

Deploy SDN-based DMM

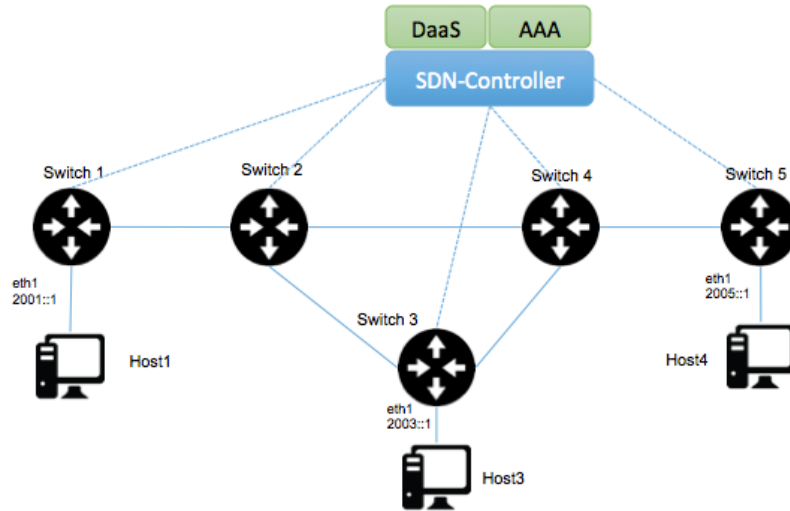


Figure 1. Testbed deployment.

- Our implementation includes 5 different files: *sdn_dmm.py*, *mobility_tracker* (directory), *sdn_dmm_mininet.py*, *run_conf_hi.py* (i=1, 3, 5), and *set_autoconf.sh*.
 - o *sdn_dmm.py* - the main application file. It is used to deploy the SDN-based DMM application running on the top of Ryu platform. This application is responsible for DMM functionality such as: authentication, HNP allocation and location update.
 - o *mobility_tracker.py* (mobilityPackage/mobility_tracker.py)- plays the role of a database (Binding Cache Entry - BCE) in S-DMM.
 - o *sdn_dmm_mininet.py* - is a python script to launch Mininet environment including 5 switches, and 3 hosts (as illustrated in Figure.1.). It is also responsible for simulating the mobility of the host 3 (from S3 to S1 and then back to S3).
 - o *run_conf_hi.py* - is a script to configure the host (e.g., set default route).
 - o *set_autoconf.sh* - is a script to enable the IPv6 auto-configuration of the hosts and the forwarding capability of the switches.
- To deploy our testbed, these necessary files need to be collected and placed into the corresponding place as follows:
 - o *sdn_dmm.py* in the application directory (/ryu/app)
 - o mobilityPackage in the library (/ryu/lib)
 - o *sdn_dmm_mininet.py* in the example directory (/mininet/examples)
 - o *run_conf_hi.py* in the example directory (/mininet/examples)
 - o *set_autoconf.sh*

Launch the environment

Open a new terminal and run the configuration script:

```
$ sudo sh set_autoconf.sh
```

Launch Mininet topology:

```
$ cd mininet/examples/  
$ sudo python sdn_dmm_mininet.py
```

Wait until Mininet's prompt appears and launch SDN-based DMM implementation (Ryu's App):

```
$ cd ryu
$ ./bin/ryu-manager --verbose --observe-links ryu/app/sdn dmm.py
```

Wait a few seconds until Ryu finishes the configuration process. Then, from the Mininet terminal, configure the host with the following commands (to obtain IPv6 address and configure the default route):

```
Mininet# h1 ./run_conf h1.py
Mininet# h3 ./run_conf h3.py
Mininet# h5 ./run_conf h5.py
```

Get the h3's IPv6 address (h3_IPv6_Addr):

```
Mininet# h3 ifconfig
```

Ping h3 from h5

From Mininet terminal, create a new h5 terminal:

```
Mininet# xterm h5
```

From h5's new terminal, ping h3:

```
h5# ping6 h3_IPv6_Addr
```

Mobility scenario

H3 first moves from S3 to S1 and then back to S3 (It is noted that the mobility is executed by using the command **exit** from Mininet terminal)

H3 moves from S3 to S1

```
Mininet# exit
```

H3 moves back to S3

```
Mininet# exit
```

Finally, from Mininet terminal type **exit**, wait a few seconds, and then type **exit** to exit the experimentation.

```
Mininet# exit
Mininet# exit
```