Anjuman-i-Islam’s

M. H. Saboo Siddik College Of Engineering

Computer Engineering Department

|  |
| --- |
| Anjuman-i-Islam’s  M. H. Saboo Siddik College Of Engineering  Computer Engineering Department  **Lab Manual**  **Network Programming Laboratory**  (2015 Rev)  Semester VI Computers  Prepared By: Er. Asadullah Shaikh  ER. Farhana Siddiqui |

**INDEX**

|  |  |
| --- | --- |
| EXP. NO. | EXPERIMENT NAME |
| 1 | Study of Networking Commands (Ping, Tracert, TELNET,  nslookup, netstat, ARP, RARP) and Network Configuration  Files. |
| 2 | Linux Network Configuration.  i. Configuring NIC’s IP Address.  ii. Determining IP Address and MAC Address using  if-config command.  iii. Changing IP Addess using ifconfig.  iv. Static IP Address and Configuration by Editing.  v. Determining IP Address using DHCP.  vi. Configuring Hostname in /etc/hosts file. |
| 3 | Setting up multiple IP Addresses on a single LAN. |
| 4 | Using netstat and route commands to do the following.  i. View current routing table.  ii. Add and delete routes.  iii. Change default gateway. |
| 5 | Using GUI configuration Tools to add /configure Ethernet Card. |
| 6 | Configuring Linux as a router by enabling IP Forwarding. |
| 7 | Configuring remote login Services, telnet & ssh.  i. To install and configure TELNET server.  ii. To set up SSH and connect to remote machine. |
| 8 | To configure Linux FTP server using VSFTPD.  i. Set up anonymous access of FTP server.  ii. Enable individual logins and add FTP users with Readonly access.  iii. Transfer Files. |
| 9 | To install and configure DHCP server and client. |
| 10 | To install and configure Web server. |

**Exp 1:** Study of Network Commands(Ping, Tracert,Telnet nslookup,netstat,arp,rarp) and network configuration files.

### *Ping*

### Check whether the local network interface is up and running

To check whether the local network is up and running using any one of the following methods.

#### Ping localhost using zero (0)

This is probably the easiest and simplest way to ping a local host

$ ping 0

PING 0 (127.0.0.1) 56(84) bytes of data.

64 bytes from 127.0.0.1: icmp\_seq=1 ttl=64 time=0.024 ms

^C

#### Ping localhost using name

$ ping localhost

### Increase or Decrease the Time Interval Between Packets

By default ping waits for 1 second before sending the next packet. You can increase or decrease this using option -i as shown below.

#### Increase Ping Time Interval

Example: Wait for 5 seconds before sending the next packet.

$ ping -i 5 192.168.2.1

#### Decrease Ping Time Interval

Example: Wait 0.1 seconds before sending the next packet.

# ping -i 0.1 IP

Note: Only super user can specify interval less than 0.2 seconds. If not, you’ll get the following error message.

$ ping -i 0.1 127.0.0.1

PING 0 (127.0.0.1) 56(84) bytes of data.

ping: cannot flood; minimal interval, allowed for user, is 200ms

### Send N packets and stop

Send N packets specified with -c option and then stop. This way the ping command can exit automatically instead of pressing CTRL+C to exit.In the following example, ping command sends 5 packets, and waits for response from the destination host. Ping will exit after receiving the response or error.

$ ping -c 5 google.com

### Show Version and Exit

Display the current version of ping program using -V option.

$ ping -V

ping utility, iputils-sss20071127

* **Print Only Ping Command Summary Statistics**

$ ping -c 5 -q 127.0.0.1

PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.

--- 127.0.0.1 ping statistics ---

5 packets transmitted, 5 received, 0% packet loss, time 3999ms

rtt min/avg/max/mdev = 0.041/0.049/0.055/0.009 ms

* **Send N packets and stop**

$ ping -c 4 google.com

PING google.com (74.125.135.100) 56(84) bytes of data.

64 bytes from plus.google.com (74.125.135.100): icmp\_req=1 ttl=128 time=251 ms

64 bytes from plus.google.com (74.125.135.100): icmp\_req=2 ttl=128 time=180 ms

64 bytes from plus.google.com (74.125.135.100): icmp\_req=3 ttl=128 time=179 ms

64 bytes from plus.google.com (74.125.135.100): icmp\_req=4 ttl=128 time=179 ms

--- google.com ping statistics ---

4 packets transmitted, 4 received, 0% packet loss, time 3005ms

rtt min/avg/max/mdev = 179.569/197.734/251.433/31.005 ms

* **Timeout**

The following example will ping for 5 seconds. i.e ping command will exit after 5 seconds irrespective of how many packets are sent or received.

$ ping -w 5 localhost

### Flood the network

Super users can send hundred or more packets per second using -f option. It prints a ‘.’ when a packet is sent, and a backspace is printed when a packet is received.

As shown below, ping -f has sent more than 400,000 packets in few seconds.

# ping -f localhost

PING localhost (127.0.0.1) 56(84) bytes of data.

.^C

--- localhost ping statistics ---

427412 packets transmitted, 427412 received, 0% packet loss, time 10941ms

rtt min/avg/max/mdev = 0.003/0.004/1.004/0.002 ms, ipg/ewma 0.025/0.004 ms

### e) Find out the IP address

You can identify the ip-address using the host name as shown below.

$ ping -c 1 google.com

### f) Print Only Ping Command Summary Statistics

Use option -q to view only the ping statistics summary as shown below.

$ ping -c 5 -q 127.0.0.1

PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.

--- 127.0.0.1 ping statistics ---

5 packets transmitted, 5 received, 0% packet loss, time 3998ms

rtt min/avg/max/mdev = 0.047/0.053/0.061/0.009 ms

### g) Change Ping Packet Size

You can change the packet size of ping command using -s option.

Example: Change the default packet size from 56 to 100.

$ ping -s 100 localhost

PING localhost (127.0.0.1) 100(128) bytes of data.

108 bytes from localhost (127.0.0.1): icmp\_seq=1 ttl=64 time=0.022 ms

108 bytes from localhost (127.0.0.1): icmp\_seq=2 ttl=64 time=0.021 ms

108 bytes from localhost (127.0.0.1): icmp\_seq=3 ttl=64 time=0.020 ms

^C

--- localhost ping statistics ---

3 packets transmitted, 3 received, 0% packet loss, time 1998ms

rtt min/avg/max/mdev = 0.020/0.021/0.022/0.000 ms

#### Ping Packet Size

In the above example, when we set the packet size to 100, it displays ’128 bytes’ in the output. This is because of the Ping packet header size, which is 28 bytes. So, if you specify the packet size as 100, 28 bytes for header will be added to it and 128 bytes will be sent.

Ping Bytes Sent = Ping Packet Size + Ping Header Packet Size (28 bytes)

### h) Specify path for ping to send the packet

You can also specify through which path the ping should send the packet to destination.

$ ping hop1 hop2 hop3 .. hopN destination

$ ping 192.168.3.33 192.168.7.1 192.168.4.45

### i) Record and print route of how ECHO\_REQUEST sent and ECHO\_REPLY received

It records, and prints the network route through which the packet is sent and received. This is useful for network engineers who wish to know how the packet is sent and received.

$ ping -R 192.168.1.63

PING 192.168.1.63 (192.168.1.63) 56(84) bytes of data.

64 bytes from 192.168.1.63: icmp\_seq=1 ttl=61 time=2.05 ms

RR: 192.168.9.118

192.168.3.25

192.168.10.35

192.168.1.26

192.168.1.63

192.168.1.63

192.168.10.4

192.168.3.10

192.168.4.25

64 bytes from 192.168.1.63: icmp\_seq=2 ttl=61 time=2.00 ms (same route)

***Traceroute / MTR***

# mtr google.com

# mtr --curses google.com

### Omit Reverse DNS using –no-dns

MTR finds the hostname of each router/node by using Reverse DNS Lookup. If you want to avoid doing a reverse DNS lookup, use –no-dns option.

mtr --curses --no-dns google.com

### Execute mtr in Report Mode using –report

Instead of running MTR in interactive mode, you can run it in report mode using –report. In report mode, mtr will run for the number of cycles ( default 10 ), and then prints the statistics and exit. This mode will be useful for generating statstics about network quality.

$ mtr --no-dns --report google.com

HOST: lakshmanan Loss% Snt Last Avg Best Wrst StDev

1.|-- 192.168.1.1 0.0% 10 2.5 3.0 2.4 4.2 0.6

2.|-- 10.228.129.9 0.0% 10 235.0 74.4 34.0 235.0 67.9

3.|-- 10.228.149.14 0.0% 10 154.8 65.5 38.7 154.8 34.8

4.|-- 116.202.226.145 0.0% 10 60.9 66.9 48.2 102.4 15.4

5.|-- 116.202.226.17 0.0% 10 54.1 65.1 36.0 194.5 46.8

6.|-- 72.14.215.234 0.0% 10 44.5 78.8 39.2 252.7 64.5

7.|-- 72.14.232.110 0.0% 10 55.7 66.4 39.1 179.8 41.8

8.|-- 66.249.94.72 0.0% 10 68.9 90.3 68.9 133.6 18.6

9.|-- 72.14.233.105 0.0% 10 68.8 76.3 68.8 92.2 7.3

10.|-- 173.194.38.162 0.0% 10 88.7 107.3 72.2 293.1 65.8

In the above example, mtr run for 10 cycles and collected the statistics. Users can change the number of cycles using the -c option.

### Verify Packet Loss

There is a common practice among the service provides to “Rate Limit” the ICMP traffic. This can provide an illusion of packet loss, when in fact there is no loss. To verify whether the loss is real or due to rate limiting, check the “Loss%” of the next hop. If it shows 0.0%, then you can be sure that the “Loss%” reported is due to the ICMP rate limiting and not actual loss.

10.|-- 209.85.250.237 0.0% 10 85.6 97.5 76.0 172.0 27.2

11.|-- 209.85.250.203 100.0 10 0.0 0.0 0.0 0.0 0.0

12.|-- 74.125.135.138 0.0% 10 77.2 107.3 77.2 219.5 43.5

In the above output, though it show 100.0% Loss between hop 10 and 11, the next hop 12, reports 0.0% packets loss, which means the Loss reported on hop 11, is only due to ICMP rate limiting.

asad # mtr -r -c 1 google.com

mtr -rwc 100 12.34.56.78

***NSLOOKUP***

### nslookup – Simple Example

nslookup followed by the domain name will display the “A Record” ( IP Address ) of the domain.

