

Assignment 4

Dynamic Host Configuration Protocol

By Ziming Song

zs2815@nyu.edu

Part 1: Configuring DHCP

R4:

`sudo nano /etc/dhcp/dhcpd.conf`

add below lines at the end of the file:

```
subnet 10.10.11.16 netmask 255.255.255.240 {  
    range 10.10.11.17 10.10.11.30;  
    option subnet-mask 255.255.255.240;  
    option routers 10.10.11.16;  
    option broadcast-address 10.10.11.31;  
    default-lease-time 300;  
    max-lease-time 7200;  
}
```

`sudo nano /etc/network/interfaces`

```
student@CN-R4:~$ cat /etc/network/interfaces  
# interfaces(5) file used by ifup(8) and ifdown(8)  
auto lo  
iface lo inet loopback  
  
auto eth0  
iface eth0 inet static  
  
auto eth1  
iface eth1 inet static  
  
auto eth2  
iface eth2 inet static  
    address 10.10.11.17  
    netmask 255.255.255.240  
    network 10.10.11.16  
    broadcast 10.10.11.31
```

Part 2: Server Interface

```
student@CN-R4:~$ cat /etc/default/isc-dhcp-server
# Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dhcp-server)

# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
#DHCPDv4_CONF=/etc/dhcp/dhcpd.conf
#DHCPDv6_CONF=/etc/dhcp/dhcpd6.conf

# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
#DHCPDv4_PID=/var/run/dhcpd.pid
#DHCPDv6_PID=/var/run/dhcpd6.pid

# Additional options to start dhcpd with.
# Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
# Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACESv4="eth2"
INTERFACESv6=""
```

