

Desi Tambunan	161402005
Sonia Elisa Telaumbanua	161402045
Novalina Simbolon	161402077
Chyntia Claudia	161402101
Tirza Priskila Kinanti Sibuea	161402110

## Administrasi & Desain Jaringan

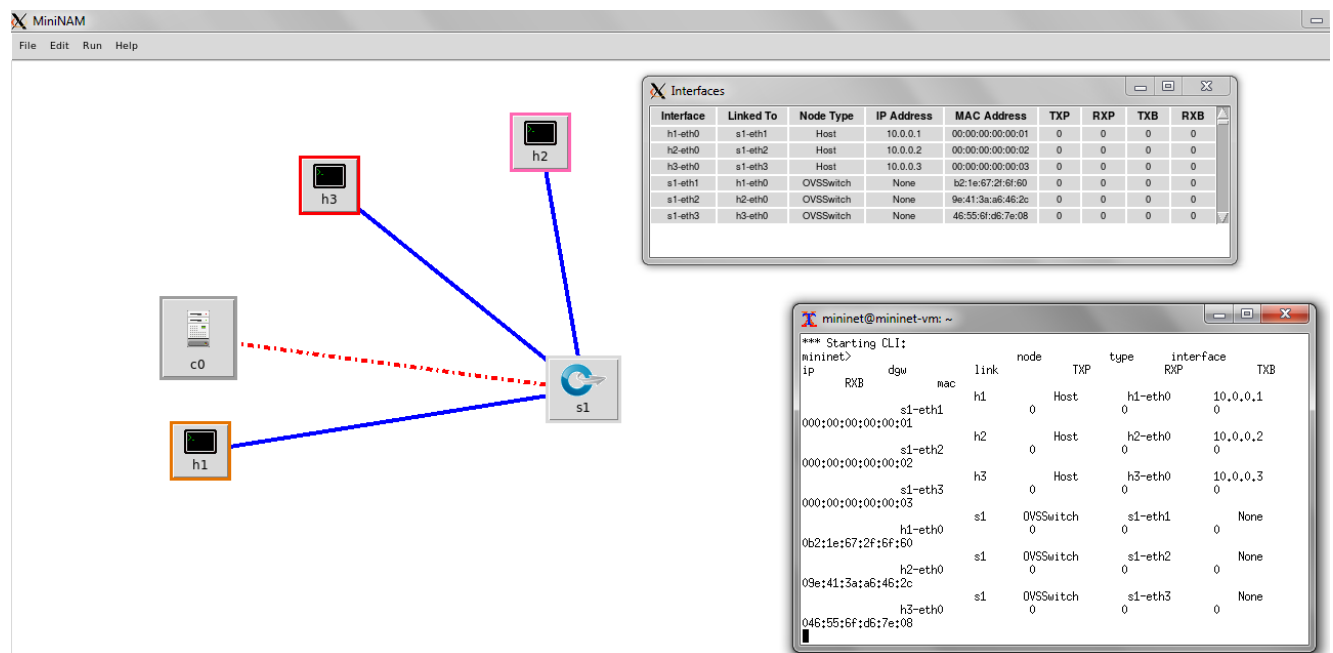
### OPENFLOW

OpenFlow adalah sebuah protokol yang memungkinkan pengaturan penjaluran dan pengiriman paket ketika melalui sebuah *switch*. Dalam sebuah jaringan konvensional, setiap *switch* hanya berfungsi meneruskan paket yang lewat ke port yang sesuai tanpa dapat membedakan tipe protokol data yang dikirimkan, misalnya *elastic* atau *inelastic traffic*.

