

🚀 Assignment 8 — Secure GUI Access via SSH (X11) or VNC

Secure Shell Protocol



Experiment: Basic Linux Data Networking Commands

Aim

To study and execute basic data networking commands in Linux using the command line interface.

Objectives

- To understand Linux network configuration.
 - To test network connectivity.
 - To diagnose network-related issues.
 - To access and transfer data between systems using networking tools.
-

Requirements

- Linux Operating System (Ubuntu/Debian/Fedora/Kali etc.)
 - Terminal access
 - Basic knowledge of Linux commands
 - Internet connection (optional)
-

Theory

Data networking in Linux is performed using built-in terminal commands. These commands help configure systems, test connections, and troubleshoot network problems.

Command	Purpose
<code>ifconfig / ip addr</code>	Shows network interface configuration
<code>ping</code>	Tests connectivity to another host
<code>hostname</code>	Displays system hostname
<code>traceroute</code>	Shows the route packets take
<code>netstat</code>	Displays active connections
<code>nslookup</code>	Queries DNS information
<code>ssh</code>	Secure remote login
<code>scp</code>	Secure file transfer

Procedure

Case 1: Friend's ubuntu and Ubuntu on the same network

Step 1: Enable SSH on Ubuntu

On Ubuntu, install and enable SSH:

```
sudo apt update
sudo apt install openssh-server
sudo systemctl enable ssh
sudo systemctl start ssh
```

```
ujjwaltyagi@ujjwaltyagi:~$ sudo useradd -m frienduser || true
[sudo] password for ujjwaltyagi:
useradd: user 'frienduser' already exists
ujjwaltyagi@ujjwaltyagi:~$ sudo systemctl enable --now ssh
Synchronizing state of ssh.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable ssh
```

Check it's running:

```
sudo systemctl status ssh
```

Step 2: Find Ubuntu's local IP address

Run:

```
hostname -I
```

```
ujjwaltyagi@ujjwaltyagi:~$ hostname -I  
10.0.2.15 fd17:625c:f037:2:8b91:9627:6829:959b fd17:625c:f037:2:a00:27ff:fe34:53  
ac
```

You'll get something like 192.168.1.42.

Step 3: Connect from friends ubuntu

On your friends ubuntu, open Terminal and run:

```
ssh username@192.168.1.42
```

Replace username with your Ubuntu username.

Enter your password when prompted — you're in!

```
ujjwaltyagi@ujjwaltyagi:~$ ssh vaibahvg@10.0.2.15  
Enter passphrase for key '/home/ujjwaltyagi/.ssh/id_ed25519':  
vaibahvg@10.0.2.15's password:  
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-29-generic x86_64)
```

```
* Documentation: https://help.ubuntu.com  
* Management: https://landscape.canonical.com  
* Support: https://ubuntu.com/pro
```

Expanded Security Maintenance for Applications is enabled.

```
132 updates can be applied immediately.  
76 of these updates are standard security updates.  
To see these additional updates run: apt list --upgradable
```

```
Last login: Mon Nov  3 17:08:14 2025 from 10.0.2.15  
vaibahvg@ujjwaltyagi:~$ whoami  
vaibahvg
```

Step 4: Create a File on Remote Computer

```
echo "This file was created remotely using SSH" > ~/ssh_test.txt
```