Part 3: Verifying DHCP and Wireshark

sudo systemctl status isc-dhcp-server.service

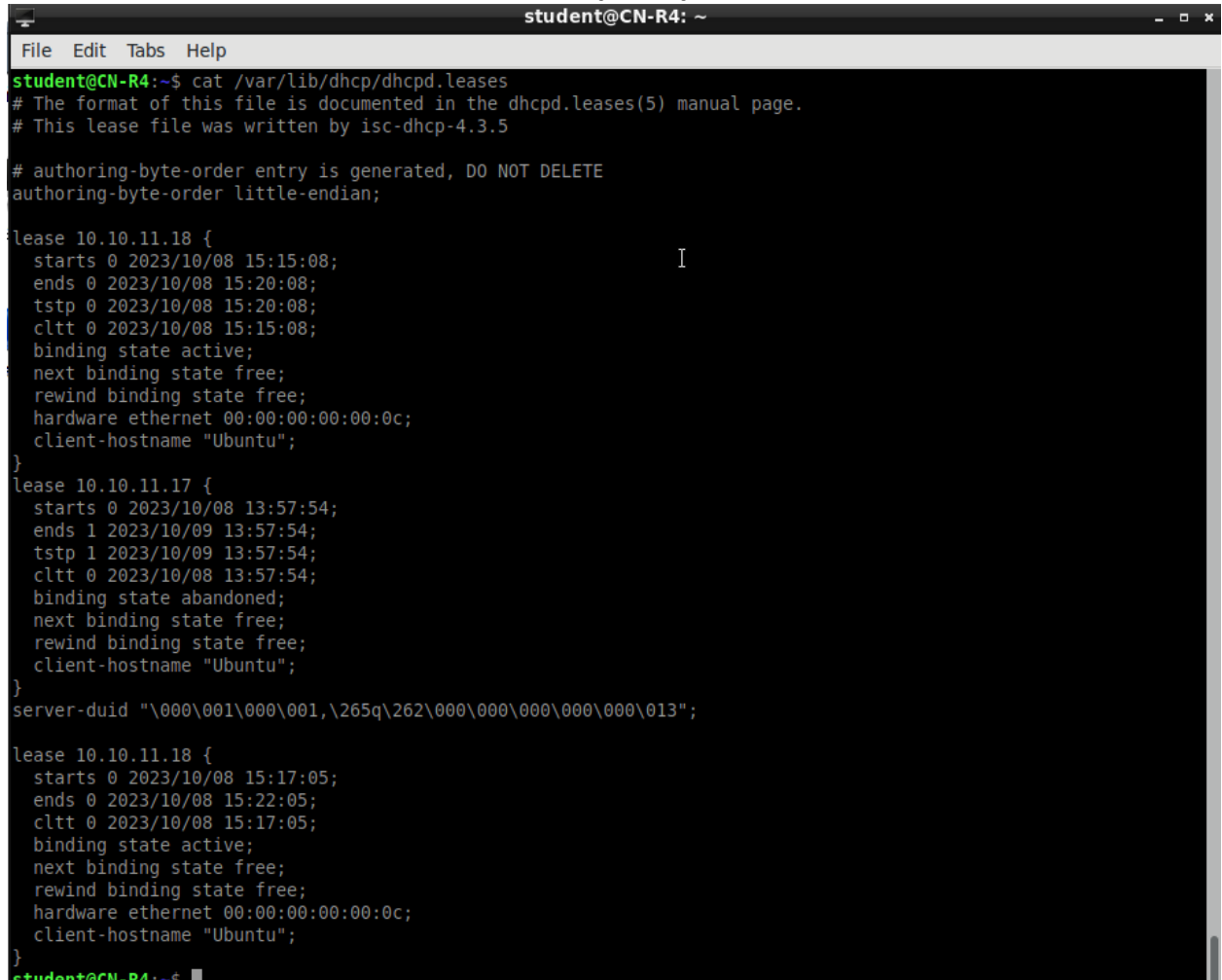
Screenshot:

```
● isc-dhcp-server.service - ISC DHCP IPv4 server
   Loaded: loaded (/lib/systemd/system/isc-dhcp-server.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2023-10-08 08:15:23 PDT; 6s ago
     Docs: man:dhcpd(8)
  Main PID: 1766 (dhcpd)
    Tasks: 1 (limit: 4670)
   Memory: 9.9M
    CGroup: /system.slice/isc-dhcp-server.service
            └─1766 dhcpd -user dhcpd -group dhcpd -f -4 -pf /run/dhcp-server/dhcpd.pid -cf /etc/dhcp/dhcpd.conf

Oct 08 08:15:23 CN-R4 dhcpd[1766]: All rights reserved.
Oct 08 08:15:23 CN-R4 dhcpd[1766]: For info, please visit https://www.isc.org/software/dhcp/
Oct 08 08:15:23 CN-R4 dhcpd[1766]: Wrote 2 leases to leases file.
Oct 08 08:15:23 CN-R4 dhcpd[1766]: Listening on LPF/eth2/00:00:00:00:00:0b/10.10.11.16/28
Oct 08 08:15:23 CN-R4 sh[1766]: Listening on LPF/eth2/00:00:00:00:00:0b/10.10.11.16/28
Oct 08 08:15:23 CN-R4 sh[1766]: Sending on LPF/eth2/00:00:00:00:00:0b/10.10.11.16/28
Oct 08 08:15:23 CN-R4 sh[1766]: Sending on Socket/fallback/fallback-net
Oct 08 08:15:23 CN-R4 dhcpd[1766]: Sending on LPF/eth2/00:00:00:00:00:0b/10.10.11.16/28
Oct 08 08:15:23 CN-R4 dhcpd[1766]: Sending on Socket/fallback/fallback-net
Oct 08 08:15:23 CN-R4 dhcpd[1766]: Server starting service.
```

Submission

- a. The leases file on R4 found in `/var/lib/dhcp/dhcpd.leases`.