Berikut contoh tahap-tahapannya;

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

Membuat 3 *host* dengan 1 *switch*



2. Manual Flow Entry

```

mininet@mininet-vm:~$ sudo python ~/MiniNAM/MiniNAM.py --topo single,3 --mac --
switch ovsk --controller remote

mininet> sh ovs-ofctl dump-flows s1

```

```
mininet@mininet-vm: ~  
mininet@mininet-vm:~$ sudo python ~/MiniNAM/MiniNAM.py --topo single,3 --mac --swi  
tch ovsk --controller remote  
*** Creating network  
*** Adding controller  
Unable to contact the remote controller at 127.0.0.1:6653  
Connecting to 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:  
mininet>
```

```
mininet@mininet-vm: ~  
(h1, s1) (h2, s1) (h3, s1)  
*** Configuring hosts  
h1 h2 h3  
*** Starting controller  
c0  
*** Starting 1 switches  
s1 ...  
*** Starting CLI:  
mininet> sh ovs-ofctl dump-flows s1  
NXST_FLOW reply (xid=0x4):  
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=6.21 ms  
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.713 ms  
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.076 ms  
  
--- 10.0.0.2 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2002ms  
rtt min/avg/max/mdev = 0.076/2.334/6.215/2.756 ms  
mininet>
```

