

DAFTAR PUSTAKA

Adel Nadjaran Toosi, Jungmin Son, dkk, “CLOUDS-Pi: A Low-Cost Raspberry-Pi based Testbed for Software-Defined-Networking in Cloud Data Centers,” 2017

Ahmad Mahardika Fatahillah, “nalisis Perbandingan Performansi Jaringan Pada Arsitektur *Software-Defined Network* (SDN) dan Konvensional”, 2016

Eko Pramono, Ema Utami, dkk, “Prototipe Infrastruktur Software-Defined Network Dengan Protokol *OpenFlow* Menggunakan Ubuntu Sebagai Kontroler,” 2014.

Eri Prasetyo, Amir Hamzah, dkk., “ Analisis Quality of Service (QoS) Kinerja Point to Point Protocol Over Ethernet (PPPOE) dan Point to Point Tunneling Protocol (PPTP). 2016

Desianto Abdillah, Izzatul Ummah, “Perancangan Simulasi Jaringan Virtual Berbasis Software-Define Networking,” 2016.

<https://www.raspberrypi.org/products> diakses pada tanggal 15 Oktober 2018 21.09 (online)

<https://angryip.org> diakses pada tanggal 28 Oktober 2018 13:52 (online)

<https://iperf.fr/> diakses tanggal 29 Oktober 2018 20:32 (online)

Krishna Varaynya Chivukula, “Monitoring and Analysis of CPU load relationships between Host and Guests in a Cloud Networking Infrastructure”, 2015

Open Networking Foundation, “Software-Defined Networking: The New Norm for Networks - ONF White Paper,” 2012.

Open Networking Foundation, “*OpenFlow* Switch Specification Version 1.5.1 (Protocol version 0x06),” 2015

OpenvSwitch, “ OpenvSwitch Realeases 2.10.0,” 2018

Open Networking Foundation, “ SDN Architecture – Issue 1,” 2014

Oracel Virtualitation, “Oracle Data Sheet-Oracle VM VirtualBox”,2018 (online)

[Oracel Virtual Box, “ User Manual Version 5.1.22,” 2017 \(online\)](#)

Peter Membrey, Tim Verhoeven, dkk, “The Definitive Guide to CentOS,” 2009

Rohmat Tulloh, Ridha Muldina Negara, dkk, “Simulasi Virtual Local Area Network (VLAN) Berbasis Software Defined Network (SDN) Menggunakan POX Controller,” 2015

Yasir Arafad, “Modul Pembelajaran Pemograman Pada Mikrokontroler Dengan Menggunakan *Raspberry PI*,” 2017.

LAMPIRAN

1. Screenshoot Data TCP menggunakan iPerf

```
pi@Host02: ~
permitted by applicable law.
Last login: Sat Oct 27 08:17:08 2018

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

pi@Host02:~$ sudo su
root@Host02:/home/pi# cd iperf-2.0.5
root@Host02:/home/pi/iperf-2.0.5# iperf -s
-----
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
-----
[ 4] local 192.168.2.2 port 5001 connected with 192.168.2.1 port 41872
[ ID] Interval      Transfer      Bandwidth
[ 4] 0.0-10.2 sec  4.38 MBytes  3.58 Mbits/sec
^Croot@Host02:/home/pi/iperf-2.0.5# iperf -s -i 1
-----
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
-----
[ 4] local 192.168.2.2 port 5001 connected with 192.168.2.1 port 41880
[ ID] Interval      Transfer      Bandwidth
[ 4] 0.0- 1.0 sec  440 KBytes    3.60 Mbits/sec
[ 4] 1.0- 2.0 sec  438 KBytes    3.59 Mbits/sec
[ 4] 2.0- 3.0 sec  433 KBytes    3.54 Mbits/sec
[ 4] 3.0- 4.0 sec  440 KBytes    3.60 Mbits/sec
[ 4] 4.0- 5.0 sec  436 KBytes    3.57 Mbits/sec
[ 4] 5.0- 6.0 sec  434 KBytes    3.56 Mbits/sec
[ 4] 6.0- 7.0 sec  428 KBytes    3.51 Mbits/sec
[ 4] 7.0- 8.0 sec  424 KBytes    3.48 Mbits/sec
[ 4] 8.0- 9.0 sec  428 KBytes    3.51 Mbits/sec
[ 4] 9.0-10.0 sec  440 KBytes    3.60 Mbits/sec
[ 4] 10.0-11.0 sec 440 KBytes    3.60 Mbits/sec
[ 4] 11.0-12.0 sec 443 KBytes    3.63 Mbits/sec
[ 4] 12.0-13.0 sec 438 KBytes    3.59 Mbits/sec
[ 4] 13.0-14.0 sec 431 KBytes    3.53 Mbits/sec
[ 4] 14.0-15.0 sec 437 KBytes    3.58 Mbits/sec
[ 4] 15.0-16.0 sec 438 KBytes    3.59 Mbits/sec
[ 4] 16.0-17.0 sec 431 KBytes    3.53 Mbits/sec
[ 4] 17.0-18.0 sec 438 KBytes    3.59 Mbits/sec
[ 4] 18.0-19.0 sec 436 KBytes    3.57 Mbits/sec
[ 4] 19.0-20.0 sec 423 KBytes    3.46 Mbits/sec
[ 4] 0.0-20.3 sec 8.62 MBytes   3.56 Mbits/sec

pi@Host01: ~
pi@Host01:~$ sudo su
root@Host01:/home/pi# cd iperf-2.0.5
root@Host01:/home/pi/iperf-2.0.5# iperf -c 192.168.2.2
-----
Client connecting to 192.168.2.2, TCP port 5001
TCP window size: 43.8 KByte (default)
-----
[ 3] local 192.168.2.1 port 41872 connected with 192.168.2.2 port 5001
[ ID] Interval      Transfer      Bandwidth
[ 3] 0.0-10.1 sec  4.38 MBytes  3.64 Mbits/sec
root@Host01:/home/pi/iperf-2.0.5# iperf -c 192.168.2.2 -i 20
connect failed: Connection refused
root@Host01:/home/pi/iperf-2.0.5# iperf -c 192.168.2.2 -i 1 -t 20
connect failed: Connection refused
root@Host01:/home/pi/iperf-2.0.5# iperf -c 192.168.2.2 -i 1
connect failed: Connection refused
root@Host01:/home/pi/iperf-2.0.5# iperf -c 192.168.2.2 -i 1 -t 20
-----
Client connecting to 192.168.2.2, TCP port 5001
TCP window size: 43.8 KByte (default)
-----
[ 3] local 192.168.2.1 port 41880 connected with 192.168.2.2 port 5001
[ ID] Interval      Transfer      Bandwidth
[ 3] 0.0- 1.0 sec  512 KBytes    4.19 Mbits/sec
[ 3] 1.0- 2.0 sec  512 KBytes    4.19 Mbits/sec
[ 3] 2.0- 3.0 sec  384 KBytes    3.15 Mbits/sec
[ 3] 3.0- 4.0 sec  384 KBytes    3.15 Mbits/sec
[ 3] 4.0- 5.0 sec  512 KBytes    4.19 Mbits/sec
[ 3] 5.0- 6.0 sec  384 KBytes    3.15 Mbits/sec
[ 3] 6.0- 7.0 sec  512 KBytes    4.19 Mbits/sec
[ 3] 7.0- 8.0 sec  384 KBytes    3.15 Mbits/sec
[ 3] 8.0- 9.0 sec  384 KBytes    3.15 Mbits/sec
[ 3] 9.0-10.0 sec  512 KBytes    4.19 Mbits/sec
[ 3] 10.0-11.0 sec 384 KBytes    3.15 Mbits/sec
[ 3] 11.0-12.0 sec  512 KBytes    4.19 Mbits/sec
[ 3] 12.0-13.0 sec  384 KBytes    3.15 Mbits/sec
[ 3] 13.0-14.0 sec  384 KBytes    3.15 Mbits/sec
[ 3] 14.0-15.0 sec  512 KBytes    4.19 Mbits/sec
[ 3] 15.0-16.0 sec  384 KBytes    3.15 Mbits/sec
[ 3] 16.0-17.0 sec  512 KBytes    4.19 Mbits/sec
[ 3] 17.0-18.0 sec  384 KBytes    3.15 Mbits/sec
[ 3] 18.0-19.0 sec  384 KBytes    3.15 Mbits/sec
[ 3] 19.0-20.0 sec  512 KBytes    4.19 Mbits/sec
[ 3] 0.0-20.2 sec  8.62 MBytes   3.58 Mbits/sec
root@Host01:/home/pi/iperf-2.0.5#
```