A terminal window titled 'student@CN-R4: ~' with a menu bar (File, Edit, Tabs, Help). The terminal displays the output of the command 'cat /var/lib/dhcp/dhcpd.leases'. The output shows DHCP lease information for three IP addresses: 10.10.11.18, 10.10.11.17, and 10.10.11.18. Each lease entry includes details such as start/end times, timestamps, client identifiers, and binding states. The file also contains a server-uid and a comment about the format. The terminal cursor is at the end of the last line.

```
student@CN-R4:~$ cat /var/lib/dhcp/dhcpd.leases
# The format of this file is documented in the dhcpd.leases(5) manual page.
# This lease file was written by isc-dhcp-4.3.5

# authoring-byte-order entry is generated, DO NOT DELETE
authoring-byte-order little-endian;

lease 10.10.11.18 {
    starts 0 2023/10/08 15:15:08;
    ends 0 2023/10/08 15:20:08;
    tstp 0 2023/10/08 15:20:08;
    cltt 0 2023/10/08 15:15:08;
    binding state active;
    next binding state free;
    rewind binding state free;
    hardware ethernet 00:00:00:00:00:0c;
    client-hostname "Ubuntu";
}
lease 10.10.11.17 {
    starts 0 2023/10/08 13:57:54;
    ends 1 2023/10/09 13:57:54;
    tstp 1 2023/10/09 13:57:54;
    cltt 0 2023/10/08 13:57:54;
    binding state abandoned;
    next binding state free;
    rewind binding state free;
    client-hostname "Ubuntu";
}
server-uid "\000\001\000\001,\265q\262\000\000\000\000\000\013";

lease 10.10.11.18 {
    starts 0 2023/10/08 15:17:05;
    ends 0 2023/10/08 15:22:05;
    cltt 0 2023/10/08 15:17:05;
    binding state active;
    next binding state free;
    rewind binding state free;
    hardware ethernet 00:00:00:00:00:0c;
    client-hostname "Ubuntu";
}
student@CN-R4:~$
```

b. Your configuration for the DHCP.

```
student@CN-R4:~$ cat /etc/dhcp/dhcpd.conf
#
# Sample configuration file for ISC dhcpd for Debian
#
# Attention: If /etc/ltsp/dhcpd.conf exists, that will be used as
# configuration file instead of this file.
#
#
# The ddns-updates-style parameter controls whether or not the server will
# attempt to do a DNS update when a lease is confirmed. We default to the
# behavior of the version 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)
ddns-update-style none;

# option definitions common to all supported networks...
#option domain-name "example.org";
#option domain-name-servers ns1.example.org, ns2.example.org;
#option domain-name "example.org";
#option domain-name-servers ns1.exaple.org, ns2.exaple.org;

#default-lease-time 600;
#max-lease-time 7200;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
#authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
log-facility local7;

# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.

#subnet 10.152.187.0 netmask 255.255.255.0 {
#}
```

```
# which we don't really recommend.
```

```
#subnet 10.254.239.32 netmask 255.255.255.224 {  
#  range dynamic-bootp 10.254.239.40 10.254.239.60;  
#  option broadcast-address 10.254.239.31;  
#  option routers rtr-239-32-1.example.org;  
#}
```

```
# A slightly different configuration for an internal subnet.
```

```
#subnet 10.5.5.0 netmask 255.255.255.224 {  
#  range 10.5.5.26 10.5.5.30;  
#  option domain-name-servers ns1.internal.example.org;  
#  option domain-name "internal.example.org";  
#  option subnet-mask 255.255.255.224;  
#  option routers 10.5.5.1;  
#  option broadcast-address 10.5.5.31;  
#  default-lease-time 600;  
#  max-lease-time 7200;  
#}
```

```
# Hosts which require special configuration options can be listed in  
# host statements.  If no address is specified, the address will be  
# allocated dynamically (if possible), but the host-specific information  
# will still come from the host declaration.
```

```
#host passacaglia {  
#  hardware ethernet 0:0:c0:5d:bd:95;  
#  filename "vmunix.passacaglia";  
#  server-name "toccata.fugue.com";  
#}
```

I

```
# Fixed IP addresses can also be specified for hosts.  These addresses  
# should not also be listed as being available for dynamic assignment.  
# Hosts for which fixed IP addresses have been specified can boot using  
# BOOTP or DHCP.  Hosts for which no fixed address is specified can only  
# be booted with DHCP, unless there is an address range on the subnet  
# to which a BOOTP client is connected which has the dynamic-bootp flag  
# set.
```

```
#host fantasia {  
#  hardware ethernet 08:00:07:26:c0:a5;  
#  fixed-address fantasia.fugue.com;
```

```
student@CN-R4: ~  
File Edit Tabs Help  
# hardware ethernet 08:00:07:26:c0:a5;  
# fixed-address fantasia.fugue.com;  
#}  
  
# You can declare a class of clients and then do address allocation  
# based on that.  The example below shows a case where all clients  
# in a certain class get addresses on the 10.17.224/24 subnet, and all  
# other clients get addresses on the 10.0.29/24 subnet.  
  
#class "foo" {  
#  match if substring (option vendor-class-identifier, 0, 4) = "SUNW";  
#}  
  
#shared-network 224-29 {  
#  subnet 10.17.224.0 netmask 255.255.255.0 {  
#    option routers rtr-224.example.org;  
#  }  
#  subnet 10.0.29.0 netmask 255.255.255.0 {  
#    option routers rtr-29.example.org;  
#  }  
#  pool {  
#    allow members of "foo";  
#    range 10.17.224.10 10.17.224.250;  
#  }  
#  pool {  
#    deny members of "foo";  
#    range 10.0.29.10 10.0.29.230;  
#  }  
#}  
  
subnet 10.10.11.16 netmask 255.255.255.240 {  
  range 10.10.11.17 10.10.11.30;  
  option subnet-mask 255.255.255.240;  
  option routers 10.10.11.16;  
  option broadcast-address 10.10.11.31;  
  default-lease-time 300;  
  max-lease-time 7200;  
}  
student@CN-R4:~$
```

