ARQUITECTURAS DE REDES AVANZADAS
PRÁCTICA 2
Balanceo de Carga con SDN
ENERO 2015

José Luis Cánovas Sánchez
joseluis.canovas2@um.es
48636907A

Resumen

Índice

1. Introducción

2. Topología mininet

```
1 # -*- coding: utf-8 -*-
2
3 from mininet.topo import Topo
4
5 class MyTopo (Topo):
6
```

```
    def __init__ (self):
        Topo.__init__( self )

        # Add switches
        sw_clients = self.addSwitch('s1')
        sw_servers = self.addSwitch('s2')

        # Add clients
        c1 = self.addHost('cli_1')
        c2 = self.addHost('cli_2')
        c3 = self.addHost('cli_3')
        c4 = self.addHost('cli_4')
        c5 = self.addHost('cli_5')
        c6 = self.addHost('cli_6')

        # Add servers
        s1 = self.addHost('srv_1', ip='10.0.0.101', mac='00:00:00:00:01:01')
        s2 = self.addHost('srv_2', ip='10.0.0.102', mac='00:00:00:00:01:02')
        s3 = self.addHost('srv_3', ip='10.0.0.103', mac='00:00:00:00:01:03')
        s4 = self.addHost('srv_4', ip='10.0.0.104', mac='00:00:00:00:01:04')

        # Add links
        self.addLink(sw_clients, sw_servers, port2=1)

        self.addLink(c1, sw_clients)
        self.addLink(c2, sw_clients)
        self.addLink(c3, sw_clients)
        self.addLink(c4, sw_clients)
        self.addLink(c5, sw_clients)
        self.addLink(c6, sw_clients)

        self.addLink(s1, sw_servers, port2=2)
        self.addLink(s2, sw_servers, port2=3)
        self.addLink(s3, sw_servers, port2=4)
        self.addLink(s4, sw_servers, port2=5)


topos = {'mytopo': lambda: MyTopo()}
```

3. Controlador POX con l2_learning

```
$sudo mn --custom topo.py --topo mytopo --controller remote --test pingall
        *** Creating network
        *** Adding controller
        *** Adding hosts:
        cli_1 cli_2 cli_3 cli_4 cli_5 cli_6 srv_1 srv_2 srv_3 srv_4
        *** Adding switches:
```

```
s1 s2
*** Adding links:
(cli_1, s1) (cli_2, s1) (cli_3, s1) (cli_4, s1) (cli_5, s1)
 (cli_6, s1) (s1, s2) (srv_1, s2) (srv_2, s2) (srv_3, s2) (srv_4, s2)
*** Configuring hosts
cli_1 cli_2 cli_3 cli_4 cli_5 cli_6 srv_1 srv_2 srv_3 srv_4
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...
*** Waiting for switches to connect
s1 s2
*** Ping: testing ping reachability
cli_1 -> cli_2 cli_3 cli_4 cli_5 cli_6 srv_1 srv_2 srv_3 srv_4
cli_2 -> cli_1 cli_3 cli_4 cli_5 cli_6 srv_1 srv_2 srv_3 srv_4
cli_3 -> cli_1 cli_2 cli_4 cli_5 cli_6 srv_1 srv_2 srv_3 srv_4
cli_4 -> cli_1 cli_2 cli_3 cli_5 cli_6 srv_1 srv_2 srv_3 srv_4
cli_5 -> cli_1 cli_2 cli_3 cli_4 cli_6 srv_1 srv_2 srv_3 srv_4
cli_6 -> cli_1 cli_2 cli_3 cli_4 cli_5 srv_1 srv_2 srv_3 srv_4
srv_1 -> cli_1 cli_2 cli_3 cli_4 cli_5 cli_6 srv_2 srv_3 srv_4
srv_2 -> cli_1 cli_2 cli_3 cli_4 cli_5 cli_6 srv_1 srv_3 srv_4
srv_3 -> cli_1 cli_2 cli_3 cli_4 cli_5 cli_6 srv_1 srv_2 srv_4
srv_4 -> cli_1 cli_2 cli_3 cli_4 cli_5 cli_6 srv_1 srv_2 srv_3
*** Results: 0\% dropped (90/90 received)
*** Stopping 1 controllers
c0
*** Stopping 11 links
...........
*** Stopping 2 switches
s1 s2
*** Stopping 10 hosts
cli_1 cli_2 cli_3 cli_4 cli_5 cli_6 srv_1 srv_2 srv_3 srv_4
*** Done
completed in 6.529 seconds
```

4. Construcción del balanceador de carga

4.1. Módulo de balanceo en POX

4.2. Modificación topología

4.3. Ejecución y prueba de PING

4.4. Tráfico y flujos

Referencias