Lab ศึกษาการทำงานของ Software-defined networking (SDN) และ OpenFlow ด้วย Mininet วิเชียร ปรีดาลัมพะบุตร wpreeda@tot.co.th สถาบันวิชาการทีโอที 21 ตุลาคม 2559

บทนำ

ในการทดลองนี้ เราจะศึกษาการทำงานของ Software-defined networking (SDN), และ OpenFlow โดยเรา จะใช้ simulation software ที่เรียกว่า Mininet ที่สามารถ สร้าง 3 ส่วน (layer) ได้แก่ infrastructure layer ที่ ประกอบด้วย virtual host, virtual link และ virtual Switch, control layer และ application layer โดย Mininet ถูกสร้าง บน server ที่เป็น Virtual Machine หรือ VM ที่ถูกจำลองด้วย VirtualBox บน laptop



รูปแสดงการใช้ Mininet สร้าง SDN architecture

<u>หมายเหตุ</u>ในการทดลองนี้ เราใช้ Mininet สำหรับ สร้าง infrastructure layer ได้แก่ virtual switch และ virtual host ,โดยใน control layer จะใช้ ptcp และ pox controller

<u>วัตถุประสงค์</u>

เพื่อทดสอบและศึกษาการทำงานของ SDN และ OpenFlow โดยใช้ mininet สร้าง 3 ส่วน (layer) ได้แก่ infrastructure layer ที่ประกอบด้วย virtual host, virtual link และ virtual Switch, control layer และ application layer บน laptopที่ทำงานด้วย windows ทั้งนี้จะเป็นการสะดวกเพราะใช้laptopเพียงเครื่องเดียว

<u>อุปกรณ์ที่ใช้ในการทดลอง</u>

- 1. Laptop ที่ทำงานด้วย Winodws 7
- 2. ใช้ VirtualBox เป็น virtual machine software
- 3. ใช้ Mininet VM Images ที่เป็น Mininet 2.2.0 บน Ubuntu version 14.04 -32 bit สำหรับ Windows users ที่ใช้ VirtualBox

<u>ขั้นตอนในการทดลอง</u>

1. ที่ laptop ให้ติดตั้ง VirtualBox

ที่ laptop ให้ติดตั้ง VirtualBox , กรณีของ Windows ท่านสามารถ download ชุด package program ได้จาก website ของ VirtualBox ได้ที่ <u>https://www.virtualbox.org/</u> (ในที่นี้เรา downloadw ไว้ให้ท่านแล้วที่ Desktop ชื่อ SDN Part 1 โดยทำตามขั้นตอนดังนี้:

click 2ครั้งที่ desktop ชื่อ SDN Part 1, click ที่ floder ชื่อMininet และ มา click 2 ครั้ง ที่ไฟล์ VirtualBox-5.2.4... เพื่อติดตั้ง



รอสักครู่จนกว่า จะได้ตามรูปข้างล่าง และให้ click ที่ next



จะได้ตามรูปข้างล่าง และให้ click ที่ next



จะได้ตามรูปข้างล่าง จากนั้นให้ click ที่ next

Oracle VM VirtualBox 5.0.8 Setup			×
Custom Setup			
Select the way you want features to be ins	talled.		
Please choose from the options below:			
Create a shortcut on the desktop			
Create a shortcut in the Quick Launch B	ar		
Register file associations			
			\rangle
Version 5.0.8	< Back	Next >	Cancel

จะได้ตามรูปข้างล่าง จากนั้นให้ click ที่ yes

词 Oracle VM VirtualBox 5.0.8	
	Warning: Network Interfaces Installing the Oracle VM VirtualBox 5.0.8 Networking feature will reset your network connection and temporarily disconnect you from the network. Proceed with installation now?
Version 5.0.8	Yes No

จะได้ตามรูปข้างล่าง จากนั้นให้ click ที่ install

🔡 Oracle VM VirtualBox 5.0.8 Setup	P
Ready to Install	
The Setup Wizard is ready to begin	in the Custom installation.
Click Install to begin the installatio installation settings, click Back. Cli	on. If you want to review or change any of your ick Cancel to exit the wizard.
Version 5.0.8	< Back Install Cancel

จากนั้นให้รอสักครู่ และ ถ้ามีการถามจาก windows เพื่อขอติดตั้งอุปกรณ์ต่างๆ เช่น USB ก็ให้ click ที่ install เพื่ออนุญาต จากนั้นจะได้ตามหน้าจอข้างล่าง และ click ที่ finish

당 Oracle VM VirtualBox 5.0.8 Setup		
	Oracle VM VirtualBox 5.0.8 installation is complete.	
	Click the Finish button to exit the Setup Wizard.	
	Start Oracle VM VirtualBox 5.0.8 after installation	
Version 5.0.8	< Back Finish Car	ncel



จากนั้นปิดการใช้งาน โดย click ที่ เครื่องหมาย × มุมบนขวา

2. ติดตั้ง Mininet

การติดตั้งใช้งาน Mininet ทำได้หลายวิธี ในที่นี้จะขอแนะนำ วิธีที่ง่ายที่สุดเรียกว่า Mininet VM (virtual machine) โดยทำการ download Mininet VM image จาก website <u>http://mininet.org/download/</u>, (ใน ที่นี้ได้ download ไฟล์ไว้ไห้แล้วชื่อ mininet-2.2.0-150106-ubuntu-14.04-server-i386.zip ≈ 800 MB อยู่ใน directoty ชื่อ Intro SDN & NFV บน Desktop)

ให้ทำการ unzip ไฟล์ mininet-2.2.0-150106-ubuntu-14.04-server-i386.zip โดยการ click ไปที่ directory ชื่อ SDN Part 1



ใช้เม้าส์click ขวาไปที่ไฟล์ชื่อ mininet-2.2.0-150106-ubuntu-14.04-server-i386.zip

🔾 🗸 📕 🕨 Intro-S	DN&NFV >	- 4 ₇	Search Intro-SDN&N
Organize 🔻 🛛 🛜 Op	en 🔻 Share with 🔻 Burn New folder		
🔆 Favorites	Name	Date modified	Туре
📃 Desktop	imininet-2.2.1-150420-ubuntu-14.04-serv	6/28/2016 5:42 PM	Compressed (zipp
📕 Downloads	PUTTY PUTTY	12/10/2011 12:35	Application
🖳 Recent Places	🗿 VirtualBox-5.0.22-108108-Win	6/18/2016 1:57 PM	Application
	Xming-6-9-0-31-setup	8/6/2016 11:30 PM	Application

click ที่ Extract All..

		Open Open in new windo Extract All 7-Zip CRC SHA สแกนด้วย Microsoft Open with	w Security Essentials	>	
		Share with Restore previous ve Send to Cut Copy	rsions	•	
Organize Favorites	Share with	Create shortcut Delete Rename Properties		rch Intro-SDN&NF\ I	✓ ✓
Desktop	 mininet-2.2.1-x PUTTY VirtualBox-5.0.2 Xming-6-9-0-3: 	22-108108-Win 1-setup	 0/20/2010 542 FW 12/10/2011 12:35 6/18/2016 1:57 PM 8/6/2016 11:30 PM 	Application Application Application Application	997,136 KB 472 KB 110,729 KB 2,154 KB

Click ที่ extract เพื่อขยายไฟล์ที่ถูกบีบอัดออกมาใช้งานปกติ

G 🏦 Extract Compressed (Zipped) Folders	
Select a Destination and Extract Files	
Files will be extracted to this folder:	
-PC\Desktop\Intro-SDN&NFV\mininet-2.2.1-150420-ubuntu-14.04-server-amd64	Browse
Show extracted files when complete	
	Extract Cancel

Click 2 ครั้งไปที่ directory mn-master-trusty64server-

						-		2.
	Intro >	mininet-2.	2.1-150420-ubuntu-14.04	-server-amdb4	▼ * ĵ	Search mininet-2	2.2.1-150420-ubunt	U)
Organize 🔻	Include in	library 🔻	Share with B	New folder	110-1	T		•
Favorites		Name	earter truth 64 conver 150	Uater	16 11.29 AM	Type File folder	Size	
Download	ls	Ja mn-r	naster-trusty04server-130	420-00-11 0/3/2	IVIA 00,11,00 AIVI	rife tolder		

จะเห็น ไฟล์ mininet 2 ไฟล์ ที่จะถูกนำไปสร้างเป็น Virtual Machine (VM) ซึ่งก็คือ server ที่ใช้ ubuntu 14.04 เป็น operating system พร้อมกับ ติดตั้ง mininet โดยเราจะสร้าง VM ด้วย VirtualBox

Solution → Solutio	inet-2.2	.1 ▶ mn-master-tru	sty64server-150420-00-11	-02 - 47	Search mn-master-trus	tyб4server-150.
Organize 🔻 Inclu	ude in li	brary 🔻 Share with	▼ Burn New fo	older	=	-
☆ Favorites	-	Name	^	Date modified	Туре	Size
🧮 Desktop	=	💗 mininet-2.2.1-15042	20-ubuntu-14.04-serv	8/9/2016 11:38 AM	Open Virtualizatio	4 KB
〕 Downloads		隊 mininet-vm-x86_64	l -	8/9/2016 11:38 AM	Virtual Machine Di	2,845,504 KB

เปิดการใช้งาน VirtualBox โดย click 2 ครั้ง ที่รูป Oracle VM VirtualBox



จากนั้น click ที่ menu File แล้ว click เลือก Import Appliance ตามรูป

ſ	3r	acle VM VirtualBox Manager		
ra	File	Machine Help		
l	S	Preferences	Ctrl+G	
L	n	Import Appliance	Ctrl+I	
L	R	Export Appliance	Ctrl+E	ral
l	6 7	Virtual Media Manager Network Operations Manager	Ctrl+D	base System: Ubuntu (32-bit)
L	9	Check for Updates		em
		Reset All Warnings		ry: 128 MB



จากนั้น ใช้ เม้าส์ชี้ไปที่ สัญญลักษณ์รูปแฟ้ม เพื่อเลือก virtual appliance file เพื่อ import เข้ามาใช้งาน

Click ไปที่ desktop

	🧊 Please choose a virtual ap	pliance file to import		x			
Appl	Search Documents > • • •						
Virtua (OVF)	Organize 🔻 New folde	:= ▼					
	Favorites	Documents library Includes: 2 locations	Arrange by:	Folder 🔻			
	🚺 Downloads 📃 Recent Places 🔤	Name	Date modified	Туре			

Click ไปที่ directory Intro-SDN&NFV



Click ไปที่ directory mininet-2.2.0-150106-ubuntu-14.04-serv...