c. Screenshot of ifconfig on Ubuntu.

```
student@Ubuntu:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.10.11.18 netmask 255.255.255.240 broadcast 10.10.11.31
    inet6 fe80::200:ff:fe00:c prefixlen 64 scopeid 0x20<link>
    ether 00:00:00:00:00:0c txqueuelen 1000 (Ethernet)
    RX packets 58 bytes 5995 (5.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 50 bytes 8932 (8.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

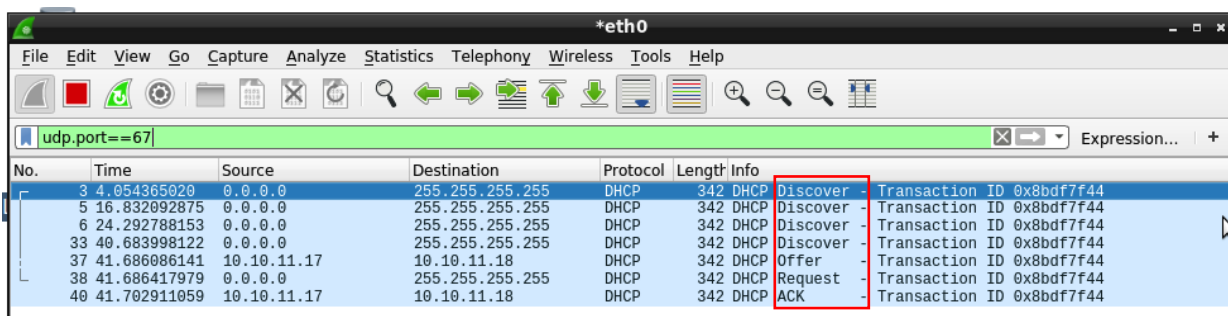
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 60 bytes 4974 (4.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 60 bytes 4974 (4.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

d. Screenshot showing Ubuntu pinging R4.

```
student@Ubuntu:~$ ping 10.10.11.17
PING 10.10.11.17 (10.10.11.17) 56(84) bytes of data.
64 bytes from 10.10.11.17: icmp_seq=1 ttl=64 time=0.421 ms
64 bytes from 10.10.11.17: icmp_seq=2 ttl=64 time=0.310 ms
64 bytes from 10.10.11.17: icmp_seq=3 ttl=64 time=0.448 ms
^C
--- 10.10.11.17 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 29ms
rtt min/avg/max/mdev = 0.310/0.393/0.448/0.059 ms
```

e. Screenshot showing Wireshark DHCP messages (4 Types).

4 Types: Discover, Offer, Request, ACK



No.	Time	Source	Destination	Protocol	Length	Info
3	4.054365020	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x8bdf7f44
4	4.054365020	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x8bdf7f44
6	24.292788153	0.0.0.0	255.255.255.255	DHCP	342	DHCP Offer - Transaction ID 0x8bdf7f44
33	40.683998122	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x8bdf7f44
37	41.686086141	10.10.11.17	10.10.11.18	DHCP	342	DHCP Offer - Transaction ID 0x8bdf7f44
38	41.686417979	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x8bdf7f44
40	41.702911059	10.10.11.17	10.10.11.18	DHCP	342	DHCP ACK - Transaction ID 0x8bdf7f44