

Realizing Network Slicing

Topology Slicing

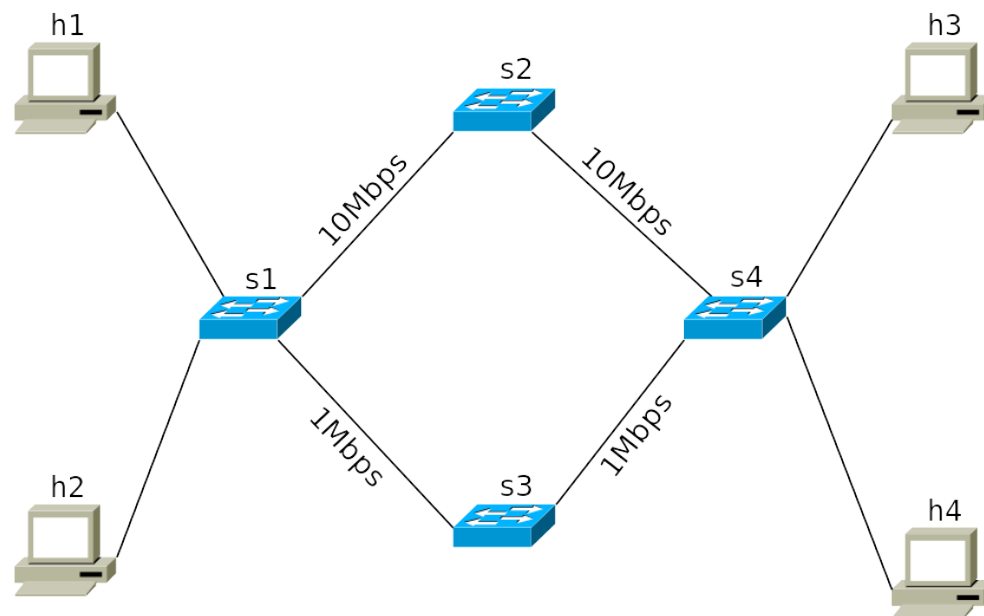
Network divided in upper slice and lower slice

Service Slicing

Network divided in two slices

for two services:

- *Video traffic slice (UDP:9999)*
- *Other traffic slice*





Realizing Network Slicing

Common parts

- ◆ Ryu controller script (***_slicing.py**)
- ◆ Python Script with Mininet Library to create network topology (**network.py**)
- ◆ Files placed in folder **~/comnetsemu/app/realizing_network_slicing**

Starting Ryu Controller

ryu-manager topology_slicing.py &
ryu-manager service_slicing.py &

Starting the network

sudo python3 network.py

Useful Mininet commands

links
net
pingall
ping
iperf
xterm



Realizing Network Slicing

Testing the network on both scenarios

- Verify connectivity (ping/pingall)
- Verify bandwidth (e.g. type “iperf h1 h3” and “iperf h2 h4” on Mininet console)
- Verify flows (e.g. use “ovs-ofctl dump-flows s1” on an extra VLAB terminal)

Specific testing service slicing

- Open mininet terminals for h1 and h3 (xterm)
- Configure h3 as server: “iper -s -u -p 9999 -b 10M”
- Configure h1 as client: “iper -c 10.0.0.3 -u -p 9999 -b 10M -t 10 -i 1”



Exercise: Enhancing Network Slicing

Enhanced Service Slicing

*Network divided in three slices
for three services:*

- *Video traffic slice (UDP:9999)*
- *Secure traffic (TCP:8888)*
- *Other traffic slice*

Hints:

- Modify network topology: add an extra path (providing 5Mbps) with a new switching device
- Update the Python controller manage the new slice