$ nslookup redhat.com

Server: 192.168.19.2

Address: 192.168.19.2#53

Non-authoritative answer:

Name: redhat.com

Address: 209.132.183.181

In the above output, server refers to the IP address of the DNS server. Then the below section provides the “A Record” ( IP Address ) of the domain “redhat.com”.

### Query the MX Record using -query=mx

MX ( Mail Exchange ) record maps a domain name to a list of mail exchange servers for that domain. The MX record tells that all the mails sent to “@redhat.com” should be routed to the Mail server in that domain.

nslookup -query=mx redhat.com

Server: 192.168.19.2

Address: 192.168.19.2#53

Non-authoritative answer:

redhat.com mail exchanger = 10 mx2.redhat.com.

redhat.com mail exchanger = 5 mx1.redhat.com.

Authoritative answers can be found from:

mx2.redhat.com internet address = 66.187.233.33

mx1.redhat.com internet address = 209.132.183.28

### Query the NS Record using -query=ns

NS ( Name Server ) record maps a domain name to a list of DNS servers authoritative for that domain. It will output the name serves which are associated with the given domain.

**nslookup -type=ns redhat.com**

Server: 192.168.19.2

Address: 192.168.19.2#53

Non-authoritative answer:

redhat.com nameserver = ns4.redhat.com.

redhat.com nameserver = ns2.redhat.com.

redhat.com nameserver = ns1.redhat.com.

redhat.com nameserver = ns3.redhat.com.

Authoritative answers can be found from:

ns4.redhat.com internet address = 209.132.188.218

ns2.redhat.com internet address = 209.132.183.2

ns1.redhat.com internet address = 209.132.186.218

ns3.redhat.com internet address = 209.132.176.100

### Query the SOA Record using -query=soa

SOA record ( start of authority ), provides the authoritative information about the domain, the e-mail address of the domain admin, the domain serial number, etc…

$ nslookup -type=soa redhat.com

Server: 192.168.19.2

Address: 192.168.19.2#53

Non-authoritative answer:

redhat.com

origin = ns1.redhat.com

mail addr = noc.redhat.com

serial = 2012071601

refresh = 300

retry = 180

expire = 604800

minimum = 14400

Authoritative answers can be found from:

ns1.redhat.com internet address = 209.132.186.218

* mail addr – specifies the mail address of the domain admin ( noc@redhat.com )
* serial – sort of revision numbering system. The standard convention is to use “YYYYMMYYNN” format. ( 2012-07-16. 01 will be incremented, if more than one edit has taken place on a same day )
* refresh – specifies ( in seconds ), when the secondary DNS will poll the primary to see if the serial number has been increased. If increased, secondary will make a new request to copy the new zone file.
* retry – specifies the interval to re-connect with the Primary DNS
* expire – specifies the time that the secondary DNS will keep the cached zone file as valid
* minimum – specifies the time that the secondary DNS should cache the zone file

### View available DNS records using -query=any

We can also view all the available DNS records using -query=any option.

$ nslookup -type=any google.com

Server: 192.168.19.2

Address: 192.168.19.2#53

Non-authoritative answer:

Name: google.com

Address: 173.194.35.7

Name: google.com

Address: 173.194.35.8

google.com nameserver = ns1.google.com.

google.com nameserver = ns2.google.com.

google.com

origin = ns1.google.com

mail addr = dns-admin.google.com

serial = 2012071701

refresh = 7200

retry = 1800

expire = 1209600

minimum = 300

google.com mail exchanger = 20 alt1.aspmx.l.google.com.

google.com mail exchanger = 30 alt2.aspmx.l.google.com.

google.com mail exchanger = 40 alt3.aspmx.l.google.com.

google.com mail exchanger = 50 alt4.aspmx.l.google.com.

google.com mail exchanger = 10 aspmx-v4v6.l.google.com.

google.com has AAAA address 2a00:1450:4002:801::1004

Authoritative answers can be found from:

ns4.google.com internet address = 216.239.38.10

ns3.google.com internet address = 216.239.36.10

### Reverse DNS lookup

You can also do the reverse DNS look-up by providing the IP Address as argument to nslookup.

$ nslookup 209.132.183.181

Server: 192.168.19.2

Address: 192.168.19.2#53

Non-authoritative answer:

181.183.132.209.in-addr.arpa name = origin-www2.redhat.com.

### Using Specific DNS server

Instead of using default DNS server’s for querying, you can also specify a particular name server to resolve the domain name.

$ nslookup redhat.com ns1.redhat.com

Server: 209.132.186.218

Address: 209.132.186.218#53

Name: redhat.com

Address: 209.132.183.181

In the above command, we have used the ns1.redhat.com as the DNS server. Here you may notice that, we don’t get any “Non-authoritative answer:” header, since ns1.redhat.com has all the zone information of redhat.com

### Change the port number to connect with

By default DNS servers uses the port number 53. If for any reasons, the port number got changed, then we can specify the port number using -port option

$ nslookup -port 56 redhat.com

### Change timeout interval to wait for a reply

You can change the default timeout to wait for a reply using -timeout option.

$ nslookup -timeout=10 redhat.com

### Enabling debug mode using -debug

You can turn on/off the debugging using -debug option in the command line

$ nslookup -debug redhat.com

Server: 192.168.19.2

Address: 192.168.19.2#53

------------

QUESTIONS:

redhat.com, type = A, class = IN

ANSWERS:

-> redhat.com

internet address = 209.132.183.181

ttl = 5

AUTHORITY RECORDS:

ADDITIONAL RECORDS:

------------

Non-authoritative answer:

Name: redhat.com

Address: 209.132.183.181

# *Netstat*

#### 1. Listing all the LISTENING Ports of TCP and UDP connections

Listing all ports (both TCP and UDP) using netstat -a option.

#### 2. Listing TCP Ports connections

Listing only TCP (Transmission Control Protocol) port connections using netstat -at.

#### 3. Listing UDP Ports connections

Listing only UDP (User Datagram Protocol ) port connections using netstat -au.

#### 4. Listing all LISTENING Connections

Listing all active listening ports connections with netstat -l.

#### 5. Listing all TCP Listening Ports

Listing all active listening TCP ports by using option netstat -lt.

#### 6. Listing all UDP Listening Ports

Listing all active listening UDP ports by using option netstat -lu.

#### 7. Listing all UNIX Listening Ports

Listing all active UNIX listening ports using netstat -lx.

#### 8. Showing Statistics by Protocol

Displays statistics by protocol. By default, statistics are shown for the TCP, UDP, ICMP, and IP protocols. The -s parameter can be used to specify a set of protocols.

#### 9. Showing Statistics by TCP Protocol

Showing statistics of only TCP protocol by using option netstat -st.

#### 10. Showing Statistics by UDP Protocol netstat -su

#### 11. Displaying Service name with PID netstat -tp

#### 13. Displaying Kernel IP routing

#### Showing Network Interface Transactions

#### Netstat -i netstat -ie

#### Finding Listening Programs

netstat -ap | grep http

### List out only listening connections

$ netstat -tnl

#### Print active connections

$ netstat -atnp | grep ESTA

If you want to check if a server like http,smtp or ntp is running or not, use grep again.

netstat -aple | grep http

**Displaying IP socket status with netstat**

**# netstat --inet -n**

**netstat -p -e --inet 74.125.68.95**

**Displaying IP socket status details with netstat**

**netstat -p -e --inet –74.125.68.95**

**Displaying the main routing table with netstat**

**# netstat -rn**

**Displaying the routing cache with netstat**

**netstat -Mn**

***ARP***

**This command is more advanced, yet still exceptionally useful when configuring networks. It observes and alters so called ARP table. ARP stands for Address Resolution Protocol. ARP table defines relationships between MAC address and IP address. In particular, for every IP address, it defines appropriate MAC address. This used when computer decides to send packet to certain IP address and it has to find MAC address for the IP address. This is when ARP table becomes useful. Computer checks if IP address is in the table and if so picks MAC address from it. If IP address is not in the table, computer uses ARP protocol to find it. arp command used to observe and manually alter ARP table entries.**

### Add new ARP entry

There are two things that you would probably like to do with ARP table; add and remove entries. This is how you add a new entry.

arp -s <ip address> <hardware address>

Again, hardware address is mostly Ethernet MAC address, but it is not always necessarily true.

### Delete ARP entry[BACK TO TOC](http://www.alexonlinux.com/useful-linux-networking-commands#table_of_contents)

arp -d <ip address>

arp -d <hardware address>

Both forms of this command delete the specified ARP address. First uses hostname or IP address to identify the ARP entry that we would like to delete. Second uses hardware address to identify appropriate ARP entry.

**Display entry for host eris:**

**arp -a eris**

Set a permanent cache entry for host **illuminati**, whose hardware address you know:

**arp -s illuminati 00:05:23:73:e6:cf**

Set an ARP proxy for host **fnord** using the eth0 interface's hardware address:

**arp -Ds fnord eth0 pub**

Remove the **fnord** ARP proxy:

**arp -i eth0 -d fnord pub**

## I. How Routing is Done?

### 1. Display Existing Routes

route command by default will show the details of the kernel routing table entries. In this example, the ip-address of the system where the route command is being executed is 192.168.1.157

$ route

Kernel IP routing table

Destination Gateway Genmask Flags Metric Ref Use Iface

192.168.1.0 \* 255.255.255.0 U 0 0 0 eth0

The above command shows that if the destination is within the network range 192.168.1.0 – 192.168.1.255, then the gateway is \*, which is 0.0.0.0.

When packets are sent within this IP range, then the MAC address of the destination is found through ARP Protocol and the packet will be sent to the MAC address.

If you don’t know what ARP is, you should first understand how ARP protocol works.

In order to send packets to destination which is not within this ip range, the packets will be forwarded to a default gateway, which decides further routing for that packet. We will see this shortly.

By default route command displays the host name in its output. We can request it to display the numerical IP address using -n option as shown below.

$ route -n

Kernel IP routing table

Destination Gateway Genmask Flags Metric Ref Use Iface

192.168.1.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0

0.0.0.0 192.168.1.10 0.0.0.0 UG 0 0 0 eth0

### 2. Adding a Default Gateway

We can specify that the packets that are not within the network has to be forwarded to a Gateway address.

The following route add command will set the default gateway as 192.168.1.10.

$ route add default gw 192.168.1.10

Now the route command will display the following entries.