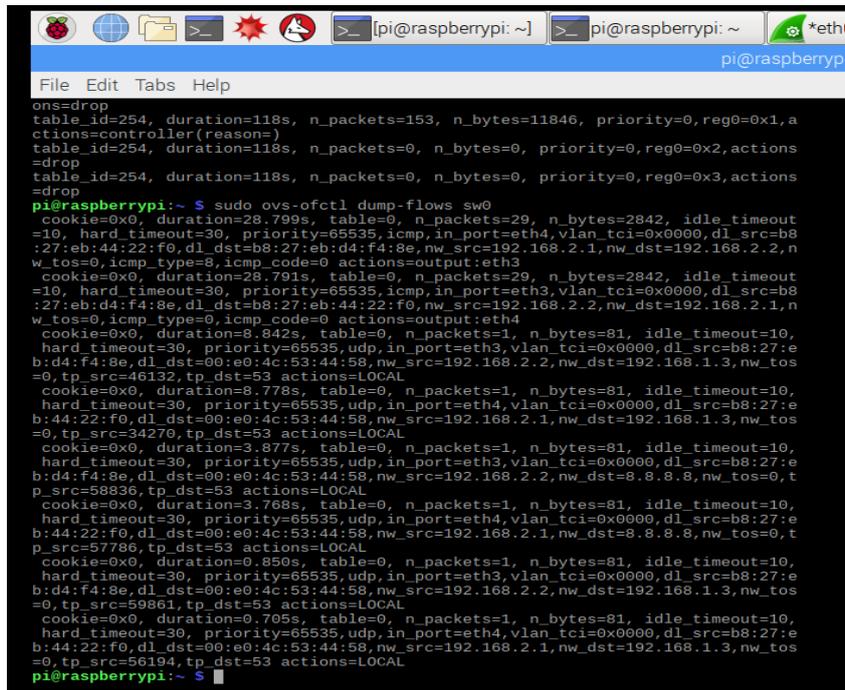
2. Screenshot Data UDP menggunakan iPerf

```
pi@Host02: ~
^[[A^Croot@Host02:/home/pi/iperf-2.0.5# iperf -s -i 1 -u
iperf: ignoring extra argument -- 1
-----
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 160 KByte (default)
-----
[ 3] local 192.168.2.2 port 5001 connected with 192.168.2.1 port 39918
[ ID] Interval      Transfer      Bandwidth      Jitter    Lost/Total Datagrams
[ 3] 0.0-20.0 sec  2.50 MBytes  1.05 Mbits/sec  0.603 ms  0/ 1785 (0%)
^Croot@Host02:/home/pi/iperf-2.0.5# iperf -s -i 1 -u
-----
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 160 KByte (default)
-----
[ 3] local 192.168.2.2 port 5001 connected with 192.168.2.1 port 44048
[ ID] Interval      Transfer      Bandwidth      Jitter    Lost/Total Datagrams
[ 3] 0.0- 1.0 sec  129 KBytes  1.06 Mbits/sec  0.573 ms  0/ 90 (0%)
[ 3] 1.0- 2.0 sec  129 KBytes  1.06 Mbits/sec  0.618 ms  0/ 90 (0%)
[ 3] 2.0- 3.0 sec  128 KBytes  1.05 Mbits/sec  0.627 ms  0/ 89 (0%)
[ 3] 3.0- 4.0 sec  128 KBytes  1.05 Mbits/sec  0.633 ms  0/ 89 (0%)
[ 3] 4.0- 5.0 sec  128 KBytes  1.05 Mbits/sec  0.580 ms  0/ 89 (0%)
[ 3] 5.0- 6.0 sec  128 KBytes  1.05 Mbits/sec  0.587 ms  0/ 89 (0%)
[ 3] 6.0- 7.0 sec  128 KBytes  1.05 Mbits/sec  0.616 ms  0/ 89 (0%)
[ 3] 7.0- 8.0 sec  129 KBytes  1.06 Mbits/sec  0.596 ms  0/ 90 (0%)
[ 3] 8.0- 9.0 sec  128 KBytes  1.05 Mbits/sec  0.627 ms  0/ 89 (0%)
[ 3] 9.0-10.0 sec  128 KBytes  1.05 Mbits/sec  0.614 ms  0/ 89 (0%)
[ 3] 10.0-11.0 sec 128 KBytes  1.05 Mbits/sec  0.609 ms  0/ 89 (0%)
[ 3] 11.0-12.0 sec 128 KBytes  1.05 Mbits/sec  0.629 ms  0/ 89 (0%)
[ 3] 12.0-13.0 sec 128 KBytes  1.05 Mbits/sec  0.607 ms  0/ 89 (0%)
[ 3] 13.0-14.0 sec 129 KBytes  1.06 Mbits/sec  0.619 ms  0/ 90 (0%)
[ 3] 14.0-15.0 sec 128 KBytes  1.05 Mbits/sec  0.605 ms  0/ 89 (0%)
[ 3] 15.0-16.0 sec 128 KBytes  1.05 Mbits/sec  0.587 ms  0/ 89 (0%)
[ 3] 16.0-17.0 sec 128 KBytes  1.05 Mbits/sec  0.605 ms  0/ 89 (0%)
[ 3] 17.0-18.0 sec 128 KBytes  1.05 Mbits/sec  0.601 ms  0/ 89 (0%)
[ 3] 18.0-19.0 sec 128 KBytes  1.05 Mbits/sec  0.590 ms  0/ 89 (0%)
[ 3] 19.0-20.0 sec 129 KBytes  1.06 Mbits/sec  0.615 ms  0/ 90 (0%)
[ 3] 0.0-20.0 sec  2.50 MBytes  1.05 Mbits/sec  0.615 ms  0/ 1785 (0%)
root@Host02:/home/pi/iperf-2.0.5# ifconfig
```

```
pi@Host01: ~
root@Host01:/home/pi/iperf-2.0.5# iperf -c 192.168.2.2 -i 1 -t 20 -u
-----
Client connecting to 192.168.2.2, UDP port 5001
Sending 1470 byte datagrams, IPG target: 11215.21 us (kalman adjust)
UDP buffer size: 160 KByte (default)
-----
[ 3] local 192.168.2.1 port 39918 connected with 192.168.2.2 port 5001
[ ID] Interval      Transfer      Bandwidth
[ 3] 0.0- 1.0 sec  131 KBytes  1.07 Mbits/sec
[ 3] 1.0- 2.0 sec  128 KBytes  1.05 Mbits/sec
[ 3] 2.0- 3.0 sec  128 KBytes  1.05 Mbits/sec
[ 3] 3.0- 4.0 sec  128 KBytes  1.05 Mbits/sec
[ 3] 4.0- 5.0 sec  128 KBytes  1.05 Mbits/sec
[ 3] 5.0- 6.0 sec  128 KBytes  1.05 Mbits/sec
[ 3] 6.0- 7.0 sec  129 KBytes  1.06 Mbits/sec
[ 3] 7.0- 8.0 sec  128 KBytes  1.05 Mbits/sec
[ 3] 8.0- 9.0 sec  128 KBytes  1.05 Mbits/sec
[ 3] 9.0-10.0 sec  128 KBytes  1.05 Mbits/sec
[ 3] 10.0-11.0 sec 128 KBytes  1.05 Mbits/sec
[ 3] 11.0-12.0 sec 128 KBytes  1.05 Mbits/sec
[ 3] 12.0-13.0 sec 129 KBytes  1.06 Mbits/sec
[ 3] 13.0-14.0 sec 128 KBytes  1.05 Mbits/sec
[ 3] 14.0-15.0 sec 128 KBytes  1.05 Mbits/sec
[ 3] 15.0-16.0 sec 128 KBytes  1.05 Mbits/sec
[ 3] 16.0-17.0 sec 128 KBytes  1.05 Mbits/sec
[ 3] 17.0-18.0 sec 128 KBytes  1.05 Mbits/sec
[ 3] 18.0-19.0 sec 129 KBytes  1.06 Mbits/sec
[ 3] 19.0-20.0 sec 128 KBytes  1.05 Mbits/sec
[ 3] 0.0-20.0 sec  2.50 MBytes  1.05 Mbits/sec
[ 3] Sent 1785 datagrams
[ 3] WARNING: did not receive ack of last datagram after 10 tries.
```

```
pi@Host01: ~  
root@Host01:/home/pi/iperf-2.0.5# ping -c 20 192.168.2.2  
PING 192.168.2.2 (192.168.2.2) 56(84) bytes of data.  
64 bytes from 192.168.2.2: icmp_seq=1 ttl=64 time=16.1 ms  
64 bytes from 192.168.2.2: icmp_seq=2 ttl=64 time=4.39 ms  
64 bytes from 192.168.2.2: icmp_seq=3 ttl=64 time=3.95 ms  
64 bytes from 192.168.2.2: icmp_seq=4 ttl=64 time=3.18 ms  
64 bytes from 192.168.2.2: icmp_seq=5 ttl=64 time=3.30 ms  
64 bytes from 192.168.2.2: icmp_seq=6 ttl=64 time=3.34 ms  
64 bytes from 192.168.2.2: icmp_seq=7 ttl=64 time=2.68 ms  
64 bytes from 192.168.2.2: icmp_seq=8 ttl=64 time=3.09 ms  
64 bytes from 192.168.2.2: icmp_seq=9 ttl=64 time=3.30 ms  
64 bytes from 192.168.2.2: icmp_seq=10 ttl=64 time=3.27 ms  
64 bytes from 192.168.2.2: icmp_seq=11 ttl=64 time=3.19 ms  
64 bytes from 192.168.2.2: icmp_seq=12 ttl=64 time=3.95 ms  
64 bytes from 192.168.2.2: icmp_seq=13 ttl=64 time=3.05 ms  
64 bytes from 192.168.2.2: icmp_seq=14 ttl=64 time=3.16 ms  
64 bytes from 192.168.2.2: icmp_seq=15 ttl=64 time=3.64 ms  
64 bytes from 192.168.2.2: icmp_seq=16 ttl=64 time=3.16 ms  
64 bytes from 192.168.2.2: icmp_seq=17 ttl=64 time=3.40 ms  
64 bytes from 192.168.2.2: icmp_seq=18 ttl=64 time=2.92 ms  
64 bytes from 192.168.2.2: icmp_seq=19 ttl=64 time=3.11 ms  
64 bytes from 192.168.2.2: icmp_seq=20 ttl=64 time=3.59 ms  
--- 192.168.2.2 ping statistics ---  
20 packets transmitted, 20 received, 0% packet loss, time 19028ms  
rtt min/avg/max/mdev = 2.686/3.992/16.103/2.805 ms  
root@Host01:/home/pi/iperf-2.0.5#
```

3. Screenshoot Data ICMP



```
pi@raspberrypi: ~  
File Edit Tabs Help  
ons=drop  
table_id=254, duration=118s, n_packets=153, n_bytes=11846, priority=0, reg0=0x1, actions=controller(reason=)  
table_id=254, duration=118s, n_packets=0, n_bytes=0, priority=0, reg0=0x2, actions=drop  
table_id=254, duration=118s, n_packets=0, n_bytes=0, priority=0, reg0=0x3, actions=drop  
pi@raspberrypi: ~ $ sudo ovs-ofctl dump-flows sw0  
cookie=0x0, duration=28.799s, table=0, n_packets=29, n_bytes=2842, idle_timeout=10, hard_timeout=30, priority=65535, icmp, in_port=eth4, vlan_tci=0x0000, dl_src=b8:27:eb:d4:f4:8e, dl_dst=b8:27:eb:d4:f4:8e, nw_src=192.168.2.1, nw_dst=192.168.2.2, nw_tos=0, icmp_type=8, icmp_code=0 actions=output:eth3  
cookie=0x0, duration=28.791s, table=0, n_packets=29, n_bytes=2842, idle_timeout=10, hard_timeout=30, priority=65535, icmp, in_port=eth3, vlan_tci=0x0000, dl_src=b8:27:eb:d4:f4:8e, dl_dst=b8:27:eb:d4:f4:8e, nw_src=192.168.2.2, nw_dst=192.168.2.1, nw_tos=0, icmp_type=0, icmp_code=0 actions=output:eth4  
cookie=0x0, duration=8.842s, table=0, n_packets=1, n_bytes=81, idle_timeout=10, hard_timeout=30, priority=65535, udp, in_port=eth3, vlan_tci=0x0000, dl_src=b8:27:eb:d4:f4:8e, dl_dst=00:e0:4c:53:44:58, nw_src=192.168.2.2, nw_dst=192.168.1.3, nw_tos=0, tp_src=46132, tp_dst=53 actions=LOCAL  
cookie=0x0, duration=8.778s, table=0, n_packets=1, n_bytes=81, idle_timeout=10, hard_timeout=30, priority=65535, udp, in_port=eth4, vlan_tci=0x0000, dl_src=b8:27:eb:d4:f4:8e, dl_dst=00:e0:4c:53:44:58, nw_src=192.168.2.1, nw_dst=192.168.1.3, nw_tos=0, tp_src=34270, tp_dst=53 actions=LOCAL  
cookie=0x0, duration=3.877s, table=0, n_packets=1, n_bytes=81, idle_timeout=10, hard_timeout=30, priority=65535, udp, in_port=eth3, vlan_tci=0x0000, dl_src=b8:27:eb:d4:f4:8e, dl_dst=00:e0:4c:53:44:58, nw_src=192.168.2.2, nw_dst=8.8.8.8, nw_tos=0, tp_src=58836, tp_dst=53 actions=LOCAL  
cookie=0x0, duration=3.768s, table=0, n_packets=1, n_bytes=81, idle_timeout=10, hard_timeout=30, priority=65535, udp, in_port=eth4, vlan_tci=0x0000, dl_src=b8:27:eb:d4:f4:8e, dl_dst=00:e0:4c:53:44:58, nw_src=192.168.2.1, nw_dst=8.8.8.8, nw_tos=0, tp_src=57786, tp_dst=53 actions=LOCAL  
cookie=0x0, duration=0.850s, table=0, n_packets=1, n_bytes=81, idle_timeout=10, hard_timeout=30, priority=65535, udp, in_port=eth3, vlan_tci=0x0000, dl_src=b8:27:eb:d4:f4:8e, dl_dst=00:e0:4c:53:44:58, nw_src=192.168.2.2, nw_dst=192.168.1.3, nw_tos=0, tp_src=59861, tp_dst=53 actions=LOCAL  
cookie=0x0, duration=0.705s, table=0, n_packets=1, n_bytes=81, idle_timeout=10, hard_timeout=30, priority=65535, udp, in_port=eth4, vlan_tci=0x0000, dl_src=b8:27:eb:d4:f4:8e, dl_dst=00:e0:4c:53:44:58, nw_src=192.168.2.1, nw_dst=192.168.1.3, nw_tos=0, tp_src=56194, tp_dst=53 actions=LOCAL  
pi@raspberrypi: ~ $
```

