Linux Networking Guide for Beginners

Table of Contents

---

1. Basic Networking Concepts

What You Need to Know

Network: Computers connected together to share information.

IP Address: A unique number that identifies your computer on a network.

- Example: `192.168.1.10`
- Like a home address for your computer

Port: A specific "door" on your computer for different services.

- Example: Port 80 for websites, Port 22 for SSH
- Think of IP as the building address, port as the apartment number

Gateway/Router: The device that connects your network to the internet.

- Usually has an IP like `192.168.1.1`

DNS: Translates website names (google.com) to IP addresses (142.250.185.46).

Interface: Your network connection (wired or wireless).

- Common names: `eth0`, `wlan0`, `enp3s0`

---

2. Network Interfaces

Viewing Your Network Interfaces

See all network connections:

```
# Modern command - shows all interfaces
ip link show

# See IP addresses too
ip addr show

# Shorter version
ip a
```

What you'll see:

- `lo`: Loopback (your computer talking to itself) - always at 127.0.0.1
- `eth0` or `enp3s0`: Wired connection
- `wlan0` or `wlp2s0`: Wireless connection

Check if interface is working:

```
# Look for "state UP" - means it's active
ip link show eth0

# Brief status of all interfaces
ip -br link show
```

---

3. IP Addresses and Configuration

Understanding IP Addresses

Private IP ranges (used in home/office networks):

- `192.168.0.0` to `192.168.255.255` (most common)

- `10.0.0.0` to `10.255.255.255`
- `172.16.0.0` to `172.31.255.255`

Subnet notation:

- `/24` means 256 addresses (like 192.168.1.0 to 192.168.1.255)
- `/16` means 65,536 addresses
- Most home networks use `/24`

Viewing Your IP Configuration

```
# See your IP address
ip addr show

# Just show the IP (cleaner)
ip addr show | grep "inet "

# See your gateway (router)
ip route show

# Get just the gateway IP
ip route | grep default
```

Example output:

```
inet 192.168.1.100/24    # Your IP address
default via 192.168.1.1  # Your gateway
```

---

4. Assigning IP Addresses

Two Ways to Get an IP Address

1. **DHCP (Automatic)** - Router assigns IP automatically
2. **Static (Manual)** - You set a fixed IP address

Method 1: Using DHCP (Automatic)

For most systems using NetworkManager:

```
# Get automatic IP from router
sudo nmcli device connect eth0

# Renew DHCP lease
sudo nmcli connection up "Wired connection 1"

# Check if DHCP worked
ip addr show eth0
```

Using older dhclient command:

```
# Request IP from DHCP server
sudo dhclient eth0

# Release current IP
sudo dhclient -r eth0

# Renew IP
sudo dhclient eth0
```

Method 2: Setting a Static IP Address

Temporary static IP (resets on reboot):

```
# Step 1: Remove current IP
sudo ip addr flush dev eth0

# Step 2: Add your chosen IP
sudo ip addr add 192.168.1.100/24 dev eth0

# Step 3: Bring interface up
sudo ip link set eth0 up

# Step 4: Add default gateway
sudo ip route add default via 192.168.1.1

# Step 5: Verify
ip addr show eth0
```

Permanent static IP using NetworkManager:

```
# Step 1: Show existing connections
nmcli connection show

# Step 2: Modify connection (replace "Wired connection 1" with your connection
name)
sudo nmcli connection modify "Wired connection 1" \
    ipv4.method manual \
    ipv4.addresses 192.168.1.100/24 \
    ipv4.gateway 192.168.1.1 \
    ipv4.dns "8.8.8.8 8.8.4.4"

# Step 3: Restart connection
sudo nmcli connection down "Wired connection 1"
sudo nmcli connection up "Wired connection 1"

# Step 4: Verify
ip addr show
```

Permanent static IP by editing config file:

```
# For Ubuntu/Debian - edit netplan
sudo nano /etc/netplan/01-netcfg.yaml
```

Add this content (adjust to your needs):

```
network:
  version: 2
  ethernets:
    eth0:
      addresses:
        - 192.168.1.100/24
      gateway4: 192.168.1.1
      nameservers:
        addresses: [8.8.8.8, 8.8.4.4]
```

```
# Apply changes
sudo netplan apply

# Verify
ip addr show eth0
```

Switching Back to DHCP

```
# Using NetworkManager
sudo nmcli connection modify "Wired connection 1" ipv4.method auto
sudo nmcli connection down "Wired connection 1"
sudo nmcli connection up "Wired connection 1"
```

Quick IP Assignment Reference

```
# See current IP
ip addr show

# Get automatic IP
sudo dhclient eth0

# Set static IP (temporary)
sudo ip addr add 192.168.1.100/24 dev eth0

# Remove IP
sudo ip addr del 192.168.1.100/24 dev eth0

# Add gateway
sudo ip route add default via 192.168.1.1

# Remove gateway
sudo ip route del default
```

---

5. Understanding and Managing Ports

What Are Ports?

Ports are numbered endpoints (0-65535) that let different services run on the same computer.