$ route

Kernel IP routing table

Destination Gateway Genmask Flags Metric Ref Use Iface

192.168.1.0 \* 255.255.255.0 U 0 0 0 eth0

default gateway.co.in 0.0.0.0 UG 0 0 0 eth0

Now we have just added a default gateway to our machine. To verify whether it is working properly, ping some external host (for example, google.com) to send ICMP packet.

$ ping www.google.com

The following is the sequences of evets that happens when the above ping command is executed.

1. First it will query the DNS server to obtain the ip-address of google.com ( for example: 74.125.236.34 )
2. The destination address ( 74.125.236.34 ) is not within the network range.
3. So, in Layer-3 (IP header) the DESTINATION IP will be set as “74.125.236.34″.
4. In Layer-2, the DESTINATION MAC address will be the filled in as the MAC address of the default gateway ( 192.168.1.10′s MAC ). The MAC will be found by using ARP as described earlier.
5. When the packet is sent out, the network switch ( which works on Layer-2 ), send the packet to the default gateway since the destination MAC is that of the gateway.
6. Once the gateway receives the packet, based on its routing table, it will forward the packets further.

The above 2 examples would have given a good idea about how routing is done within a network. Now we will see other command line options available with route command.

### List Kernel’s Routing Cache Information

Kernel maintains the routing cache information to route the packets faster. We can list the kernel’s routing cache information by using the -C flag.

$ route -Cn

Kernel IP routing cache

Source Destination Gateway Flags Metric Ref Use Iface

192.168.1.157 192.168.1.51 192.168.1.51 0 0 1 eth0

192.168.1.157 74.125.236.69 192.168.1.10 0 0 0 eth0

.

.

.

### Reject Routing to a Particular Host or Network

Sometimes we may want to reject routing the packets to a particular host/network. To do that, add the following entry.

$ route add -host 192.168.1.51 reject

As you see below, we cannot access that particular host (i.e .51 host that we just rejected).

$ ping 192.168.1.51

connect: Network is unreachable

However we can still access other hosts in the network (for example, .52 host is still accessible).

$ ping 192.168.1.53

PING 192.168.1.53 (192.168.1.53) 56(84) bytes of data.

64 bytes from 192.168.1.53: icmp\_seq=1 ttl=64 time=7.77 ms

If you want to reject an entire network ( 192.168.1.1 – 192.168.1.255 ), then add the following entry.

$ route add -net 192.168.1.0 netmask 255.255.255.0 reject

Now, you cannot access any of the host in that network (for example: .51, .52, .53, etc.)

$ ping 192.168.1.51

connect: Network is unreachable

$ ping 192.168.1.52

connect: Network is unreachable

$ ping 192.168.1.53

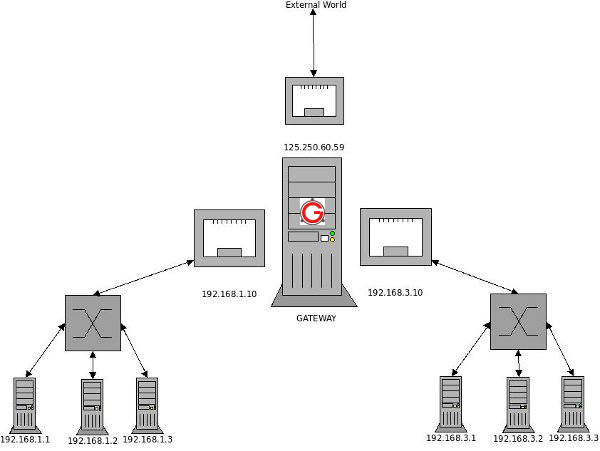
connect: Network is unreachable

## II. A Sample Network Architecture (to understand routing)

Let us use the following sample network architecture for the rest of the examples.

In the diagram below, we have 2 individual networks ( 192.168.1.0 and 192.168.3.0, with subnet mask of 255.255.255.0 ).

We also have a “GATEWAY” machine with 3 network cards. 1st card is connected to 192.168.1.0, 2nd card is connected to 192.168.3.0, and the 3rd card is connected to the external world.



### Make 192.168.3.\* Accessible from 192.168.1.\*

Now we need to add a routing entry such that we are able to ping 192.168.3. series ip-addresses from 192.168.1. series. The common point we have is the GATEWAY machine.

So, on each machine in 192.168.1.\* network a default gateway will be added as shown below.

$ route add default gw 192.168.1.10

Now when 192.168.1.1 pings 192.168.3.1, it will go to the GATEWAY via 192.168.1.10.

In GATEWAY, add the following routing entry.

$ route add -net 192.168.3.0 netmask 255.255.255.0 gw 192.168.3.10

Now all the packets addressed to 192.168.3.\* network will be forwarded via the 192.168.3.10 interface, which then delivers the packets to the addressed machine.

### 6. Make 192.168.1.\* Accessible from 192.168.3.\*

It is very similar to what we did earlier.

So, on each machine in 192.168.3.\* network a default gateway will be added as shown below.

$ route add default gw 192.168.3.10

In GATEWAY, add the following routing entry.

$ route add -net 192.168.1.0 netmask 255.255.255.0 gw 192.168.1.10

Now 192.168.3.\* machines can ping 192.168.1.\* machines.

### Allow Internet Access ( External World )

In the previous two example, we have interconnected the 2 different networks.Now we need to access the internet from these 2 different networks. For that, we can add a default routing ( when no routing rule matches ) to the 125.250.60.59 which is connected to the external world as follows.

$ route add default gw 125.250.60.59

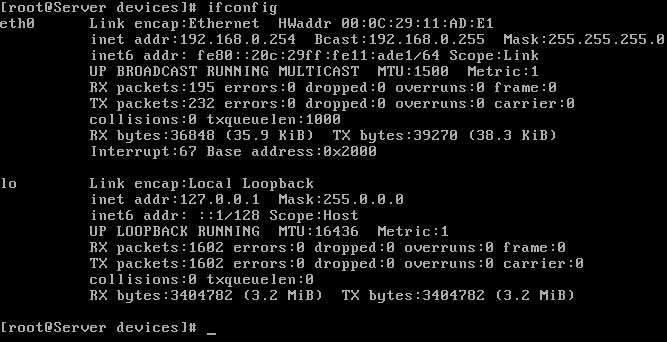
This is how it works:

1. Now when you try to access the internet (for example: ping google.com) from any of these machines (for example, from 192.168.3.2), the following is the sequence of events that happens.
2. Since the destination (google.com) is not within 3.\* series, it will be forwarded to GATEWAY via 3.10 interface
3. In GATEWAY, it checks whether the destination is within 1.\* range. In this example, it is not.
4. It then checks whether the destination is within 2.\* range. IN this example, it is not
5. Finally, it takes the default route to forward the packets (i.e using the 125.250.60.59 interface, which is connected to the external world).

EXP 2: Linux Network Configuration.

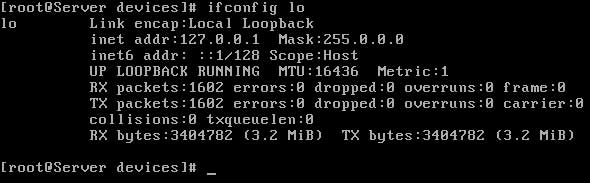
i. Configuring NIC’s IP Address.

Every node participating in networking needs a valid IP address. On Linux command prompt IP address is assigned by a network configuration window. This window can be invoked by selecting **network configuration** sub menu form **setup** command or directly executing **system-config-network** commands.Like ifconfig.



The **ifconfig** command will display the configuration of all active Ethernet card. Without specifying any parameter this command will show all active Ethernet card. if you want to see the configuration of any specific Ethernet card then use the name of that card as the command line arguments. for example to show the IP configuration on loop back Interface execute this command

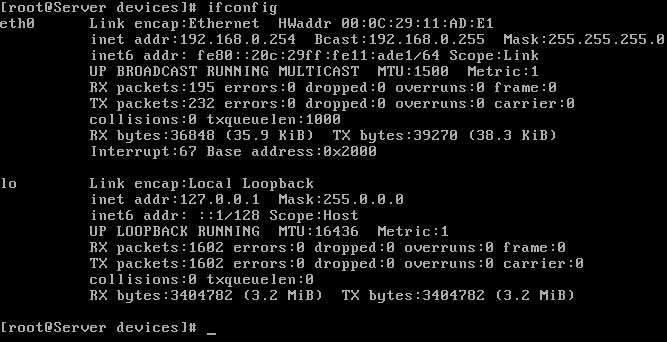
#ifconfig lo



ii. Determining IP Address and MAC Address using if-config command

Use ifconfig Command to Find Out IP Address and MAC address

You can use the ifconfig command to view the IP and mac address of your own computer or an ethernet adapter by  
# ifconfig  
# ifconfig eth0  
Sample outputs:



iii. Changing IP Addess using ifconfig.

You can change ip address using ifconfig command itself. To set IP address 192.168.1.5, enter command:  
# ifconfig eth0 192.168.1.5 netmask 255.255.255.0 up  
# ifconfig eth0

iv. Static IP Address and Configuration by Editing.

ifconfig eth0 192.168.0.10 netmask 255.255.255.0 up

This sets the IP until reboot. If you want the IP configuration to be persistent, edit /etc/network/interfaces like this:

editor /etc/network/interfaces

auto eth0

iface eth0 inet static

address 192.168.0.10

netmask 255.255.255.0

v. Determining IP Address using DHCP.

Edit /etc/network/interfaces like this:

auto lo eth0 eth0:0

iface lo inet loopback

iface eth0 inet dhcp

iface eth0:0 inet static

address ...

netmask ...

vi. Configuring Hostname in /etc/hosts file.

ou can add the host name and aliases of the storage system in the /etc/hosts file. You can use the setup command to rewrite the /etc/hosts file.

During setup, if you enable IPv6 on the storage system and configure IPv6 addresses for your network interfaces, these IPv6 addresses are also added to the/etc/hosts file.

**Step**

1. From a workstation that has access to your storage system, edit the /etc/hosts file. Add the following line to the /etc/hosts file:

***IP\_address host\_name aliases***

*IP\_address* is the IP address of the host.

*host\_name* is the name of the host.

*aliases* are the alias names for the host.

**Example**

To add a host name, myhost, with an IP address 192.0.2.16, add the following line in the /etc/hosts file:

**192.0.2.16 myhost newhost myhost-e0a**

newhost and myhost-e0a are the alias names for myhost.