4. Screenshot tabel flow

```
hard_timeout=30, priority=65535,arp,dl_port=LOCAL,vlan_tci=0x0000,dl_src=00:00:00:00:00:00,dl_dst=b8:27:eb:d4:f4:8e,arp_spa=192.168.1.3,arp_tpa=192.168.2.2,arp_op=2,actions=output:eth2
pi@raspberrypi:~$ sudo ovs-ofctl dump-flows sw0
cookie=0x0, duration=0.869s, table=0, n_packets=1, n_bytes=81, idle_timeout=10, hard_timeout=30, priority=65535,udp,in_port=eth2,vlan_tci=0x0000,dl_src=b8:27:eb:d4:f4:8e,dl_dst=00:00:00:00:00:00,mw_src=192.168.2.2,mw_dst=192.168.1.3,mw_tos=0,tp_src=47374,tp_dst=53 actions=LOCAL
cookie=0x0, duration=7.080s, table=0, n_packets=1, n_bytes=81, idle_timeout=10, hard_timeout=30, priority=65535,udp,in_port=eth1,vlan_tci=0x0000,dl_src=b8:27:eb:d4:22:f0,dl_dst=00:00:00:00:00:00,mw_src=192.168.2.1,mw_dst=8.8.8.8,mw_tos=0,tp_src=52153,tp_dst=53 actions=LOCAL
cookie=0x0, duration=4.061s, table=0, n_packets=1, n_bytes=81, idle_timeout=10, hard_timeout=30, priority=65535,udp,in_port=eth1,vlan_tci=0x0000,dl_src=b8:27:eb:d4:22:f0,dl_dst=00:00:00:00:00:00,mw_src=192.168.2.1,mw_dst=192.168.1.3,mw_tos=0,tp_src=59649,tp_dst=53 actions=LOCAL
cookie=0x0, duration=3.831s, table=0, n_packets=1, n_bytes=81, idle_timeout=10, hard_timeout=30, priority=65535,udp,in_port=eth2,vlan_tci=0x0000,dl_src=b8:27:eb:d4:f4:8e,dl_dst=00:00:00:00:00:00,mw_src=192.168.2.2,mw_dst=8.8.8.8,mw_tos=0,tp_src=56544,tp_dst=53 actions=LOCAL
cookie=0x0, duration=3.758s, table=0, n_packets=5, n_bytes=490, idle_timeout=10, hard_timeout=30, priority=65535,icmp,in_port=eth1,vlan_tci=0x0000,dl_src=b8:27:eb:d4:22:f0,dl_dst=b8:27:eb:d4:f4:8e,mw_src=192.168.2.1,mw_dst=192.168.2.2,mw_tos=0,icmp_type=8,icmp_code=0 actions=output:eth2
cookie=0x0, duration=3.749s, table=0, n_packets=5, n_bytes=490, idle_timeout=10, hard_timeout=30, priority=65535,icmp,in_port=eth2,vlan_tci=0x0000,dl_src=b8:27:eb:d4:f4:8e,dl_dst=b8:27:eb:d4:22:f0,mw_src=192.168.2.2,mw_dst=192.168.2.1,mw_tos=0,icmp_type=0,icmp_code=0 actions=output:eth1
cookie=0x0, duration=0.809s, table=0, n_packets=1, n_bytes=81, idle_timeout=10, hard_timeout=30, priority=65535,udp,in_port=eth2,vlan_tci=0x0000,dl_src=b8:27:eb:d4:f4:8e,dl_dst=00:00:00:00:00:00,mw_src=192.168.2.2,mw_dst=192.168.1.3,mw_tos=0,tp_src=39837,tp_dst=53 actions=LOCAL
pi@raspberrypi:~$
```

5. Screenshot Data load Average

```
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\AURAEVA>cd Documents\Rizky

C:\Users\AURAEVA\Documents\Rizky>cd Documents\Rizky
The system cannot find the path specified.

C:\Users\AURAEVA\Documents\Rizky>python test_ssh.py
2018-12-27 12:37:50.400779
top - 18:20:37 up 2:03, 2 users, load average: 0.23, 0.43, 0.47

Tasks: 164 total, 3 running, 161 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.6 us, 1.9 sy, 0.0 ni, 96.4 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st

2018-12-27 12:37:52.046367
top - 18:20:38 up 2:03, 2 users, load average: 0.37, 0.46, 0.48

Tasks: 164 total, 1 running, 163 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.6 us, 1.9 sy, 0.0 ni, 96.4 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st

2018-12-27 12:37:53.359253
top - 18:20:40 up 2:03, 2 users, load average: 0.37, 0.46, 0.48

Tasks: 164 total, 2 running, 162 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.6 us, 1.9 sy, 0.0 ni, 96.4 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st

2018-12-27 12:37:54.686676
top - 18:20:41 up 2:03, 2 users, load average: 0.37, 0.46, 0.48

Tasks: 164 total, 2 running, 162 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.6 us, 1.9 sy, 0.0 ni, 96.4 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st

2018-12-27 12:37:56.110792
top - 18:20:42 up 2:03, 2 users, load average: 0.37, 0.46, 0.48

Tasks: 164 total, 1 running, 163 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.6 us, 1.9 sy, 0.0 ni, 96.4 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st

2018-12-27 12:37:57.472590
top - 18:20:44 up 2:03, 2 users, load average: 0.34, 0.45, 0.48
```

```
Tasks: 164 total, 1 running, 163 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.6 us, 1.9 sy, 0.0 ni, 96.4 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st

2018-12-27 12:37:57.472590
top - 18:20:44 up 2:03, 2 users, load average: 0.34, 0.45, 0.48

Tasks: 164 total, 1 running, 163 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.6 us, 1.9 sy, 0.0 ni, 96.4 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st

2018-12-27 12:37:58.744292
top - 18:20:45 up 2:03, 2 users, load average: 0.34, 0.45, 0.48

Tasks: 164 total, 1 running, 163 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.6 us, 1.9 sy, 0.0 ni, 96.4 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st

2018-12-27 12:38:00.051288
top - 18:20:46 up 2:03, 2 users, load average: 0.34, 0.45, 0.48

Tasks: 164 total, 2 running, 162 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.6 us, 1.9 sy, 0.0 ni, 96.4 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st

2018-12-27 12:38:01.564581
top - 18:20:48 up 2:03, 2 users, load average: 0.34, 0.45, 0.48

Tasks: 164 total, 3 running, 161 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.6 us, 1.9 sy, 0.0 ni, 96.4 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st

2018-12-27 12:38:02.915513
top - 18:20:49 up 2:03, 2 users, load average: 0.79, 0.54, 0.51

Tasks: 164 total, 2 running, 162 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.6 us, 1.9 sy, 0.0 ni, 96.4 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st

2018-12-27 12:38:04.236542
top - 18:20:50 up 2:03, 2 users, load average: 0.79, 0.54, 0.51

Tasks: 164 total, 1 running, 163 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.6 us, 1.9 sy, 0.0 ni, 96.4 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st
```

```
❏ Select Command Prompt
2018-12-27 12:38:09.530321
top - 18:20:56 up 2:03, 2 users, load average: 0.89, 0.57, 0.52
Tasks: 164 total, 1 running, 163 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.7 us, 1.9 sy, 0.0 ni, 96.3 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st
2018-12-27 12:38:10.843164
top - 18:20:57 up 2:03, 2 users, load average: 0.89, 0.57, 0.52
Tasks: 164 total, 2 running, 162 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.7 us, 1.9 sy, 0.0 ni, 96.3 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st
2018-12-27 12:38:12.154491
top - 18:20:58 up 2:04, 2 users, load average: 0.98, 0.59, 0.52
Tasks: 164 total, 2 running, 162 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.7 us, 1.9 sy, 0.0 ni, 96.3 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st
2018-12-27 12:38:13.451191
top - 18:21:00 up 2:04, 2 users, load average: 0.98, 0.59, 0.52
Tasks: 164 total, 1 running, 163 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.7 us, 1.9 sy, 0.0 ni, 96.3 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st
2018-12-27 12:38:14.754508
top - 18:21:01 up 2:04, 2 users, load average: 0.98, 0.59, 0.52
Tasks: 164 total, 2 running, 162 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.7 us, 1.9 sy, 0.0 ni, 96.3 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st
2018-12-27 12:38:16.060791
top - 18:21:02 up 2:04, 2 users, load average: 0.98, 0.59, 0.52
Tasks: 164 total, 1 running, 163 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.7 us, 1.9 sy, 0.0 ni, 96.3 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st
2018-12-27 12:38:17.331287
top - 18:21:03 up 2:04, 2 users, load average: 0.90, 0.58, 0.52
Tasks: 164 total, 1 running, 163 sleeping, 0 stopped, 0 zombie
```

```
2018-12-27 12:38:18.575153
top - 18:21:05 up 2:04, 2 users, load average: 0.90, 0.58, 0.52
Tasks: 164 total, 1 running, 163 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.7 us, 1.9 sy, 0.0 ni, 96.3 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st
2018-12-27 12:38:19.857613
top - 18:21:06 up 2:04, 2 users, load average: 0.90, 0.58, 0.52
Tasks: 164 total, 2 running, 162 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.7 us, 1.9 sy, 0.0 ni, 96.3 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st
2018-12-27 12:38:21.163219
top - 18:21:07 up 2:04, 2 users, load average: 0.90, 0.58, 0.52
Tasks: 164 total, 1 running, 163 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.7 us, 1.9 sy, 0.0 ni, 96.3 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st
2018-12-27 12:38:22.466329
top - 18:21:09 up 2:04, 2 users, load average: 0.83, 0.57, 0.52
Tasks: 164 total, 1 running, 163 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.7 us, 1.9 sy, 0.0 ni, 96.3 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st
2018-12-27 12:38:23.768254
top - 18:21:10 up 2:04, 2 users, load average: 0.83, 0.57, 0.52
Tasks: 164 total, 1 running, 163 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.7 us, 1.9 sy, 0.0 ni, 96.3 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st
2018-12-27 12:38:25.010979
top - 18:21:11 up 2:04, 2 users, load average: 0.83, 0.57, 0.52
Tasks: 164 total, 1 running, 163 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.7 us, 1.9 sy, 0.0 ni, 96.3 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st
2018-12-27 12:38:26.312801
top - 18:21:12 up 2:04, 2 users, load average: 0.83, 0.57, 0.52
```

```

2018-12-27 12:38:26.312801
top - 18:21:12 up 2:04, 2 users, load average: 0.83, 0.57, 0.52
Tasks: 164 total, 2 running, 162 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.7 us, 1.9 sy, 0.0 ni, 96.3 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st
2018-12-27 12:38:27.618552
top - 18:21:14 up 2:04, 2 users, load average: 1.08, 0.63, 0.54
Tasks: 164 total, 2 running, 162 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.7 us, 1.9 sy, 0.0 ni, 96.3 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st
2018-12-27 12:38:28.940408
top - 18:21:15 up 2:04, 2 users, load average: 1.08, 0.63, 0.54
Tasks: 164 total, 2 running, 162 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.7 us, 1.9 sy, 0.0 ni, 96.3 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st
2018-12-27 12:38:30.245633
top - 18:21:16 up 2:04, 2 users, load average: 1.08, 0.63, 0.54
Tasks: 164 total, 2 running, 162 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.7 us, 1.9 sy, 0.0 ni, 96.3 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st
2018-12-27 12:38:31.568335
top - 18:21:18 up 2:04, 2 users, load average: 1.08, 0.63, 0.54
Tasks: 164 total, 1 running, 163 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.7 us, 1.9 sy, 0.0 ni, 96.3 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st
2018-12-27 12:38:32.866122
top - 18:21:19 up 2:04, 2 users, load average: 1.00, 0.62, 0.53
Tasks: 164 total, 1 running, 163 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.7 us, 1.9 sy, 0.0 ni, 96.3 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st
2018-12-27 12:38:34.208836
top - 18:21:20 up 2:04, 2 users, load average: 1.00, 0.62, 0.53

