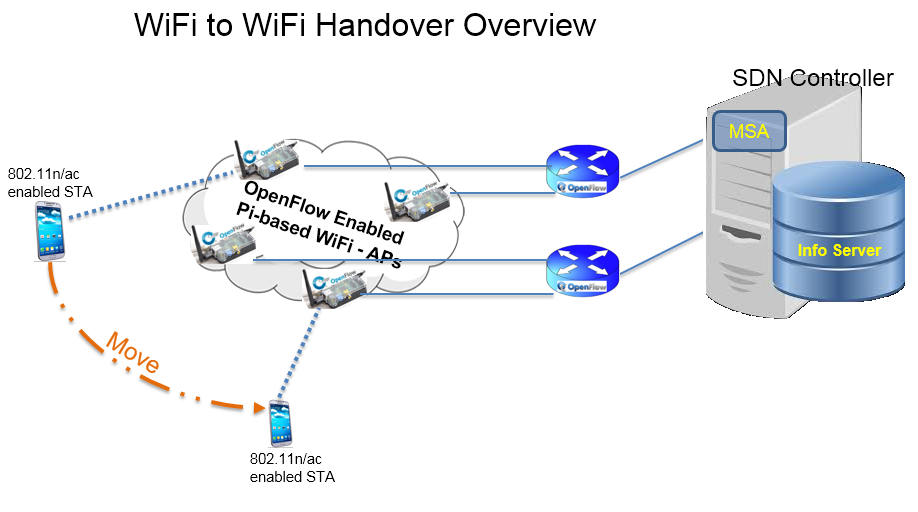
**Mobility management**

**Introduction:** This module is implemented as an application called **Mobility Service Agent (MSA)** over ONOS controller to supports a network-based mobility management. **Mobility Service Agent (MSA),** includes **two major sub-modules currently - Info Server (IS) and WiFi packet parsing (Probe Request parsing)**. **MSA** parses the OpenFlow Packet\_In, creates and assigns a unique Pseudo Static Access Point ID (PSAP) to each client and sends the PSAP info in the Packet\_Out to the client. The **Info Server** maintains the PSAP and corresponding client’s MAC-address. The **WiFi packet parsing** is a test stub component implemented to parse the injected Probe Request WiFi Packet. It can be extended further to parse all the WiFi Packets (for e.g. Beacon).

The implemented application and sub-modules are tested in the Mininet emulated environment. The main feature to here is to show the concept of Network controlled mobility in WiFi environment. Usually, the WiFi connection and mobility is client (station) controlled/driven. We wanted to make it more intelligent and network controlled with the help of Pseudo Static Access Point (PSAP) concept.

