

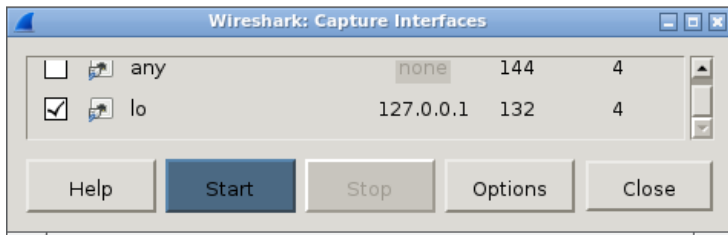


Universidade Federal da Bahia
MATE18 – Tópicos em Redes de Computadores III
Oficina de OpenFlow/SDN

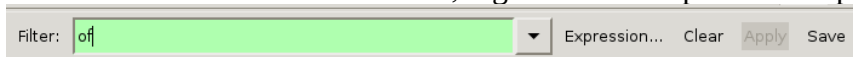
Professor: Leobino Sampaio <leobino@ufba.br>
Instrutor: Italo Valcy <italovalcy@ufba.br>

Prática 01 – Captura de pacotes OpenFlow no Wireshark

1. Inicie o Wireshark e escolha a opção Capture > Interfaces. Então escolha a interface “Loopback (lo)” e clique em “Start” para começar a captura. Caso você não tenha permissão, terá que iniciar o Wireshark como usuário “root”.



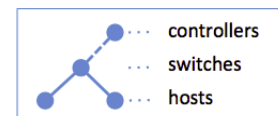
2. Na barra de filtros do Wireshark, digite o filtro de protocolo OpenFlow “of” e depois “Apply”



3. Agora deixe o Wireshark executando e inicie um terminal (e.g. xterm). Em seguida digite o comando “sudo mn” para iniciar o Mininet em sua topologia padrão (exemplificada na direita):

```
mininet@oficina-openflow-ufba:~$ sudo mn
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>
```

> sudo mn



4. Volte ao Wireshark e observe as mensagens geradas:

No.	Time	Source	Destination	Protocol	Length	Info
398	196.00123200	127.0.0.1	127.0.0.1	OF 1.0	74	of_hello
400	196.00889300	127.0.0.1	127.0.0.1	OF 1.0	74	of_hello
402	196.03564600	127.0.0.1	127.0.0.1	OF 1.0	74	of_features_request
404	196.03600000	127.0.0.1	127.0.0.1	OF 1.0	146	of_features_reply
405	196.03714100	127.0.0.1	127.0.0.1	OF 1.0	78	of_set_config
407	196.12597300	127.0.0.1	127.0.0.1	OF 1.0	130	of_port_status
408	196.13685900	127.0.0.1	127.0.0.1	OF 1.0	130	of_port_status
410	201.00085400	127.0.0.1	127.0.0.1	OF 1.0	74	of_echo_request
411	201.00306600	127.0.0.1	127.0.0.1	OF 1.0	74	of_echo_reply

Ao clicar em uma mensagem você pode visualizar mais detalhes sobre o protocolo na tela abaixo. Por exemplo, clique na mensagem de “of_features_reply” e veja os campos do protocolo Openflow:

No.	Time	Source	Destination	Protocol	Length	Info
12	1.461817000	127.0.0.1	127.0.0.1	OpenFlow	242	Type: OFPT_FEATURES_REPLY
13	1.462738000	127.0.0.1	127.0.0.1	OpenFlow	78	Type: OFPT_SET_CONFIG

```

> capabilities: 0x00000000
< actions: 0x0000c700
    ...0 = Output to switch port: False
    ...0 = Set the 802.1q VLAN id: False
    ...0 = Set the 802.1q priority: False
    ...0 = Strip the 802.1q header: False
    ...0 = Ethernet source address: False
    ...0 = Ethernet destination address: False
    ...0 = IP source address: False
    ...0 = IP destination address: False
    ...1 = IP ToS (DSCP field, 6 bits): True

```

Observe e analise o conteúdo das mensagens trocadas pelo Switch OF e o Controlador, a saber:

1. Hello (switch => controlador)
2. Hello (controlador => switch)
3. OF Features Request (controlador => switch)
4. OF Set Config (controlador => switch)
5. OF Features Reply (switch => controlador)
6. OF Echo Request (switch => controlador)
7. OF Echo Reply (controlador => switch)

5. Retorne ao console do Mininet e execute um PING do H1 para H2, com o seguinte comando: “h1 ping -c4 h2”. Exemplo:

```

mininet> h1 ping -c4 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=14.1 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=1.38 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.276 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.174 ms

--- 10.0.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 0.174/3.990/14.123/5.869 ms
mininet>

```

Para reflexão: por que o primeiro pacote teve um tempo muito superior aos demais? Quais fatores levaram a esse comportamento?