The following is a sample /etc/hosts file:

#Auto-generated by setup Tue Apr 21 17:41:40 IST 2009

127.0.0.1 localhost

192.0.2.16 myhost myhost-e0a

# 0.0.0.0 myhost-e0b

# 0.0.0.0 myhost-e0c

# 0.0.0.0 myhost-e0d

The following is a sample /etc/hosts file in which an IPv6 address is also configured for the interface e0a:

#Auto-generated by setup Tue Apr 21 17:41:40 IST 2009

127.0.0.1 localhost

192.0.2.16 myhost myhost-e0a

2001:0db8::95 myhost myhost-e0a

# 0.0.0.0 myhost-e0b

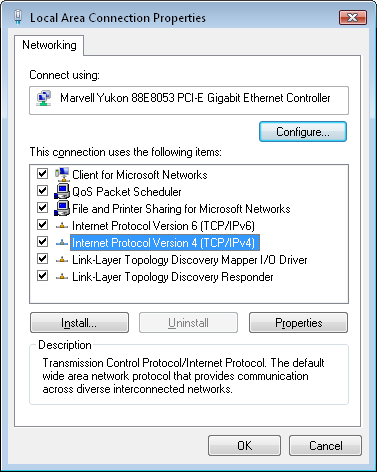
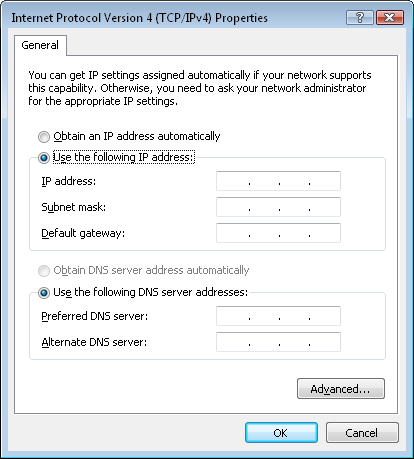
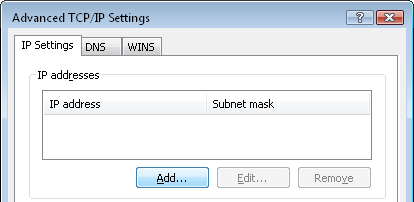
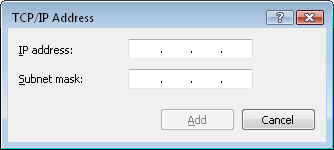
# 0.0.0.0 myhost-e0c

# 0.0.0.0 myhost-e0d

EXP 3: Setting up multiple IP Addresses on a single LAN.

There are several ways to set up multiple IP addresses on a computer:   
  
1. To have multiple network interface cards (NICs) on your computer and to assign a different IP address to each card.   
2. To assign multiple IP addresses to a single NIC.   
3. To combine 2 previous options: have multiple NICs with multiple IPs assigned to one or more of them.   
  
By default, each network interface card (NIC) has its own unique IP address. However, you can assign multiple IP addresses to a single NIC.

### How to assign multiple IP addresses to the same NIC

If you want to assign more than one IP address to a network card on Windows Vista, follow the steps below.   
  
1. Choose **Settings -> Network Connections** on the Windows **Start** menu.   
2. Right-click on the **Local Area Connection**, choose **Properties**.  
  
  
  
3. Highlight **Internet Protocol (TCP/IP)**, click **Properties**.  
  
  
  
4. If you use **DHCP**, you should disable it: click **Use the following IP address** and enter IP address, Subnet mask and Default gateway.  
5. Click **Advanced...** at the bottom.  
  
   
  
6. Enter additional IP addresses: click the **Add...** button and enter a new IP address and Subnet mask.  
  


Exp 4: Using netstat and route commands to do the following.

i. View current routing table.

When you invoke **netstat** with the –r flag, it displays the kernel routing table in the way we've been doing with **route**. On vstout, it produces:

|  |
| --- |
| # **netstat -nr**  Kernel IP routing table  Destination Gateway Genmask Flags MSS Window irtt Iface  127.0.0.1 \* 255.255.255.255 UH 0 0 0 lo  172.16.1.0 \* 255.255.255.0 U 0 0 0 eth0  172.16.2.0 172.16.1.1 255.255.255.0 UG 0 0 0 eth0 |

The –n option makes **netstat** print addresses as dotted quad IP numbers rather than the symbolic host and network names. This option is especially useful when you want to avoid address lookups over the network (e.g., to a DNS or NIS server).

The second column of **netstat** 's output shows the gateway to which the routing entry points. If no gateway is used, an asterisk is printed instead. The third column shows the “generality” of the route, i.e., the network mask for this route. When given an IP address to find a suitable route for, the kernel steps through each of the routing table entries, taking the bitwise AND of the address and the genmask before comparing it to the target of the route.

The fourth column displays the following flags that describe the route:

G: The route uses a gateway.

U: The interface to be used is up.

H: Only a single host can be reached through the route. For example, this is the case for the loopback entry 127.0.0.1.

D: This route is dynamically created. It is set if the table entry has been generated by a routing daemon like **gated** or by an ICMP redirect message .

M: This route is set if the table entry was modified by an ICMP redirect message.

!: The route is a reject route and datagrams will be dropped.

The next three columns show the MSS, Window and irtt that will be applied to TCP connections established via this route. The MSS is the Maximum Segment Size and is the size of the largest datagram the kernel will construct for transmission via this route. The Window is the maximum amount of data the system will accept in a single burst from a remote host. The acronym irtt stands for “initial round trip time.” The TCP protocol ensures that data is reliably delivered between hosts by retransmitting a datagram if it has been lost. The TCP protocol keeps a running count of how long it takes for a datagram to be delivered to the remote end, and an acknowledgement to be received so that it knows how long to wait before assuming a datagram needs to retransmitted; this process is called the round-trip time. The initial round-trip time is the value that the TCP protocol will use when a connection is first established. For most network types, the default value is okay, but for some slow networks, notably certain types of amateur packet radio networks, the time is too short and causes unnecessary retransmission. The irtt value can be set using the **route** command. Values of zero in these fields mean that the default is being used.

Finally, the last field displays the network interface that this route will use.

ii. Add and delete routes.

You might want to add or delete routes in your routing table depending on the changes in your network. You can use the route command to modify the routing table.

**Step**

1. Depending on whether you want to add or delete a route from the routing table, perform the following step:

| **If you want to...** | **Enter the following command...** |
| --- | --- |
| Add a route | **route add *destination* [*gateway metric*]**  *destination* is the IP address or host name of the destination for which the route is being added or deleted.  *gateway* is the gateway for the specified *destination*.  *metric* indicates the number of hops to the *destination*. The value of *metric* should be greater than zero when the route to the destination is through the *gateway*. The value of *metric* is zero when the *destination* is on a directly-attached network. |
| Delete a route | **route delete *destination* [*gateway metric*]**  **Attention:** You must not delete a cloned route (denoted by the C flag) from the routing table; if you do, the network connectivity to that subnet is lost. If you have deleted a cloned route, you must add the route again to the routing table in either of the following ways:   * + Bring the interface that connects to the particular subnet first to the down state and then to the up state.   You can change the state of the interface by using the ifconfig command.   * + Delete and reconfigure the IP address on the interface that connects to the particular subnet. |

For more information about the route command and options, see the na\_route(1) man page.

**Example**

To add a destination with the IP address 192.0.2.25 to the routing table, enter the following command:

**route add 192.0.2.25 gateway.com 1**

You can verify that the route to this destination is added to the routing table by using the netstat -rn or route -sn command, as shown in the following output:

system1> netstat -rn

Routing tables

Internet:

Destination Gateway Flags Refs Use Interface

default 192.0.2.1 UGS 4 184855 e0a

127.0.0.1 127.0.0.1 UH 0 0 lo

192.0.2/24 link#11 UC 2 1238 e0a

192.0.2.1 0:d0:d3:0:30:0 UHL 0 40 e0a

192.0.2.23 0:1:30:b8:30:c0 UHL 1 0 e0a

192.0.2.25 192.0.2.1 UHL 0 1285 lo

**route add -net 127.0.0.0**

adds the normal loopback entry, using netmask 255.0.0.0 (class A net, determined from the destination address) and associated with the "lo" device (assuming this device was prviously set up correctly with **[ifconfig](http://linux.about.com/library/cmd/blcmdl8_ifconfig.htm)**(8)).

**route add -net 192.56.76.0 netmask 255.255.255.0 dev eth0**

adds a route to the network 192.56.76.x via "eth0". The Class C netmask modifier is not really necessary here because 192.\* is a Class C IP address. The word "dev" can be omitted here.

**route add default gw mango-gw**

adds a default route (which will be used if no other route matches). All packets using this route will be gatewayed through "mango-gw". The device which will actually be used for that route depends on how we can reach "mango-gw" - the static route to "mango-gw" will have to be set up before.

**route add ipx4 sl0**

Adds the route to the "ipx4" host via the SLIP interface (assuming that "ipx4" is the SLIP host).

**route add -net 192.57.66.0 netmask 255.255.255.0 gw ipx4**

This command adds the net "192.57.66.x" to be gatewayed through the former route to the SLIP interface.

**route add -net 224.0.0.0 netmask 240.0.0.0 dev eth0**

This is an obscure one documented so people know how to do it. This sets all of the class D (multicast) IP routes to go via "eth0". This is the correct normal configuration line with a multicasting kernel.

**route add -net 10.0.0.0 netmask 255.0.0.0 reject**

This installs a rejecting route for the private network "10.x.x.x."

**Ubuntu route del command guide**

Ubuntu route del command is used to delete a route from Ubuntu routing table. Here is a route del step by step guide:

First, print current routing table with route command to check which route to be removed.

