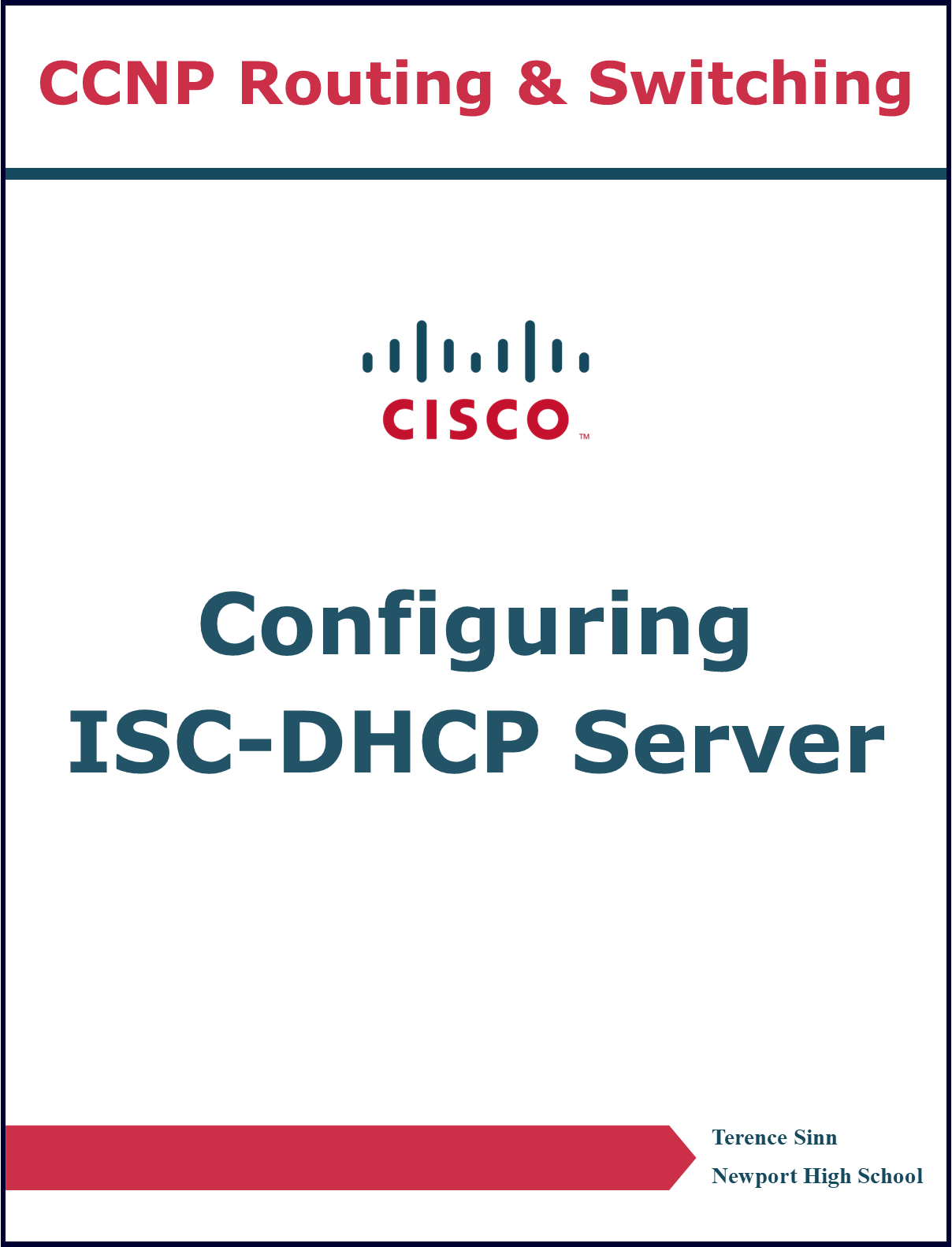
****

**Purpose**

In many modern networks, it is impractical to manually configure the IP addresses for every device that is connected. Especially with the rise of wireless networks and short-term network connections, a Dynamic Host Configuration Protocol (DHCP) server can act as a centralized point that automatically manages all the IP addressing. Though DHCP servers can be implemented within many home routers, the performance and customizability can be increased by having a dedicated server, especially with larger and larger devices that need to be configured. This lab configures an open-source implementation of a DHCP server running on Linux to server multiple clients.