Volte ao Wireshark e verifique os pacotes capturados. Foi possível visualizar todo o fluxo de funcionamento do OpenFlow (e.g. Packet-In, Packet-Out, Flow-Modify)? Foi possível visualizar o tráfego ICMP?

Reinicie a captura do Wireshark porém agora escolha a interface “any” (Capture > Interfaces). Além disso, altere o filtro do Wireshark para o seguinte: **of or icmp and not of10.echo_reply.type and not of10.echo_request.type**

Faça novamente o teste de PING no mininet e observe agora os pacotes no Wireshark.

Prática 02: Criar uma topologia linear (3 switches) e configurar regras OpenFlow manualmente para encaminhamento L2 entre os switches, a fim de que todos os hosts tenham conectividade entre si.

1. Em um terminal (xterm), execute o seguinte comando para criação da topologia:

sudo mn --topo linear,3 --mac --arp --controller none

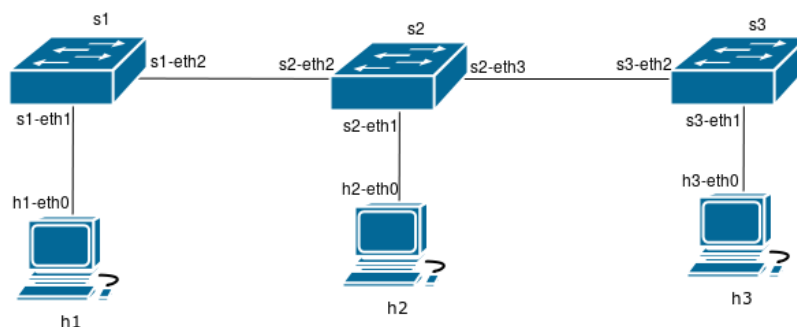
2. Nesse modo de funcionamento, sem o controlador, precisaremos criar as regras manualmente nos switches para que eles possam encaminhar o tráfego. O primeiro passo é levantar algumas informações para criação das regras OpenFlow, por exemplo: quais as interfaces de cada switch – nome e número OpenFlow. Para isso, vamos executar o comando “sh ovs-ofctl show SWITCH” para visualizar as portas de cada switch. Exemplo:

```
mininet> sh ovs-ofctl show s1
OFPT_FEATURES_REPLY (xid=0x2): dpid:0000000000000001
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:c6:d8:d9:b6:6d:51
  config: 0
  state: 0
  current: 10GB-FD COPPER
  speed: 10000 Mbps now, 0 Mbps max
2(s1-eth2): addr:da:50:fd:fc:f6:30
  config: 0
  state: 0
  current: 10GB-FD COPPER
  speed: 10000 Mbps now, 0 Mbps max
LOCAL(s1): addr:86:e9:33:ed:2a:45
  config: 0
  state: 0
  speed: 0 Mbps now, 0 Mbps max
OFPT_GET_CONFIG_REPLY (xid=0x4): frags=normal miss_send_len=0
mininet>
```

Outro comando interessante para essa tarefa é visualizar as ligações entre os equipamentos. Para isso, execute o seguinte comando no console do Mininet: **net**

```
mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s2-eth1
h3 h3-eth0:s3-eth1
s1 lo: s1-eth1:h1-eth0 s1-eth2:s2-eth2
s2 lo: s2-eth1:h2-eth0 s2-eth2:s1-eth2 s2-eth3:s3-eth2
s3 lo: s3-eth1:h3-eth0 s3-eth2:s2-eth3
mininet>
```

Pela saída acima, podemos ver que o “h1” está conectado à porta s1-eth1 do switch “s1” e que o switch “s1” está conectado ao switch “s2” através das portas s1-eth2 e s2-eth2, respectivamente. A topologia criada assemelha-se à apresentada abaixo:



3. Agora tente executar o comando de PING entre os hosts, através do comando “pingall” no console do Mininet. Este comando não deve funcionar, mas por quê?

4. Criação das regras de encaminhamento OpenFlow através do comando ovs-ofctl. Para isso vejamos a relação entre interfaces e conexões nos switches, conforme ilustração da topologia acima. Por exemplo, para configurar a comunicação entre o h1 e h2, executamos no console do Mininet:

```
mininet> sh ovs-ofctl add-flow s1
in_port=1,dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:02,actions=output:2
mininet> sh ovs-ofctl add-flow s1
in_port=2,dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01,actions=output:1
mininet> sh ovs-ofctl add-flow s2
in_port=1,dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01,actions=output:2
mininet> sh ovs-ofctl add-flow s2
in_port=2,dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:02,actions=output:1
```

Agora execute comandos equivalentes para viabilizar a comunicação entre os demais nós.