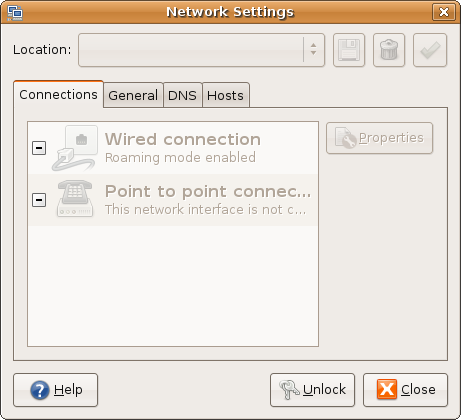
luzar@ubuntu:~$ route  
Kernel IP routing table  
Destination Gateway Genmask Flags Metric Ref Use Iface  
192.168.44.0 \* 255.255.255.0 U 0 0 0 eth0  
10.21.35.0 \* 255.255.255.0 U 0 0 0 eth0  
default 192.168.44.2 0.0.0.0 UG 100 0 0 eth0  
luzar@ubuntu:~$

Next, let's delete the network route in blue. Below is an example of Ubuntu route del command used to remove a route from network:

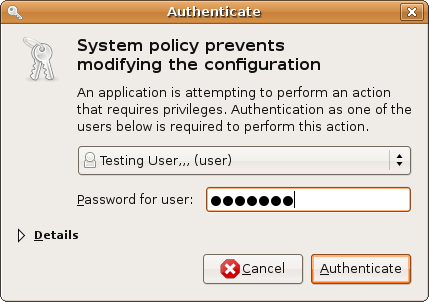
luzar@ubuntu:~$ sudo route del -net 10.21.35.0 netmask 255.255.255.0  
 dev eth0   
luzar@ubuntu:~$ route   
Kernel IP routing table  
Destination Gateway Genmask Flags Metric Ref Use Iface  
localnet \* 255.255.255.0 U 0 0 0 eth0  
default mygateway1.ar7 0.0.0.0 UG 100 0 0 eth0luzar@ubuntu:~$

Exp 5: Using GUI configuration Tools to add /configure Ethernet Card.

1. From your menu bar, go to System->Administration->Network

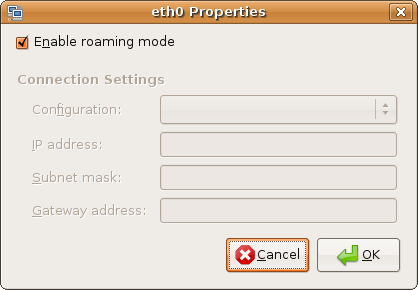


You will need to click the Unlock button and enter your password to make changes (assuming you have admin/sudo privileges on your user account - see [RootSudo](https://help.ubuntu.com/community/RootSudo)).



# Configuration

Wired connections generally work out of the box, but if the connection fails, select your "Wired connection" and hit the Properties button. Now check the box for "Enable roaming mode".



Wireless connections can prove a little more tricky, and are usually easily configured using nm-applet from the tray area. If you would rather use this method, select "Wireless connection" and hit the Properties button. Enter your wireless network's ESSID and WEP/WPA password (if applicable). Under **Connection Settings**, you will most likely want to select "Automated Configuration (DHCP)" for the "Configuration" option.

1. The preferred method is to "Enable roaming mode" - this often fixes problems when trying to connect to networks using nm-applet.

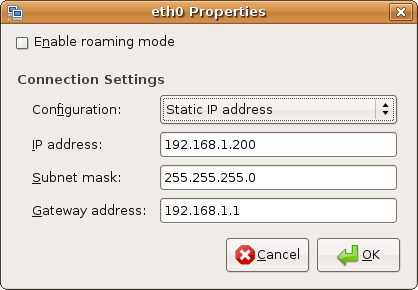
## Advanced Configuration

Please consult your router's manual for help with accessing router configuration pages, which may be needed for using advanced networking options. Most of these are available in PDF format online, just Google your router's make/model.

### Static IP address

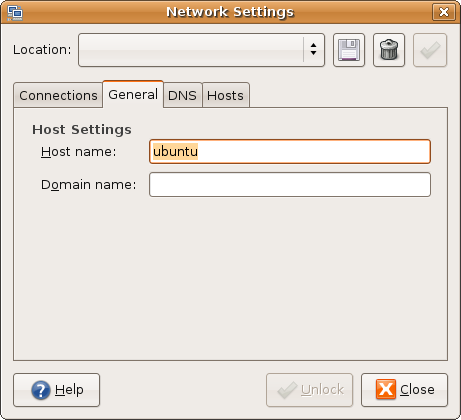
This applies to both wired and wireless network settings. First, be sure that roaming mode is disabled. I have provided you with an example using a Linksys router, but every make and model router can be a little different:

1. **Configuration:** Static IP Address
2. **IP address:** 192.168.1.200 - the last octet of the IP address should usually be between 150 and 255, which is outside the DHCP range of a standard Linksys router.
3. **Subnet mask:** 255.255.255.0
4. **Gateway address:** 192.168.1.1 - this is usually your router's internal IP address.



### General tab

Here you can set your computer's Host name and Domain name. These are needed when sharing files, folders, and printers between computers on a network. All computers should be on the same domain, and your host name is your computer's unique identifier.



### DNS tab

Most people will not want to fiddle with this unless they are using their own DNS servers, or their computer is not auto-detecting them. Since this varies between ISPs and networks, we will not cover examples here.



### Hosts tab

This is data stored in the /etc/hosts file. Here is a generic example, most computers will have very similar values:

127.0.0.1 localhost

127.0.1.1 your\_computer\_name

::1 ip6-localhost ip6-loopback

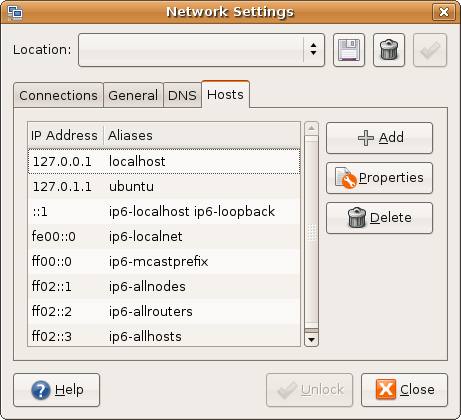
fe00::0 ip6-localnet

ff00::0 ip6-mcastprefix

ff02::1 ip6-allnodes

ff02::2 ip6-allrouters

ff02::3 ip6-allhosts



Exp 6: Configuring Linux as a router by enabling IP Forwarding.

By default any modern Linux distributions will have **IP Forwarding disabled**. This is normally a good idea, as most peoples will not need IP Forwarding, but if we are setting up a **Linux router/gateway** or maybe a **VPN server** (pptp or ipsec) or just a plain **dial-in server** then we will need to **enable forwarding**. This can be done in several ways that I will present bellow.

### Check if IP Forwarding is enabled

We have to query the **sysctl kernel**value **net.ipv4.ip\_forward** to see if forwarding is enabled or not: Using **sysctl**:

sysctl net.ipv4.ip\_forward

net.ipv4.ip\_forward = 0

or just checking out the value in the **/proc** system:

cat /proc/sys/net/ipv4/ip\_forward

0

As we can see in both the above examples this was disabled(as show by the **value 0**).

### Enable IP Forwarding on the fly

As with any sysctl kernel parameters we can change the value of **net.ipv4.ip\_forward** on the fly (without rebooting the system):

sysctl -w net.ipv4.ip\_forward=1

or

echo 1 > /proc/sys/net/ipv4/ip\_forward

the setting is changed instantly; the result will not be preserved after rebooting the system.

### Permanent setting using /etc/sysctl.conf

If we want to make this configuration permanent the best way to do it is using the file**/etc/sysctl.conf** where we can add a line containing **net.ipv4.ip\_forward = 1**

/etc/sysctl.conf:

net.ipv4.ip\_forward = 1

if you already have an entry net.ipv4.ip\_forward with the value 0 you can change that 1.

To enable the changes made in sysctl.conf you will need to run the command:

sysctl -p /etc/sysctl.conf

On RedHat based systems this is also enabled when restarting the network service:

service network restart

and on Debian/Ubuntu systems this can be also done restarting the procps service:

/etc/init.d/procps.sh restart

### Using distribution specific init scripts

Although the methods presented above should work just fine and you would not need any other method of doing this, I just wanted to note that there are also other methods to enable IP Forwarding specific to some Linux distributions. For example Debian based distributions might use the setting:

/etc/network/options:

ip\_forward=no

set it to yes and restart the network service. Also RedHat distributions might set this using:

/etc/sysconfig/network:

FORWARD\_IPV4=true

and again restart the network service.

Regardless the method you have used once you have completed this you can check it out using the same method shown above:

sysctl net.ipv4.ip\_forward

net.ipv4.ip\_forward = 1

cat /proc/sys/net/ipv4/ip\_forward

1

If the result is **1** then the Linux system will start forwarding IP packets even if they are not destined to any of its own network interfaces.

Exp 7: Configuring remote login Services, telnet & ssh.

1. To install and configure TELNET server.

Telnet service is to provide a text-oriented communications and this service is used on internet/LAN using a virtual terminal connection. Telnet by default uses 23 port number.

* 1. Install telnet use this command in terminal(Applications/Accessories/Terminal):

sudo apt-get install xinetd telnetd

* 1. Edit /etc/inetd.conf using your favourite file editor with root permission,add this line:

telnet stream tcp nowait telnetd /usr/sbin/tcpd /usr/sbin/in.telnetd

* 1. Edit /etc/xinetd.conf,make its content look like following:

# Simple configuration file for xinetd  
#  
# Some defaults, and include /etc/xinetd.d/  
defaults  
{  
# Please note that you need a log\_type line to be able to use log\_on\_success  
# and log\_on\_failure. The default is the following :  
# log\_type = SYSLOG daemon info  
instances = 60  
log\_type = SYSLOG authpriv  
log\_on\_success = HOST PID  
log\_on\_failure = HOST  
cps = 25 30  
}

* 1. You can change telnet port number by edit /etc/services with this line:

telnet 23/tcp

* 1. If you’re not satisfied with default configuration.create etc/xinetd.d/telnet, add following:

# default: on  
# description: The telnet server serves telnet sessions; it uses  
# unencrypted username/password pairs for authentication.  
service telnet  
{  
disable = no  
flags = REUSE  
socket\_type = stream  
wait = no  
user = root  
server = /usr/sbin/in.telnetd  
log\_on\_failure += USERID  
}

add these lines as you like:

only\_from = 192.168.120.0/24 #Only users in 192.168.120.0 can access to  
only\_from = .bob.com #allow access from bob.com  
no\_access = 192.168.120.{101,105} #not allow access from the two IP.  
access\_times = 8:00-9:00 20:00-21:00 #allow access in the two times  
......

1. Use this command to start telnet server:

sudo /etc/init.d/xinetd restart

### Let Telnet Listen On Another TCP Port

Letting telnet run on an alternate TCP port doesn't encrypt the traffic, but it makes it less likely to be detected as telnet traffic. Remember that this isn't a foolproof strategy; good port scanning programs can detect telnet and other applications running on alternative ports.