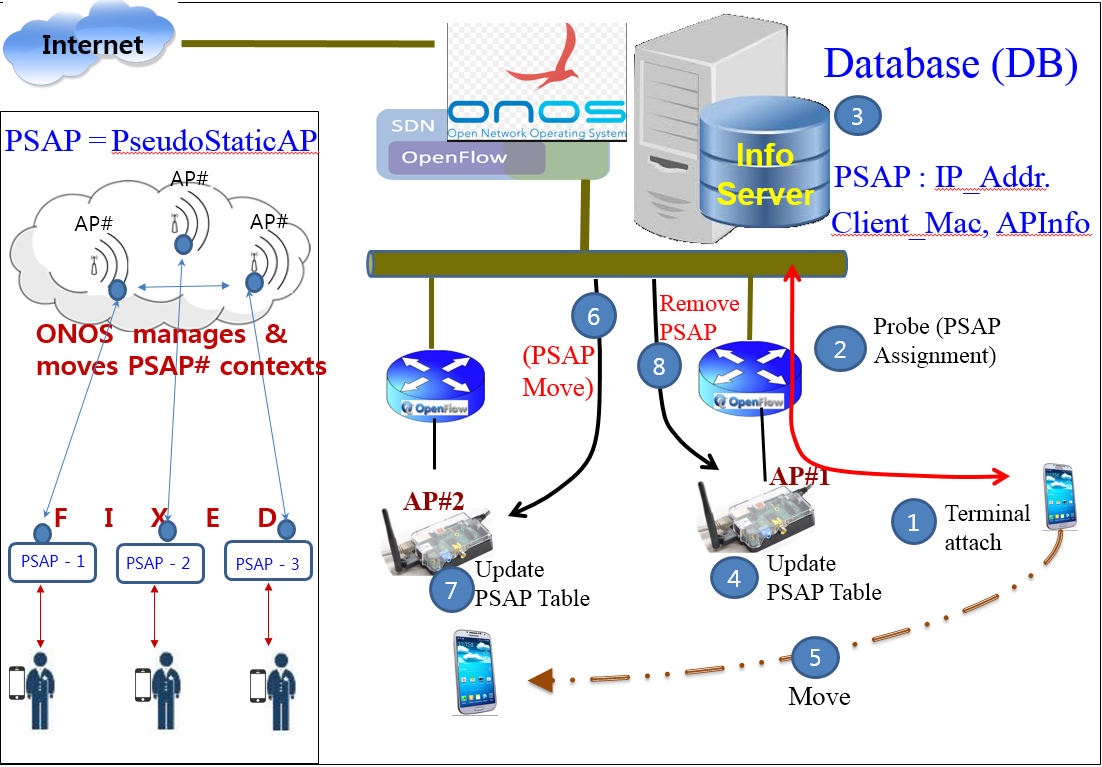
**Architecture and Operation:**



**Figure 1:** Network architecture of WiFi to WiFi handover in enterprise network

Figure 1 shows the architectural overview. The OpenFlow enabled Pi-boards are the WiFi APs which are connected with the ONOS Controller via certain OpenFlow switches. The ONOS controller hosts the MSA application that realizes the network controlled mobility in WiFi. The functionality of MSA application and its sub-modules are tested on emulated Mininet environment currently. The testbed development (Pi-board based customized APs to support PSAP) is still in progress.

Figure 2 shows the concept of PSAP, the sequence of events and how it creates the mobility perception. 1. A host is trying to attach with the AP by sending the PROBE-REQUEST. The request is received by MSA application (in ONOS) as an OpenFlow PACKET-IN message. 2. and 3. MSA parses it and extracts the host’s MAC-Address, creates a unique Pseudo Static Access Point (PSAP) ID sends it out to the AP via OpenFlow PACKET-OUT message after



**Figure 2:** Architecture implemented in first year of OpenWinCon project to achieved seamless handover

storing the required information in the Info Server (IS). 4. The AP updates its data structure by storing the PSAP ID to confirm that the host is now connected/associated. 5. The host is now moving far from the current AP and approaching towards a new AP. 6 and 7. The MSA keeps receiving the RSSI information of this host and upon realizing this fact, it moves the PSAP ID from previous AP to the new AP. 8. After successful updation of PSAP information in the new AP, the PSAP ID in old AP is removed. This way, the network controlled mobility is achieved.

**Evaluation:**

We used the local VMs and Mininet to simulate the mobility. Figure 3 shows our evaluation setups.

- Local simulation using two different Virtual Machines (VMs).

- Created a network topology in Mininet in a Virtual Machine (VM) having IP address: IP.