Í	Please choose a virtual ap	pliance file to import		-	
	G ♥ ICT-SDN-	Feb-59 🕨	▼ ⁴ 9	Search ICT-SDN	V-Feb-59
l	Organize 🔻 New folde	r /	>		≣ ▼ 🗍
ſ	🔶 Favorites	Name	Date modified	Туре	Size
l	Nesktop	🎉 mininet-2.2.0-150106-ubuntu-14.04-serv	1/6/2016 10:32 AM	File folder	
	Downloads				

Click ไปที่ directoty ย่อยชื่อ mn-2.20-trusty32server-150106-02-57-00

 Please choose a virtual appliance file to import VICT-SDN-F > mininet-2.2.0-150106-ubuntu-14.04-server-i386 > VICT-SDN-F > mininet-2.2.0 								
Organize 🔻 New folder								
★ Favorites Name	Date modified Type Size							
Desktop	5-02-57-00 1/6/2016 10:36 AM File folder							

จากนั้น click ไปที่ชื่อไฟล์ mininet-2.2.0-150106-ubuntu-14.04-ser จากนั้น click ที่ open

ſ	🔆 Favorites	^	N	ame	Date modified	Туре	Si
4	ktop 📃 Desktop		-	mininet-2.2.0-150106-ubuntu-14.04-serv	1/6/2015 3:05 AM	Open Virtualizatio	
ſ	🐌 Downloads		_	\sim			
ų	🔚 Recent Places	-		2			
h		=					
	🥽 Libraries						
h	Documents						
	🌙 Music						
Ľ	Pictures						
L	Videos						
L					^		
L	💷 Computer	Ψ.	•			.	
L		File na	me:	mininet-2.2.0-150106-ubuntu-14.04-server-i3	86 🗸	pen Virtualization Fe	orma
					2	Open	Са

Click ที่ next



Click ที่ import

Description	Configuration
Virtual System 1	
🍪 Name	Mininet-VM
📃 Guest OS Type	쭏 Ubuntu (32-bit)
CPU	1
RAM	1024 MB
🔗 USB Controller	
🛃 Network Adapter	Intel PRO/1000 M Server (82545EM)
Reinitialize the MAC address	of all network cards
	Restore Defaults Import Cano

รอจนกว่าการติดตั้งจะแล้วเสร็จ



เมื่อติดตั้งแล้วเสร็จจะได้ตามรูป ขณะนี้ virtual machine (VM) ได้ถูกสร้างแล้วชื่อ Mininet-VM



จากนั้นเพื่อให้สามารถเข้าใช้งาน virtual machine ได้ง่ายขึ้น ให้ติดตั้งเพิ่มเติมดังนี้, เนื่องจากอาจมีการติดตั้ง VM หลายชนิด ดังนั้นให้เลือกที่ (1) Mininet-VM, (2) setting, (3) network, (4) adapter 2, (5) check เลือก ค่า enable network adapter , (6) เลือกค่า Host-only Adapter และ (7) เลือกค่า OK



หมายเหตุ: Adapter 1 โดย default ถูก set ให้ทำหน้าที่ NAT เพื่อให้ VM สามารถเข้าถึง internet

Adapter 2 ถูกset เป็น host only interface เพื่อให้ program อื่นๆ เช่น putty ที่ run บน คอมพิวเตอร์จริง (host computer) ติดต่อกับ VM หรือ virtual machine ด้วย SSH (Secure Shell) เป็น รูปแบบการรับส่งข้อมูลระหว่างclient กับ serverโดยข้อมูลมีการเข้ารหัสลับ.

3. เตรียมความพร้อม การใช้งาน mininet

ในบางครั้งเราอาจติดตั้ง VM หลายชนิดไว้ใช้งาน ในที่นี้ มี VM ชนิดเดียว, ให้เลือก VM ชื่อ Mininet-VM , จากนั้น click ที่ start

🐧 Ora	acle VM Vi	rtualBox	Manager	\wedge				
File	Machine	Help						
E MA	6	\diamond	\Rightarrow					🙆 Det
New	Settings	Discard	Start					
Mininet-VM				General		📃 Pre	view	
			$\overline{\langle}$	Na	ame: perating System:	Mininet-VM Ubuntu (32-bit)		

รอจนกว่า virtual machine (VM) จะทำงานเสร็จ ,

หน้าจอ login จะมองเห็นไม่ชัด ให้ click เครื่องหมาย X ที่อยู่มุมขวาของ แถบเมนูที่มาปิดทับ login

พิมพ์ ค่า logi n	: mininet, Password: mininet
Mininet-V	/M [Running] - Oracle VM VirtualBox
File Machi	ine View Input Devices Help
mininet-um Password: Last login: Welcome to	login: mininet Fri Jan 8 22:23:34 PST 2016 on tty1 Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic i686)
ใช้คำสั่ง ifconf	ig เพื่อดูค่า ip address ของเครื่อง VM มีค่า 192.168.56.101
mininet@mir	ninet-vm:\$ ifconfie
eth0 L i U R T C R	ink encap:Ethernet HWardr 08:00:27:48:e7:ca inet addr:192.168.56.101 Bcast:192.168.56.255 Mask:255.255.255.0 JP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1 XX packets:1 errors:0 dropped:0 overruns:0 frame:0 XX packets:1 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 XX bytes:590 (590.0 B) TX bytes:342 (342.0 B)
lo L i U	link encap:Local Loopback inet addr:127.0.0.1 Mask:255.0.0.0 JP LOOPBACK RUNNING MTU:65536 Metric:1

และให้พิมพ์คำสั่ง sudu dhclient eth1 เพื่อกำหนด ip address ให้กับ เครื่อง VM สามารถติดต่อ internet

mininet@mininet-vm:~\$ sudo dhclient eth1

(โดยสามารถตรวจสอบด้วยคำสั่ง mininet@mininet-vm:\$ ifconfig)

จะพบว่าเครื่องVM ที่ eth0 มีค่า ip address :192.168.56.101 ส่วนที่ eth1 มีค่า ip address :10.0.2.15 <u>หมายเหตุ</u> หน้าจอของ virtual machine นี้ที่ผ่าน console ของ VirtualBox ต่อจากนี้ไปเราจะย่อ และไม่ใช้ งานอีกต่อไป โดย click ที่เครื่องหมาย – ที่มุมขวาบน, แต่ยังไม่ปิดการใช้งาน

<u>ข้อสังเกตุ</u>

ก. การสั่ง execute commands หรือ ให้ คำสั่งทำงานนั้น จะอยู่ในรูป

mininet@mininet-vm:~\$ sudo command (ในที่นี้ command คือ คำสั่งต่าง ๆ เช่น dhclient, nano, ovs-ofctl เป็นต้น)

นั่นคือจะใช้ sudo นำหน้าคำสั่งต่างๆที่จะใช้ ทั้งนี้ sudo หมายถึง user เข้าใช้งานด้วยชื่อ root และ การ ได้รับสิทธิใช้งานสูงสุด

ข. เครื่องหมายบน command line มีข้อแตกต่างกันดังนี้:

Mininet> command (คำสั่ง เช่น pingall, sudo mn, iperf ดังจะได้กล่าวต่อไป)

เป็นคำสั่งหรือ commands ที่ถูกใช้นำหน้า ด้วย mininet> จะถูก run บน mininet console

\$ command

Command ที่ถูกใช้นำหน้าด้วย \$ จะถูก run ที่ VM ภายใต้สิทธิการใช้งาน user ธรรมดา

command

Command ที่ถูกใช้นำหน้าด้วย # จะถูก run ที่ VM (virtual machine) ภายใต้สิทธิการใช้งาน sudo

3.1 การใช้ mininet บน VM ด้วย SSH terminal (หรือ ด้วยputty ที่ใช้ SSH สำหรับ connection)

ตามที่ได้กล่าวไว้ก่อนหน้านี้ว่าการใช้ mininet บน console ของ VirtualBox โดยตรงค่อนข้างยุ่งยาก เช่น ไม่สามารถใช้งาน copy-and-paste ไป มา ระหว่าง host computer กับ VM, ดังนั้นเราจะใช้งาน mininet บน VM ต่อจากนี้ไปเราจะใช้ด้วย SSH terminal นั่นคือใช้ด้วย putty โดย click ที่ putty ตามรูป



จะได้ ตามรูปข้างล่าง จากนั้น (1) เลือก session, (2) พิมพ์ค่า IP address ของ VM คือ 192.168.56.101, (3) เลือก connection แบบ SSH (ค่า default ของ port คือ 22), และ (4) click ที่ Open