**Background Information**

Dynamic Host Configuration Protocol (DHCP) is an important IP service that automatically configures IP addressing of devices that connect to the network. Its use comes from the fact that in modern networks, the network administrator may not have access to every device that connects to the network. This can be seen, for example, in the free internet serviced at malls. In a traditional network, the IP addressing and the subnet mask must be configured on the device before it can connect to the network, requiring a network admin to do this manually. This would be infeasible to do manually, especially with people coming in and out of the mall, constantly connecting and disconnecting from the network. Another problem occurs when two devices are manually configured for the same IP address. Then, the network will not know how to reach the two devices correctly. A DHCP server would solve this by letting a server handle all new and incoming devices on the network as well as automatically terminating their IP addresses after they disconnect.

DHCP uses 4 important steps: discover, offer, request, and acknowledgment. During the discover phase, the new connecting device probes the network for a DHCP server. As a result, the server responds with an offer to the new device for a new IP address. Then the device sends a request to the server to accept this offer. Finally, the server acknowledges this acceptance and records down that the IP address has been assigned to this new device. The assignment of IP addressing follows a leasing scheme, meaning that after a predetermined period of time, the server will automatically get rid of this assignment entry. For a device to keep its IP address, it would have to be relatively active on the network such that it can refresh the expiration timer on its IP address lease.

For this lab, the Internet Systems Consortium’s (ISC) open-source implementation of DHCP is used. This is because an open-source implementation can be installed on nearly all devices, including Linux distributions. The advantage of this is that a DHCP server is not very resource intensive, only sending a couple of messages per transactions. This means that being able to install it on a lightweight OS like Linux will improve the performance and stability of the server. It also allows it to be easily installed on a server hypervisor or virtual machine, allowing a single device to act as multiple separate servers because of its open source and lightweight nature.

**Lab Summary**

Two routers were connected to each other. One router acts as an external network whereas the other router acts as the default gateway for a local area network. In this local area network, there is a windows PC running virtual box. It has a virtual machine running Ubuntu that simulates a host connecting to the network. There is also another virtual machine running Ubuntu that simulates the DHCP server running the open source ISC implementation. The files /etc/dhcp/dhcpd.conf and /etc/default/isc-dhcp-server were modified in configuration.

**Lab Commands**

**Linux Ubuntu Commands**

**sudo su**

Enables root access on the Ubuntu server

**sudo apt install isc-dhcp-server**

Installs the isc-dhcp-server on the Ubuntu server.

**sudo systemctl restart isc-dhcp-server.service**

Restarts the isc-dhcp-server. This command is needed to apply any changes to configuration files.

**dhcpd.conf File Configuration**

**default-lease-time <time in seconds>;**

Sets the DHCP server’s lease time in seconds. A typical time is 600 seconds.

**max-lease-time <time in seconds>;**

Sets the DHCP server’s maximum lease-time in seconds. A typical time is 7200 seconds.

**subnet <subnet IP address> netmask <subnet mask> {**

Enables root access on the Ubuntu server

**range <start IP address> <end IP address>;**

Sets the range of IP addresses that will be assigned by the DHCP server.

**option subnet-mask <subnet mask>;**

Sets the subnet mask of the network the DHCP server will serve.