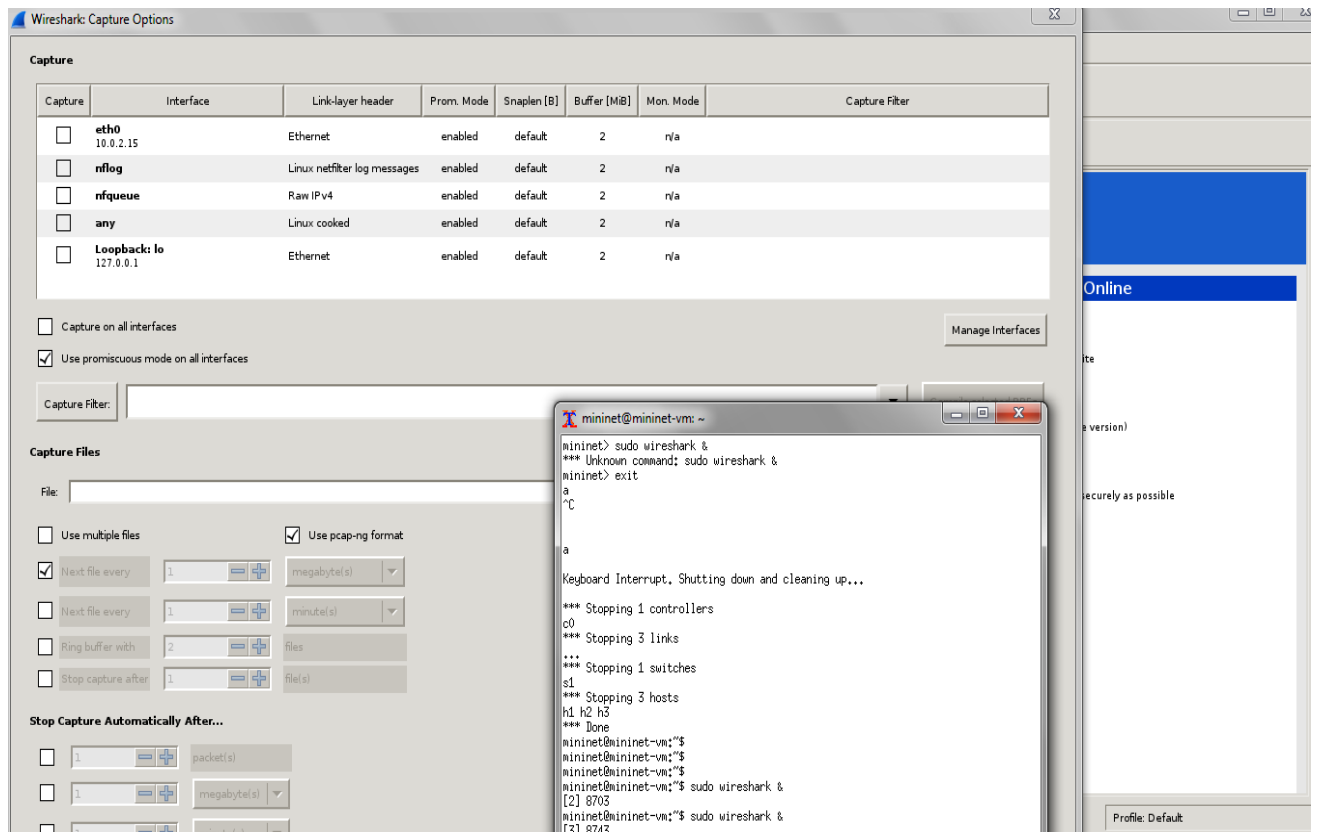
```

mininet@mininet-vm: ~
rtt min/avg/max/mdev = 0.074/3.474/9.584/4.329 ms
mininet> sh ovs-ofctl add-flow s1 in_port=1,actions=output:2
mininet> sh ovs-ofctl add-flow s1 in_port=2,actions=output:1
mininet> sh ovs-ofctl dump-flows s1
NXST_FLOW reply (xid=0x4):
  cookie=0x0, duration=47.957s, table=0, n_packets=1, n_bytes=42, idle_timeout=60,
  idle_age=47, priority=65535,arp,in_port=1,vlan_tci=0x0000,dl_src=00:00:00:00:00:01
  ,dl_dst=00:00:00:00:00:02,arp_spa=10.0.0.1,arp_tpa=10.0.0.2,arp_op=2 actions=output:2
  cookie=0x0, duration=52.979s, table=0, n_packets=1, n_bytes=42, idle_timeout=60,
  idle_age=52, priority=65535,arp,in_port=2,vlan_tci=0x0000,dl_src=00:00:00:00:00:02
  ,dl_dst=00:00:00:00:00:01,arp_spa=10.0.0.2,arp_tpa=10.0.0.1,arp_op=2 actions=output:1
  cookie=0x0, duration=47.96s, table=0, n_packets=1, n_bytes=42, idle_timeout=60, i
  dle_age=47, priority=65535,arp,in_port=2,vlan_tci=0x0000,dl_src=00:00:00:00:00:02,
  dl_dst=00:00:00:00:00:01,arp_spa=10.0.0.2,arp_tpa=10.0.0.1,arp_op=1 actions=output:1
  cookie=0x0, duration=18.725s, table=0, n_packets=0, n_bytes=0, idle_age=18, in_po
  rt=1 actions=output:2
  cookie=0x0, duration=8.331s, table=0, n_packets=0, n_bytes=0, idle_age=8, in_port
  =2 actions=output:1
  cookie=0x0, duration=52.976s, table=0, n_packets=3, n_bytes=294, idle_timeout=60,
  idle_age=50, priority=65535,icmp,in_port=1,vlan_tci=0x0000,dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:02,nw_src=10.0.0.1,nw_dst=10.0.0.2,nw_tos=0,icmp_type=8,i
  cmp_code=0 actions=output:2
  cookie=0x0, duration=52.972s, table=0, n_packets=3, n_bytes=294, idle_timeout=60,
  idle_age=50, priority=65535,icmp,in_port=2,vlan_tci=0x0000,dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01,nw_src=10.0.0.2,nw_dst=10.0.0.1,nw_tos=0,icmp_type=0,i
  cmp_code=0 actions=output:1
mininet>

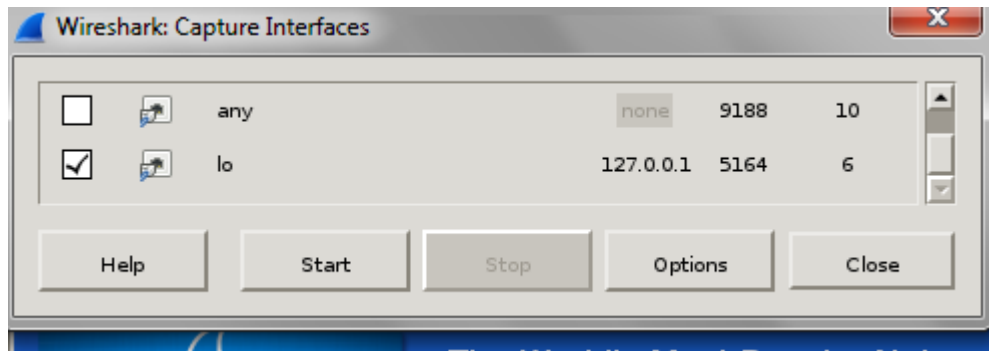
```

### 3. Wireshark

mininet@mininet-vm:~\$ sudo wireshark &



Klik 'Capture' lalu pada *menubar* di atas lalu pilih 'Capture Interfaces' dan pilih (centang) 'lo' kemudian mulai ('Start').