Puttry Configuration (2) Categony: (2) Basic options for your Put I TY session Specify the destination you van to connect to Host Name (or IP address) Port 192.168.56.101 22 Connection type: Behaviour Translation Selection Colours Connection Data Proxy Telnet Rlogin SSH Serial Close window on exit: Aways Never Only on clean exit	(1)	
Category: (2) Basic options for your Pull TY session Logging Terminal Keyboard Bell Features Window Appearance Behaviour Translation Selection Colours Connection Data Proxy Telnet Rlogin SSH Setial	Reputry Configuration	
Basic options for your Pu I TY session Specify the destination you want to connect to Keyboard Bell Features Window Appearance Behaviour Translation Selection Colours Proxy Telnet Rlogin SSH Serial	Category:	(2)
 Logging Terminal Keyboard Bell Features Window Appearance Behaviour Translation Selection Colours Default Settings Load Save Default Settings Load Save Delete Rlogin SSH Serial 	Session	Basic options for your Pu I TY session
 □ Terminal □ Keyboard □ Bell □ Features □ Window □ Appearance □ Behaviour □ Translation □ Selection □ Colours □ Data □ Proxy □ Telnet □ Raw □ Telnet □ Rogin □ SSH □ SSH □ Close window on exit: ○ Always ○ Never ○ Only on clean exit 	Logging	Specify the destination you want to connect to
 Keyboard Bell Features Window Appearance Behaviour Translation Selection Colours Connection Data Proxy Telnet Rlogin SSH Serial 		Host Name (or IP address) Port
Bell Features Features Features Appearance Appearance Appearance Behaviour Translation Selection Colours Data Data Default Settings Proxy Telnet Rlogin Ssh Serial Close window on exit: Appearance Gold to the stored session	···· Keyboard	
Features	···· Bell	192.168.36.101
 Window Appearance Behaviour Translation Selection Colours Connection Data Proxy Telnet Rlogin SSH Save Default Settings Load Save Delete Close window on exit: Always Never Only on clean exit 	- Features	Connection type:
Appearance Behaviour Translation Selection Colours Ornection Data Proxy Telnet Rlogin SSH Serial Close window on exit: Always Never Only on clean exit	Window	🔘 Raw 🔘 Telnet 🔘 Rlogin 💿 SSH 🕥 Serial
	···· Appearance	l and save or delete a stored session
Translation Selection Colours Colours Data Data Proxy Telnet Rlogin Serial Serial Close window on exit: Always Never Only on clean exit	···· Behaviour	
Selection Colours Connection Outro Data Proxy Telnet Rlogin SSH Serial Close window on exit: Always Never Only on clean exit	···· Translation	Saved Sessions \sim (3)
Colours Connection Default Settings Load Save Delete Delete (4) Close window on exit: Always ● Never Only on clean exit	Selection	
 □- Connection □ Data □ Proxy □ Telnet □ Rlogin □ SSH □ Serial Close window on exit: ○ Always ○ Never ⓒ Only on clean exit 	Colours	Default Settings
Close window on exit: Always ○ Never ◎ Only on clean exit		
Proxy Telnet Rlogin SSH Close window on exit: Always ○ Never ◎ Only on clean exit	···· Data	Save
Telnet Rlogin SSH (4) Close window on exit: Always ○ Never ◎ Only on clean exit	···· Proxy	
Rlogin SSH Serial Close window on exit:	···· Telnet	Delete
(4) Close window on exit: Always Never Only on clean exit	···· Rlogin	
Close window on exit: Always Never Only on clean exit	⊞. SSH	(4)
C Always Never O Only on clean exit	····· Serial	Close window on exit:
		🔘 Always 💿 Never 💿 Only on clean exit
About Open Cancel	About	Open Cancel

จะได้ว่าเราสามารถใช้งาน mininet บน VM ได้ตามรูปข้างล่าง

```
mininet@mininet-vm: ~
login as: mininet
mininet@192.168.56.101's password:
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic i686)
* Documentation: https://help.ubuntu.com/
Last login: Wed Jan 6 15:44:29 2016
mininet@mininet-vm:~$
```

นั่นคือต่อจากนี้ไปการใช้งานใดๆบน mininet จะถูกใช้งานผ่านหน้าจอนี้เราเรียกว่าผ่านทาง SSH terminal (หรือ ใช้ผ่าน PUTTY)

3.2 การเตรียมการตรวจจับข่าวสารว่าเป็นชนิดไหน ที่จุดต่างๆ (ตามลูกศรตามรูปข้างบน) ด้วย Wireshark

ก. ติดตั้ง Xming server , click 2 ครั้งที่ ไฟล์ชื่อ Xming-6-9-0-31-setup ที่อยู่ใน directory ชื่อ Intro SDN&NFV จากนั้นทำการติดตั้งโดยใช้ค่า default ทั้งหมด, จากนั้นเปิดการใช้งาน Xming server ตามรูป

โดยการทำดังนี้ (1) click ที่ รูป windows, (2) click ที่รูป Xming, (3) click ที่รูป, (4) ถ้ามี logo ของ Xming แสดงว่า ขณะนี้ Xming server กำลังทำงานอยู่

Calculator Calculator WinSCP (2) Sticky Notes Xming Xming	Control Paner Devices and Printers Default Programs Help and Support	(4)
Search programs and files	Shut down 🕨	Customize
		EN (3)

ข. การสร้าง SSH terminal ที่มีการ <u>enable X11 forwarding</u>

ให้เปิดการใช้งาน putty ขึ้นใหม่ บน หน้าจอ desktop, เพื่อเปิดการทำงาน (enable) x11 forwarding โดยทำตามขั้นตอนดังนี้: (1) click ที่ SSH, (2) click ที่ X11, (3) check ที่ Enable X11 forwarding



จากนั้นใช้ (1) ใช้เม้าส์เลื่อน ที่แถบเลื่อนหน้าจอลงมา, (2) click ที่ session , (3) พิมพ์ค่า IP address ของ VM คือ 192.168.56.101 (4) เลือก SSH จากนั้น click open



จากนั้น พิมพ์ login : mininet , password: mininet

ให้ พิมพ์ mininet@mininet-vm:~\$ sudo wireshark & จากนั้น กด enter ได้ตามรูปข้างล่าง

login as: mininet mininet@192.168.56.101's password: Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic i686) * Documentation: https://help.ubuntu.com/ Last login: Wed Jan 6 21:27:48 2016 /usr/bin/xauth: file /home/mininet/.Xauthority does not exist mininet@mininet-vm:~\$ sudo wireshark &

แม้ว่าได้มีข้อความ Lua: error during loading ก็ตาม ให้ click ok



จากนั้นจะได้ข้อความเตือนอีกครั้งให้ click ok

Xming X	
<u>a</u>	Running as user "root" and group "root". This could be dangerous. If you're running Wireshark this way in order to perform live capture, you may want to be aware that there is a better way documented at
	/usr/share/doc/wireshark-common/README.Debian
	Don't show this message again.
	ок

จะได้ ตามรูป ให้ (1) click ที่ capture, (2) เลื่อนเม้าส์มาที่click ที่ interface



ใช้เม้าส์เลื่อนที่ แถบเลื่อนเพื่อเลือกตรวจวัด(capture) ข่าวสารที่ interface ที่ต้องการ ในรูปเป็นตัวอย่างการเลือกวัดที่ lo (loopback ของ VM)

-		Wires	hark: C	apture In	terfaces	_	_				x
e :di				any			no	ne	3555	27	
1			1	lo			127.(0.0.1	1846	18	<u>×</u>
-	Help				Start	Stop		Opti	ons	Close	

4. แนะนำ openflow

OpenFlow เป็น protocol ที่กล่าวถึงขั้นตอน,รูปแบบ,การรับส่งข้อมูลการเชื่อมต่อแบบมาตราฐาน (open interface) ระหว่าง SDN controller กับ SDN switch สำหรับการควบคุมการทำงานของ switches, routers, และ access points ที่อยู่ระยะไกล ด้วย Flow table ทั้งนี้ด้วยแนวคิดเดิมการควบคุมการส่งผ่าน, หรือการ forward data ของ SDN switch จะอยู่ในตัวอุปกรณ์เหล่านั้น แต่ด้วยแนวใหม่ของsdnที่ต้องการแยกการควบคุม การส่งผ่าน (control plane) ของ switch ออกจากตัว switch ที่มีไว้เพื่อรับส่งข้อมูลอย่างเดียว (data plane) โดยทำงานตามFlow table ที่ถูกประมวลผลโดย SDN controller

5. ใช้งาน Mininet

ในการศึกษาการทำงานของ OpenFlow จำเป็นต้องสร้างการเชื่อมต่อของอุปกรณ์หลายชนิดที่ประกอบกันขึ้น ในโครงข่าย ได้แก่ virtual hosts, links, และ switches เป็นต้น และ เนื่องจาก เราจะสร้างโครงข่าย ที่ไม่ได้ใช้ อุปกรณ์จริง แต่เราใช้ simulation software ชื่อ Mininet ถูกจำลองด้วย VirtualBox ทำงานบน Laptop เครื่อง เดียว ในที่นี้ Mininet สร้างโครงข่ายเสมือนที่ประกอบด้วย virtual links, virtual hosts, virtual openflow switches.