Step 5: Verify the File**

```
cat ~/ssh_test.txt
```

Step 6: Delete the File

```
rm ~/ssh_test.txt
```

Step 7: Exit SSH

```
exit
```

```
vaibahvg@ujjwaltyagi:~$ cat ~/.ssh/authorized_keys
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAIIF2VoISW9box0H9etpllHkPwkz0KyysCxt9f04bbXDu
ujjwaltyagi@ujjwaltyagi
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCrj9pWlhK52jqKxYIwGZTR202UUPI0DzUXvrb3Ua
hTBzy7oBbj2w0e7dTqaFojwf0s4xR39o3hy8mIUvU8S4cZs26R8iGwz/c0dDFYzhSYJ0W8WofsdXfnwf
4N6PDcYgKFHHkHu/kI1e1k54xkjYGK+5z0jaJS5K+YVCn6uhdFTLB+yC5JaFG0v8B8fltuuvkseepE5/
Tevg1D+MjzD9Z+XLxUEgP+N/MGwdcsHohmz3ezuKtTVIc0Kwc2N9ykwT0z6xsnXE1m9CY/hfhtgBz2gV
oujFsGgZ7BBG7Tk0IAp4QLBFXAeVq62U9/WWEldku6nV2qhyH5sy1UZ14k5B vaibahvg@ujjwaltyagi
i
vaibahvg@ujjwaltyagi:~$ exit
logout
Connection to 10.0.2.15 closed.
ujjwaltyagi@ujjwaltyagi:~$
```

Case 2: Ubuntu and Ubuntu on different networks (e.g., home ↔ office or remote server)

◊ Option 1: Connect to a remote Ubuntu server (e.g., cloud VM)

Step 1: View IP Address and Network Interfaces

```
ip addr show
```

```
ujjwaltyagi@ujjwaltyagi:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host noprefixroute
            valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:34:53:ac brd ff:ff:ff:ff:ff:ff
        inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
            valid_lft 74084sec preferred_lft 74084sec
        inet6 fd17:625c:f037:2:8b91:9627:6829:959b/64 scope global temporary dynamic
            valid_lft 86232sec preferred_lft 14232sec
        inet6 fd17:625c:f037:2:a00:27ff:fe34:53ac/64 scope global dynamic mngtmpaddr
            valid_lft 86232sec preferred_lft 14232sec
        inet6 fe80::a00:27ff:fe34:53ac/64 scope link
            valid_lft forever preferred_lft forever
ujjwaltyagi@ujjwaltyagi:~$
```

Step 2:Display hostname

```
hostname
```

```
ujjwaltyagi@ujjwaltyagi:~$ hostname
ujjwaltyagi
```

step 3:Test Network connectivity(ping)

```
ping google.com -c 4
```

```
ujjwaltyagi@ujjwaltyagi:~$ ping google.com -c 4
PING google.com (142.250.76.78) 56(84) bytes of data.
64 bytes from tzdela-aq-in-f14.1e100.net (142.250.76.78): icmp_seq=1 ttl=255 time=55.6 ms
64 bytes from tzdela-aq-in-f14.1e100.net (142.250.76.78): icmp_seq=2 ttl=255 time=20.2 ms
64 bytes from tzdela-aq-in-f14.1e100.net (142.250.76.78): icmp_seq=3 ttl=255 time=18.4 ms
64 bytes from tzdela-aq-in-f14.1e100.net (142.250.76.78): icmp_seq=4 ttl=255 time=29.2 ms

--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3510ms
rtt min/avg/max/mdev = 18.406/30.845/55.581/14.850 ms
```

Step 4: Trace Route to Remote Host

```
traceroute google.com
```

```
ujjwaltyagi@ujjwaltyagi:~$ traceroute google.com
traceroute to google.com (142.250.76.78), 30 hops max, 60 byte packets
 1 _gateway (10.0.2.2)  8.226 ms  1.374 ms  1.060 ms

 2 * * *
 3 * * *
 4 * * *
 5 * * *
 6 * * *
 7 * * *
 8 * * *
 9 * * *
10 * * *
11 * * *
12 * * *
13 * * *
14 * * *
15 * * *
16 * * *
17 * * *
18 * * *
19 * * *
20 * * *
21 * * *
22 * * *
```

