

---

## A. Project Objectives

The objectives of this project are:

- Learning a widely-used computer network emulation (simulation) tool, Mininet
- Understanding the simulation of *Software-Defined Network* (SDN), virtual switches, and SDN controllers
- Understanding how a small-scale cloud data center hardware setup operates.

## B. Downloading and Installing Software

Mininet is a network emulator targeted for SDN-based cloud network simulation. It uses simulated hosts, virtual switches, and links on a single Linux kernel. With Mininet, one can use a variety of networking protocols and send packets through what seems like a real Ethernet interface, with any typical given link speed and delay. Packets are processed by an emulated Ethernet switch or router and queueing packets can be analyzed. Since Mininet is an emulator, any real-time program including Web servers, TCP window monitoring programs, or Wireshark can run on it. Software-defined network (SDN) that are developed in Mininet can be transferred to hardware OpenFlow switches for packet forwarding. Servers in Mininet, can create virtual machines (VMs), and thus this emulator can be used in cloud computing. Any Mininet topology can be connected to an SDN controller like OpenDaylight or Floodlight and the default port is 6633.

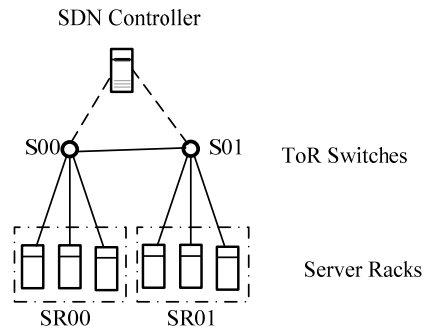
Start learning the basics of Mininet with Chapter 17 of the textbook or google it. Download its open source and install the tool. Mininet commands are preceded by a `mininet>` prompt. To use virtualization in Mininet, tools like VMware or VirtualBox must be installed. When Mininet runs on VirtualBox, for example, a prompt such as `mininet@mininet -vm:-$` is created.

## C. Experiment: Simulation of a Small Scale SDN-Based Cloud Data Center

### **Simulation 1 Implement 6 Servers, in 2 Racks, with 2 Switches and an SDN Controller.**

Implement the following data center that includes 6 servers (hosts) connected in 2 server racks forming a small data center network. Run Mininet and:

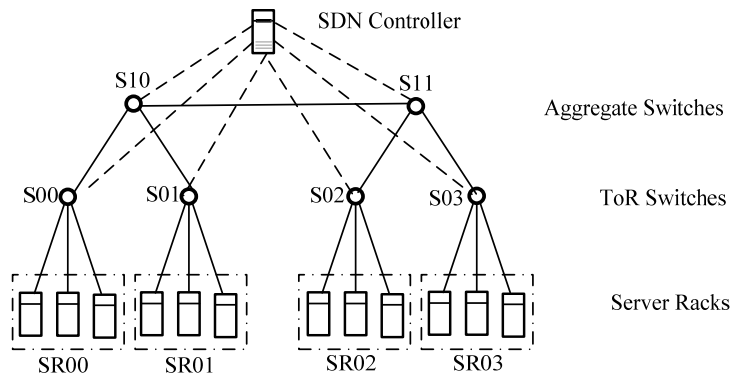
- Conduct a test using `Pingall` command and report the average round trip time (RTT). Show the snapshot of your result for all hosts.
- Comment specifically on an RTT over a connection from a server in S00 rack to a server in S01 rack.
- Experience `iperf` a server in S00 rack to a server in S01 rack. Show the snapshot of your result.
- Experience Wireshark on a server in S00 rack to a server in S01 rack. Show the snapshot of your result.



### Simulation 2 Implement 12 Servers in 4 Racks, with 6 Switches, and an SDN Controller.

Implement the following data center that includes 12 servers (hosts) in 4 server racks connected in a slightly different version of fully connected network. Run Mininet and:

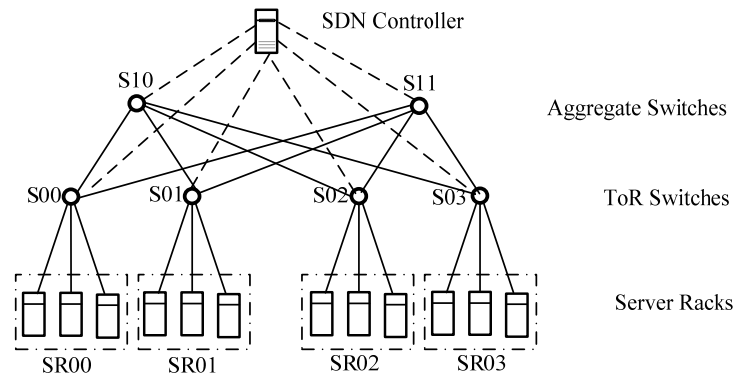
- Conduct a test using `Pingall` command and report the average round trip time (RTT). Show the snapshot of your result for all hosts.
- Comment specifically on an RTT over a connection from a server in S00 rack to a server in S03 rack.
- Experience `iperf` a server in S00 rack to a server in S03 rack. Show the snapshot of your result.
- Experience Wireshark on a server in S00 rack to a server in S03 rack. Show the snapshot of your result.



### Simulation 3 Implement 12 Servers in 4 Racks, with 6 Switches, and an SDN Controller (Fully Connected).

Implement the following data center that includes 12 servers (hosts) in 4 server racks connected in a fully connected network. Run Mininet and:

- Conduct a test using `Pingall` command and report the average round trip time (RTT). Show the snapshot of your result for all hosts.
- Comment specifically on an RTT over a connection from a server in S00 rack to a server in S03 rack.
- Experience `iperf` a server in S00 rack to a server in S03 rack. Show the snapshot of your result.
- Experience Wireshark on a server in S00 rack to a server in S03 rack. Show the snapshot of your result.



#### **D. Project Grading, Report, and Due**

**Project Team:** Every student works on the project individually and report independently.

**Grading:** As stated in the green-sheet: This project account 10% of the total course grade.

**Report and Tasks:** Run the simulation and show your log sample parameters. Collect all the necessary simulation results and present them in your report.

**Due and submission:** Project due is Thursday, May 14<sup>th</sup>. Upload your softcopy of the project report onto Canvas.