mininet@mininet-vm:~\$ sudo mn help บอกรูปแบบการใช้งานคำสั่ง mn

ในการศึกษาของเราจะสร้างโครงข่ายเสมือนที่ประกอบด้วย 3 virtual hosts, 1 OpenFlow switch, 1 OpenFlow controller ดังรูปในหน้าที่แล้ว ที่เราสามารถสร้างโครงข่ายเสมือนนี้ ผ่านทางSSH terminal ด้วยคำสั่งดังนี้: อธิบายการใช้คำสั่ง ในแต่ละส่วนย่อยดังนี้

sudo mn เริ่มใช้งาน Mininet และ การ run Mininet ต้องใช้ sudo เพื่อใช้สิทธิของ root

- topo single, 3 บอก mininet ว่า ให้สร้างการเชื่อมต่อ (topology)ของ switch แบบ single หรือ ชั้น เดียว โดย switch เชื่อมต่อไปยัง 3 virtual hosts
- mac บอก Mininet ให้กำหนด แต่ละvirtual host มีค่า mac address ที่เรียงลำดับกัน และตรงกันกับ IP address ของแต่ละ virtual host เอง
- - switch ovsk ใช้ Open vSwitch ใน kernel mode ของ switch
- controller remote เป็นการบอก mininet ว่า OpenFlow switches แต่ละตัว จะติดต่อกับ SDN controller ที่อยู่ภายนอก Mininet
 โครงข่ายจำลองที่ถูกสร้างขึ้นด้วย mininet

ยังไม่ถูกเชื่อมต่อกับ controller

ผลการใช้คำสั่งข้างบนได้ผลดังนี้ :

```
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6633
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1)
*** Configuring hosts
h1 h2 h3
*** Starting controller
c0
*** Starting 1 switches
s1
*** Starting CLI:
                              หน้าจอของ SSH terminal ที่เริ่มต้นด้วย mininet> เรียกว่า
mininet>
```

Mininet console

5.1 คำสั่งเบื้องต้นของ Mininet

เนื่องจากเราจะใช้งาน Mininet ในการทดลองนี้เป็นส่วนใหญ่ จึงนับว่าเป็นประโยชน์ที่จะเรียนรู้คำสั่งใช้งาน เฉพาะของ Mininet พอสมควรดังนี้, **โดยคำสั่งต่างๆจะถูก run บน Mininet console** ดูรายการของ nodes ที่ถูกสร้างขึ้น, ใช้คำสั่ง nodes

```
mininet> nodes
available nodes are:
c0 h1 h2 h3 s1
```

C0 => OpenFlow controller

h1, h2, h3 => virtual host1, virtual host2, virtual host3

s1=> virtual OpenFlow switch

- ดู รายการของคำสั่ง ที่สามารใช้งานได้ , ใช้คำสั่ง help

```
mininet> help
Documented commands (type help <topic>):
dpctl help link noecho pingpair py swite
dump intfs links pingall ports sh x
exit iperf net pingellf.
                                                              switch
                        pingall ports sh
pingallfull px sour
                                                      source xterm
You may also send a command to a node using:
 <node> command {args}
For example:
 mininet> h1 ifconfig
The interpreter automatically substitutes IP addresses
for node names when a node is the first arg, so commands
like
 mininet> h2 ping h3
should work.
Some character-oriented interactive commands require
noecho:
 mininet> noecho h2 vi foo.py
However, starting up an xterm/gterm is generally better:
 mininet> xterm h2
mininet>
```

 ทำการ run คำสั่งเดียวที่ node, ให้ใส่ชื่อ node ไว้หน้าคำสั่ง ตัวอย่างการใช้ เช่น ต้องการทราบ IP address ของ virtual host 1, ให้ใช้คำสั่ง h1 ifconfig

mininet>	h1 ifconfig
h1-eth0	Link encap:Ethernet HWaddr 00:00:00:00:00:01
	inet addr:10.0.0.1 Bcast:10.255.255.255 Mask:255.0.0.0
	inet6 addr: fe80::200:ff:fe00:1/64 Scope:Link
	UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
	RX packets:0 errors:0 dropped:0 overruns:0 frame:0
	TX packets:9 errors:0 dropped:0 overruns:0 carrier:0
	collisions:0 txqueuelen:1000
	RX bytes:0 (0.0 B) TX bytes:738 (738.0 B)
10	Link encap:Local Loopback
	inet addr:127.0.0.1 Mask:255.0.0.0
	inet6 addr: ::1/128 Scope:Host
	UP LOOPBACK RUNNING MTU:65536 Metric:1
	RX packets:0 errors:0 dropped:0 overruns:0 frame:0
	TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
	collisions:0 txqueuelen:0
	RX bytes: 0 (0 0 B) TX bytes: 0 (0 0 B)

 มีทางเลือกให้ใช้ในบางกรณี ที่ต้องการ run คำสั่งโต้ตอบ กับvirtual host เพื่อตรวจหาข้อบกพร่อง ที่ output เราสามารถใช้คำสั่ง xterm กับ virtual host เดียวหรือมากกว่า เช่น xterm h1 h2

(โดยก่อนใช้ คำสั่ง xterm บน SSH terminal จะต้องทำตามข้อ 3.2 ก. เปิดการทำงาน Xming server, และข้อ 3.2 ข. การสร้าง SSH terminal ที่มีการ Enable X11 forwarding)

mininet> xterm h1 h2

ได้ผลตามรูปข้างล่างคือ มี หน้าจอ ปรากฎขึ้นมา 2 หน้าจอ (เราป้อนค่า h1 h2) เพื่อให้เราใช้คำสั่งบน virtual host คือ h1 และ h2 ได้โดยตรง และหน้าจอทั้งสองนี้เรียกว่า **xterm terminal**



จากตัวอย่างเราต้องการทราบ IP address ของ h1และ h2 โดยใช้คำสั่ง ifconfig , การเรียกใช้งาน หน้าจอของ h1, h2 จะถูกเรียกใช้ โดย click ที่ รูปของ Xming server, ในที่นี้ท่านสามารถปิดการใช้งาน ของหน้าจอ h1 และ h2 ได้ เนื่องจากคำสั่งส่วนใหญ่เราจะใช้ผ่าน Mininet console

 ในบางครั้ง ถ้า Mininet ทำงานไม่ถูกต้อง และจำเป็นต้อง restart, ในขั้นแรกให้ท่าน ออกจากการใช้
 Mininet (โดยการใช้ exit command หรือ control-D) และพยายามลบสถานะและการประมวลผลด้วย การใช้คำสั่ง sudo mn –c

```
mininet@mininet-vm:~$ sudo mn -c
*** Removing excess controllers/ofprotocols/ofdatapaths/pings/noxes
killall controller ofprotocol ofdatapath ping nox core lt-nox core ovs-openflowd
ovs-controller udpbwtest mnexec ivs 2> /dev/null
killall -9 controller ofprotocol ofdatapath ping nox core lt-nox core ovs-openfl
owd ovs-controller udpbwtest mnexec ivs 2> /dev/null
pkill -9 -f "sudo mnexec"
*** Removing junk from /tmp
rm -f /tmp/vconn* /tmp/vlogs* /tmp/*.out /tmp/*.log
*** Removing old X11 tunnels
*** Removing excess kernel datapaths
ps ax | egrep -o 'dp[0-9]+' | sed 's/dp/nl:/'
*** Removing OVS datapathsovs-vsctl --timeout=1 list-br
ovs-vsctl --timeout=1 list-br
*** Removing all links of the pattern foo-ethX
ip link show | egrep -o '([- .[:alnum:]]+-eth[[:digit:]]+)'
*** Killing stale mininet node processes
pkill -9 -f mininet:
*** Shutting down stale tunnels
pkill -9 -f Tunnel=Ethernet
pkill -9 -f .ssh/mn
rm -f ~/.ssh/mn/*
*** Cleanup complete.
```

6. ตัวอย่าง การใช้ ovs-ofctl

(ขออ้างถึงขณะนี้เรายังคง เปิดการใช้งาน Mininet บน VM ด้วย putty ผ่าน SSH อยู่หนึ่งหน้าจอ ซึ่งมีการ run คำสั่ง **sudo** mn -topo=single,3 --mac -switch=ovsk -controller=remote ใช้งานอยู่ตามในข้อ5)

ovs-ofctl เป็น คำสั่ง ที่อำนวยความสะดวกสำหรับสั่ง (configure) ให้ Open vSwitch ทำงานตามที่ต้องการ หรือ สอบถาม สถานะการทำงานของ Open vSwitch และเช่น สามารถสั่งให้ Open vswitchทำงานตามที่ ต้องการโดย ส่งผ่านข้อมูลไปที่ตารางการทำงาน หรือ flow table ของ Open Vswitch นอกจากนี้เราจะพบว่า คำสั่งนี้ (ovs-ofctl) มีประโยชน์โดยเฉพาะเอาไว้หาข้อบกพร่อง (debugging) โดยการดู สถานะของ switch port (ช่อง interface), flow state, flow counters, และ ในการทดลองที่จะได้กล่าวต่อไปเราใช้คำสั่งนี้เพื่อสั่ง Open vswitch ให้ทำงานตามที่ต้องการแบบง่ายๆ โดยการเพิ่ม flow ด้วยคำสั่งทีละบรรทัด <u>สร้างหน้าจอใหม่สำหรับการใช้งาน Mininet บน VM ด้วย putty ผ่าน SSH (ตามข้อ3.1)</u> และ run คำสั่ง ovsofctl show s1 ดังนี้: (ต้องมี sudo นำหน้าคำสั่ง ovs-ofctl มิฉะนั้น จะไม่ได้รับอนุญาตให้ใช้คำสั่งนี้)