```

```

2018-12-27 12:38:44.878114
top - 18:21:31 up 2:04, 2 users, load average: 1.31, 0.70, 0.56
Tasks: 164 total, 1 running, 163 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.7 us, 1.9 sy, 0.0 ni, 96.2 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st
2018-12-27 12:38:46.156505
top - 18:21:32 up 2:04, 2 users, load average: 1.31, 0.70, 0.56
Tasks: 164 total, 2 running, 162 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.8 us, 1.9 sy, 0.0 ni, 96.2 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st
2018-12-27 12:38:47.472771
top - 18:21:34 up 2:04, 2 users, load average: 1.37, 0.72, 0.57
Tasks: 164 total, 2 running, 162 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.8 us, 2.0 sy, 0.0 ni, 96.2 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st
2018-12-27 12:38:48.852365
top - 18:21:35 up 2:04, 2 users, load average: 1.37, 0.72, 0.57
Tasks: 164 total, 1 running, 163 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.8 us, 2.0 sy, 0.0 ni, 96.2 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st
2018-12-27 12:38:50.203288
top - 18:21:36 up 2:04, 2 users, load average: 1.37, 0.72, 0.57
Tasks: 164 total, 1 running, 163 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.8 us, 2.0 sy, 0.0 ni, 96.2 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st
2018-12-27 12:38:51.557935
top - 18:21:38 up 2:04, 2 users, load average: 1.37, 0.72, 0.57
Tasks: 164 total, 1 running, 163 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.8 us, 2.0 sy, 0.0 ni, 96.1 id, 0.0 wa, 0.0 hi, 0.1 si, 0.0 st
2018-12-27 12:38:52.869551
top - 18:21:39 up 2:04, 2 users, load average: 1.42, 0.74, 0.58

```

6. Modul Praktikum

CHAPTER 2
IMPLEMENTASI *OPENFLOW* PADA *RASPBERRY PI*

1. Tujuan Praktikum

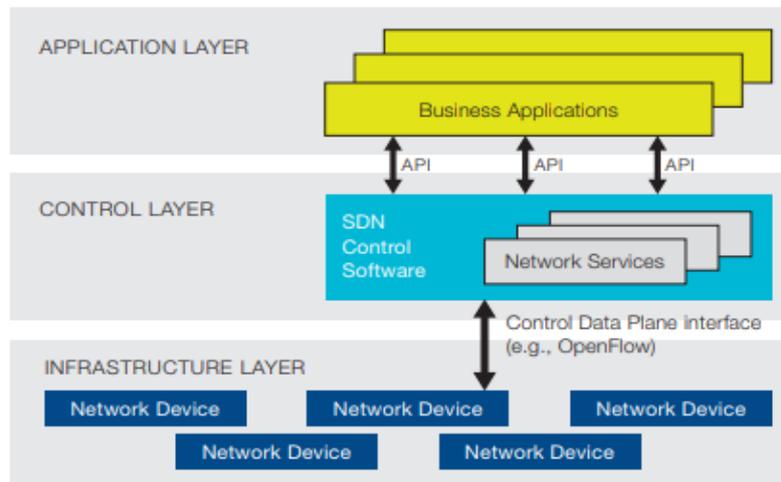
Tujuan dari praktikum ini adalah untuk mengetahui dan mempelajari cara kerja dari SDN (Software Defined Network), *OpenFlow* dan POX controller yang di terapkan pada topologi sederhana. Serta mengetahui kinerja *OpenFlow* pada *Raspberry PI*.

2. Teori Dasar

2.1. Software Defined Network

Software Defined Network (SDN) adalah jaringan komputer yang memisahkan *controller* dan *data plane* dalam perangkat jaringan konvensional. Berkat pemisahan ini, SDN dapat memberikan tampilan jaringan terpusat secara logis dalam satu titik pengelolaan. Hal ini dicapai melalui *open interface* dan abstraksi fungsi tingkat rendah dan mengubah

jaringan menjadi *platform* yang dapat diprogram secara dinamis menyesuaikan tingkah lakunya.



Gambar dibawah ini adalah arsitektur SDN, memiliki 3 layer yaitu :

Gambar 2.1. Arsitektur SDN

- a. Application layer : berupa aplikasi yang dapat secara langsung dan eksplisit mendefinisikan network requirement dan network behavior yang diinginkan. Layer ini berkomunikasi dengan Control Layer melalui NorthBound Interface (NBI).
- b. Control layer : entitas kontrol yang memiliki tugas mentranslasikan network requirement yang telah didefinisikan oleh application layer menjadi instruksi-instruksi untuk infrastructure layer dan memberikan abstract view yang dibutuhkan application layer (abstract view yaitu informasi statistik dan event yang terjadi di jaringan).

- c. Infrastructure layer : terdiri dari elemen jaringan yang dapat menerima instruksi dari Control Layer. Interface antara Controller layer dan Data Plane disebut SouthBound Interface (SBI) atau Control-To-Data-Plane Interface (CDPI).

Keuntungan menggunakan Software Defined Network yaitu :

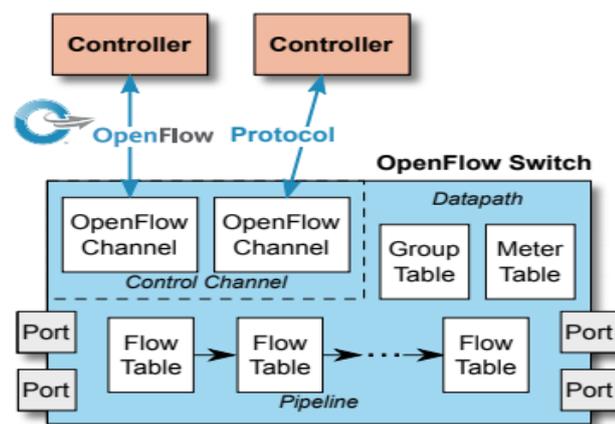
6. Operator jaringan dan administrator dapat secara terprogram mengkonfigurasi jaringan yang disederhanakan. Jadi tidak perlu menkonfigurasi yang ribuan perangkat.
7. Dapat merubah tingkah laku jaringan secara real time serta dapat menyebarkan aplikasi baru dan layanan jaringan dalam hitungan jam atau hari.
8. Dapat memberikan fleksibilitas kepada pengelola jaringan untuk mengkonfigurasi, mengelola, mengamankan dan mengoptimalkan sumberdaya jaringan melalui program SDN yang dinamis dan otomatis
9. Mendukung satu set API yang memungkinkan untuk mengimplementasikan layanan jaringan umum, termasuk routing, multicast, keamanan, kontrol akses, manajemen bandwidth, rekayasa lalu lintas, kualitas layanan, prosesor dan optimasi penyimpanan, penggunaan

energi, dan semua bentuk kebijakan manajemen, kustom disesuaikan untuk memenuhi tujuan bisnis

10. SDN membuat jaringan tidak begitu banyak “application-aware” sebagai “application-customized” dan aplikasi tidak begitu banyak “network-aware” sebagai “network-capability-aware”. Akibatnya, komputasi, penyimpanan, dan sumber daya jaringan dapat dioptimalkan.

2.2. OpenFlow

Pada tahun 2008 Stanford University menyimpulkan bahwa jaringan menjadi infrastruktur yang sangat penting. Namun, inovasi jaringan sangat terhambat oleh karena itu sebagai solusi ide virtualisasi jaringan di usulkan di bagian produksi dan eksperimetal untuk membuat penelitian *OpenFlow*.