Common ports you should know:

- **22**: SSH (remote login)
- **80**: HTTP (websites)
- **443**: HTTPS (secure websites)
- **21**: FTP (file transfer)
- **25**: SMTP (email sending)
- **3306**: MySQL database

- **5432**: PostgreSQL database
- **8080**: Alternative web port

Viewing Open Ports

See what ports are listening (accepting connections):

```
# Modern command - shows all listening ports
sudo ss -tulpn

# Explanation:
# -t: TCP ports
# -u: UDP ports
# -l: listening ports only
# -p: show program name
# -n: show numbers, not names

# Just TCP listening ports
sudo ss -tlpn

# Just UDP listening ports
sudo ss -ulpn
```

Using older netstat command:

```
# Show listening ports with programs
sudo netstat -tulpn

# Show all connections
sudo netstat -an
```

Example output:

```
Proto  Local Address    Foreign Address   State    PID/Program
tcp    0.0.0.0:22       0.0.0.0:*         LISTEN   1234/sshd
tcp    127.0.0.1:3306   0.0.0.0:*         LISTEN   5678/mysqld
```

This shows:

- SSH running on port 22, accessible from anywhere (0.0.0.0)
- MySQL on port 3306, only accessible locally (127.0.0.1)

Testing if a Port is Open

Test connection to a port:

```
# Using telnet
telnet 192.168.1.100 80

# Using netcat (nc)
nc -zv 192.168.1.100 80

# Test multiple ports
nc -zv 192.168.1.100 20-25

# Test port with timeout
nc -zv -w 5 192.168.1.100 22
```

Test from outside (check if port is reachable):

```
# Test if SSH port is open on remote server
nc -zv example.com 22

# Test web server
nc -zv google.com 80
```

Opening/Closing Ports with Firewall

Using UFW (Ubuntu Firewall - easiest for beginners):

```
# Check firewall status
sudo ufw status

# Allow incoming connections on port 80
sudo ufw allow 80/tcp

# Allow SSH (port 22)
sudo ufw allow ssh

# Allow port range
sudo ufw allow 8000:8100/tcp

# Allow from specific IP
sudo ufw allow from 192.168.1.50 to any port 22

# Block a port
sudo ufw deny 23/tcp

# Delete a rule
sudo ufw delete allow 80/tcp

# Enable firewall
sudo ufw enable

# Disable firewall
sudo ufw disable
```

Using firewalld (CentOS/RHEL/Fedora):

```
# Check status
sudo firewall-cmd --state

# Allow port
sudo firewall-cmd --permanent --add-port=80/tcp

# Allow service
sudo firewall-cmd --permanent --add-service=http

# Remove port
sudo firewall-cmd --permanent --remove-port=80/tcp

# Reload firewall
sudo firewall-cmd --reload

# List allowed ports
sudo firewall-cmd --list-ports
```

Finding What's Using a Port

```
# Find what program is using port 80
sudo lsof -i :80

# Alternative with ss
sudo ss -tlpn | grep :80

# Find all ports used by a program
sudo lsof -i -P | grep ssh

# Kill a program using a port (use PID from above)
sudo kill 1234
```

Port Usage Examples

Check if web server is running:

```
sudo ss -tlpn | grep :80
```

Check if SSH is accessible:

```
nc -zv localhost 22
```

See all established connections:

```
ss -tn
```

Count connections on port 80:

```
sudo ss -tn | grep :80 | wc -l
```

---

6. Testing Network Connectivity

Basic Connectivity Tests

Ping - test if host is reachable:

```
# Ping Google's DNS (tests internet)
ping -c 4 8.8.8.8

# Ping website (tests DNS too)
ping -c 4 google.com

# Ping gateway (tests local network)
ping -c 4 192.168.1.1

# Continuous ping (stop with Ctrl+C)
ping google.com
```

What ping results mean:

- Response with time = Working
- "Request timeout" = Not reachable or blocking pings
- "Unknown host" = DNS problem
- "Network unreachable" = No internet connection

Traceroute - see path to destination:

```
# Show route to Google
traceroute google.com

# Faster (skip DNS lookups)
traceroute -n 8.8.8.8

# Limit to 15 hops
traceroute -m 15 google.com
```

Step-by-Step Network Diagnosis

```
# 1. Check if interface is up
ip link show

# 2. Check if you have an IP
ip addr show

# 3. Ping your gateway
ping -c 3 192.168.1.1

# 4. Ping external IP
ping -c 3 8.8.8.8

# 5. Test DNS
ping -c 3 google.com
```

If step fails, that's where your problem is!

---

7. DNS Basics

What is DNS?