เราใช้คำสั่ง ovs-ofctl show s1 ก็เพื่อดูสถานะของ port ของ Open vswitch หรือ ก็คือ S1 นั่นเอง ที่ถูกเรา สร้างด้วยคำสั่ง **sudo** mn --topo single,3 --mac --switch ovsk --controller remote

```
mininet@mininet-vm:~$ sudo ovs-ofctl show s1
n_tables:254, n_buffers:256
capabilities: FLOW_STATS TABLE_STATS PORT_STATS QUEUE_STATS ARP_MATCH_IP
actions: OUTPUT_SET_VLAN_VID_SET_VLAN_PCP_STRIP_VLAN_SET_DL_SRC_SET_DL_DST
SET_NW_SRC_SET_NW_DST_SET_NW_TOS_SET_TP_SRC_SET_TP_DST_ENQUEUE
 1(s1-eth1): addr:ca:9b:e1:61:8e:7f
     config:
                  0
     state:
                  0
     current:
                  10GB-FD COPPER
     speed: 10000 Mbps now, 0 Mbps max
 2(s1-eth2): addr:36:4e:3e:4b:24:39
     config:
                  0
     state:
                  0
     current: 10GB-FD COPPER
     speed: 10000 Mbps now, 0 Mbps max
 3(s1-eth3): addr:c6:a5:cf:1a:e9:f4
     config:
                0
                  0
     state:
     current: 10GB-FD COPPER
     speed: 10000 Mbps now, 0 Mbps max
 LOCAL(s1): addr:e6:73:69:ed:ae:ae
     config: PORT_DOWN
state: LINK_DOWN
     speed: 0 Mbps now, 0 Mbps max
OFPT GET CONFIG REPLY (xid=0x4): frags=normal miss send len=0
```

โดยคำสั่ง show จะเชื่อมต่อไปยัง switch และ แสดงสถานะ(state) ของ port ของ switch (s1),และค่า สมรรถนะ(capabilities)อื่นๆของ switch (s1)

คำสั่งที่มีประโยชน์อีกคำสั่งคือ ovs-ofctl dump-flows s1 (ต้องมี sudo นำหน้าคำสั่ง ovs-ofctl มิฉะนั้น จะ ไม่ได้รับอนุญาตให้ใช้คำสั่งนี้)

```
mininet@mininet-vm:~$ sudo ovs-ofctl dump-flows s1
NXST FLOW reply (xid=0x4):
```

option ที่เราใส่ คือ dump-flows ก็เพื่อต้องการดูค่าใน flow-table ของ Open Vswitch S1 ในที่นี้ ได้ค่าใน flow-table ว่างเปล่า <u>ทั้งนี้เนื่องจากเรายังไม่ได้ สั่งให้ controller ทำงานเลย</u>

หมายเหตุ: .กรณีตัวอย่างข้างบน, ovs-ofctl ติดต่อ Open vSwitch ที่อยู่ภายในของVM ดังนั้นการติดต่อก็จะ ใช้ชื่อ S1 ได้, แต่ในกรณีที่เราจะติดต่อกับเครื่อง OpenFlow Switch ที่มี hardware จริง และกำลังทำงานอยู่ และเราต้องการทราบสถานะของ OpenFlow Switch ดังกล่าวแล้วจะใช้คำสั่งดังนี้: \$ ovs-ofctl dump-flows tcp : {IP address} : {port}

โดย {IP address} คือ IP address ของ interface ที่ใช้ทำ management ของ switch และ {port}

ก็จะเป็น listening/management port ของ OpenFlow ที่เป็น passive

6.1 ทดสอบคำสั่ง Ping

ไปที่ Mininet console ที่ยังคงเปิดการใช้งานอยู่ และให้ทดลองใช้ คำสั่ง ping จาก h1 ไปยัง h2 ดังนั้นที่ Mininet console จะได้ดังนี้:

mininet> h1 ping -c3 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
From 10.0.0.1 icmp_seq=1 Destination Host Unreachable
From 10.0.0.1 icmp_seq=2 Destination Host Unreachable
From 10.0.0.1 icmp_seq=3 Destination Host Unreachable
--- 10.0.0.2 ping statistics --3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 2000ms
pipe 3
mininet>

สังเกตว่า ชื่อของ host h2 ถูกแทนด้วย IP address 10.0.0.2 อย่างอัตโนมัติ เมื่อ run คำสั่ง ping ใน Mininet console

คำถาม การใช้คำสั่ง ping ครั้งนี้ h1 ติดต่อ h2 ได้หรือไม่ ? ทำไม ?

ตามที่ท่านได้ทราบก่อนหน้านี้แล้วว่า flow table ของ switch s1 ว่างเปล่า, นอกจากนี้ switch s1 ก็ยังไม่ได้ถูก เชื่อมต่อไปยัง controller ใดๆ ดังนั้น switch s1 จึงยังไม่รู้ว่าจะต้องทำอะไร เมื่อมี ข้อมูลเข้ามายัง switch s1จึง ทำให้ การping ครั้งนี้ไม่ประสบความสำเร็จ

ให้เราใช้ ovs-ofctl เพื่อติดตั้ง หรือเพิ่ม flow บน switch S1 ที่จำเป็นด้วยcommand-line, ที่ SSH terminal ใช้คำสั่งดังนี้:

```
mininet@mininet-vm:~$ sudo ovs-ofctl add-flow s1 in_port=1,action=output:2
mininet@mininet-vm:~$ sudo ovs-ofctl add-flow s1 in_port=2,action=output:1
```

ด้วยการใช้คำสั่งข้างบน จะบอกให้ switch ทำการส่งผ่าน (forward) ข้อมูลที่เข้ามาทาง port 1 พาออกไปทาง port 2 และในทำนองเดียวกันเมื่อมี ข้อมูลเข้ามาทาง port 2 แล้ว switch ก็จะพาออกไปทาง port 1, เราสามารถพิสูจน์ความถูกต้องการทำงานของ flow-table ของ switch s1 ได้โดยใช้คำสั่งดังนี้:

```
mininet@mininet-vm:~$ sudo ovs-ofctl dump-flows s1
NXST_FLOW reply (xid=0x4):
    cookie=0x0, duration=619.451s, table=0, n_packets=0, n_bytes=0, idle_age=619, in_port=1 actions=output:2
    cookie=0x0, duration=575.683s, table=0, n_packets=0, n_bytes=0, idle_age=575, in port=2 actions=output:1
```

ไปที่ Mininet console (SSH terminal) และให้ทดลองใช้ คำสั่ง ping จาก h1 ไปยัง h2 อีกครั้ง ดังนั้นที่ Mininet console จะได้ดังนี้

```
mininet> h1 ping -c3 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=1.90 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.092 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.091 ms
--- 10.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 0.091/0.696/1.905/0.854 ms
```

<u>คำถาม</u> การใช้คำสั่ง ping ครั้งนี้ h1 ติดต่อ h2 ได้หรือไม่ ?

ให้ตรวจสอบ flow-table อีกครั้ง และดูค่าสถิติ สำหรับในแต่ละบรรทัดของ flow, และค่าเหล่านี้เป็นไปตามที่ ท่านคาดหวังไว้หรือไม่ โดยอาศัยการping

จากนั้นให้ออกจาก Mininet โดยการใช้คำสั่ง

mininet> exit

จากนั้นใส่คำสั่ง restart Mininet

```
mininet@mininet-vm:~$ sudo mn -c
```

เป็นการยกเลิกการสร้าง โครงข่ายจำลอง

7. เปิดการใช้งาน Controller และ ใช้ Wireshark ดู ข่าวสารที่รับส่งในช่วงเริ่มการทำงานของ controller.

ให้ทำการเปิดการใช้งานของ wireshark โดยทำตามข้อ 3.2 ทั้งข้อ ก และข, จากนั้นที่ Wireshark ให้ตั้งค่าที่ช่อง ของ filter โดยการพิมพ์ of (เป็นตัวย่อของOpenFlow) ตามในรูปเพื่อดูข่าวสารที่ใช้ควบคุมของ OpenFlow ที่ รับส่งระหว่าง controller กับ switch

T	he W	Viresha	ark N	letwork /	Analyzer	[Wiresha	rk 1.10.6	(v1.10	.6 from n	naster-1	.10)]					×
File	Edit	View	Go	Capture	Analyze	Statistics	Telephor	ny Tools	Internals	Help				\square		
		۲	4		٦		4301 930 9335	X	2	0			¢			•
Filt	er:	of [\leq								▼ E>	pression	Cle	ar Apply Sav	e	

จากนั้น click ที่ apply ตามรูปข้างบนเพื่อทำการ filter ข่าวสารที่ถูกบันทึกทั้งหมด

จากนั้นทำตามข้อ 3.2 ข เพื่อวัดชนิดข่าวสารที่ interface lo, โดยใช้ เม้าส์เลือกที่ capture, interface, จากนั้น วัดที่ interface lo นั่นคือขณะนี้ Wireshark กำลังตรวจจับข่าวสารที่ interface lo ของ VM

ให้เริ่ม การทำงานของ controller อย่างง่าย ด้วยคำสั่ง controller ptcp:6633 & ที่ SSH terminal ,

```
mininet@mininet-vm:~$ controller ptcp:6633 &
[2] 6132
```

เป็นการ สั่งให้ openflow controller ทำงาน ที่ port 6633 , ip=127.0.0.1,

ดูวิธีใช้ mininet@mininet-vm:~\$ controller --help , ptcp ใช้ passive openflow connection

