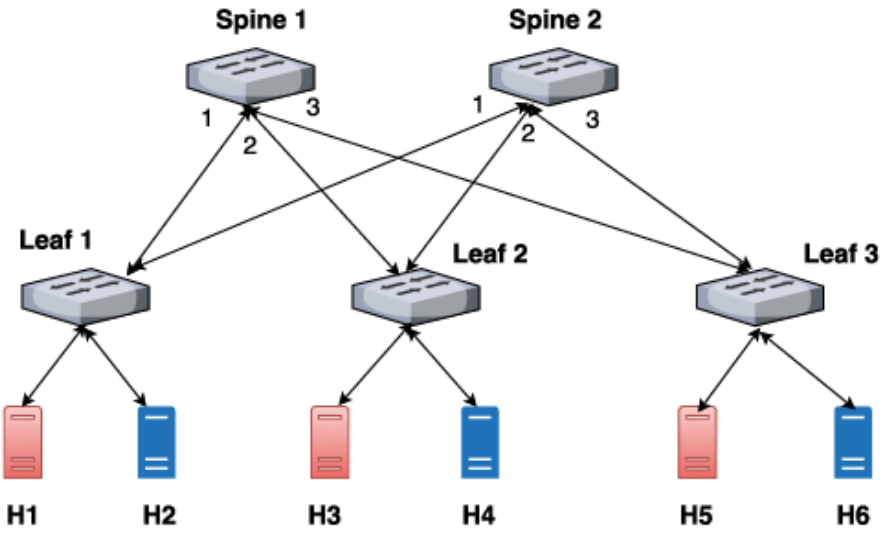
Programmable Networks

A.A. 2022/23

Project (part 1): SDN and Openflow lab

A common datacenter network is represented by a L2 leaf and spine topology where:

- racks hosting the servers are connected to the leaf switches
- the gateway to access the public network is connected to one (or more) spine switch



The traffic flows interesting a datacenter network can be grouped into two categories:

- north to south (N2S), which is traffic from/to the public network and directed/generated by servers
- east to west (E2W), i.e., traffic exchanged between different servers hosted in the datacenter

If the network is managed according to the classical L2 paradigm, then there is a large waste of bandwidth, due to the fact that the Spanning Tree Protocol (SPT) must be used in order to remove loops in the physical topology. It implies that the capacity of the network links that do not belong to the spanning tree is wasted.

To overcome this limitation we can rely on Openflow switches.
In this context, the requirement is to define and develop a control logic that handles the routing in the datacenter network in a reactive fashion: when a new traffic flow is started, then the controller chooses a path between source and destination and configures the involved switches. The selected path should be the one achieving the highest possible throughput. Once the traffic flow expires, the flow rules must be erased.

Specifically, the following points should be implemented:

- create a leaf and spine topology in Kathara, using OVS switches as network nodes. Connect a host to each leaf node (all the hosts must be in the same network). Also create a separate management network connecting the switches with the SDN controller
- in the controller, integrate and develop the following functionalities:
    - a host and network discovery component
    - a component that track the current occupation (in terms of number of flows) of each link
    - a component that compute and install the “max throughput” path for an new E2W traffic flow
    - create a component that answers to ARP requests (assume that the gateway is at 10.0.0.1)