

CS 164 - Project

The goal of this project is to implement your own DHCP [1] server. There are no language constraints; you may use any programming language of your choice. However, this guide assumes the use of Python 3.

Preliminary Task

Use the provided `mytopo0.py` to setup a simple point-to-point client-server topology. Run the following command:

```
$ sudo mn --custom=mytopo0.py --topo=mytopo
```

Check the IP addresses of the interfaces on the client. Run the following command in the mininet console:

```
mininet> client ip address
mininet> server ip address
```

You should observe that there are no IPv4 addresses on the `client-eth0` and `server-eth0` interfaces. However, make a note of the MAC address associated with `client-eth0`.

Assign a static IPv4 address to `server-eth0`. Run the following commands in the mininet console:

```
mininet> server ip address add 192.168.0.1/24 dev server-eth0
mininet> server ip route add default dev server-eth0
```

Open a new terminal that runs within the server's network namespace. Run the following command on the mininet console:

```
mininet> xterm server
```

Run the given `dhserver.py` script on this terminal. Use the following command:

```
# python dhserver.py
```

Use the `dhclient` [2] command to get an IPv4 address on the `client-eth0` interface (This will not run successfully). Run the following command in the mininet console:

```
mininet> client dhclient -4 -v client-eth0
```

You will observe that this process tries to obtain an IPv4 address for the client-eth0 interface. You should also observe that the dhserver.py script finishes execution and prints out the MAC address of client-eth0. Use the starter code in dhserver.py to help you complete the rest of the project (if you choose to use Python 3 for your implementation).

The rest of your preliminary task is as follows. Modify the dhserver.py file to implement a rudimentary DHCP server. This will include:

1. Maintaining a pool of IPv4 addresses. These should belong to the same subnet as the IPv4 address of server-eth0. You may change this IPv4 address (statically) to use a different subnet.
2. Parsing the DHCP header and constructing an appropriate response.

Final Task

Use the provided mytopo1.py to set up a simple topology containing a single server and multiple clients. Run the following command:

```
$ sudo mn --custom=mytopo0.py --topo=mytopo
```

Run your implementation of a DHCP server on the server and use the dhclient command to get an IPv4 address on the eth0 interface of all clients.

References

1. https://en.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol
2. <https://linux.die.net/man/8/dhclient>