จากนั้นให้สร้างโครงข่ายจำลองที่เคยสร้างแล้ว (openflow switch 1 ตัว, virtual hosts 3ตัว, และ controller) ด้วยคำสั่ง sudo mn -topo=single,3 -mac -switch=ovsk -controller=remote , สังเกตว่าโครงสร้าง จำลองที่ถูกสร้างขึ้นนั้น สามารถเชื่อมต่อกับ controller ได้สำเร็จ (Adding controller)

```
mininet@mininet-vm:~$ sudo mn --topo=single,3 --mac --switch=ovsk --controller=remote
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1)
*** Configuring hosts
h1 h2 h3
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>
```

โดยที่ไม่ต้องเพิ่ม flow การทำงานใดๆลงบน flow-table ของ switch ในข้อ 6. (ด้วยคำสั่ง sudo ovs-ofctl add-flow s1 in_port=1, action=output:2) โดยเราจะเห็น กลุ่มของข่าวสารที่แสดงโดย Wireshark เกิดขึ้น ดังนี้ (เนื่องมาจากการใช้คำสั่ง \$ controller ptcp:6633 &)

💰 Ca	Capturing from Loopback: lo [Wireshark 1.10.6 (v1.10.6 from master-1.10)]												
File E	dit View Go Cap	pture Analyze Statistics	ēlephony Tools Internals ⊦	telp									
0				o, 🤶	🔿 🥺 ዥ 🕹 目	•							
Filter	of			• E	Expression Clear Apply Save								
No.	Time	Source	Destination	Protoco	Lengtl Info	~							
11	.25 30.546031000	127.0.0.1	127.0.0.1	0F 1.0	74 of_hello								
11	.27 30.546747000	127.0.0.1	127.0.0.1	0F 1.0	74 of_hello								
13	.29 30.547343000	127.0.0.1	127.0.0.1	0F 1.0	74 of_features_request								
13	.31 30.547496000	127.0.0.1	127.0.0.1	0F 1.0	290 of_features_reply								
13	.32 30.547751000	127.0.0.1	127.0.0.1	0F 1.0	78 of_set_config								
12	259 35.546627000	127.0.0.1	127.0.0.1	0F 1.0	74 of_echo_request								
12	260 35.547039000	127.0.0.1	127.0.0.1	0F 1.0	74 of_echo_reply	-							
D Era	ame 1125: 74 byt	es on wire (592 bits).	74 bytes captured (592	(bits) on (interface 0								
D Et	nernet II, Src:	00:00:00 00:00:00 (00:	00:00:00:00:00) Dst: 0	0: 00: 00 00	:00:00 (00:00:00:00:00:00)								
D In	ternet Protocol	Version 4, Src: 127.0.	0.1 (127.0.0.1), Dst: 1	.27.0.0.1 (:	127.0.0.1)								
Transmission Control Protocol, Src Port: 47169 (47169), Dst Port: 6633 (6633), Seg: 1, Ack: 1, Len: 8													
D Op	enFlow												
\sum													
5													
0000	00 00 00 00 00	00 00 00 00 00 00 00 00	08 00 45 c0	E.									
0020	00 01 b8 41 19	e9 5f 63 d1 01 07 22	01 c7 80 18A c										
0030	00 56 fe 30 00	00 01 01 08 0a 00 06	ld 6e 00 06 .V.O	n									
0040	ld 6e 01 00 00	08 00 00 00 01	. n										

ข่าวสารของ OpenFlow ที่รับส่งระหว่าง controller กับ switch เริ่มจากการแลกเปลี่ยนข่าวสาร Hello, ตาม ด้วย Feature_request, Feature_reply, set_config, echo_request และ echo_reply โดยการ click ที่รูป สามเหลี่ยม ข้างหน้าของ OpenFlow ก็จะแสดงรายละเอียดเพื่อดูข้อมูลของส่วนต่างๆใน OpenFlow เพิ่มเติม เป็นต้น

- ให้ ดูข่าวสารของ OpenFlow จากการใช้คำสั่ง Ping,

ก่อนอื่นที่ช่อง filter ของ Wireshark เราจะกรองเพื่อไม่ให้ Wireshark แสดงข่าวสาร OpenFlow ที่เป็นชนิด of_echo_request และ of_echo_replay (ซึ่งเป็นข่าวสารของ OpenFlow ที่ใช้ตรวจสอบการมีอยู่ระหว่าง controller และ switch) โดยพิมพ์ค่าดังต่อไปนี้ลงในช่องของ filter

(of and not (of10.echo_request.type or of10.echo_reply.type)

6	Capt	uring	from I	Loopba	k: lo [Nireshark	: 1.10.6 (\	/1.10.6 f	rom ma	ster-1.10)]						
File	Edi	t View	/ Go	Capture	Analyze	Statistic	s Telepho	ny Tools	Interna	als Help							
	Ð	۲	\mathcal{L}		٦	F		X	Z	🎤	>		4	7	₹		•
Filt	:er:	of an	d not (of10.ec	o_reques	t.type or	of10.echo_	reply.typ	pe)		▼ E×	pression	Cle	ar Ap	oply S	ave	

จากนั้นให้ลบค่าต่างๆ บน ARP table ของ host h1และ h2 (arp table บอกค่า MAC address ของ host ปลายทางและบอกว่าอยู่ interface ไหน) โดยใช้คำสั่ง ที่ Mininet consoleดังนี้:

```
mininet> h1 ip -s -s neigh flush all
mininet> h2 ip -s -s neigh flush all
```

ใช้คำสั่ง ping โดย ping จาก h1 ไปยัง h2 จำนวน 1 ครั้ง, โดยทำที่ Mininet console ดังนี้

```
mininet> h1 ping -c1 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=2.15 ms
```

<u>คำถาม</u>

ท่านคิดว่า การ ping ครั้งนี้ จาก host h1 ไปยัง host h2 ทำไม จึงping สำเร็จ ?

และเมื่อใช้ Wireshark ตรวจจับก็จะได้ OpenFlow ชนิดต่างๆ เช่น of_packet_in , of_packet_out, of flow add เป็นต้น ตามรูปข้างล่าง

🔏 Capti	uring from Loo	pback: lo [Wireshark 1	10.6 (v1.10.6 from master	r-1.10)]			x
File Edit	: View Go Cap	oture Analyze Statistics	Telephony Tools Internals	Help			
0				S (👄 ዥ 🕹 🗐	•
Filter:	of and not (of1(0.echo_request.type or of1	0.echo_reply.type)	• E	Expression	Clear Apply Save	
No.	Time	Source	Destination	protocol			
		Jource	Descinación	Protoco	Lengti	Info	
109572	5387.2397930	00:00:00_00:00:01	Broadcast	0F 1.0	Lengti 126	of_packet_in	
109572 109573	5387.2397930 5387.2399960	00: 00: 00_00: 00: 01 127. 0. 0. 1	Broadcast 127.0.0.1	0F 1.0 0F 1.0	Lengti 126 90	Info of_packet_in of_packet_out	
109572 109573 109575	5387.2397936 5387.2399966 5387.2402986	00: 00: 00_00: 00: 01 127. 0. 0. 1 00: 00: 00_00: 00: 02	Broadcast 127.0.0.1 00:00:00:00:00:01	OF 1.0 OF 1.0 OF 1.0	Lengtl 126 90 126	Into of_packet_in of_packet_out of_packet_in	
109572 109573 109575 109576	5387. 2397930 5387. 2399960 5387. 2402980 5387. 2402980	00: 00: 00_00: 00: 01 127. 0. 0. 1 00: 00: 00_00: 00: 02 127. 0. 0. 1	Broadcast 127.0.0.1 00:00:00_00:00:01 127.0.0.1	OF 1.0 OF 1.0 OF 1.0 OF 1.0 OF 1.0	Lengtl 126 90 126 146	Into of_packet_in of_packet_out of_packet_in of_flow_add	
109572 109573 109575 109576 109577	5387. 2397930 5387. 2399960 5387. 2402980 5387. 2404060 5387. 2409670	00: 00: 00_00: 00: 01 127. 0. 0. 1 00: 00: 00_00: 00: 02 127. 0. 0. 1 10. 0. 0. 1	Description Broadcast 127.0.0.1 00:00:00_00:00:01 127.0.0.1 10.0.0.2	OF 1.0 OF 1.0 OF 1.0 OF 1.0 OF 1.0 OF 1.0	Lengtl 126 90 126 146 182	Into of_packet_in of_packet_out of_packet_in of_flow_add of_packet_in	

คำถาม

1. ให้ใช้ wireshark วัดและดูค่าอย่างละเอียด เพื่อหาว่า ARP_request, ARP_reply อยู่ที่ packet ไหน

(แนะ ดูที่ of_packet_in,)

2. ดูว่า Openflow Controller สั่งให้ switch S1 ทำงาน ตามที่ controller ต้องการอยู่ที่ packet ไหน

(แนะ ดูที่ of_packet_add,)

ให้ออกจาก mininet ด้วย exit และเป็นการยกเลิกการสร้าง โครงข่ายจำลองด้วย sudo mn -c

8. การสร้างโครงขายเสมือนด้วยMininet ที่ใช้ GUI (Graphical User Interface)

ในที่นี้ตามในข้อ 5. เราได้สร้างโครงข่ายเสมือน ที่ประกอบด้วย 3 virtual hosts, 1 OpenFlow switch, 1 OpenFlow controller ผ่านทางSSH terminal ด้วยคำสั่ง:

```
mininet@mininet-vm:~$ sudo mn --topo single,3 --mac --switch ovsk --controller remote
```