1) Edit your /etc/services file and add an entry for a new service. Call it stelnet.

# Local services

stelnet 7777/tcp # "secure" telnet

2) Copy the telnet configuration file called /etc/xinetd.d/telnet and call it /etc/xinetd.d/stelnet:

[root@bigboy tmp]# cp /etc/xinetd.d/telnet /etc/xinetd.d/stelnet

3) Edit the new /etc/xinetd.d/stelnet file. Make the new service stelnet and add a port statement for TCP port 7777.

# default: on

# description: The telnet server serves telnet sessions

# unencrypted username/password pairs for authentication.

service stelnet

{

flags = REUSE

socket\_type = stream

wait = no

user = root

server = /usr/sbin/in.telnetd

log\_on\_failure += USERID

disable = no

port = 7777

}

4) Use chkconfig to activate stelnet.

[root@bigboy tmp]# chkconfig stelnet on

5) Check to make sure your server is now listening on port 7777 with the netstat command.

[root@bigboy tmp]# netstat -an | grep 777

tcp 0 0 0.0.0.0:7777 0.0.0.0:\* LISTEN

[root@bigboy tmp]#

You should now be able to log in to the new stelnet server on port 7777. This is done using the telnet command with the TCP port as the second argument.

[root@smallfry tmp]# telnet 192.168.1.100 7777

Trying 192.168.1.100...

Connected to 192.168.1.100.

Escape character is '^]'.

Fedora Core release 2 (Tettnang)

Kernel 2.6.8-1.521 on an i686

login:

### Let Telnet Allow Connections From Trusted Addresses

You can restrict telnet logins access to individual remote servers by using the only\_from keyword in the telnet configuration file. Here's how.

1) Add a list of trusted servers to the /etc/xinetd.d/telnet file separated by spaces:

# default: on

# description: The telnet server serves telnet sessions

# unencrypted username/password pairs for authentication.

service telnet

{

flags = REUSE

socket\_type = stream

wait = no

user = root

server = /usr/sbin/in.telnetd

log\_on\_failure += USERID

disable = no

only\_from = 192.168.1.100 127.0.0.1 192.168.1.200

}

2) Restart telnet.

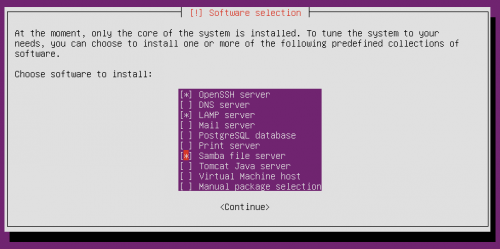
1. To set up SSH and connect to remote machine.

SSH (“secure shell”) offers a powerful collection of tools for remote control of networked computers and transfer of [data](http://www.htpcbeginner.com/install-ssh-server-on-ubuntu-1204/) between networked computers. In this post, I will explain how to install SSH server on Ubuntu system. It should work on all recent releases of Ubuntu, including 12.04 Precise Pangolin. This post is an and updated and follow-up on my previous post on how to[install SSH server](http://www.htpcbeginner.com/connecting-to-ubuntu-server-using-ssh-and-putty/) on Ubuntu. To *install SSH server* on Linux, most users resort to [OpenSSH](http://www.openssh.com/" \t "_blank). It offers a secure encrypted connection to remotely administer systems, with protection from eavesdropping and is the single most important tool for Linux administration.

## Install SSH Server

If you install Ubuntu [server](http://www.htpcbeginner.com/install-ssh-server-on-ubuntu-1204/), you can choose to install OpenSSH server from the tasksel screen as shown in the picture below. In fact, on Ubuntu server install you can run sudo tasksel command any time to install SSH server and more.

### Install SSH Server

[](http://www.htpcbeginner.com/images/2012/04/Tasksel.png)

**Ubuntu Tasksel – Package Selection**

On Ubuntu desktop or server [install](http://www.htpcbeginner.com/install-ssh-server-on-ubuntu-1204/), you can also directly install the OpenSSH server package as you would install any other package. To install SSH server on Ubuntu, all you have to do is to install the OpenSSH server package that is readily available through the respositories. Use the following command to setup SSH server:

sudo apt-get install openssh-server

**Configure SSH Server**

To **install SSH server** is very easy but making it much more [secure](http://www.htpcbeginner.com/install-ssh-server-on-ubuntu-1204/) requires a bit more work. After the installation is complete, edit the /etc/ssh/sshd\_config file. But before you start editing any configuration file, create backup of the original file:

sudo cp -a /etc/ssh/sshd\_config /etc/ssh/sshd\_config\_backup

Now, use the following command to edit the file:

sudo nano /etc/ssh/sshd\_config

The first thing you will want to edit is the port on which your SSH server listens. By default SSH server listens on port 22. Everybody knows that. Therefore, to secure your connection it is always advisable to run your SSH server on a non-standard port. So edit the following section to choose a random port number:

# What ports, IPs and protocols we listen for

Port 2122

To increase your security further you can optional customize a couple more settings. The first isPermitRootLogin. Set this to no to disallow anybody to login as root, significantly reducing the chance of serious changes by hackers.

# Authentication:

LoginGraceTime 120

PermitRootLogin no

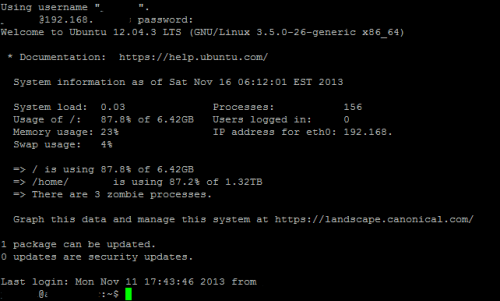
The second optional change to increase security is to list the users who are allowed to [access](http://www.htpcbeginner.com/install-ssh-server-on-ubuntu-1204/) the system remotely through SSH. To do this, add the following line to the end of the sshd\_config file:

AllowUsers user1 user2

Replace user1 and user2 with the actual usernames.

The only other thing you might want to ensure is that X11Forwarding is set to “yes”. This will allow you to run the softwares on your remote server on your local system. After you install SSH server and make any changes to the [configuration](http://www.htpcbeginner.com/install-ssh-server-on-ubuntu-1204/) file (sshd\_config) you will have to restart the service. Use the following command to restart SSH:

sudo service ssh restart

  
**Secure Shell Ubuntu**

There you go, [install SSH server](https://help.ubuntu.com/lts/serverguide/openssh-server.html) on your Ubuntu system and start enjoying your remote access.

Exp 8: To configure Linux FTP server using VSFTPD

1. Set up anonymous access of FTP server
2. Enable individual logins and add FTP users with Read only access
3. Transfer files

**Step 1:** Download vsftpd package and install it.

**Step 2:** Take the backup of Original vsftpd.conf file

asad-Vostro-1440 asad # cp /etc/vsftpd.conf /etc/vsftpd.bak.

**Step 3:** asad-Vostro-1440 asad # nano /etc/vsftpd.conf

Enable the given below bolded parameters in /etc/vsftpd.conf file.

listen=YES  
**anonymous\_enable=NO**  
**local\_enable=YES  
write\_enable=YES**  
dirmessage\_enable=YES  
use\_localtime=YES  
xferlog\_enable=YES  
connect\_from\_port\_20=YES  
**chroot\_local\_user=YES**  
secure\_chroot\_dir=/var/run/vsftpd/empty  
pam\_service\_name=vsftpd  
rsa\_cert\_file=/etc/ssl/certs/ssl-cert-snakeoil.pem  
rsa\_private\_key\_file=/etc/ssl/private/ssl-cert-snakeoil.key  
**allow\_writeable\_chroot=YES**

**Step 4:** Now restart the vsftpd service

asad-Vostro-1440 asad

# service vsftpd restart

**Step 5:** Create a file /etc/vsftpd.chroot\_list and give system username in file which you want to provide the chroot ftp access.

**Step 6 :**asad-Vostro-1440 asad

# touch /etc/vsftpd.chroot\_list

asad-Vostro-1440 asad

# nano /etc/vsftpd.chroot\_list (add user name)

asad-Vostro-1440 asad

# nano /etc/vsftpd.conf

chroot\_list\_file=/etc/vsftpd.chroot\_list

Restart the vsftpd service

#service vsftpd restart

**Step 7 :** Now check your FTP login through command line or FTP client.

waqqas@asad-Vostro-1440 ~ $ mkdir waq

waqqas@asad-Vostro-1440 ~ $ touch f1

waqqas@asad-Vostro-1440 ~ $ ftp 127.0.01

your output will be:

Connected to 127.0.01.

220 Welcome to Dell FTP service.

Name (127.0.01:waqqas): waqqas

331 Please specify the password.

Password:

230 Login successful.

Remote system type is UNIX.

Using binary mode to transfer files.

ftp> cd /home/waq

550 Failed to change directory.

ftp> cd /waq

250 Directory successfully changed.

ftp> put f1

local: f1 remote: f1

200 PORT command successful. Consider using PASV.

150 Ok to send data.

226 Transfer complete.

ftp> ls

200 PORT command successful. Consider using PASV.

150 Here comes the directory listing.

-rw------- 1 1001 1001 0 Feb 11 21:54 f1

226 Directory send OK.

ftp> get f1

local: f1 remote: f1

200 PORT command successful. Consider using PASV.

150 Opening BINARY mode data connection for f1 (0 bytes).

226 Transfer complete.

ftp> quit

221 Goodbye.

waqqas@asad-Vostro-1440 ~ $ ls

f1 waq

waqqas@asad-Vostro-1440 ~ $ ls

f1 waq

waqqas@asad-Vostro-1440 ~ $ ftp 127.0.01

Connected to 127.0.01.

220 Welcome to Dell FTP service.

Name (127.0.01:waqqas): waqqas

331 Please specify the password.

Password:

230 Login successful.

Remote system type is UNIX.

Using binary mode to transfer files.

ftp> ls

200 PORT command successful. Consider using PASV.

150 Here comes the directory listing.

-rw-rw-r-- 1 1001 1001 0 Feb 11 21:56 f1

drwxrwxr-x 2 1001 1001 4096 Feb 11 21:54 waq

226 Directory send OK.

Finish up by uncommenting command to chroot\_local\_user. When this line is set to Yes, all the local users will be jailed within their chroot and will be denied access to any other part of the server.

chroot\_local\_user=YES

Save and Exit that file.