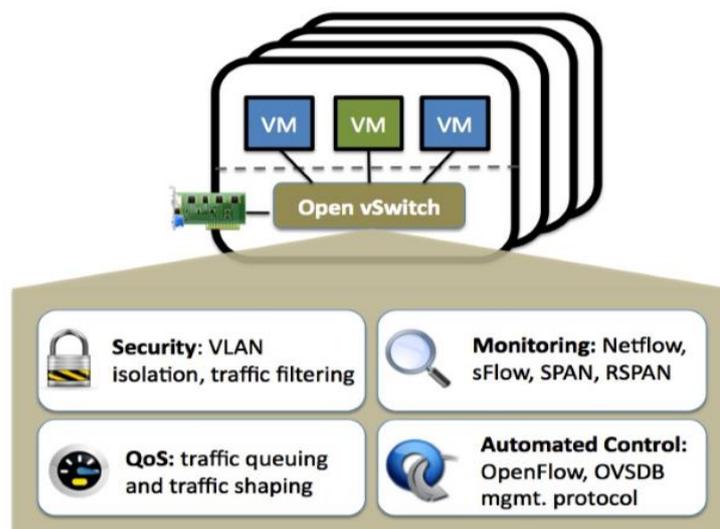
Gambar 2.2. Komponen utama dari Switch *OpenFlow*

Komponen utama switch *OpenFlow* v.1.5.1 adalah sebagai berikut

- e. *OpenFlow Channel* : Interface yang menghubungkan setiap *OpenFlow* switch ke *OpenFlow* controller. Melalui interface ini dapat mengkonfigurasi controller, mengelola switch, menerima event dari switch dan mengirim paket keluar. Pengontrol channel switch dapat mendukung single *OpenFlow* channel dengan tunggal controller atau beberapa *OpenFlow* channel yang memungkinkan beberapa controller untuk berbagi management switch.
- f. *Group Table* : Sebuah group table terdiri group entri. Dimana flow entri memiliki kemampuan untuk mengarahkan group untuk memungkinkan *OpenFlow* mewakili metode forwarding.
- g. *Meter Table* : Terdiri dari entri meter yang berfungsi untuk mendefinisikan flow meter. Flow meter memungkinkan *OpenFlow* untuk mengimplementasikan berbagai simple operasi QoS, seperti rate-limiting dan dapat di kombinasikan port queue untuk mengimplementasikan kerangka kerja QoS yang kompleks, seperti Diffserv.
- h. *Flow Table* : Terdiri dari match fields berfungsi untuk mencocokkan paket, priority untuk mencari kecocokan dengan flow entry, counter untuk melakukan update saat pencocokan, instruction untuk untuk memodifikasi set action atau pemrosesan pipeline, timeout adalah jumlah waktu maksimum atau idle sebelum flow kadaluarsa oleh switch, cookie adalah

nilai data opaque yang dipilih controller, dan flag berfungsi untuk mengubah manajemen flow entrie.

2.3. OpenvSwitch



Gambar 2.3. OpenvSwitch

OpenvSwitch adalah perangkat lunak multilayer yang berlisensi dibawah lisensi opensource apache2. Tujuannya adalah untuk mengimplementasikan platform switch yang mendukung standar interface

management dan membuka fungsi forwarding ke ekstensi yang terprogram dan terkontrol.

Openvswitch sangat berfungsi untuk switch virtual di VM. Selain mengekspos standar control dan interface visibilitas ke virtual networking layer, juga di rancang untuk mendukung distribusi di beberapa server fisik. OpenvSwitch mendukung beberapa teknologi virtualisasi berbasis linux termasuk Xen/Xen Server, KVM dan Virtual Box.

Sebagian besar kode ditulis dalam bahasa C dan mudah dipindah ke bahasa lain. OpenvSwitch yang rilis saat ini mendukung fitur-fitur berikut :

- j. Standard 802.1Q model VLAN dengan trunk dan access ports
- k. NIC bonding dengan atau tanpa LACP pada upstream switch
- l. NetFlow, sFlow(R), dan pencerminan untuk peningkatan visibility
- m. QoS (Quality of Service) konfigurasi, dan policing
- n. Geneve, GRE, VXLAN, STT, dan LISP tunneling
- o. 802.1ag manajemen kesalahan konektivitas
- p. *OpenFlow* 1.0 ditambahkan banyak ekstensi
- q. Konfigurasi database transactional dengan bahasa C dan Python bindings

- r. Kinerja forwarding menjadi lebih baik dengan menggunakan modul kernel linux (hanya mendukung versi linux 3.10 dan versi lebih baru)

OpenvSwitch juga dapat beroperasi sepenuhnya di userspace tanpa bantuan dari modul kernel. Implementasi userspace harus lebih mudah ke port daripada switch berbasis kernel. OpenvSwitch didalam userspace dapat mengakses linux atau perangkat DPDK (Data Plane Development Kit). OpenvSwitch dengan userspace data path dan perangkat non DPK dianggap sebagai eksperimental.

Komponen utama OpenvSwitch yaitu :

- g. ovs-vswitchd : Sebuah daemon yang mengimplementasikan switch, bersama dengan modul kernel linux untuk flowbased switch.
- h. ovsdb-server : Server data base yang ringan bahwa ovs-vswitchd memiliki pertanyaan untuk memperoleh konfigurasinya.
- i. ovs-dpctl : alat untuk mengkonfigurasi modul kernel switch
- j. script and spec: Membangun RPM untuk Citrix XenServer dan Red Hat Enterprise Linux. RPM XenServer memungkinkan OpenvSwitch untuk di instal pada host Citrix XenServer sebagai pengganti drop-in untuk switch.
- k. ovs-vsctl : Kegunaanya untuk query dan memperbarui konfigurasi ovs-vswitchd.

1. `ovs-appctl` : Mengirimkan perintah untuk menjalankan daemon `openvswitch`

OpenvSwitch juga menyediakan beberapa tool, yaitu :

- c. `ovs-ofctl` : Kegunaanya untuk query dan mengontrol switch *OpenFlow* dan controller
- d. `ovs-pki` : Kegunaanya untuk membuat dan mengelola infrastruktur `public-key`.

3. Pengenalan Controller

Controller SDN adalah entitas perangkat lunak yang memiliki kontrol eksklusif atas serangkaian abstraksi dari sumber daya data plane. Controller SDN juga dapat menawarkan contoh model informasi yang diabstraksi, setidaknya ke satu klien. Kontroler SDN dapat diimplementasikan dengan sejumlah komponen perangkat lunak, yang berada pada sejumlah platform fisik.

POX adalah platform yang digunakan untuk pengembangan dan pemodelan pada network kontrol software. POX menggunakan python dalam bahasa pemrogramannya. POX bekerja pada layer control plan sebagai sebuah network controller. Dalam arsitektur RouteFlow POX berada pada RFPProxy yang bertanggung jawab untuk interaksi dengan *OpenFlow* switch (datapath) melalui protokol *OpenFlow*.

4. Pengenalan Sistem Operasi CentOS

CentOS (Community ENTERprise Operating System) adalah sebuah operasi sistem bawaan linux. Awalnya dibangun dengan nama Red Hat Enterprise Linux (RHEL). Untuk mengunduh RHEL harus memiliki akun berlangganan RHEL yang aktif dan tidak murah. RHEL sepenuhnya didasarkan pada perangkat lunak open source. Sehingga RHEL berkewajiban untuk melepaskan semua sumber kodenya kepada siapa saja yang berlangganan.

Namun, RHEL membuat sumber kode yang tersedia bagi siapa saja. Dengan menggunakan undang-undang perlindungan merek dagang untuk mencegah orang membangun RHEL versi lain. Kemudian mendistribusikannya dengan berbagai cara dan dilakukan banyak penelitian untuk menghapus semua logo merek dagang Red Hat dan menjadikan namanya CentOS. CentOS dapat diunduh secara gratis dari situs web CentOS.

CentOS dan RHEL memiliki banyak kesamaan, Versi baru RHEL dirilis setiap 18-24 bulan. Sedangkan CentOS sekitar sebulan setelah tanggal rilis RHEL karena proyek CentOS harus membangun dan melakukan pengujian kembali. Proyek CentOS terdiri dari sukarelawan yang melakukan rekondisi ini, menguji pembaruan, dan memberikan dukungan.

Pembaruan kecil juga dilakukan setiap enam bulan atau lebih, dan ini termasuk semua pembaruan yang dari rilis awal, dukungan perangkat

keras tambahan, dan setiap peningkatan teknologi atau fitur baru. CentOS memiliki komunitas dunia yang sangat kuat dengan lebih dari 6 juta sistem terdaftar. Jadi banyak sistem yang diinstal dari disk yang sama dan banyak perusahaan yang mengatur repositori lokal mereka sendiri untuk pembaruan, sehingga diperkirakan ada lebih banyak sistem CentOS di luar perkiraan.

Berikut perangkat lunak repository yang tersedia :

3. The CentOS Plus repository : berisi paket yang menggantikan paket asli yang dibangun kembali dari kode sumber RHEL. Ini memecah kompatibilitas biner, jadi bisa mendapatkan perangkat lunak baru yang tidak tersedia.
4. The CentOS Extras repository : berisi paket yang digunakan selain paket asli. Menggunakan repositori ini tidak akan merusak kompatibilitas biner, karena tidak akan memperbarui paket asli.

Tentu saja, ada beberapa kerugian ketika menggunakan CentOS di atas RHEL, yaitu :

4. CentOS sebagai proyek komunitas tidak dapat memberikan tingkat dukungan yang sama dengan entitas komersial seperti Red Hat
5. CentOS harus menerima pembaruan RHEL sebelum pembaruan CentOS dapat dibuat. Namun, penundaan ini tidak signifikan, dan meskipun Proyek CentOS memiliki target 72 jam, pembaruan biasanya dirilis dalam 24 jam.

6. CentOS tidak memiliki semua sertifikasi perangkat keras dan perangkat lunak yang tersedia di RHEL
7. CentOS tidak menawarkan program ganti rugi jika ada IP atau kode yang melanggar.
- 8.

5. Pengenalan *Raspberry PI*

5.1. *Raspberry PI*

Raspberry PI adalah sebuah SBC (Single Board Computer) seukuran kartu kredit yang dikembangkan oleh Yayasan *Raspberry PI* di Inggris (UK) dengan maksud untuk memicu pengajaran ilmu pengetahuan dasar komputer disekolah-sekolah. *Raspberry PI* diproduksi melalui lisensi manufaktur yang berkaitan dengan elemen 14/Premier Farnell dan RS komponen.