ไปแล้วนั้น ในที่นี้เราจะสร้างโครงข่ายเดียวกัน (ตามในข้อ 5 การใช้ Mininet หน้า 14) แต่ใช้ GUI , โดยที่เราจะ ต้องทำในข้อ 3.2ก.คือติดตั้งXming serverและทำในข้อ 3.2ข คือการ enable X11 forward บน ssh terminal,

ดังนั้นที่ SSH terminal (ทีได้ enable X11 forwarding) ให้พิมพ์คำสั่งดังนี้:

```
mininet@mininet-vm:~$ sudo ~/mininet/examples/miniedit.py
```

และจะได้ตามรูปข้างล่าง(หน้าจอว่างเปล่า), จากนั้น ให้เราเลือก อุปกรณ์ที่เราต้องการสร้าง ก็เอาเม้าส์เลื่อนมาที่ รูป s1 จากนั้นให้ click ลงไปที่รูปของ switch s1 (1), จากนั้น ใช้เม้าส์เลื่อนมาที่หน้าจอ พื้นที่ว่างทางขวามือ แล้ว click ลงไป ก็จะได้รูป switch s1 ปรากฏขึ้นมา(2), ทำในทำนองเดียวกันนี้ กับอุปกรณ์อื่นๆ (host h1,host h2,host h3, controller c0)จนครบทุกตัว, จากนั้น ใช้เม้าส์ มา click ที่ link (3), จากนั้น สมมุติต้องการสร้าง link ระหว่าง S1 กับ h1 ก็เอาเม้าส์มา click ที่ s1 โดยกดเม้าส์ค้างไว้ แล้วลากเม้าส์มาที่ h1 แล้วปล่อยการกดปุ่ม ของเม้าส์ (4) ก็จะได้ link ที่เชื่อมต่อระหว่าง S1 กับ h1 ทำเช่นนี้จนกว่าได้สร้าง link ครบตามรูปข้างล่าง



การใช้ MiniEdit GUI เพื่อสร้างโครงข่ายที่ประกอบด้วย Switch, Host, และ controller จะมีการ configure ค่า preference เพิ่มเติม , โดย click ที่ edit แล้ว click ที่ preference



จะได้ตามรูปข้างล่าง , ให้ click เลือก Start CLI แล้ว Click OK, ตามรูป

Preferences IP Base: IP Base: IDefault Terminal: xterm Start CLI: V Default Switch: Open vSwitch Open vSwitch Open Flow 1.0: V OpenFlow 1.1: OpenFlow 1.2: OpenFlow 1.3: dpctl port: OK SFlow Profile for Open vSwitch Target: Sampling: 400 Header: 128 Polling: 30 NetFlow Profile for Open vSwitch Target: Active Timeout: 600 Add ID to Interface: OK	×
---	-------

ใช้เม้าส์ชี้ไปที่ controller แล้วก็กดเม้าส์ค้างไว้ แล้วเลื่อนมาชี้ที่ properties แล้วปล่อย



<u>หมายเหตุ</u> หน้าจอของ MiniEdit จะถูกซ่อนอยู่ให้ click ไปที่ Xming server ก็จะได้ หน้าจอ MiniEdit

เลือก controller type เป็น remote controller, จากนั้น click ที่ OK



จากนั้นให้มา click ที่ file, click ที่ save จะได้ตามรูปข้างล่างใส่ชื่อไฟล์เช่น ในที่นี้ตั้งชื่อเป็น test1, click ที่ save

File Edit Run Help	
k	Save the topology as
	Directory: /home/mininet
	Cache pox
	initial content in the second content is the second content in the second content in the second content is the second content in the second content in the second content is the second content in the second content in the second content is the second content in the second content is the second content in the second content in the second content is the second content in the second content is the second content is the second content in the second content is the second content is the second content in the second content is the second content in the second content is the second content is the second content in the second content is the second content is the second content in the second content is the second content is the second content in the second content is the second content in the second content is the second content is the second content in the second content is the second content is the second content in the second content is the second content in the second content is the second content is the second content in the second content is the second content is the second content in the second content is the second content in the second content in the second content is the second content in the second content in the second content is the second content in the second content in the second content is the second content in the
	□ mininet □ oflops
	oftest
	File name: test1
	Files of type: Mininet Topology (*.mn) Cancel

สั่ง ให้โครงข่ายที่เราสร้างขึ้นมา ทำงาน

ให้เลื่อน เม้าส์มา พร้อมกับ clickที่ RUN เลื่อน เม้าส์ลงมาพร้อมกับ click ที่ run อีกครั้ง



จากนั้นให้ เปิดหน้าจอ SSH terminal (ใช้เม้าส์ ไปที่ รูปจอซ้อนกัน 2 เครื่อง ที่menu bara่างสุดเพื่อเลือกจอ SSH terminal ที่มีการ run Mininet) จะได้หน้าจอ Mininet console ตามรูป

```
NOTE: PLEASE REMEMBER TO EXIT THE CLI BEFORE YOU PRESS THE STOP BUTT
ting will prevent MiniEdit from quitting and will prevent you from st
network again during this sessoin.
*** Starting CLI:
mininet>
```

มาถึงขณะนี้ ที่ OpenFlow Switch ยังไม่ได้เชื่อมต่อไปยัง controller ดังนั้น ถ้า ping จาก h1 ไปยัง h2 จำนวน 1 ครั้ง การping ในครั้งนี้ ก็จะไม่สำเร็จ เนื่องจาก switch S1 ไม่รู้ว่าจะทำอย่างไรเมื่อมี ข้อมูลเข้ามาที่ switch S1

```
mininet> h1 ping -c1 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
From 10.0.0.1 icmp_seq=1 Destination Host Unreachable
```

้ในขั้นต่อไปจะ สาธิตแนวคิดพื้นฐานของ software-defined networking (SDN) ที่ใช้ SDN controller ชื่อ POX

POX controller สั่งให้ switch S1ทำงาน เป็น Hub

โดยใช้คำสั่งดังนี้ ที่ SSH terminal (เปิดหน้าจอ putty ใหม่)

mininet@mininet-vm:~\$ sudo ~/pox/pox.py forwarding.hub

เป็นการบอก POX sdn controller ทำงาน โดยสั่งให้ switch S1 ทำงาน เป็น hub, อย่างไรก็ตาม switch อาจ ใช้เวลาสักครู่ เพื่อเชื่อมต่อไปยัง controller และให้คอยจนกว่า switch S1 เชื่อมกับ controller โดยหน้าจอของ POX บนจอ SSH terminal จะแสดงดังรูป

```
POX 0.2.0 (carp) / Copyright 2011-2013 James McCauley, et al.
INFO:forwarding.hub:Hub running.
INFO:core:POX 0.2.0 (carp) is up.
INFO:openflow.of_01:[None 1] closed
INFO:openflow.of_01:[00-00-00-00-01 2] connected
INFO:forwarding.hub:Hubifying 00-00-00-00-01
```

ตรวจสอบการทำงานของ switch s1 (ถูกสั่งให้ทำหน้าที่เป็น hub) ด้วย tcpdump

เราจะตรวจสอบว่า host แต่ละตัวสามารถ ping ถึงกันและกันได้, โดย host ทุกตัวจะมองเห็น ข่าวสารชนิด เดียวกัน ซึ่งก็คือการทำงานของ hub, ในที่นี้เราจะสร้าง terminal ของ host แต่ละตัวด้วยคำสั่งที่ mininet console ตามข้างล่างนี้

mininet> xterm h1 h2 h3

ใช้เม้าส์ชี้ไปที่ รูป Xming Server ที่ tool bar บรรทัดล่างสุดของ windows พร้อม click ไปที่ host แต่ละตัว และจัดให้ทั้ง 3 หน้าจอของ host แสดงผลพร้อมกันตามรูป

X "Node: h1"	🛣 "Node: h2"
root@mininet-vm:**# []	root@mininet-vm:~*# [
X "Node: h3"	
root@mininet-vm; [~] # ∎	

ที่หน้าจอของ host h2, ใช้คำสั่ง #tcpdump

ที่หน้าจอของ host h3, ใช้คำสั่ง #tcpdump

ที่หน้าจอของ host h1, ใช้คำสั่ง #ping -c1 10.0.0.2

คำสั่ง ping จะถูกส่งไปยัง controller , ที่จะทำให้ คำสั่ง ping ดังกล่าวถูก flood คือ ถูกส่งออกทุก port ของ switch ได้แก่ port 2, 3 แต่จะไม่ถูกส่งออก port 1 ซึ่งคำสั่ง ping ถูกส่งเข้ามาทาง port นี้ เราจะมองเห็นว่า

ข่าวสารของทั้ง ARP และ ICMP packets เดียวกันนี้ จะปรากฎพร้อมกันบนหน้าจอของ host h2, host h3

ที่สอดคล้องกับคำสั่ง ping ที่เราใช้ อันนี้เป็นการแสดงการทำงานของ hub นั่นคือ hub จะส่ง packets ออกไปยัง ทุกport ของ hub ยกเว้น port ที่ hub ได้รับ packet เข้ามา

🏋 "Node: h1"

```
root@mininet-vm:~# ping -c1 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=16.5 ms
```

--- 10.0.0.2 ping statistics ---1 packets transmitted, 1 received, 0% packet loss, time Oms rtt min/avg/max/mdev = 16.535/16.535/16.535/0.000 ms