Because of a recent vsftpd upgrade, vsftpd is "refusing to run with writable root inside chroot". A handy way to address this issue to is to take the following steps:

1. Create a new directory within the user's home directory

mkdir /home/*username*/files

1. Change the ownership of that file to root

chown root:root /home/*username*

1. Make all necessary changes within the "files" subdirectory

Then, as always, restart:

sudo service vsftpd restart

## Step Three—Access the FTP server

Once you have installed the FTP server and configured it to your liking, you can now access it.

You can reach an FTP server in the browser by typing the domain name into the address bar and logging in with the appropriate ID. Keep in mind, you will only be able to access the user's home directory.

ftp://example.com

Alternatively, you can reach the FTP server on your virtual server through the command line by typing:

ftp example.com

Fix permissions for ‘s home directory:

chmod a-w /home/user2/

Make a new directory for uploading files:

mkdir /home/user2/files  
chown user2:user2 /home/user2/files/

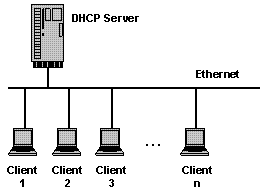
Exp 9: To install and configure DHCP server and client.

Address Pool

This method entails defining a pool (sometimes also called a range or scope) of IP addresses from which DHCP clients are supplied their configuration properties dynamically and on a fist come first serve basis. When a DHCP client is no longer on the [network](http://www.ubuntugeek.com/how-to-install-and-configure-dhcp-server-in-ubuntu-server.html) for a specified period, the configuration is expired and released back to the address pool for use by other DHCP Clients.

MAC Address

This method entails using DHCP to identify the unique [hardware](http://www.ubuntugeek.com/how-to-install-and-configure-dhcp-server-in-ubuntu-server.html) address of each network card connected to the network and then continually supplying a  constant configuration each time the DHCP client makes a request to the DHCP server using that network device.



Install DHCP server in ubuntu

sudo apt-get install dhcp3-server

This will complete the installation.

Configuring DHCP server

If you have two network cards in your ubuntu server you need to select which interface you want to use for  DHCP server listening.By default it listens to eth0.

You can change this by editing  /etc/default/dhcp3-server [file](http://www.ubuntugeek.com/how-to-install-and-configure-dhcp-server-in-ubuntu-server.html)

sudo vi /etc/default/dhcp3-server

Find this line

INTERFACES="eth0″

Replace with the following line

INTERFACES="eth1″

Save and exit.This is optional.

Next you need to make a backup copy of /etc/dhcp3/dhcpd.conf file

cp /etc/dhcp3/dhcpd.conf /etc/dhcp3/dhcpd.conf.back

Edit /etc/dhcp3/dhcpd.conf file using the following command

sudo vi /etc/dhcp3/dhcpd.conf

Using address pool method

You need to change the following sections in /etc/dhcp3/dhcpd.conf file

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

option subnet-mask 255.255.255.0;  
option broadcast-address 192.168.1.255;  
option routers 192.168.1.254;  
option domain-name-servers 192.168.1.1, 192.168.1.2;  
option domain-name "yourdomainname.com";

subnet 192.168.1.0 netmask 255.255.255.0 {  
range 192.168.1.10 192.168.1.200;  
}

save and exit the file

This will result in the DHCP server giving a client an IP address from the range 192.168.1.10-192.168.1.200 . It will lease an IP address for 600 seconds if the client doesn't ask for a specific time frame. Otherwise the maximum (allowed) lease will be 7200 seconds. The server will also "advise" the client that it should use 255.255.255.0 as its subnet mask, 192.168.1.255 as its broadcast address, 192.168.1.254 as the router/gateway and 192.168.1.1 and 192.168.1.2 as its DNS [servers](http://www.ubuntugeek.com/how-to-install-and-configure-dhcp-server-in-ubuntu-server.html).

Using MAC address method

This method is you can reserver some of the machines or all the machines with fixed ip address.In the following example i am using fixed ip address for server1,server2,printer1 and printer2

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

option subnet-mask 255.255.255.0;  
option broadcast-address 192.168.1.255;  
option routers 192.168.1.254;  
option domain-name-servers 192.168.1.1, 192.168.1.2;  
option domain-name "yourdomainname.com";

subnet 192.168.1.0 netmask 255.255.255.0 {  
range 192.168.1.10 192.168.1.200;  
}

host server1 {  
hardware ethernet 00:1b:63:ef:db:54;  
fixed-address 192.168.1.20;  
}

host server2 {  
hardware ethernet 00:0a:95:b4:d4:b0;  
fixed-address 192.168.1.21;  
}

host printer1 {  
hardware ethernet 00:16:cb:aa:2a:cd;  
fixed-address 192.168.1.22;  
}

host printer2 {  
hardware ethernet 00:0a:95:f5:8f:b3;  
fixed-address 192.168.1.23;  
}

Now you need to restart dhcp server using the following command

sudo /etc/init.d/dhcp3-server restart

Configure Ubuntu DHCP Client

If you want to configure your ubuntu desktop as DHCP client following this procedure

You need to open /etc/network/interfaces file

sudo vi /etc/network/interfaces

make sure you have the following lines (eth0 is an example)

auto lo eth0  
iface eth0 inet dhcp  
iface lo inet loopback

Save and exit the file

You need to restart networking services using the following command

sudo /etc/init.d/networking restart

How to find DHCP server IP address

You need to use the following commands

sudo dhclient

or

tail -n 15 /var/lib/dhcp3/dhclient.\*.leases

## **Installation**

To install DHCP server on Ubuntu 14.04 LTS, enter the following command:

sudo apt-get install isc-dhcp-server -y

## Configuration

DHCP server configuration is not that difficult. First, we have to assign on what interfaces should the DHCP server (dhcpd) serve DHCP requests. In my case, I have only one Interface on my system (eth0), so I assigned eth0.

To do that, edit file /etc/default/isc-dhcp-server,

sudo nano /etc/default/isc-dhcp-server

Assign the network interface:

[...]

INTERFACES="eth0"

Save and close the file.

Now, edit dhcpd.conf file,

sudo nano /etc/dhcp/dhcpd.conf

Make the changes as shown below.

Set the domain name and domain-name servers:

[...]

# option definitions common to all supported networks...

option domain-name "unixmen.local";

option domain-name-servers server.unixmen.local;

[...]

If this DHCP server is the official DHCP server for the local network, you should uncomment the following line:

[...]

authoritative;

[...]

Define the sunbet, range of ip addresses, domain and domain name servers like below:

[...]

# A slightly different configuration for an internal subnet.

subnet 192.168.1.0 netmask 255.255.255.0 {

range 192.168.1.20 192.168.1.30;

option domain-name-servers server.unixmen.local;

option domain-name "unixmen.local";

option routers 192.168.1.1;

option broadcast-address 192.168.1.255;

default-lease-time 600;

max-lease-time 7200;

}

[...]

If you want to assign a fixed IP address to your client, you should enter it’s MAC id and the IP address in the following directive. For example, I want to assign a fixed IP address 192.168.1.15 to my Ubuntu client, hence I modified the following directive as shown below.

[...]

host ubuntu-client {

hardware ethernet 00:22:64:4f:e9:3a;

fixed-address 192.168.1.15;

}

[...]

After making all the changes you want, save and close the file. Be mindful that if you have another unused entries on the dhcpd.conf file, comment all of them. Otherwise, you’ll get issues while starting dhcp service.

Now, restart dhcp service and make it to start automatically on every reboot.

sudo service isc-dhcp-server restart

Likewise, you can start/stop dhcp service as shown below:

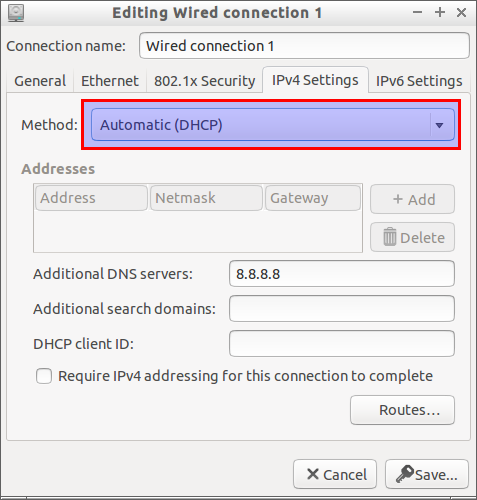
sudo service isc-dhcp-server start

sudo service isc-dhcp-server stop

## Configure Clients

Now, go to the client configuration network settings and change the IP settings to Automatic (DHCP).

Here is my Lubuntu 14.04 settings:



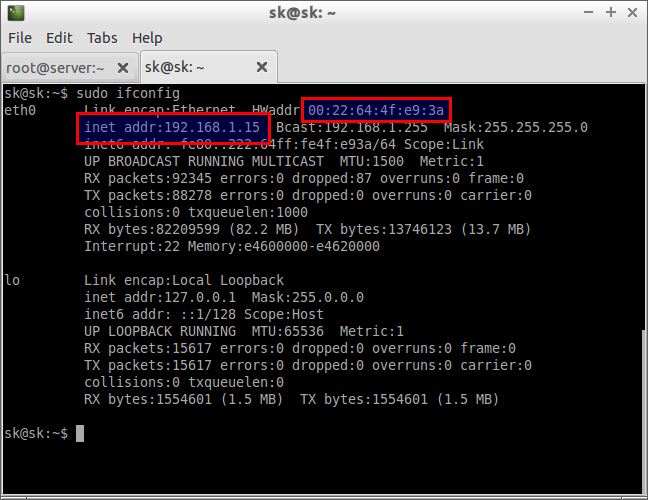
Restart the network or reboot the client system to get IP address automatically from the DHCP server.

Now, you should see the IP address has been automatically assigned to the clients from the DHCP server.

Run the following command from the client system Terminal:

sudo ifconfig

Sample output:



As you see in the above picture, My ubuntu client system which has MAC id 00:22:64:4f:e9:3a gets a fixed IP address 192.168.1.15 from the DHCP server.

# apt-get install isc-dhcp-server  
Sample outputs:

Reading package lists... Done

Building dependency tree

Reading state information... Done

Suggested packages:

isc-dhcp-server-ldap

The following NEW packages will be installed:

isc-dhcp-server

0 upgraded, 1 newly installed, 0 to remove and 11 not upgraded.