**option routers <default gateway IP address>;**

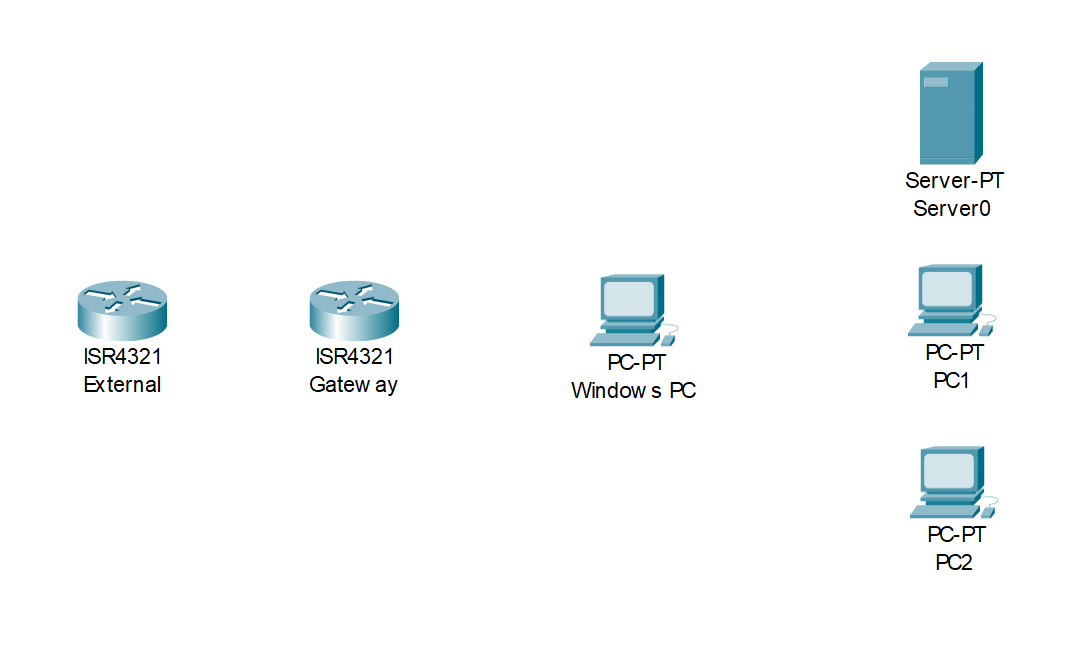
Sets the default gateway address for the network the DHCP server will serve.

**option domain-name-servers <DNS IP Address>;**

Sets the DNS address for the network the DHCP server will serve.

**}**

**Network Diagram with IP’s**



G0/0/1: 1.1.1.2/24

G0/0/0: 192.168.0.254/24

G0/0/1:

1.1.1.1/24

Ethernet:

192.168.0.16/24

Enp0s3

192.168.0.254/24

Enp0s3

192.168.0.11/24

Enp0s3

192.168.0.5/24

|  |  |  |
| --- | --- | --- |
| **Device Name** | Interface | IPv4 Address |
| **Gateway** | G0/0/0 | 192.168.0.254/24 |
|  | G0/0/1 | 1.1.1.1/24 |
|  |  |  |
| **External** | G0/0/1 | 1.1.1.2/24 |
|  |  |  |
| **DHCP Server** | Enp0s3 | 192.168.0.253/24 |
|  |  |  |
| **Windows PC** | Ethernet | 192.168.0.16/24 |
|  |  |  |
| **PC1** | Enp0s3 | 192.168.0.11/24 |
|  |  |  |
| **PC2** | Enp0s3 | 192.168.0.5/30 |
|  |  |  |