🏋 "Node: h2"

root@mininet-vm:~# tcpdump -xx -n -i h2-eth0 topdump: verbose output suppressed, use -v or -vv for full protocol decode listening on h2-eth0, link-type EN10MB (Ethernet), capture size 65535 bytes 17:41:33.6666681 IP 10.0.0.1 > 10.0.0.2: ICMP echo request, id 2411, seq 1, lengt h 64 0×0000: 267b cd89 c355 7a27 70b3 3fde 0800 4500 0x0010: 0054 9cad 4000 4001 89f9 0a00 0001 0a00 0x0020: 0002 0800 9fd2 096b 0001 4dab 9556 77bc 0x0030t 0900 0809 0a0b 0c0d 0e0f 1011 1213 1415 0x0040: 1617 1819 1a1b 1c1d 1e1f 2021 2223 2425 0x0050; 2627 2829 2a2b 2c2d 2e2f 3031 3233 3435 0x0060: 3637 17:41:33.666741 IP 10.0.0.2 > 10.0.0.1: ICMP echo reply, id 2411, seq 1, length 64 0×0000: 7a27 70b3 3fde 267b cd89 c355 0800 4500 0054 efd2 0000 4001 76d4 0a00 0002 0a00 0x0010: 0x0020; 0001 0000 a7d2 096b 0001 4dab 9556 77bc 0x0030: 0900 0809 0a0b 0c0d 0e0f 1011 1213 1415 1617 1819 1a1b 1c1d 1e1f 2021 2223 2425 0x0040: 0x0050: 2627 2829 2a2b 2c2d 2e2f 3031 3233 3435 0x0060: 3637 17:41:38.649206 ARP, Request who-has 10.0.0.2 tell 10.0.0.1, length 28 267b cd89 c355 7a27 70b3 3fde 0806 0001 0800 0604 0001 7a27 70b3 3fde 0a00 0001 0x0000: 0x0010: 0000 0000 0000 0a00 0002 0x0020: 17:41:38.649235 ARP, Reply 10.0.0.2 is-at 26:7b:cd:89:c3:55, length 28 7a27 70b3 3fde 267b cd89 c355 0806 0001 0800 0604 0002 267b cd89 c355 0a00 0002 0x0000: 0x0010: 0x0020: 7a27 70b3 3fde 0a00 0001

🏋 "Node: h3"

root@mininet-vm:"# tcpdump -xx -n -i h3-eth0 topdump: verbose output suppressed, use -v or -vv for full protocol decode listening on h3-eth0, link-type EN10MB (Ethernet), capture size 65535 bytes 17:41:33.666677 IP 10.0.0.1 > 10.0.0.2: ICMP echo request, id 2411, seq 1, lengt h 64 0x0000: 267b cd89 c355 7a27 70b3 3fde 0800 4500 0054 9cad 4000 4001 89f9 0a00 0001 0a00 0x0010: 0x0020: 0002 0800 9fd2 096b 0001 4dab 9556 77bc 0900 0809 0a0b 0c0d 0e0f 1011 1213 1415 0x0030: 1617 1819 1a1b 1c1d 1e1f 2021 2223 2425 0x0040± 0x0050; 2627 2829 2a2b 2c2d 2e2f 3031 3233 3435 0x0060: 3637 17:41:33.669220 IP 10.0.0.2 > 10.0.0.1: ICMP echo reply, id 2411, seq 1, length 64 0x0000: 7a27 70b3 3fde 267b cd89 c355 0800 4500 0x0010: 0054 efd2 0000 4001 76d4 0a00 0002 0a00 0001 0000 a7d2 096b 0001 4dab 9556 77bc 0x0020: 0x0030: 0900 0809 0a0b 0c0d 0e0f 1011 1213 1415 0x0040: 1617 1819 1a1b 1c1d 1e1f 2021 2223 2425 2627 2829 2a2b 2c2d 2e2f 3031 3233 3435 0x0050: 0x0060: 3637 17:41:38.649204 ARP, Request who-has 10.0.0.2 tell 10.0.0.1, length 28 0x0000: 267b cd89 c355 7a27 70b3 3fde 0806 0001 0x0010: 0800 0604 0001 7a27 70b3 3fde 0a00 0001 0000 0000 0000 0a00 0002 0x0020: 17:41:38.650248 ARP, Reply 10.0.0.2 is-at 26:7b:cd:89:c3:55, length 28 0x0000: 7a27 70b3 3fde 267b cd89 c355 0806 0001 0x0010: 0800 0604 0002 267b cd89 c355 0a00 0002 0x0020: 7a27 70b3 3fde 0a00 0001

ิตรวจสอบประสิทธิภาพการทำงานของ controller ที่ถูกกำหนดให้ทำหน้าที่เป็น Hub

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3
h2 -> h1 h3
h3 -> h1 h2
*** Results: 0% dropped (6/6 received)
```

ใช้คำสั่ง iperf เพื่อทดสอบสมรรถนะการส่งข้อมูลของ switch s1

```
mininet> iperf
*** Iperf: testing TCP bandwidth between h2 and h1
*** Results: ['1.49 Gbits/sec', '1.49 Gbits/sec']
```

ต่อไปจะให้ POX controller ทำหน้าที่เป็น Layer 2 learning switch โดยดำเนินการดังนี้:

จาก ที่หน้าจอเดิมของ POX controller ที่ทำหน้าที่เป็น Hub ตามรูปข้างล่าง

```
POX 0.2.0 (carp) / Copyright 2011-2013 James McCauley, et al.
DEBUG:core:POX 0.2.0 (carp) going up...
DEBUG:core:Running on CPython (2.7.6/Mar 22 2014 22:59:38)
DEBUG:core:Platform is Linux-3.13.0-24-generic-i686-with-Ubuntu-14.04-trusty
INFO:core:POX 0.2.0 (carp) is up.
DEBUG:openflow.of_01:Listening on 0.0.0.0:6633
INFO:openflow.of_01:[00-00-00-00-01 1] connected
DEBUG:misc.of tutorial:Controlling [00-00-00-00-01 1]
```

<u>ให้พิมพ์ ctrl + C</u>จะได้หน้าจอตามรูปข้างล่าง

```
INFO:openflow.of_01:[00-00-00-00-01 1] connected
DEBUG:misc.of_tutorial:Controlling [00-00-00-00-01 1]
^CINFO:core:Going down...
INFO:openflow.of_01:[00-00-00-00-01 1] disconnected
INFO:core:Down.
mininet@mininet-vm:~$
```

POX sdn controller สั่งให้ switch S1 ทำงาน เป็น Layer 2 learning switch

มาที่หน้าจอ putty ทีโดยใช้คำสั่งดังนี้ ที่ SSH terminal

mininet@mininet-vm:~\$ sudo ~/pox/pox.py forwarding.12_learning

้อย่างไรก็ตาม switch อาจใช้เวลาสักครู่ เพื่อเชื่อมต่อไปยัง controller และให้คอยจนกว่า switch S1 เชื่อมกับ

controller โดยหน้าจอของ POX บนจอ SSH terminal จะแสดงดังรูป

```
mininet@mininet-vm:~$ sudo ~/pox/pox.py forwarding.l2_learning
POX 0.2.0 (carp) / Copyright 2011-2013 James McCauley, et al.
INFO:core:POX 0.2.0 (carp) is up.
INFO:openflow.of_01:[00-00-00-00-01 1] connected
```

ตรวจสอบประสิทธิภาพการทำงานของ POX controller ที่ถูกกำหนดให้ทำหน้าที่เป็น layer 2 learning switch

```
mininet> pingall
*** Ping: testing ping reachability
h3 -> h2 h1
h2 -> h3 h1
h1 -> h3 h2
*** Results: 0% dropped (6/6 received)
```

<u>คำถาม</u>ให้ตรวจสอบด้วยว่า switch S1 มีการ flood ข้อมูลหรือไม่ เมื่อ switch s1 ถูกสั่งด้วย POX SDN

controller ให้ทำงานเป็น Layer 2 Learning Switch

ใช้คำสั่ง iperf เพื่อทดสอบสมรรถนะการส่งข้อมูลของ switch s1



มาที่ หน้าจอ putty ให้ออกจาก mininet

```
mininet> exit
```



ที่หน้าจอ putty สั่งหยุดการทำงานของ mininet server

mininet@mininet-vm:~\$ sudo shutdown -h now

<u>คำถาม</u> ด้วยโครงข่ายเสมือนที่ถูกสร้างขึ้นด้วคำสั่ง "sudo mn -topo=single,3 -mac -switch=ovsk controller=remote" เชือมต่อกับ controller 3 ชนิด ptcp:6633, POX ทำหน้าที่เป็น Hub ,และ POX ทำหน้าที่เป็น layer 2 learning switch ท่านคิดว่า controller ชนิดใดที่ทำให้ switch s1 ทำงานมีประสิทธิภาพ มากที่สุด และทำไมจึงเป็นเช่นนั้น

<u>เอกสารอ้างอิง</u>

- 1. <u>http://mininet.org</u>/ แนะนำการการติดตั้ง พร้อมตัวอย่าง การใช้งานเริ่มต้น
- 2. https://github.com/mininet/mininet/wiki/Introduction-to-Mininet แนะนำ Mininet
- 3. <u>https://github.com/mininet/openflow-tutorial/wiki</u> แนะนำ openflow-tutorial
- N. McKeown, T. Anderson, H. Balakrishnan, G. ParulKar, L. Peterson, J. Rexford, S. Shenker, J. Turner, "OpenFlow: Enabling Innovation in the Campus Network", ACM SIGCOMM Computer Communication Review, vol. 38, no. 2, p. 6, 2008.