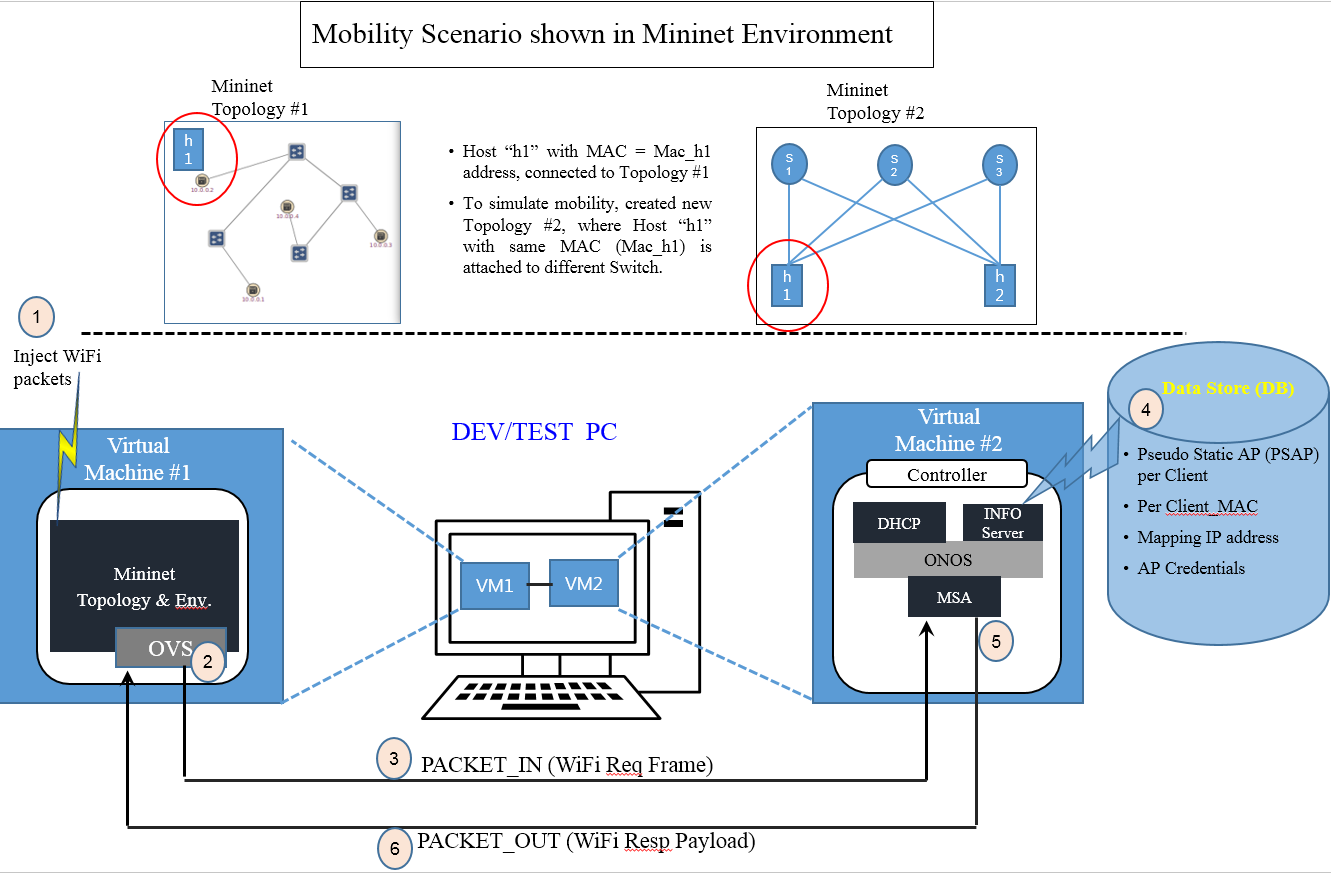
- A remotely running ONOS controller (running in a separate virtual machine that controls the network topology, created in the Mininet.

- The scapy based packet injector script developed to inject the 802.11 probe request frame is executed in Mininet (through Xterm h1).

- Once the Mininet receives the Packet-Out (Probe Response message) from the ONOS controller, we change the Mininet topology.

- A new topology simulates the roaming scenario, where the host ‘h1’ is now moved/located. The remote ONOS controller is running intact in other VM.

- Now, when we send the Probe Request (packet injected through Mininet Xterm), we can see that ONOS controller assigns the same PSAP ID to the Terminal having MAC address: 00:00:00:00:00:01.

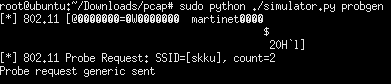


**Figure 3:** Mobility scenario emulated in Mininet environment.

**Command:**

* In one VM:
  + Cd ~/onos
  + Check bash\_profile. Notice that all setup, cell information is clearly shown.
  + Ob && onos-setup-karaf clean <IP> (In which we are running the controller).
  + Ok clean <IP>
  + Check the apps which are running ($onos> apps –a –s)
  + $onos> App activate org.onosproject.msa
  + $onos> app deactivate org.onosproject.fwd
* In other VM, we need to have the scapy and Mininet installed. Run the Mininet.
  + Cd ~/mininet
  + Sudo mn –topo linear,4 –mac –switch ovs –controller=remote,i=<IP> /\*(controller’s IP).\*/
  + Pingall
  + Open a terminal in mininet (assuming our host is h1) – mininet> xterm h1
  + In this terminal (xterm h1), run the simulator.py script with the following command:
    - Sudo python ./simulator.py probgen

This command will inject a dummy PROBE REQUEST packet and we can see the output like this:



* + - By running the wireshark, we can see the PACKET-IN and PACKET-OUT properly sent/received at Mininet & ONOS.
    - Now, stop the mininet script and create a different topology. Now, this simulates the mobility scenario. The host h1 will be connected in different way. Now, when we run the xterm h1 and so the same steps again, we can see that MSA assigns the same PSAP for the h1 host as the MAC-Addr is same.