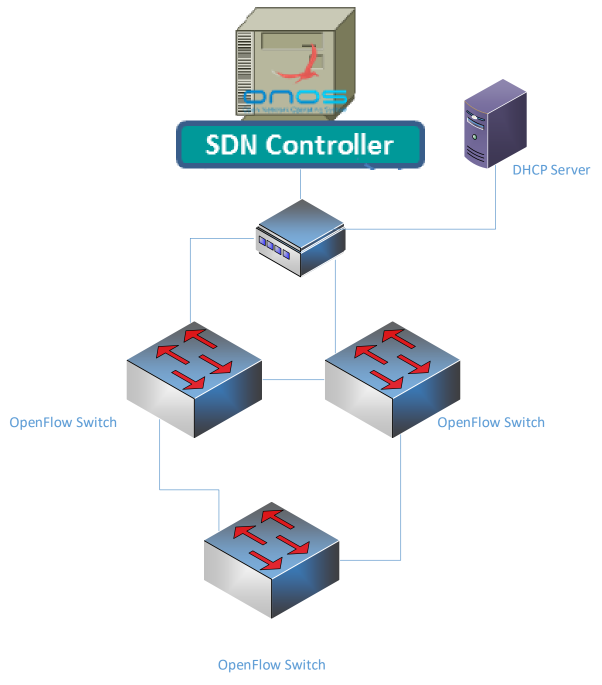
Software Defined Network (SDN) In-Band Boot Strapping

Abstract:

In traditional networks, the control software is distributed across all devices, which run routing protocols to compute forwarding state. An advantage of this design is that legacy networks can use the in-band for the control plane. This means that control plane packets are carried over the same data plane network as with the regular traffic. Because legacy routing protocols are designed to exchange data between neighboring routers and do not require global state, they are self-stabilizing in the sense that they can automatically bootstrap the network and converge to a valid operating state from any initial conditions.

In contrast, Software Defined Networking (SDN) is based on the separation of the control plane from the data plane. The network control logic is decoupled from the switches/routers and is moved to the SDN controller. This approach provides several benefits in terms of improved flexibility and performance, ease of management and decreased operational complexity, as well as lower maintenance costs. On the other hand it introduces difficulties in the support of switch-controller communication along with the data traffic. The result is that most SDN deployments today use out-of-band control, where control plane packets are carried by a dedicated management network. Moreover, the controller configuration parameters (for example: Controller IP, Controller port #) in each switch in those deployments are static.

Such way of deployment is a major obstacle and operators are looking for a way where both the control and data planes will be transmitted on the same channel (in-band mode) and there will be no need to statically configure each deployed switch. Implementing such an in-band mode is not trivial, since switches have to search and establish a path to the controller (bootstrapping) via in-band. This becomes even more complex when the switches are not directly connected to the controller and so the control plane packets must traverse through other switches in the network.



Goals:

1. Get familiar with SDN, Open Flow – start with https://www.opennetworking.org
2. Review proposal for InBand Boot Strapping using DHCP server : https://www.researchgate.net/publication/261489665_Automatic_bootstrapping_of_OpenFlow_networks
3. Implement the proposal using Mininet , RYU SDN controller , DHCP Server and OpenFlow switches where:
    a. DHCP Server should return: Switch IP address and in addition also Controller IP address, Controller Port ID using option 43 (vendor specific).
    b. Using CPqD OpenFlow switch (per https://github.com/CPqD/ofsoftswitch13) will enable to customize the DHCP client to support option 43

    Note: Learn how to raise Mininet environment using: https://github.com/mininet/openflow-tutorial/wiki

Requirements:

Python , Internet Networking Course