**Configurations**

**Gateway Router**

**Gateway#show running-config**

Current configuration : 1532 bytes

version 15.5

service timestamps debug datetime msec

service timestamps log datetime msec

no platform punt-keepalive disable-kernel-core

hostname Gateway

boot-start-marker

boot-end-marker

vrf definition Mgmt-intf

address-family ipv4

exit-address-family

address-family ipv6

exit-address-family

no aaa new-model

subscriber templating

multilink bundle-name authenticated

license udi pid ISR4321/K9 sn FDO21491LXV

license accept end user agreement

license boot level securityk9

spanning-tree extend system-id

redundancy

mode none

vlan internal allocation policy ascending

interface GigabitEthernet0/0/0

ip address 192.168.0.254 255.255.255.0

negotiation auto

interface GigabitEthernet0/0/1

ip address 1.1.1.1 255.255.255.0

negotiation auto

interface Serial0/1/0

no ip address

shutdown

interface Serial0/1/1

no ip address

shutdown

interface GigabitEthernet0

vrf forwarding Mgmt-intf

no ip address

shutdown

negotiation auto

interface Vlan1

no ip address

shutdown

ip forward-protocol nd

no ip http server

no ip http secure-server

ip tftp source-interface GigabitEthernet0

control-plane

line con 0

stopbits 1

line aux 0

stopbits 1

line vty 0 4

login

end

**External Router**

**External#show running-config**

Current configuration : 1438 bytes

version 15.5

service timestamps debug datetime msec

service timestamps log datetime msec

no platform punt-keepalive disable-kernel-core

hostname External

boot-start-marker

boot-end-marker

vrf definition Mgmt-intf

address-family ipv4

exit-address-family

address-family ipv6

exit-address-family

no aaa new-model

subscriber templating

multilink bundle-name authenticated

license udi pid ISR4321/K9 sn FDO21491LXV

license accept end user agreement

license boot level securityk9

spanning-tree extend system-id

redundancy

mode none

vlan internal allocation policy ascending

interface GigabitEthernet0/0/0

no ip address

shutdown

negotiation auto

interface GigabitEthernet0/0/1

ip address 1.1.1.2 255.255.255.0

negotiation auto

interface Serial0/1/0

no ip address

shutdown

interface Serial0/1/1

no ip address

shutdown

interface GigabitEthernet0

vrf forwarding Mgmt-intf

no ip address

shutdown

negotiation auto

interface Vlan1

no ip address

shutdown

ip forward-protocol nd

no ip http server

no ip http secure-server

ip tftp source-interface GigabitEthernet0

ip route 192.168.0.0 255.255.255.0 GigabitEthernet0/0/1

control-plane

line con 0

stopbits 1

line aux 0

stopbits 1

line vty 0 4

login

end

**dhcpd.conf**

default-lease-time 600;

max-lease-time 7200;

subnet 192.168.0.0 netmask 255.255.255.0 {

range 192.168.0.1 192.168.0.252;

option subnet-mask 255.255.255.0;

option routers 192.168.0.254;

option domain-name-servers 8.8.8.8;

}

**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) server DHCP requests?

# Separate multiple interfaces with spaces, e.g. "eth0 eth1".

INTERFACESv4="enp0s3"

INTERFACESv6=""

**Connectivity Tests**

**IPv4 Pings from PC1**

**ubuntutest@ubuntutest-VirtualBox:~/Desktop$ ping 192.168.0.16**

PING 192.168.0.16 (192.168.0.16) 56(84) bytes of data.

64 bytes from 192.168.0.16: icmp\_seq=1 ttl=128 time=0.340 ms

64 bytes from 192.168.0.16: icmp\_seq=2 ttl=128 time=0.526 ms

64 bytes from 192.168.0.16: icmp\_seq=3 ttl=128 time=0.460 ms

64 bytes from 192.168.0.16: icmp\_seq=4 ttl=128 time=0.475 ms

--- 192.168.0.16 ping statistics ---

4 packets transmitted, 4 received, 0% packet loss, time 3067ms

rtt min/avg/max/mdev = 0.340/0.450/0.526/0.068 ms

**ubuntutest@ubuntutest-VirtualBox:~/Desktop$ ping 192.168.0.253**

PING 192.168.0.253 (192.168.0.253) 56(84) bytes of data.

64 bytes from 192.168.0.253: icmp\_seq=1 ttl=64 time=0.483 ms

64 bytes from 192.168.0.253: icmp\_seq=2 ttl=64 time=0.736 ms

64 bytes from 192.168.0.253: icmp\_seq=3 ttl=64 time=0.812 ms

64 bytes from 192.168.0.253: icmp\_seq=4 ttl=64 time=0.671 ms

--- 192.168.0.253 ping statistics ---

4 packets transmitted, 4 received, 0% packet loss, time 3066ms

rtt min/avg/max/mdev = 0.483/0.675/0.812/0.121 ms

**ubuntutest@ubuntutest-VirtualBox:~/Desktop$ ping 192.168.0.254**

PING 192.168.0.254 (192.168.0.254) 56(84) bytes of data.

64 bytes from 192.168.0.254: icmp\_seq=1 ttl=255 time=0.460 ms

64 bytes from 192.168.0.254: icmp\_seq=2 ttl=255 time=0.588 ms

64 bytes from 192.168.0.254: icmp\_seq=3 ttl=255 time=0.585 ms

64 bytes from 192.168.0.254: icmp\_seq=4 ttl=255 time=0.527 ms

--- 192.168.0.254 ping statistics ---

4 packets transmitted, 4 received, 0% packet loss, time 3051ms

rtt min/avg/max/mdev = 0.460/0.540/0.588/0.052 ms

**ubuntutest@ubuntutest-VirtualBox:~/Desktop$ ping 1.1.1.1**

PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.

