Student Name:
Student ID:

LabReSiD25
Hands-On 7

Università degli Studi di Messina

Software Defined Networking: Indirizzamento diretto

Realizzare in `mininet` una rete composta da due host interconnessi da uno switch. Aprire per ogni host una sessione `xterm` ed usare il programma `netcat` per mettersi in ascolto sulla porta `5432` del primo nodo ed instaurare una connessione TCP dal secondo nodo. Osservare il traffico generato con `Wireshark`.

Software Defined Networking: Indirizzamento indiretto

Realizzare in `mininet` una infrastruttura di rete composta da due host ed un `router`. Definire la topologia attraverso Python. Utilizzare l'oggetto Mininet ed i metodi addHost ed addLink. Settare i parametri di rete attraverso comandi unix. Validare l'infrastruttura usando i comandi dello Unix Network Toolkit, attraverso la CLI di mininet.
Parametri di rete:

- Host 1: 10.0.1.1/24
- Host 2: 10.0.2.1/24

```
from mininet.topo import Topo
from mininet.net import Mininet
from mininet.node import Node
from mininet.log import setLogLevel, info
from mininet.cli import CLI


# Classe da passare come parametro al costruttore del Router

class GenericRouter(Node):
    def config(self, **params):
        super(GenericRouter, self).config(**params)
    # Enable forwarding on the router
        self.cmd('sysctl net.ipv4.ip_forward=1')

    def terminate( self ):
        self.cmd('sysctl net.ipv4.ip_forward=0')
        super(GenericRouter, self).terminate()
```

Esercizio 3: Physical Constraints

Usando la classe `TCLink` del modulo `mininet.link` aggiungere un delay di 75ms ed una larghezza di banda di 100Mbps ad ogni link. Osservare le differenze con la topologia precedente attraverso i comandi `ping` ed `iperf`.