#### 4. Starting Controller (OF Reference Controller)

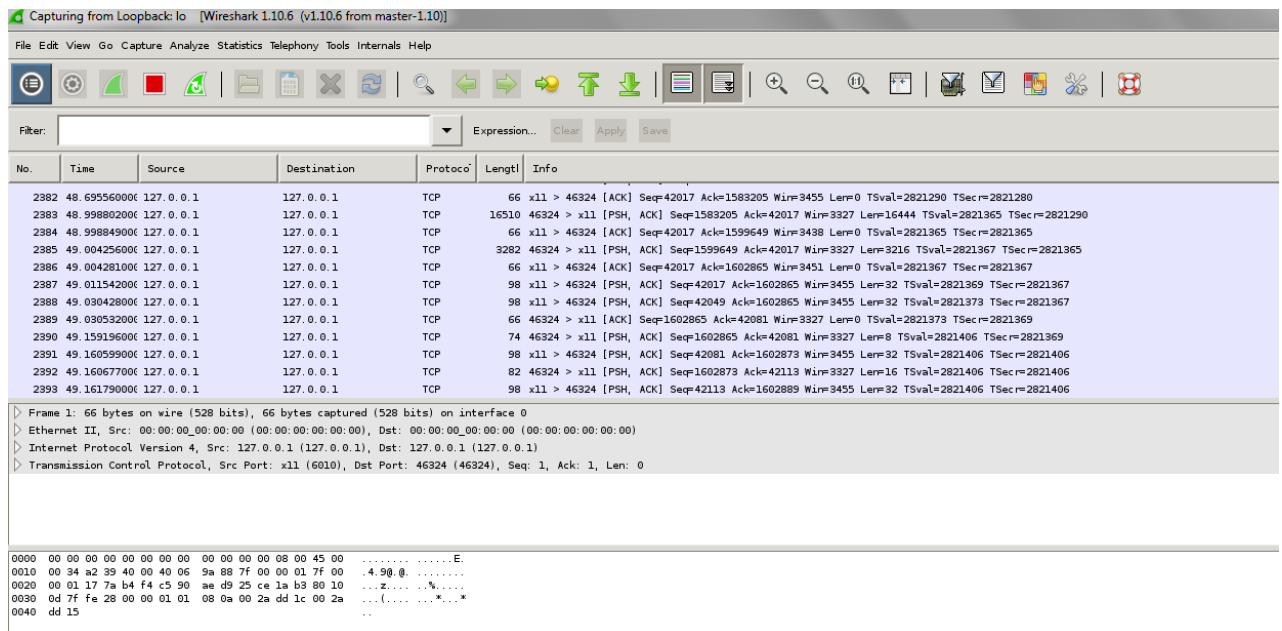
mininet@mininet-vm:~\$ controller ptcp: &

```
mininet@mininet-vm:~$ controller ptcp: &
[4] 9119
mininet@mininet-vm:~$ Apr 07 03:45:20|00001|vconn_tcp|ERR|ptcp:: bind: Address already in use
Apr 07 03:45:20|00002|controller|ERR|ptcp:: connect: Address already in use
mininet@mininet-vm:~$ sudo mn --topo single,3 --mac --switch ovsk --controller remote
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6653
Connecting to 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:
mininet>
```

