

## ***Lab Redes WAN de Banda Larga - Pox***

Neste laboratório o aplicativo Mininet da máquina virtual (login tamandua, senha ufabc) será utilizado para criar redes virtuais com controladores OpenFlow. Em todas as atividades será usada a topologia da atividade 4.1.

### **Lab 4.1**

Nesta atividade, a rede criada pelo Mininet terá uma topologia com dois nós tipo host, um nó tipo switch (Open vSwitch), irá usar uma numeração do endereço MAC conforme o endereço IP e um nó controlador conectado à porta 6633 da máquina virtual do switch. Open vSwitch é um softswitch IP com suporte a QOS e suporte ao OpenFlow. Execute os seguintes comandos numa janela lxterminal:

```
su
```

```
mn -c
```

```
mn --topo single,2 --mac --switch ovsk --controller remote
```

Numa segunda janela execute os comandos:

```
su
```

```
ovs-ofctl show s1
```

```
ovs-ofctl dump-flows s1
```

O aplicativo ovs-ofctl permite a gerência de switches OpenFlow. O comando show mostra as portas de um switch se seus dados. O comando dump-flows mostra os fluxos criados pelo controlador OpenFlow.

Na janela do Mininet execute o seguinte comando: pingall

O que aconteceu? Por quê? Qual a relação com o resultado do comando dump-flows?

Execute os seguintes comandos na segunda janela:

```
ovs-ofctl add-flow s1 in_port=1,actions=output:2
```

```
ovs-ofctl add-flow s1 in_port=2,actions=output:1
```

```
ovs-ofctl dump-flows s1
```

Na janela Mininet volte a executar o comando pingall. O que aconteceu? Na janela Mininet execute o comando iperf.

Inclua suas respostas no relatório.

## Lab 4.2

Nesta atividade a flow table do switch será programada por um controller Pox, que é um controller OpenFlow desenvolvido em Python.

Na janela Mininet execute os comandos:

```
exit
```

```
mn -c
```

na segunda janela execute os comandos:

```
cd pox
```

```
python pox.py log.level --DEBUG forwarding.hub
```

Na janela Mininet execute: mn --topo single,2 --mac --switch ovsk --controller remote

teste a conectividade com o comando pingall

Abra o arquivo ../pox/pox/forwarding/hub.py com o editor leafpad.

```
def _handle_ConnectionUp (event):
```

```
    msg = of.ofp_flow_mod()
```

```
    msg.actions.append(of.ofp_action_output(port = of.OFPP_FLOOD))
```

```
    event.connection.send(msg)
```

```
    log.info("Hubifying %s", dpidToStr(event.dpid))
```

def \_handle\_ConnectionUp é um método que irá tratar o evento conexão de um switch.

of.ofp\_flow\_mod é uma classe que representa uma mensagem para modificação de uma flow table de um switch. Possui uma lista de ações. Por default, a lista de ações é acrescentada na flow table.

ofp\_action\_output é uma classe que especifica a ação de enviar um pacote para uma dada porta de saída. event.connection.send(msg) envia a mensagem para o switch.

**def launch ():**

**core.openflow.addListenerByName("ConnectionUp", \_handle\_ConnectionUp)**

**log.info("Hub running.")**

def launch será executado na inicialização do objeto e associa o método \_handle\_ConnectionUp para tratamento dos eventos tipo ConnectionUp.

A classe of.ofp\_flow\_mod possui os seguintes campos:

- cookie (int) - identificador para essa regra de fluxo. (opcional)
- command (int) - Um dos seguintes valores:
  - OFPFC\_ADD - adiciona uma regra ao switch (padrão)
  - OFPFC\_MODIFY - modifica qualquer regra combinante
  - OFPFC\_MODIFY\_STRICT - modifica regras com valores wildcards estritos
  - OFPFC\_DELETE - deleta todas as regras combinantes
  - OFPFC\_DELETE\_STRICT - deleta regras com valores wildcards estritos
- idle\_timeout (int) - regra irá expirar se não recebe qualquer pacote que combine com ela em 'idle\_timeout' segundos. Um valor
- OFP\_FLOW\_PERMANENT significa que não há qualquer idle\_timeout (o padrão). hard\_timeout (int) - regra expirará após 'hard\_timeout' segundos. Um valor OFP\_FLOW\_PERMANENT quer dizer que ela nunca expirará (o padrão)
- priority (int) - a prioridade com que a regra combinará, quanto maior maior a prioridade. Note: combinações exatas terão maiores prioridades.
- buffer\_id (int) - um buffer no switch que a nova regra será aplicada. Use None para nenhum. Sem significado quando deleta-se regras.
- out\_port (int) - esse campo é usado para combinar comando de deletar. OFPP\_NONE pode ser usando para indicar que não há qualquer restrição.
- flags (int) - um dos seguintes valores:

- OFPFF\_SEND\_FLOW\_REM - envia uma mensagem de regra removida ao controlador quando a regra expira
- OFPFF\_CHECK\_OVERLAP - checa se há sobreposição de regras ao instalar. Se houver, um erro é enviado ao controlador
- OFPFF\_EMERG - Considere isso como uma regra emergencial e só use quando a conexão com o switche é perdida
- actions (list) - ações são definidas abaixo, cada objeto de ação desejado é adicionado a lista e são executados em ordem
- match (ofp\_match) - a estrutura de combinação para as regras (veja abaixo)
  - in\_port Porta do switch que o pacotes chegou
  - dl\_src Endereço de origem Ethernet
  - dl\_dst Endereço de destino Ethernet
  - dl\_vlan VLAN ID dl\_vlan\_pcp VLAN prioridade
  - dl\_type Ethertype / tamanho (e.g. 0x0800 = IPv4)
  - nw\_tos IP TOS/DS bits
  - nw\_proto IP protocolo (e.g., 6 = TCP) or lower 8 bits of ARP opcode
  - nw\_src Endereço de origem IP
  - nw\_dst Endereço de destino IP
  - tp\_src Porta de origem TCP/UDP
  - tp\_dst Porta de destino TCP/UDP

Para criar uma estrutura match

```
my_match = of.ofp_match(in_port = 5, dl_dst = EthAddr("01:02:03:04:05:06"))
```

Ou

```
my_match = of.ofp_match()
```

```
my_match.in_port = 5
```

```
my_match.dl_dst = EthAddr("01:02:03:04:05:06")
```

### Lab 4.3

Crie um arquivo lab4\_3.py no diretório .../pox/pox/misc que reproduz exatamente

a flow table da atividade 4.1. Teste o controlador Pox com misc.lab4\_3 no lugar de forwarding.hub. Não se esqueça de desfazer as configurações anteriores terminando a execução do controlador pox e executando o comando mn -c.

Documente os resultados e o conteúdo do arquivo do controlador no relatório.

## **Lab 4.4**

Crie um firewall que descarta pacotes UDP (protocolo 17 decimal). Você precisa especificar que o tipo de pacote é IPV4 (0x0800). Para criar uma regra da flow table que descarta um pacote crie uma regra com a lista de ações vazia.

Para testar o firewall use os comandos na janela Mininet:

```
iperf
```

```
xterm h1 h2
```

Na janela de h1 execute: iperf -s -u

Na janela de h2 execute: iperf -c 10.0.0.1 -u

Documente os resultados e o conteúdo do arquivo do controlador no relatório.

### **Referências:**

POX Wiki em <https://openflow.stanford.edu/display/ONL/POX+Wiki>

Mininet em <https://github.com/mininet/mininet/wiki/Documentation>