Gambar 5.1. *Raspberry PI*

5.2. *Raspberry PI 3 Model B*

Raspberry PI 3 adalah generasi ketiga dari *Raspberry PI*, menggantikan *Raspberry PI 2 Model B* pada Februari 2016. *Raspberry PI3* memiliki bentuk yang identik dengan *Raspberry PI 2* sebelumnya (dan *Pi 1 Model B +*) dan memiliki kompatibilitas lengkap dengan *Raspberry PI 1* dan *2*. Pada perangkat terbarunya ini Raspberry menambahkan fitur built- in *wireless* dan *processor* yang lebih bertenaga yang belum pernah dimiliki pada versi sebelumnya. Spesifikasi t dapat dilihat pada tabel 5.1

Tabel 5.1 Spesifikasi *Raspberry PI Model B*

Spesifikasi	Keterangan
SOC (System On Chip)	BCM2837
Prosesor	1.2GHz 64-bit quad-core ARMv8 CPU
Memory/RAM	1 GB SDRAM 400MHz
GPU	VideoCore IV 3D graphics core
Wireless Adapter/LAN	802.1 In Wireless LAN
Bluetooth	Bluetooth 4.1 (built in), Bluetooth Low Energy(BLE)
GPIO	40 Pin
Port USB	4 USB Port

Card Stroge	Micro SD card slot (now push-pull rather than pushpush)
Port Ethernet	1 Ethernet Port
External Audio dan Video	Full HDMI port, Camera interface (CSI), Display interface (DSI), Combined 3.5mm audio jack and composite video
Power Supply	Micro USB Power Input Up to 2.5A

5.3. Sistem Operasi Rasberry PI

Sistem operasi yang digunakan oleh *Raspberry PI* biasa disebut Rasbian. Sistem operasi bebas berbasis Debian GNU / LINUX dan dioptimalkan untuk perangkat keras *Raspberry PI* (arsitektur prosesor ARMHF). Rasbian dilengkapi dengan lebih dari 35.000 paket, atau perangkat lunak precompiled paket dalam format yang bagus untuk kemudahan instalasi pada *Raspberry PI*. Awal di rilis sejak Juni 2012, menjadi distribusi yang terus aktif dikembangkan dengan penekanan pada peningkatan stabilitas dan kinerja sebanyak mungkin. Meskipun Debian menghasilkan distribusi untuk arsitektur lengan, Rasbian hanya kompatibel dengan versi yang lebih baru dari yang digunakan pada *Raspberry PI* (ARMv7 CPU-A dan vs *Raspberry PI* ARMv6 CPU yang lebih tinggi).

6. Pengenalan Aplikasi

6.1. Oracel VM VirtualBox

Oracle VM VirtualBox adalah perangkat lunak virtualisasi cross-platform yang dapat menjalankan beberapa sistem operasi seperti windows, mac, dan linux pada saat yang bersamaan sehingga memperluas kemampuan komputer yang ada. Didesain untuk digunakan pada sistem mulai dari ultra-book hingga hardware kelas server high-end, Oracle VM VirtualBox ringan dan mudah dipasang dan digunakan.

Berikut beberapa keuntungan menggunakan Oracle virtual box :

5. Menjalankan beberapa sistem operasi secara bersamaan : VirtualBox dapat menjalankan lebih dari satu sistem operasi secara bersamaan tanpa harus reboot untuk menggunakannya
6. Instalasi perangkat lunak lebih mudah : Dengan VirtualBox, pengaturan yang begitu kompleks dapat dimasukkan ke dalam mesin virtual.
7. Pengujian dan pemulihan kerusakan : Dengan cara ini dapat dengan bebas bereksperimen dengan lingkungan komputasi. Jika ada yang salah dapat dengan mudah beralih kembali ke pengaturan sebelumnya dan menghindari backup dan restores yang terlalu sering.
8. Konsolidasi infrastruktur : Virtualisasi dapat secara signifikan mengurangi biaya perangkat keras dan listrik.

Fitur utama Oracle VM virtual box :

5. Portabilitas : VirtualBox berjalan pada sejumlah sistem operasi 32-bit dan 64-bit. VirtualBox adalah apa yang disebut "host" hypervisor atau hypervisor tipe 2. VirtualBox memerlukan sistem operasi yang sudah ada untuk diinstal. Dengan demikian dapat berjalan bersama aplikasi yang ada pada host itu.
6. Tidak diperlukan virtualisasi hardware : VirtualBox tidak memerlukan fitur prosesor yang dibangun ke dalam perangkat keras yang lebih baru seperti Intel VT-x atau AMD-V. Berbeda dengan banyak virtualisasi lainnya, dapat menggunakan VirtualBox bahkan pada perangkat keras yang lebih lama di mana fitur-fitur ini tidak ada.
7. Guest Addition : VirtualBox Guest Additions adalah paket perangkat lunak yang dapat diinstal di dalam sistem tamu yang didukung untuk meningkatkan kinerja mereka dan untuk memberikan integrasi dan komunikasi tambahan dengan sistem host.
8. Dukungan perangkat keras yang hebat. VirtualBox mendukung:
 - g. Guest Multiprocessing : VirtualBox dapat menyajikan hingga 32 CPU virtual ke setiap mesin virtual, terlepas dari berapa banyak core CPU yang secara fisik ada pada host Anda.
 - h. Dukungan perangkat USB : VirtualBox mengimplementasikan pengendali USB virtual dan memungkinkan untuk menghubungkan perangkat USB ke mesin virtual tanpa harus menginstal driver

khusus pada host. Dukungan USB tidak terbatas pada kategori perangkat tertentu.

- i. Dukungan ACPI : Advanced Configuration and Power Interface (ACPI) didukung penuh oleh VirtualBox. Ini memudahkan kloning gambar PC dari mesin asli sehingga mesin virtual pihak tiga menjadi VirtualBox. Dengan dukungan status daya ACPI yang unik, VirtualBox bahkan dapat melaporkan ke sistem operasi tamu yang mengetahui ACPI status daya dari host. Untuk sistem seluler yang menggunakan baterai, tamu dapat mengaktifkan penghematan energi dan memberi tahu pengguna tentang kekuatan yang tersisa.
- j. Resolusi Multilayer : VirtualBox mesin virtual mendukung resolusi multilayer dari layar fisik, memungkinkan untuk tersebar di sejumlah besar layar yang terpasang ke sistem host.
- k. Dukungan iSCSI internal : Fitur unik ini memungkinkan untuk menghubungkan mesin virtual langsung ke server penyimpanan iSCSI tanpa melalui sistem host. VM mengakses target iSCSI secara langsung tanpa tambahan biaya yang diperlukan untuk mem-virtualisasi hard disk dalam file container.
- l. PXE Network boot : Kartu virtual network terintegrasi dari VirtualBox sepenuhnya mendukung booting jarak jauh melalui Preboot Execution Environment (PXE).

6.2. VNC

Virtual Network Computing (VNC) merupakan sebuah software remote kontrol, dimana dengan melalui software ini suatu komputer dapat melakukan akses untuk bekerja di suatu komputer lain yang terhubung dengan jaringan. Pada grafis desktop sharing sistem yang menggunakan remote Bingkai Buffer protocol (RFB) untuk kontrol jarak jauh dengan computer lain. Ini mentransmisikan *keyboard* dan *mouse* dari satu komputer ke komputer lain, merelay grafis layar pembaruan kembali ke arah yang lain melalui jaringan. Perangkat lunak ini terdiri dari sebuah server dan aplikasi klien untuk mengontrol layar komputer lain dari jarak jauh.

6.3. Angry IP Scanner

Angry IP Scanner adalah pemindai alamat IP dan port yang sangat cepat dengan rentang alamat IP apapun. Termasuk cross platform yang sangat ringan, sehingga mudah digunakan dimana saja. Juga memiliki fungsi seperti informasi NetBIOS (nama computer, nama workgroup), rentang alamat ip favorit, deteksi server web, dll.

Hasil pemindaian dapat disimpan ke file daftar CSV, TXT, XML atau IP-Port. Dengan bantuan plugin, Angry IP Scanner dapat mengumpulkan informasi apa pun tentang IP yang dipindai. Untuk meningkatkan kecepatan pemindai, menggunakan pendekatan multithread. Yaitu sebuah thread pemindaian terpisah yang dibuat untuk setiap alamat IP yang dipindai. [15]

6.4.iPerf

iPerf dikembangkan secara oronal oleh NLANR atau DAST. Namun pada prinsipnya, iPerf dikembangkan oleh ESnet atau Lawrence Berkeley National Laboratory dan dirilis di bawah lisensi BSD three-clause.

iPerf adalah perangkat lunak untuk pengukuran aktif dari bandwidth maksimum yang dapat dicapai pada jaringan. Mendukung penyetelan berbagai parameter yang terkait dengan pengaturan waktu, buffer dan protokol (TCP, UDP, SCTP dengan IPv4 dan IPv6). Setiap tes itu melaporkan bandwidth, kerugian, dan parameter lainnya.

Berikut beberapa fitur iPerf :

7. TCP dan SCTP
 - d. Mengukur bandwidth
 - e. Melaporkan ukuran MSS/MTU dan mengukur ukuran baca
 - f. Mendukung ukuran TCP melalui socket buffer.
8. UDP
 - e. Klien dapat membuat UDP dari bandwidth yang ditentukan.
 - f. Menghitung packet loss
 - g. Menghitung delay jitter
 - h. Mampu mengirim data dalam waktu yang bersamaan
9. Dapat digunakan di Windows, Linux, Android, MacOS X, FreeBSD, OpenBSD, NetBSD, VxWorks, Solaris, dll.
10. Klien dan server dapat memiliki beberapa koneksi simultan (opsi -P).

11. Server menangani banyak koneksi, dalam kali satu tes.
12. Dapat menentukan waktu pengiriman data (opsi -t), dan jumlah data untuk ditransfer (-n atau opsi -k).

7. Persiapan Praktikum

1. Modul praktikum.
2. PC/Laptop yang dilengkapi dengan aplikasi VNC, virtual box dan angry ip scanner
3. *Raspberry PI* yang telah ter-install operasi sistem
4. Power Suply untuk *Raspberry PI*
5. Adapter converter USB ke Ethernet sebanyak 4 buah
6. Kabel Ethernet/LAN 4 buah
7. Koneksi LAN dan Internet.

8. Langkah-langkah praktikum

A. Install dan Konfigurasi Openvswitch Pada Raspberry

1. Sambungkan power suply ke *Raspberry PI*, converter usb to Ethernet dan sambungkan pula kabel Ethernet dari port Ethernet *Raspberry PI* ke laptop/pc.
2. Buka aplikasi angry ip scanner dan cari IP dari *Raspberry PI* yang di gunakan.