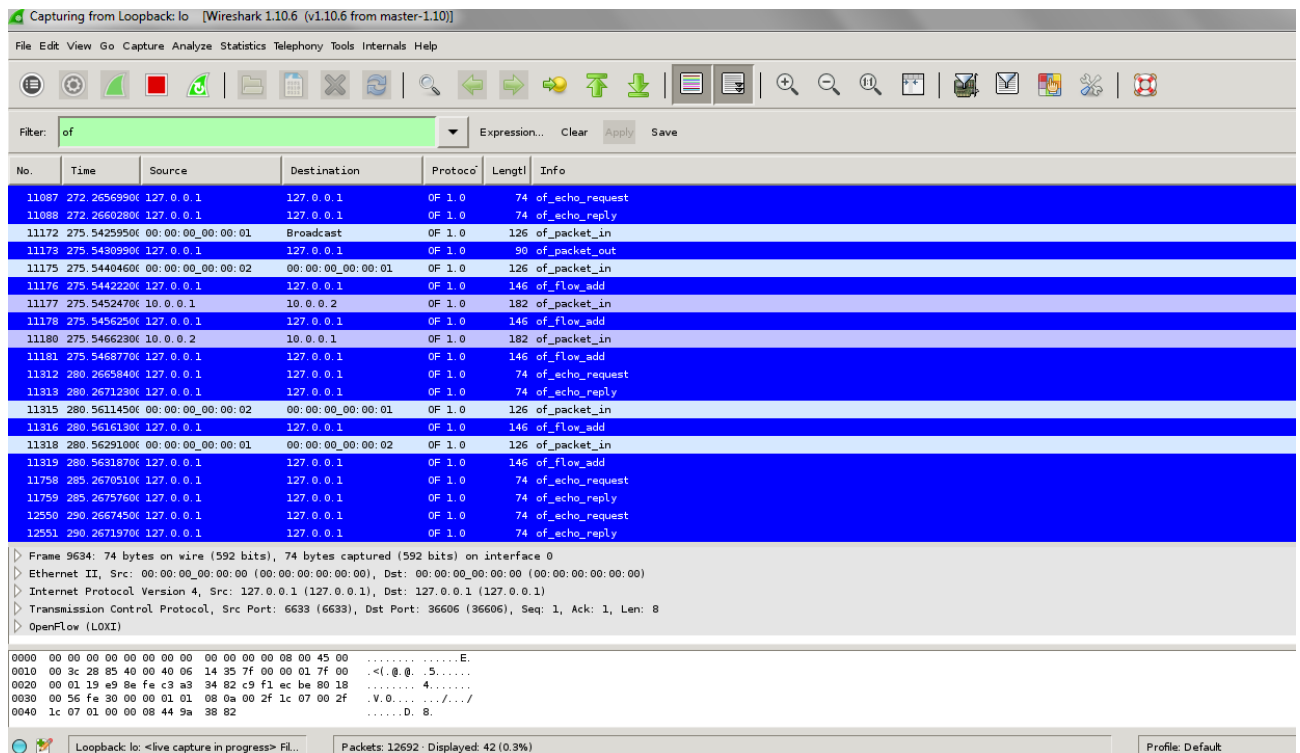
```
mininet@mininet-vm: ~
mininet> h1 ip -s -s neigh flush all
Nothing to flush.
mininet> h2 ip -s -s neigh flush all
Nothing to flush.
mininet> sh ovs-ofctl del-flows s1
mininet> h1 ping -c1 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data:
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=6.39 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 6.393/6.393/6.393/0.000 ms
mininet>
```