DNS translates names to IP addresses:

- You type: google.com
- DNS says: 142.250.185.46
- Your computer connects to that IP

View DNS Settings

```
# See DNS servers
cat /etc/resolv.conf

# See DNS for interface
nmcli device show eth0 | grep DNS
```

DNS Lookup Commands

```
# Basic lookup
nslookup google.com

# Better tool with more info
dig google.com

# Quick answer only
dig google.com +short

# Reverse lookup (IP to name)
dig -x 8.8.8.8

# Use specific DNS server
dig @8.8.8.8 google.com
```

Common DNS Servers

- Google: `8.8.8.8` and `8.8.4.4`
- Cloudflare: `1.1.1.1` and `1.0.0.1`
- Quad9: `9.9.9.9`

Temporarily Change DNS

```
# Add Google DNS (temporary - resets on reboot)
echo "nameserver 8.8.8.8" | sudo tee /etc/resolv.conf

# Using NetworkManager (permanent)
sudo nmcli connection modify "Wired connection 1" ipv4.dns "8.8.8.8 8.8.4.4"
sudo nmcli connection down "Wired connection 1"
sudo nmcli connection up "Wired connection 1"
```

---

8. Practical Examples

Example 1: Complete Network Setup

```
# Configure static IP on eth0

# Step 1: Set IP address
sudo ip addr add 192.168.1.100/24 dev eth0

# Step 2: Bring interface up
sudo ip link set eth0 up

# Step 3: Add gateway
sudo ip route add default via 192.168.1.1

# Step 4: Add DNS
echo "nameserver 8.8.8.8" | sudo tee /etc/resolv.conf

# Step 5: Test
ping -c 3 google.com
```

Example 2: Quick Network Check

```
# Create a simple network report
echo "=== Network Status ===" > network_check.txt
echo "IP Address:" >> network_check.txt
ip -br addr show >> network_check.txt
echo "" >> network_check.txt
echo "Gateway:" >> network_check.txt
ip route | grep default >> network_check.txt
echo "" >> network_check.txt
echo "DNS:" >> network_check.txt
cat /etc/resolv.conf | grep nameserver >> network_check.txt

# View report
cat network_check.txt
```

Example 3: Test Multiple Servers

```
# Test connectivity to common servers
echo "Testing Google DNS..."
ping -c 2 8.8.8.8

echo "Testing Cloudflare DNS..."
ping -c 2 1.1.1.1

echo "Testing Google.com..."
ping -c 2 google.com
```

Example 4: Check Open Ports

```
# See what services are running
echo "=== Listening Services ===" > services.txt
sudo ss -tlpn >> services.txt
cat services.txt
```

---

9. Common Problems

Problem 1: No Internet Connection

Steps to fix:

```
# 1. Check if interface is up
ip link show

# 2. Check for IP address
ip addr show

# 3. If no IP, get one from DHCP
sudo dhclient eth0

# 4. Test gateway
ping -c 3 192.168.1.1

# 5. Test internet
ping -c 3 8.8.8.8
```

Problem 2: DNS Not Working

Symptoms: Can ping 8.8.8.8 but not google.com

Fix:

```
# Add Google DNS
echo "nameserver 8.8.8.8" | sudo tee /etc/resolv.conf

# Test
nslookup google.com
```

Problem 3: Can't Access a Port

Check if port is open:

```
# See if anything is listening on port 80
sudo ss -tlpn | grep :80

# If nothing, the service isn't running
# Start the service (example for web server)
sudo systemctl start apache2

# Allow through firewall
sudo ufw allow 80/tcp
```

Problem 4: Slow Connection

```
# Test ping to gateway
ping -c 10 192.168.1.1

# Check for errors on interface
ip -s link show eth0

# Look for "errors" or "dropped" packets
```

Quick Command Reference

Most Important Commands

```
# View network interfaces
ip link show
ip addr show

# Assign IP (temporary)
sudo ip addr add 192.168.1.100/24 dev eth0
sudo ip route add default via 192.168.1.1

# Get automatic IP
sudo dhclient eth0

# Test connectivity
ping -c 4 8.8.8.8
ping -c 4 google.com

# View open ports
sudo ss -tulpn

# Test if port is open
nc -zv 192.168.1.100 80

# DNS lookup
dig google.com +short
nslookup google.com

# View gateway
ip route show

# View DNS servers
cat /etc/resolv.conf
```

Emergency Network Reset

```
# Restart networking
sudo systemctl restart NetworkManager

# Or restart interface
sudo ip link set eth0 down
sudo ip link set eth0 up

# Get new IP
sudo dhclient eth0
```

Tips for Beginners

1. **Start simple**: Always ping your gateway first (192.168.1.1)
2. **Use DHCP**: Let the router assign IPs automatically until you need static IPs
3. **Test step by step**: Gateway → Internet IP → Website name
4. **Write down your settings**: Save IP, gateway, and DNS when things work
5. **One change at a time**: Don't change multiple settings at once
6. **Restart carefully**: Some changes need service/interface restart
7. **Use Google DNS**: 8.8.8.8 is reliable for testing DNS issues

---

Common Port Numbers to Remember

Port	Service	Purpose
22	SSH	Remote terminal access
80	HTTP	Websites
443	HTTPS	Secure websites
21	FTP	File transfer
25	SMTP	Email sending
3306	MySQL	Database
5432	PostgreSQL	Database
8080	HTTP-alt	Alternative web port

Practice Exercises

Try these to build confidence:

1. Find your current IP address and gateway
2. Ping your gateway, then ping 8.8.8.8
3. Look up the IP address of google.com
4. See what ports are open on your system
5. Test if you can connect to port 80 on google.com

6. Create a network status report

Remember: You can't break anything permanently with these commands!