IP Range - Angry IP Scanner

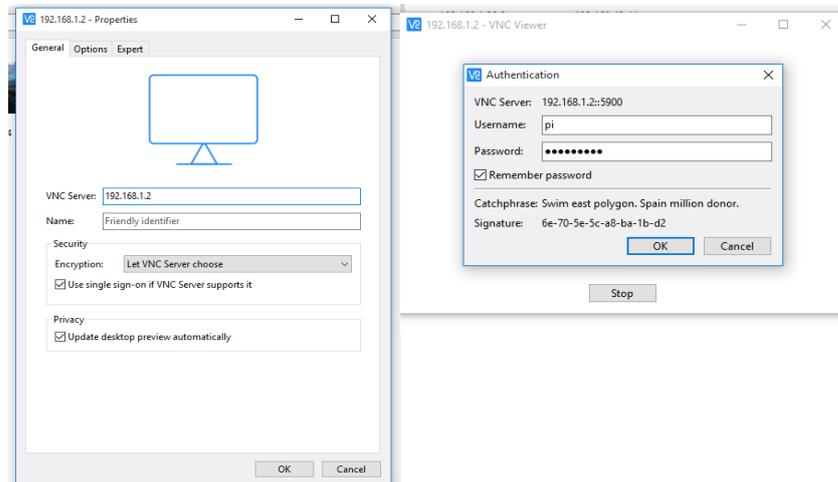
Scan Go to Commands Favorites Tools Help

IP Range: 192.168.1.1 to 192.168.1.15 IP Range [v] [g]

Hostname: DESKTOP-NVNS6DS IP: Netmask [v] [g] Start [g]

IP	Ping	Hostname	Ports [0+]
192.168.1.1	0 ms	DESKTOP-NVNS6DS...	[n/s]
192.168.1.2	0 ms	[n/a]	[n/s]
192.168.1.3	[n/a]	[n/s]	[n/s]
192.168.1.4	[n/a]	[n/s]	[n/s]
192.168.1.5	[n/a]	[n/s]	[n/s]
192.168.1.6	[n/a]	[n/s]	[n/s]
192.168.1.7	[n/a]	[n/s]	[n/s]
192.168.1.8	[n/a]	[n/s]	[n/s]
192.168.1.9	[n/a]	[n/s]	[n/s]
192.168.1.10	[n/a]	[n/s]	[n/s]
192.168.1.11	[n/a]	[n/s]	[n/s]
192.168.1.12	[n/a]	[n/s]	[n/s]
192.168.1.13	[n/a]	[n/s]	[n/s]
192.168.1.14	[n/a]	[n/s]	[n/s]
192.168.1.15	[n/a]	[n/s]	[n/s]

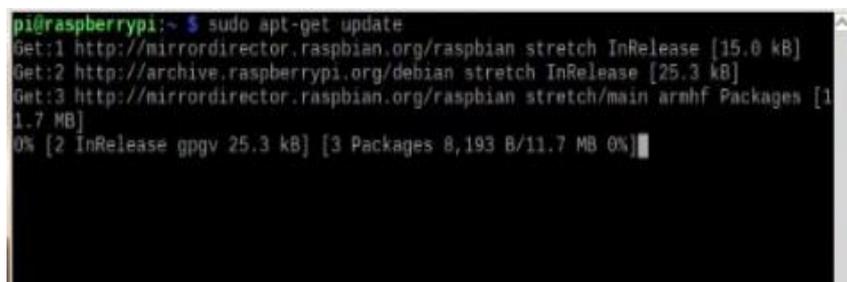
Gambar 8.1. Scan IP



3. Setelah mendapatkan IP dari *Raspberry PI*, buka aplikasi vnc dan remot *Raspberry PI* menggunakan aplikasi tersebut.

Gambar 8.2. Remote *Raspberry PI*

4. Setelah raspberry telah berhasil di remote menggunakan aplikasi vnc
Buka Terminal dan lakukan langkah berikut :
 - a. Masuk kedalam root dengan mengetikan perintah “**sudo su**”
 - b. Update *Raspberry PI* dengan mengetikan perintah “**sudo apt-get update**”



Gambar 8.3 .Update Raspberry

5. Buka browser dan buka situs openvswitch.org. setelah itu klik download dan cek openvswitch versi berapa yang ingin di install packagenya ke raspberry (pada praktikum ini menggunakan openvswitch v2.10.0)
6. Buka terminal kembali dan download openvswitch dengan mengetikan perintah **“wget <http://openvswitch.org/releases/openvswitch-2.10.0.tar.gz>”**

```

root@raspberrypi:/home/pi# wget http://openvswitch.org/releases/openvswitch-2.10.0.tar.gz
--2018-10-03 07:23:56-- http://openvswitch.org/releases/openvswitch-2.10.0.tar.gz
Resolving openvswitch.org (openvswitch.org)...

```

Gambar 8.4. Download OpenvSwitch

7. Ekstrak file yang telah di download dengan mengetikan command berikut pada terminal **“tar -xvf openvswitch-2.10.0.tar.gz”**
8. Ketik **“ls”** pada terminal dan pastikan terdapat file openvswitch
9. Memasang package python simplejson, python qt4, build essential, libssl-dev, gcc, libtool,autoconf, automake, pkg config dan package uml utilities dengan mengetikan **“apt-get install python-simplejson python-q4 libssl-dev python-twisted-conch automake autoconf gcc uml-utilities libtool build-essential pkg-config”**

```

root@raspberrypi:/home/pi/openvswitch-2.10.0# apt-get install python-simplejson python-qt4 libssl-dev python-twisted-conch automake autoconf gcc uml-utilities
libtool build-essential pkg-config
Reading package lists... Done
Building dependency tree
Reading state information... Done
autoconf is already the newest version (2.69-10).
automake is already the newest version (1:1.15-6).
build-essential is already the newest version (12.3).
gcc is already the newest version (4:6.3.0-4).
libssl-dev is already the newest version (1.1.0f-3-deb9u2).
libtool is already the newest version (2.4.6-2).
pkg-config is already the newest version (0.29-4).
python-qt4 is already the newest version (4.11.4+dfsg-2).
python-simplejson is already the newest version (3.10.0-1).
python-twisted-conch is already the newest version (1:16.6.0-2).
uml-utilities is already the newest version (20070815.1-2).
0 upgraded, 0 newly installed, 0 to remove and 166 not upgraded.
root@raspberrypi:/home/pi/openvswitch-2.10.0#

```

Gambar 8.5. Install Package

10. Mencari headers linux yang cocok “**apt-search linux headers**” dan lihat pilihan “header file for linux” (pada praktikum ini memilih header 4.9.0-6 yang paling bawah)

```
root@raspberrypi:/home/pi/openvswitch-2.10.0# apt-cache search linux-headers
aufs-dkms - DKMS files to build and install aufs
linux-headers-3.10-3-all - All header files for Linux 3.10 (meta-package)
linux-headers-3.10-3-all-armhf - All header files for Linux 3.10 (meta-package)
linux-headers-3.10-3-common - Common header files for Linux 3.10-3
linux-headers-3.10-3-rpi - Header files for Linux 3.10-3-rpi
linux-headers-3.16.0-4-all - All header files for Linux 3.16 (meta-package)
linux-headers-3.16.0-4-all-armhf - All header files for Linux 3.16 (meta-package)
linux-headers-3.16.0-4-common - Common header files for Linux 3.16.0-4
linux-headers-3.16.0-4-rpi - Header files for Linux 3.16.0-4-rpi
linux-headers-3.18.0-trunk-all - All header files for Linux 3.18 (meta-package)
linux-headers-3.18.0-trunk-all-armhf - All header files for Linux 3.18 (meta-package)
linux-headers-3.18.0-trunk-common - Common header files for Linux 3.18.0-trunk
linux-headers-3.18.0-trunk-rpi - Header files for Linux 3.18.0-trunk-rpi
linux-headers-3.18.0-trunk-rpi2 - Header files for Linux 3.18.0-trunk-rpi2
linux-headers-3.6-trunk-all - All header files for Linux 3.6 (meta-package)
linux-headers-3.6-trunk-all-armhf - All header files for Linux 3.6 (meta-package)
linux-headers-3.6-trunk-common - Common header files for Linux 3.6-trunk
linux-headers-3.6-trunk-rpi - Header files for Linux 3.6-trunk-rpi
linux-headers-4.4.0-1-all - All header files for Linux 4.4 (meta-package)
linux-headers-4.4.0-1-all-armhf - All header files for Linux 4.4 (meta-package)
linux-headers-4.4.0-1-common - Common header files for Linux 4.4.0-1
linux-headers-4.4.0-1-rpi - Header files for Linux 4.4.0-1-rpi
linux-headers-4.4.0-1-rpi2 - Header files for Linux 4.4.0-1-rpi2
linux-headers-4.9.0-6-all - All header files for Linux 4.9 (meta-package)
linux-headers-4.9.0-6-all-armhf - All header files for Linux 4.9 (meta-package)
linux-headers-4.9.0-6-common - Common header files for Linux 4.9.0-6
linux-headers-4.9.0-6-common-rt - Common header files for Linux 4.9.0-6-rt
linux-headers-4.9.0-6-rpi - Header files for Linux 4.9.0-6-rpi
linux-headers-4.9.0-6-rpi2 - Header files for Linux 4.9.0-6-rpi2
linux-headers-rpi - Header files for Linux rpi configuration (meta-package)
linux-headers-rpi-rpiv - This metapackage will pull in the headers for the raspbian kernel for the
linux-headers-rpi2 - Header files for Linux rpi2 configuration (meta-package)
linux-headers-rpi2-rpv - This metapackage will pull in the headers for the raspbian kernel for the
linux-libc-dev-alpha-cross - Linux Kernel Headers for development (for cross-compiling)
linux-libc-dev-arm64-cross - Linux Kernel Headers for development (for cross-compiling)
linux-libc-dev-armel-cross - Linux Kernel Headers for development (for cross-compiling)
linux-libc-dev-armhf-cross - Linux Kernel Headers for development (for cross-compiling)
linux-libc-dev-ppc64-cross - Linux Kernel Headers for development (for cross-compiling)
linux-libc-dev-m68k-cross - Linux Kernel Headers for development (for cross-compiling)
linux-libc-dev-mips-cross - Linux Kernel Headers for development (for cross-compiling)
```

Gambar 8.6. Mencari Header Linux

11. Instal headers dengan perintah “**apt-get install -y linux-headers-4.9.0-6-rpi**”

```
root@raspberrypi:/home/pi/openvswitch-2.10.0# apt-get install -y linux-headers-4.9.0-6-rpi
Reading package lists... Done
Building dependency tree
Reading state information... Done
linux-headers-4.9.0-6-rpi is already the newest version (4.9.82-1+deb9u3+rpi1).
```