64 bytes from 1.1.1.1: icmp\_seq=1 ttl=255 time=0.470 ms

64 bytes from 1.1.1.1: icmp\_seq=2 ttl=255 time=0.561 ms

64 bytes from 1.1.1.1: icmp\_seq=3 ttl=255 time=0.574 ms

64 bytes from 1.1.1.1: icmp\_seq=4 ttl=255 time=0.545 ms

--- 1.1.1.1 ping statistics ---

4 packets transmitted, 4 received, 0% packet loss, time 3067ms

rtt min/avg/max/mdev = 0.470/0.537/0.574/0.040 ms

**ubuntutest@ubuntutest-VirtualBox:~/Desktop$ ping 1.1.1.2**

PING 1.1.1.2 (1.1.1.2) 56(84) bytes of data.

64 bytes from 1.1.1.2: icmp\_seq=2 ttl=254 time=0.568 ms

64 bytes from 1.1.1.2: icmp\_seq=3 ttl=254 time=0.610 ms

64 bytes from 1.1.1.2: icmp\_seq=4 ttl=254 time=0.592 ms

64 bytes from 1.1.1.2: icmp\_seq=5 ttl=254 time=0.586 ms

--- 1.1.1.2 ping statistics ---

5 packets transmitted, 4 received, 20% packet loss, time 4075ms

rtt min/avg/max/mdev = 0.568/0.589/0.610/0.015 ms

**Problems**

The first problem was that the isc-dhcp-server could not be installed. The typical problem on virtual machines being unable to install a package is that the virtual machine is on bridged mode instead of NAT mode. This means that the virtual machine is choosing to use the wired LAN network instead of the typical NAT connection to the internet. This means that the virtual machine is unable to access the internet and thus is unable to install the package. In this case, the virtual machine was on NAT mode however the problem was that another application was already use apt. The most probable cause of this is that the operating system was using the apt application to install updates. Waiting for all updates to finish would likely fix this but the **sudo kill -9 <PID>** and **sudo killall**  **<process name>** commands were used to stop apt. In the first command, the process ID of apt was needed to shut it down, however apt sometimes uses multiple processes. Thus, the **sudo killall apt** is a simpler command to instantly stop apt. After this, the isc-dhcp-server could be installed.

A large problem was that devices on the network were sometimes able to ping each other and sometimes unable to ping each other. After the initial lab setup, there was full connectivity but on reboot, the virtual machines were only able to ping each other. To investigate, the devices were all reverted to static IP addressing. However, the same problem was still there. On another reboot, this lab setup did end up having connectivity, so the DHCP was implemented once more. After this, the virtual machines were still only able to ping each other. Then, the windows PC that VirtualBox is installed on top of was configured using DHCP, and the PC had full connectivity. In this case, the PC was able to ping the virtual machines however the virtual machines could not ping the PC. The lab setup was reset and a DHCP server was configured on the Cisco router instead, yielding full connectivity. After this, the dhcpd.conf file was re-examined and was verified to be correct. The inconsistency of connectivity between the same isc-dhcp configuration, using a Cisco router as the DHCP server, and static IP addressing suggested that this was not a layer 3 issue. As a result, the ARP entries were also examined. The ARP tables for some of the virtual machines were complete yet would still not be able to ping outside of the virtual machine. For some of the virtual machines, there wasn’t even an entry to the default gateway. Because the only consistent aspects of all these trials was that the virtual machines could ping each other but not outside and that the PC could ping the virtual machines but not the other way around, the problem must have been in transition between the virtual machines and the PC. For a long time, it was hypothesized that the problem was a VirtualBox bug where the bridged mode would be inconsistent. Upon further inspection, the problem ended up being with the windows PC’s firewall. The firewall was off for private networks but on for public networks. As a result, it seems the bridged adapter is inconsistent with how it treats the network, sometimes as public and sometimes as private. Turning off the firewall for both public and private finally solved the issue.

**Conclusion**

A DHCP server is an effective, scalable solution that can automatically handle IP address assignment. On top of the fact that it is quite lightweight and simple to configure, a DHCP server should be a part of every LAN network to have quick accessibility for new connections. With more and more devices moving away from standard ethernet connections and into wireless BYOD connections, a DHCP server should be treated as a requirement for any new LAN.