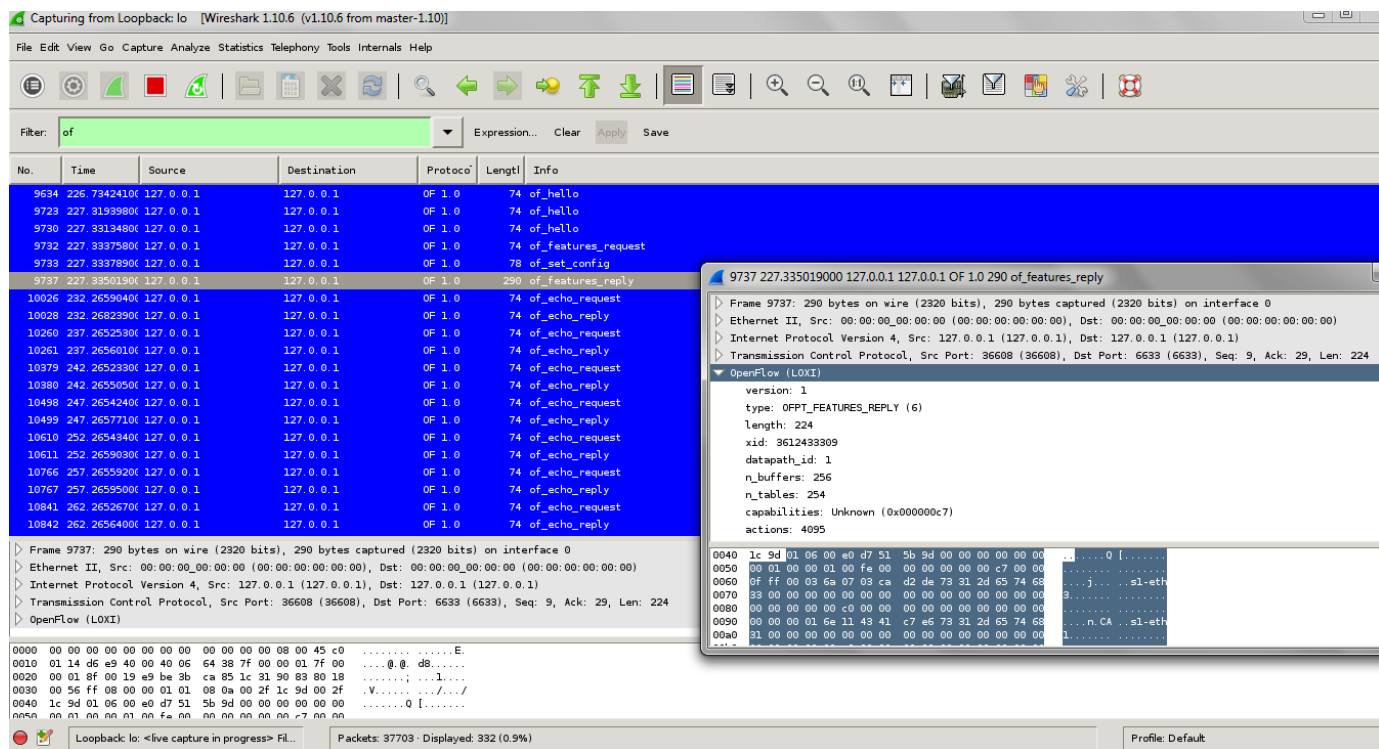
Berikut tampilan pada wireshark;



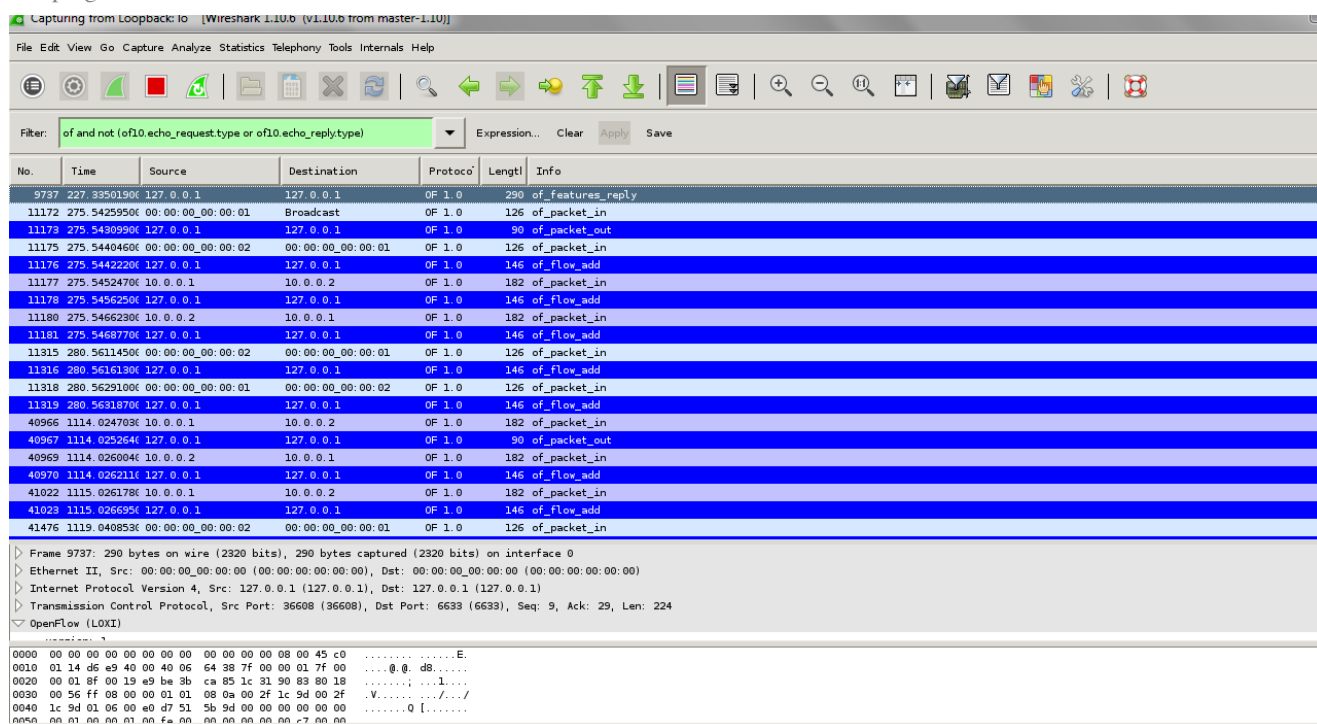
## 5. Lakukan Filter 'OF' pada Wireshark