Need to get 0 B/411 kB of archives.

After this operation, 938 kB of additional disk space will be used.

Preconfiguring packages ...

Selecting previously deselected package isc-dhcp-server.

(Reading database ... 281728 files and directories currently installed.)

Unpacking isc-dhcp-server (from .../isc-dhcp-server\_4.1.1-P1-15+squeeze8\_amd64.deb) ...

Processing triggers for man-db ...

Setting up isc-dhcp-server (4.1.1-P1-15+squeeze8) ...

Generating /etc/default/isc-dhcp-server...

Starting ISC DHCP server: dhcpdcheck syslog for diagnostics. ... failed!

failed!

invoke-rc.d: initscript isc-dhcp-server, action "start" failed.

## Configure the DHCP server

The configuration file for dhcpd is called /etc/dhcp/dhcpd.conf. The file comes with a number of global configuration options. Type the following command to edit the file:  
# vi /etc/dhcp/dhcpd.conf  
You must prevent the DHCP server from receiving DNS information from clients, set the following global option (this is a security feature):

ddns-update-style none;

You need to set your domain name and name server:

## Set a domain name for your LAN ##

option domain-name "nixcraft.net.in"*;*

## Set DNS server IP address, you can set to your ISP's dns server too or use Google DNS server##

option domain-name-servers 192.168.1.2, 192.168.1.3*;*

Increase the lease time. The time is set in seconds:

### Set the length in seconds that will be assigned to a lease if the client requesting the lease does not ask for a specific expiration time. ##

### This is used for both DHCPv4 and DHCPv6 leases (it is also known as the "valid lifetime" in DHCPv6). ###

default-lease-time 86400*;*

## Set the maximum length in seconds that will be assigned to a lease ##

max-lease-time 604800*;*

The authoritative directive should be uncommented:

authoritative*;*

The authoritative directive indicate that the DHCP server should send DHCPNAK messages to misconfigured clients. If this is not done, clients will be unable to get a correct IP address after changing subnets until their old lease has expired, which could take quite a long time. Finally, update the configuration file with your subnet as follows:

subnet 192.168.1.0 netmask 255.255.255.0 {

## dhcp start and end IP range ##

range 192.168.1.100 192.168.1.200*;*

option subnet-mask 255.255.255.0*; ## subnet*

option broadcast-address 192.168.1.255*; ## broadcast*

option routers 192.168.1.254*; ## router IP*

}

Where,

1. subnet 192.168.1.0 netmask 255.255.255.0 { - The subnet statement is used to provide dhcpd with enough information to tell whether or not an IP address is on that subnet. It may also be used to provide subnet-specific parameters and to specify what addresses may be dynamically allocated to clients booting on that subnet. Such addresses are specified using the range declaration. In this example 192.168.1.0 is the subnet-number and should be an IP address or domain name which resolves to the subnet number of the subnet being described. The netmask 255.255.255.0 should be an IP address or domain name which resolves to the subnet mask of the subnet being described. The subnet number, together with the netmask, are sufficient to determine whether any given IP address is on the specified subnet.
2. range 192.168.0.100 192.168.0.200; - For any subnet on which addresses will be assigned dynamically, there must be at least one range statement. The range statement gives the lowest and highest IP addresses in a range. All IP addresses in the range should be in the subnet in which the range statement is declared. 192.168.1.100 is the starting IP address and 192.168.1.200 is the ending IP address in this pool.
3. option subnet-mask 255.255.255.0; - Use this subnet-mask.
4. option broadcast-address 192.168.1.255; - Use this broadcast address.
5. option routers 192.168.1.254; - Use this gateway address i.e. the address of your router connected to the Internet.

Save and close the file. To [check the syntax of dhcpd.conf file for errors](http://www.cyberciti.biz/tips/check-unix-linux-configuration-file-for-syntax-errors.html), run:  
# dhcpd -t  
OR  
# dhcpd -t /etc/dhcp/dhcpd.conf

### How do I start / stop / restart the DHCP server?

Type the following commands:

service isc-dhcp-server start

service isc-dhcp-server stop

service isc-dhcp-server restart

service isc-dhcp-server status

## How do I verify that DHCP server UDP port # 67 is opened by dhcpd?

Type any one of the following command  
# netstat -tulpn | grep --color "dhcp"  
OR  
# ps aux | grep --color "[d]hcpd"  
OR  
# pgrep dhcpd  
Sample outputs:

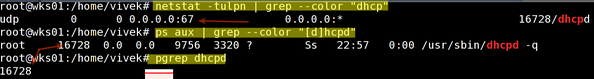


Fig.02: Verify That The DHCPD Server Is Running or Not

## Troubleshooting the DHCP server problem

By default the dhcpd will log all output using the syslog function with the log facility set to LOG\_DAEMON i.e. /var/log/syslog file:  
# tail -f /var/log/syslog  
# grep dhcpd /var/log/syslog  
You can [dump DHCP packets under Linux / UNIX for monitoring or debugging purpose using dhcpdump](http://www.cyberciti.biz/faq/linux-unix-dhcpdump-monitor-dhcp-traffic/) command as follows:  
# dhcpdump -i eth0  
OR use old good the tcpdump program:  
# tcpdump -lenx -i eth0 -s 1500 port bootps or port bootpc  
cd to /var/lib/dhcp directory to see more information about leases that the dhcp server has assigned to clients:  
# cd /var/lib/dhcp/  
# ls -l  
# vi dhcpd.leases  
# cat dhcpd.leases  
# grep 'something' dhcpd.leases

## Securing the DHCP server

Disable the dynamic DNS:

ddns-update-style none*;*

Set Deny decline messages to avoid DoS attack againest your dhcp server. The client device can send DHCPDECLINE message many times that can exhaust the DHCP server’s pool of IP addresses, causing the DHCP server to forget old address allocations:

deny declines*;*

Disable support older BOOTP clients:

deny bootp*;*

You must set valid and correct values for all the following operational directives. If you are not using NIS domain or ntp server, make sure the following options are not defined.

## see dhcpd.conf man page for more info on the directives ##

option domain-name

option domain-name-servers

option nis-domain

option nis-servers

option ntp-servers

option routers

option time-offset

In most cases you only need domain-name, domain-name-servers, and routers directives and rest should be removed to minimize information served by the dhcp server.

### How do I configure iptables to allow access to the DHCP server?

Edit your iptables scripts and add the following lines

*## Make sure you use an appropriate network block, ##*

*## and network mask, representing the machines on your ##*

*## network* ***which*** *should operate* ***as*** *clients of the dhcp serve. ##*

*## Syntax: ##*

*## /sbin/iptables -A INPUT -s net/mask -i $LAN\_IFACE -p udp --dport 67:68 --sport 67:68 -j ACCEPT ##*

*## Adjust rules* ***as*** *per your setup ##*

/sbin/iptables -A INPUT -s 192.168.1.0/24 -i eth0 -p tcp --sport 68 --dport 67 -j ACCEPT

/sbin/iptables -A INPUT -s 192.168.1.0/24 -i eth0 -p udp --sport 68 --dport 67 -j ACCEPT

## A slightly different configuration for an internal subnet

The following is a special subnet that allows to pxe network booting using tftpd server at 192.168.0.5 (please note that you need to install and configure tftpd server separately):

subnet 192.168.0.0 netmask 255.255.255.0 {

range 192.168.0.20 192.168.0.50*;*

## openbsd pxe boot file ##

filename "openbsd/pxeboot"*;*

## Debian 6 pxe boot file ##

## filename "debian6/pxelinux.0"*;*

## Freebsd pxe boot file ##

## filename "freebsd/pxeboot"*;*

## our boot server ##

next-server 192.168.0.5*;*

option subnet-mask 255.255.255.0*;*

option broadcast-address 192.168.0.255*;*

option routers 192.168.0.5*;*

}

## How do I add BOOTP support?

Each BOOTP client must be explicitly declared in the dhcpd.conf file.

## bootp my headless home router ##

host router {

hardware ethernet 08:00:2b:4c:59:23*;*

fixed-address 192.168.0.21*;*

filename "debian6/pxelinux.0"*;*

}

Exp 10 : To install and configure web server

$ sudo apt-get install apache2

Apache by default configures itself quickly so that you can open up it from the browser with the localhost url

http://localhost/

### Start/Stop apache

To start or stop apache web server from command line, use the following commands.

# Start web server

$ sudo service apache2 start

# Stop the web server

$ sudo service apache2 stop

# Restart the web server

$ sudo service apache2 restart

# Reload configuration without restarting

$ sudo service apache2 reload

The default web root directory is /var/www. So whatever files are put in this directory are accessible from the localhost url. Later we shall check how to change the default web root directory

To check what version of apache is installed use the apache2 command with the v option/

$ apache2 -v

Server version: Apache/2.4.6 (Ubuntu)

Server built: Aug 9 2013 14:28:56

#### Locate configuration files

To get more information about how exactly apache is configure on your system, use the apache2ctl command.

$ apache2ctl -V

AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message

Server version: Apache/2.4.6 (Ubuntu)

Server built: Aug 9 2013 14:28:56

Server's Module Magic Number: 20120211:23

Server loaded: APR 1.4.8, APR-UTIL 1.5.2

Compiled using: APR 1.4.8, APR-UTIL 1.5.2

Architecture: 64-bit

Server MPM: prefork

threaded: no

forked: yes (variable process count)

Server compiled with....

-D APR\_HAS\_SENDFILE

-D APR\_HAS\_MMAP

-D APR\_HAVE\_IPV6 (IPv4-mapped addresses enabled)

-D APR\_USE\_SYSVSEM\_SERIALIZE

-D APR\_USE\_PTHREAD\_SERIALIZE

-D SINGLE\_LISTEN\_UNSERIALIZED\_ACCEPT

-D APR\_HAS\_OTHER\_CHILD

-D AP\_HAVE\_RELIABLE\_PIPED\_LOGS

-D DYNAMIC\_MODULE\_LIMIT=256

-D HTTPD\_ROOT="/etc/apache2"

-D SUEXEC\_BIN="/usr/lib/apache2/suexec"

-D DEFAULT\_PIDLOG="/var/run/apache2.pid"

-D DEFAULT\_SCOREBOARD="logs/apache\_runtime\_status"

-D DEFAULT\_ERRORLOG="logs/error\_log"

-D AP\_TYPES\_CONFIG\_FILE="mime.types"

-D SERVER\_CONFIG\_FILE="apache2.conf"

.....