Introduction to the labs

Andrea Bianco, Paolo Giaccone
http://www.telematica.polito.it/

Labs

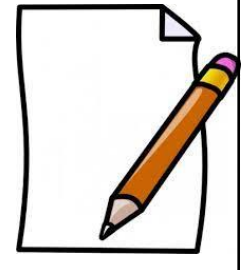
- Lab #1
  - 3 hours
  - Configuration
  - SDN and network routing
- Lab #2
  - 3 hours
  - Performance
  - QoS support: mainly scheduling
- Lab #3
  - 3 hours
  - Detailed analysis and implementation
  - Simulation of algorithms
- It is possible to do the labs at home
  - assistance is provided only in presence during the lab

Lab logistics

- LED2
  - 2nd floor south ”scavalco” on C.so Castelfidardo
- Friday 08:30-11:30 in LED2
- Please arrive 5 minutes earlier, so you can start the lab on time
- Use crownlabs
  - Working in a virtual environment
  - Use a Linux Virtual Machine

Lab detailed instructions

- 3 pdf files, one for each lab
- Print the pdf at home and bring it into the lab
  - one copy for each student
- Bring pen and papers to take notes
  - required to be able to follow the lab

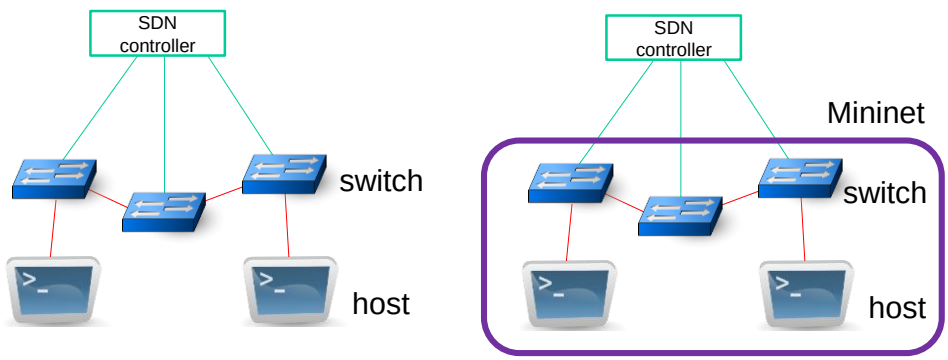


Lab #1

- Preliminary step
  - overview of the shell commands to use with the terminal application
  - consider also the hints to type quickly the commands
- Follow step by step and do not skip any step, otherwise you will not be able to follow
- Main software tool: Mininet

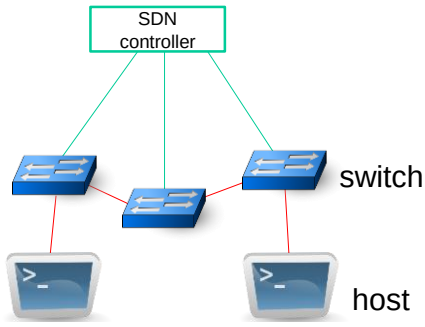
Mininet

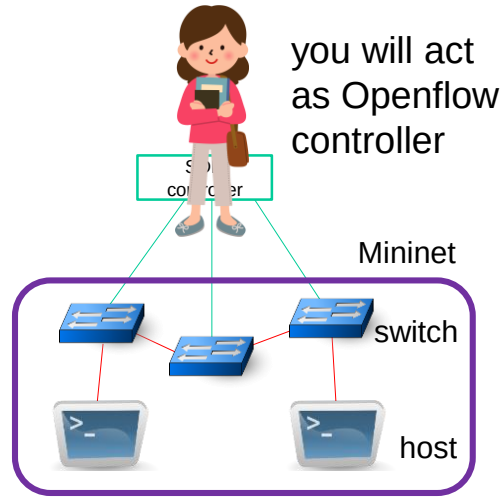
- Network emulator



Mininet

- Network emulator





TNG group - Politecnico di Torino

Introduction to SDN lab - 7

Mininet

- Network emulator
  - host
  - switch
  - SDN controller
- Linux container/process for each node
- Command line interface CLI
  - global commands for the emulator
  - local commands for the nodes

TNG group - Politecnico di Torino

Introduction to SDN lab - 8

Mininet global commands

- **nodes** display nodes
  - h1 -> host 1
  - s1 -> switch 1
- **links** display links
  - h1-eth0<->s1-eth1
- **net** display a summary of all the nodes and links
  - h1 h1-eth0:s1-eth1
- **dump** dump information about all nodes
  - <P4Host h1: h1-eth0:10.0.0.1 pid=12345>

Mininet local commands

- local commands for the nodes
  - if the first string typed a host, switch or controller name, the command is executed on that node
- **h1 ifconfig** provides the list of the network interfaces attached to h1
- **h1 ping h2** sends ICMP packets from h1 to h2
- **h1 iperf –c 10.0.0.1** uses iperf to test the bandwidth towards 10.0.0.1
- **sh** allows to run a command outside mininet, while mininet is running; e.g., mininet> **sh ls**

Network performance tool

- iperf3 to test the available bandwidth between two hosts
- client server application
  - client: generates the traffic (TCP/UDP)
  - server: receives the traffic (TCP/UDP)

- **iperf3 –c dest_IP** run the test as client
- **iperf3 –s** run the test as server
  - by default, each host is already running it in background
- many options are available: **iperf3 --help**

Lab step #1

- Become familiar with Mininet and its commands
  - two hosts and one (software) Openflow switch
  - understand the performance of software switches

Lab step #2

- Single switch topology
  - discovery in terms of node graph and IP addresses
  - test connectivity
  - add the proper match-action rules in the flow table to route the traffic
    - based only on the source port
    - based only on the destination port
  - observe the traffic through **tshark**
- Reminde the notation: IP 1.2.3.4/24 = 1.2.3.4/255.255.255.0

Lab step #3

- Two switches topology
  - discovery in terms of node graph and IP addresses
  - test connectivity and fix the routing

Lab step #4

- Multiple switches topology
  - "complex" topology
  - discovery in terms of node graph and IP addresses
  - test connectivity and fix the routing

Lab step #5

- Multipath routing
  - "complex" topology
  - route UDP and TCP flows between the same pairs of hosts on different paths

Lab step #6

- Dynamic routing
  - "complex" topology
  - a flow is rerouted from the main path to a backup one
  - flow-mod messages are issued by the controller (i.e., the student)
    - investigate whether the sequence of the messages matters or not for a completely transparent rerouting process (i.e., no losses)

Lab step #7

- Fault-tolerant rerouting (OPTIONAL)
  - detect a link failure and apply the backup path