## 6. Inspect Packet



- Filter 'of and not (of10.echo\_request.type or of10.echo\_reply.type)' pada tampilan WireShark setelah 'h1 ping -c1 h2'.



## 8. Flow Entries

```
mininet> dpctl dump-flows
```

```
mininet@mininet~vm::~$ sudo ovs-ofctl dump-flows s1
```

```

mininet@mininet-vm: ~
mininet> dpctl dump-flows
*** s1 -----
NXST_FLOW reply (xid=0x4):
  cookie=0x0, duration=244.939s, table=0, n_packets=10, n_bytes=420, idle_timeout=60, idle_age=4, priority=65535, arp, in_port=1, vlan_tci=0x0000, dl_src=00:00:00:00:01, dl_dst=00:00:00:00:00:02, arp_spa=10.0.0.1, arp_tpa=10.0.0.2, arp_op=2 actions=output:2
  cookie=0x0, duration=244.941s, table=0, n_packets=10, n_bytes=420, idle_timeout=60, idle_age=4, priority=65535, arp, in_port=2, vlan_tci=0x0000, dl_src=00:00:00:00:00:02, dl_dst=00:00:00:00:00:01, arp_spa=10.0.0.2, arp_tpa=10.0.0.1, arp_op=1 actions=output:1
  cookie=0x0, duration=248.955s, table=0, n_packets=240, n_bytes=23520, idle_timeout=60, idle_age=9, priority=65535, icmp, in_port=1, vlan_tci=0x0000, dl_src=00:00:00:00:00:01, dl_dst=00:00:00:00:00:02, nw_src=10.0.0.1, nw_dst=10.0.0.2, nw_tos=0, icmp_type=8, icmp_code=0 actions=output:2
  cookie=0x0, duration=249.956s, table=0, n_packets=241, n_bytes=23618, idle_timeout=60, idle_age=9, priority=65535, icmp, in_port=2, vlan_tci=0x0000, dl_src=00:00:00:00:00:02, dl_dst=00:00:00:00:00:01, nw_src=10.0.0.2, nw_dst=10.0.0.1, nw_tos=0, icmp_type=0, icmp_code=0 actions=output:1
mininet>

```

## 9. Benchmark kernel- vs. user-space

```

mininet@mininet-vm: ~
:00:02, dl_dst=00:00:00:00:00:01, nw_src=10.0.0.2, nw_dst=10.0.0.1, nw_tos=0, icmp_type=0, icmp_code=0 actions=output:1
mininet> iperf
*** Iperf: testing TCP bandwidth between h1 and h3
.e*** Results: ['6.26 Gbits/sec', '6.26 Gbits/sec']
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 3 links
***
*** Stopping 1 switches
s1
*** Stopping 3 hosts
h1 h2 h3
*** Done
completed in 1202.889 seconds
mininet@mininet-vm:~$ sudo mn --topo single,3 --controller remote --switch user
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6653
Connecting to 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:
mininet> sh ovs-ofctl add-flow s1 in_port=1,actions=output:3
ovs-ofctl: s1 is not a bridge or a socket
mininet> sh ovs-ofctl add-flow s1 in_port=3,actions=output:1
ovs-ofctl: s1 is not a bridge or a socket
mininet>

```