Gambar 8.7. Install Header Linux

12. Lakukan konfigurasi dengan dengan mengetikkan “./configure --with-linux=/lib/modules/4.9.0-6/build” dan dilanjutkan dengan mebuat dan menginstall yang telah di konfigurasi dengan mengetikkan “make && make install”

```
root@raspberrypi:/home/pi/openswitch-2.10.0# ./configure --with-linux=/lib/modules/4.9.0-6-rpi
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking how to create a pax tar archive... gnutar
checking for style of include used by make... GNU
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether gcc understands -c and -o together... yes
checking dependency style of gcc... gcc3
checking for gcc option to accept ISO C99... none needed
checking for g++... g++

root@raspberrypi:/home/pi/openswitch-2.10.0# make && make install
make all-recursive
make[1]: Entering directory '/home/pi/openswitch-2.10.0'
Making all in datapath
make[2]: Entering directory '/home/pi/openswitch-2.10.0/datapath'
Making all in linux
make[3]: Entering directory '/home/pi/openswitch-2.10.0/datapath/linux'
make -C /lib/modules/4.9.0-6-rpi/build M=/home/pi/openswitch-2.10.0/datapath/li
nux modules
make[4]: Entering directory '/usr/src/linux-headers-4.9.0-6-rpi'
Building modules, stage 2.
MODPOST 6 modules
make[4]: Leaving directory '/usr/src/linux-headers-4.9.0-6-rpi'
make[3]: Leaving directory '/home/pi/openswitch-2.10.0/datapath/linux'
make[2]: Leaving directory '/home/pi/openswitch-2.10.0/datapath'
make[1]: Leaving directory '/home/pi/openswitch-2.10.0'
```

Gambar 8.8. Konfigurasi

13. Masuk ke dalam file linux yang berada didirektori datapath dengan mengetikkan “cd datapath/linux” setelah itu menambahkan kernel openswitch ke kernel linux dengan perintah “modprobe openswitch”

```
root@raspberrypi:/home/pi/openswitch-2.10.0# cd datapath/linux
root@raspberrypi:/home/pi/openswitch-2.10.0/datapath/linux# modprobe openswitch
```

Gambar 8.9. Membuat kernel openswitch

14. Pastikan telah dibuat `openvswitch` dalam file `modules` dengan mengetikkan pada terminal `cat /etc/modules` jika belum terbuat ketikkan `echo "openvswitch" >> /etc/modules` dan cek kembali

```
root@raspberrypi:/home/pi/openvswitch-2.10.0/datapath/linux# cat /etc/modules
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.

i2c-dev
openvswitch
```

Gambar 8.10. Membuat Openvswitch

15. Setelah itu keluar dari file `linux` dan `datapath` dengan mengetikkan `cd ../..` dan membuat folder `ovs-vswitchd.conf` dengan cara `touch /usr/local/etc/ovs-vswitchd.conf`.

16. Membuat direktori `mkdir -p /usr/local/etc/openvswitch`.

17. Membuat database `ovsdb-tool create /usr/local/etc/openvswitch/conf.db vswitchd/vswitch.ovsschema`.

18. Membuat file script `nano script` dan isi file tersebut dengan kode di bawah ini :

```
#!/bin/bash
```

```
ovsdb-server --remote=punix:/usr/local/var/run/openvswitch/db.sock --
```

```
remote=db:Open_vSwitch,Open_vSwitch,manager_options --private-
```

```
key=db:Open_vSwitch,SSL,pr$
```

```
ovs-vswitchd --pidfile --detach
```

```
ovs-vsctl --no-wait init
```

```
ovs-vsctl show
```

19. Ketikkan pada terminal “**chmod +x script**” berguna untuk mengurangi atau menambah pemakai untuk mengakses file
20. Run file script “**./script**”
21. Konfigurasi port-port pada *Raspberry PI*
 - Port br0 : **ovs-vsctl add-br br0**
 - Port eth1 : **ovs-vsctl add-port br0 eth1** (ulangi untuk eth2,eth3,eth4)
22. Cek port yang telah di konfigurasi “**ovs-vsctl show**”

```

root@raspberrypi:/home/pi/openvswitch-2.10.0# ovs-vsctl show
996bc4c4-0504-4d87-9a8e-e874eb56d2d5
Bridge "sw0"
  Controller "tcp:192.168.1.2:6633"
  fail_mode: standalone
  Port "sw0"
    Interface "sw0"
      type: internal
  Port "eth3"
    Interface "eth3"
  Port "eth4"
    Interface "eth4"
  Port "eth2"
    Interface "eth2"
  Port "eth1"
    Interface "eth1"
  ovs_version: "2.10.0"

```

Gambar 8.11. Port yang telah di konfigurasi

23. Pastikan port dalam status up. Cek pada perintah “**ifconfig**”. Jika belum maka ketikkan perintah “**ifconfig eth1 0 up**” (begitu pula jika terjadi pada eth2, eth3, eth4)
24. Konfigurasi IP Controller “**ovs-vsctl set-controller br0 tcp:192.168.1.2**” dan pastikan konfigurasi berhasil “**ovs-vsctl show**”
25. Mengecek status controller yang sudah terhubung atau tidak terhubung “**ovs-vsctl list controller**”

```
root@raspberrypi:/home/pi/openvswitch-2.10.0# ovs-vsctl list controller
_uuid          : b22fbf47-b3c9-4d43-866b-5202efa10181
connection_mode : out-of-band
controller_burst_limit: []
controller_rate_limit: []
enable_async_messages: []
external_ids   : {}
inactivity_probe : []
is_connected   : false
local_gateway  : []
local_ip       : []
local_netmask  : []
max_backoff    : []
other_config   : {disable-in-of-band="true"}
role           : other
status         : {last_error="No route to host", sec_since_disconnect="10", state=CONNECTING}
target         : "tcp:192.168.1.2:6633"
```

Gambar 8.16. Controller Status

B. Controller POX

1. Buka Virtual Box dan aktifkan OS yang kalian telah di instal (Pada praktikum ini menggunakan CentOS)
2. Buka directory root “su” dan masukan password. Install POX Controller “git clone <http://github.com/noxrepo/pox>” jika belum terdapat package git pada CentOS, install terlebih dahulu “**sudo yum install git**”
3. Setelah Install selesai masuk ke dalam folder pox dan run “./pox.py forwarding l2._learning

```
[auraevaa@localhost ~]$ su
Password:
[root@localhost auraevaa]# cd
[root@localhost ~]# ls
anaconda-ks.cfg          initial-setup-ks.cfg  vnd-sdn-version
get-pip.py               MININET_CENTOS7
gnome-shell-extension-sublime-text  nox
[root@localhost ~]# cd MININET_CENTOS7
[root@localhost MININET_CENTOS7]# ls
loxygen  mininet  oflops  oftest  openflow  pox  rdo-release.rpm  README.md
[root@localhost MININET_CENTOS7]# cd pox
[root@localhost pox]# sudo ./pox.py forwarding.l2_learning
POX 0.5.0 (eel) / Copyright 2011-2014 James McCauley, et al.
INFO:core:POX 0.5.0 (eel) is up.
```

Gambar 8.17. Run POX Controller

C. Pengambilan Data

1. Install iPerf pada Host1 dan Host2 pada direktori root “**sudo apt-get install iperf**”

- Setelah itu jalankan iperf sebagai server di salah satu host “**iperf -s**” untuk menambahkan interfal data “**iperf -s -i 1**”

```
root@Host02:/home/pi/iperf-2.0.5# iperf -s
-----
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
-----
^Croot@Host02:/home/pi/iperf-2.0.5# iperf -s -i 1
-----
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
-----
```

Gambar 8.18. Iperf Pada Host2

- Jalankan iperf sebagai client di salah satu host “**iperf -c 192.168.2.2**” untuk menentukan waktu pengiriman data “**iperf -c 192.168.2.2 -i 1 -t 20**”

```
root@Host01:/home/pi/iperf-2.0.5# iperf -c 192.168.2.2
-----
Client connecting to 192.168.2.2, TCP port 5001
TCP window size: 43.8 KByte (default)
-----
root@Host01:/home/pi/iperf-2.0.5# iperf -c 192.168.2.2 -i 1 -t 20
-----
Client connecting to 192.168.2.2, TCP port 5001
TCP window size: 43.8 KByte (default)
-----
```

Gambar 8.19. Iperf Pada Host1

- Lakukan juga untuk UDP dengan cara yang sama pada nomor 2 dan 3 namun di tambahkan “**-u**” di akhir perintah

```
^[[A^Croot@Host02:/home/pi/iperf-2.0.5# iperf -s -i 1 -u
iperf: ignoring extra argument -- 1
-----
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 160 KByte (default)
-----
root@Host01:/home/pi/iperf-2.0.5# iperf -c 192.168.2.2 -i 1 -t 20 -u
-----
Client connecting to 192.168.2.2, UDP port 5001
Sending 1470 byte datagrams, IPG target: 11215.21 us (kalman adjust)
UDP buffer size: 160 KByte (default)
-----
```

Gambar 8.20 : Iperf untuk UDP

5. Lakukan juga dengan ICMP. Yaitu melakukan PING pada salah satu host “**ping 192.168.2.2**”. Melakukan PING dengan waktu yang di tentukan “**ping -c 20 192.168.2.2**”

```
root@Host01:/home/pi/iperf-2.0.5# ping -c 20 192.168.2.2  
PING 192.168.2.2 (192.168.2.2) 56(84) bytes of data.
```

Gambar 8.21. Ping antar host

TUGAS DAN EVALUASI

Lakukan pengujian hos-host, host-switch dan switch-controller dengan parameter sebagai berikut :

- a. TCP : Mengirim Data selama 20 detik
- b. UDP : Mengirim Data selama 20 detik
- c. Melakukan PING selama 20detik

Dari jendela pelatihan, catatlah data yang telah di dapat dari hasil pelatihan tersebut ke dalam bentuk tabel. Lakukan perhitungan dengan kategori throughput, delay, jitter paket loss dan buatlah grafik dari setiap kagetori. Kemudian jawablah pertanyaan-pertanyaan berikut :

1. Lakukan analisis terhadap parameter yang telah di hitung dan jelaskan!
2. Dari grafik yang telah di buat, jelaskan perubahan yang terjadi setiap kategorinya!
3. Jika kategori paket loss tidak 0%, maka ada beberapa total paket yang hilang?
4. Jelaskan kinerja *OpenFlow* pada raspberry dari data yang anda dapat!
5. Jelaskan kesimpulan yang anda dapatkan bedasarkan percobaan yang telah dilakukan!