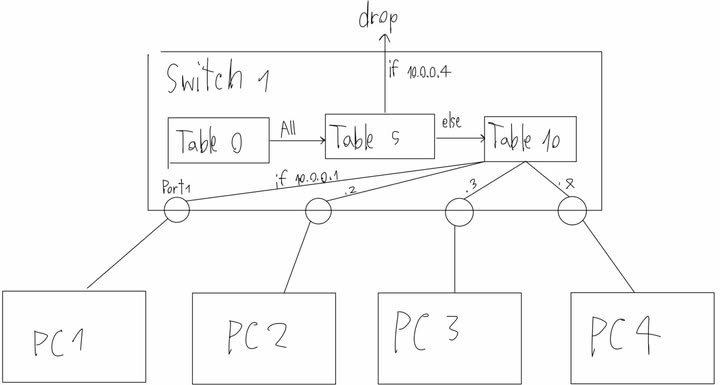
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedขั้นตอน 3 ใส่ flow, table (7 flow แรกเป็น flow ปกติ, ส่วน flow อันเดี่ยวๆคือ policy)

เป็นอันเสร็จขั้นตอนที่นี้ pc ก็สามารถปิงกันได้ตามที่เราเขียนใน .json (ปิงกันได้หมดยกเว้น pc4) มาดูตัวอย่าง .json



A screen shot of a computer code

Description automatically generatedA screen shot of a computer code

Description automatically generatedข้างบนคือแนวคิด ส่วนข้างล่างคือ table ที่เป็น .json ต่างๆ

ถ้าไม่ต้อง match กับอะไรเลยเราจะกำหนดให้ priority ในภาพนี้ก็บอกว่าไป table 10 เลย

เป็น 0 ไม่ต้อง match กับอะไรเลยมีความหมายว่า สังเกตุ table\_id ใต้ dpid เลขเปลี่ยนตาม

ทุกๆ packet หมายเลข table

ในภาพนี้คือทุก packet เข้ามาแล้วไป table 5 ต่อเลย

A screen shot of a computer code

Description automatically generatedA screen shot of a computer code

Description automatically generated

หากเจอ 10.0.0.4 ให้ drop หากเจอ mac เบอร์นี้ให้ออก port นี้

สังเกต priority 100 เพราะมีการ match 2,3,4 ก็ประมาณนี้

actions หน้าตาแบบนี้คือ drop

A screenshot of a computer code

Description automatically generated

อันนี้คือการ flood ทุกเบอร์

ตรง actions เลขนั้นก็หมายถึง flood

**BIGKUMA ปรับปรุง**

**สร้าง topo มี3แบบ Single กับ Linear กับ Tree**

**Terminal ที่ 1 สร้าง Topology**

**อันนี้คือคำสั่งสร้าง topo ด้วย mininet แบบไม่กำหนด IP เริ่มต้นเป็น 10.0.0.1/8**

**Single**

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

**Linear**

sudo mn --controller=remote,ip=127.0.0.1 --mac --swicth=ovsk,protocols=OpenFlow13 --topo=linear,4

**Tree**

sudo mn --controller=remote,ip=127.0.0.1 --mac -i 10.1.1.0/24 --topo=tree,depth=2,fanout=3

**อันนี้คือคำสั่งสร้าง topo ด้วย mininet แบบกำหนด IP**

sudo mn --controller=remote,ip=127.0.0.1 --mac -i [10.1.1.0/24](http://10.1.1.0/24) --switch=ovsk,protocols=OpenFlow13 --topo=linear,4

**อันนี้คือคำสั่งรัน topo ที่สร้างจาก pythonเข้าไปใน pathก่อน**

Sudo python test.py(ชื่อไฟล์)

**Terminal ที่ 2 รันสมอง (ryu)**

**ryu-manager ryu.app.simple\_switch\_13** แบบปกคิดว่าไม่น่าใช้ **ryu-manager ryu.app.ofctl\_rest** แบบกำหนด Flow เอง

**Terminal ที่ 3 ใส่ Flow และ Group เพื่อกำหนดการส่งข้อมูล**

**ก่อนใช้คำสั่ง curl ต้องเข้าไปใน path ของไฟล์ .json ก่อน ซึ่ง Flow และ Group ต้องสร้างเองให้ตรงตามโจทย์**

**curl -X POST** [**http://localhost:8080/stats/flowentry/add -d ‘@ชื่อไฟล์.json**](http://localhost:8080/stats/flowentry/add%20-d%20‘@ชื่อไฟล์.json)**’**

**curl -X POST** [**http://localhost:8080/stats/flowentry/add -d ‘@table20\_flow1.json**](http://localhost:8080/stats/flowentry/add%20-d%20‘@table20_flow1.json)**’**

**curl -X POST** [**http://localhost:8080/stats/groupentry/add**](http://localhost:8080/stats/groupentry/add) **-d ‘@ชื่อไฟล์.json’**

**curl -X POST** [**http://localhost:8080/stats/groupentry/add**](http://localhost:8080/stats/groupentry/add) **-d ‘@s5\_group51.json’**

**Terminal ที่ 4 ใส่ Flow และ Group เพื่อกำหนดการส่งข้อมูล**

**ดู Flow ของ Traffic**

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

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

**ดู Group ของ Traffic**

**sudo ovs-ofctl -O OpenFlow13 dump-groups ชื่อswitch**

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

**Terminal ที่ 1(mininet) คำสั่งทดสอบระบบว่า topo ของเราใช้งาน Flow กับ Group ได้ไหม**

**pingall** ทุก host ping หากัน

**h1 ping h2** ทดสอบการping รหว่างhost

**iperf h1 h2**

**links** ดูการต่อสายของ topo ว่าต่อกันแบบไหน

**Multi Table ต่อไปของอาจารย์อธิบายละเอียด**

**ใช้กับ topoนี้**

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

A close-up of a computer code

Description automatically generated

A close-up of a computer code

Description automatically generated

A close-up of a computer code

Description automatically generated

A close-up of a computer code

Description automatically generated

A close-up of a computer code

Description automatically generated

A close-up of a computer screen

Description automatically generated

A close-up of a computer code

Description automatically generated

A close-up of a computer code

Description automatically generated

A screenshot of a computer

Description automatically generated

**อันนี้โค้ดเผื่อใช้สร้าง Topo นี้เพื่อโจทย์ออก**

A diagram of a network

Description automatically generated

A screen shot of a computer program

Description automatically generated A computer screen shot of a program code

Description automatically generated

**อันนี้ใช้สร้างแบบ 1 Switch 6 Host เผื่อใช้งาน**

A screen shot of a computer

Description automatically generated

**curl -L -o Bigkuma.zip** [**https://github.com/thanormsaksudsee/SDN\_MIDTERM/archive/refs/heads/main.zip**](https://github.com/thanormsaksudsee/SDN_MIDTERM/archive/refs/heads/main.zip)