Step 5: View Active Network Ports

```
netstat -tulnp
```

```
ujjwaltyagi@ujjwaltyagi:~$ netstat -tulnp
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State      PID/Program name
tcp      0      0 127.0.0.54:53           0.0.0.0:*
                                         LISTEN
tcp      0      0 0.0.0.0:22            0.0.0.0:*
                                         LISTEN
tcp      0      0 127.0.0.53:53           0.0.0.0:*
                                         LISTEN
tcp      0      0 127.0.0.1:631          0.0.0.0:*
                                         LISTEN
tcp6     0      0 :::22                  :::*
                                         LISTEN
tcp6     0      0 ::1:631                :::*
                                         LISTEN
udp      0      0 127.0.0.54:53           0.0.0.0:*
                                         LISTEN
udp      0      0 127.0.0.53:53           0.0.0.0:*
                                         LISTEN
udp      0      0 0.0.0.0:58272          0.0.0.0:*
                                         LISTEN
udp      0      0 0.0.0.0:5353           0.0.0.0:*
                                         LISTEN
udp6     0      0 :::39957               :::*
                                         LISTEN
udp6     0      0 ::::5353              :::*
                                         LISTEN
```

Step 6: DNS Lookup

```
nslookup google.com
```

```
ujjwaltyagi@ujjwaltyagi:~$ nslookup google.com
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 142.250.76.78
Name:   google.com
Address: 2404:6800:4002:830::200e
```

Step 7: Remote Login using SSH

```
ssh user@192.168.1.10
```

```
ujjwaltyagi@ujjwaltyagi:~$ ssh vaibahvg@10.0.2.15
Enter passphrase for key '/home/ujjwaltyagi/.ssh/id_ed25519':
vaibahvg@10.0.2.15's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-29-generic x86_64)
```

```
* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/pro
```

Expanded Security Maintenance for Applications is enabled.

132 updates can be applied immediately.
 76 of these updates are standard security updates.
 To see these additional updates run: apt list --upgradable

```
Last login: Mon Nov  3 17:23:09 2025 from 10.0.2.15
```

Step 8: File Transfer using SCP

```
scp test.txt user@192.168.1.10:/home/user/
```

```
ujjwaltyagi@ujjwaltyagi:~$ echo "Hello from ujjwal" >test2.txt
ujjwaltyagi@ujjwaltyagi:~$ scp test2.txt vaibahvg@10.0.2.15:/home/vaibahvg/
sign_and_send_pubkey: signing failed for ED25519 "/home/ujjwaltyagi/.ssh/id_ed25519" from agent: agent refused operation
vaibahvg@10.0.2.15's password:
test2.txt                                100%   18      3.0KB/s   00:00
ujjwaltyagi@ujjwaltyagi:~$
```



1) Prepare & secure the remote machine (run on friend's laptop)

Run these on the friend's laptop (they must have sudo):

install SSH and optional VNC

```
sudo apt update  
sudo apt install -y openssh-server tigervnc-standalone-server
```

```
ujjwaltyagi@ujjwaltyagi:~$ sudo apt install tightvncserver xfce4 xfce4-goodies -y  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
tightvncserver is already the newest version (1:1.3.10-8).  
The following package was automatically installed and is no longer required:  
 libl1vm19  
Use 'sudo apt autoremove' to remove it.  
The following additional packages will be installed:  
 7zip accountsservice-ubuntu-schemas avahi-utils bamfdaemon bzip2 desktop-base elementary-xfce-icon-theme exo-utils  
 fonts-quicksand gir1.2-dbusmenu-glib-0.4 gnome-bluetooth gnome-screensaver greybird-gtk-theme indicator-applet  
 indicator-application indicator-appmenu indicator-bluetooth indicator-common indicator-datetime indicator-keyboard  
 indicator-messages indicator-power indicator-printers indicator-session indicator-sound jayatana libaccounts-glib0  
 libasound2-plugins libatkmm-1.6-1v5 libbamf3-2t64 libburn4t64 libcairomm-1.0-1v5 libexo-2-0 libexo-common  
 libfcitx-config4 libfcitx-gclient1 libfcitx-utils0 libgarcon-1-0 libgarcon-common libgarcon-gtk3-1-0  
 libglibmm-2.4-1t64 libgnome-panel3 libgtk-layer-shell0 libgtkmm-3.0-1t64 libgtksourceview-4-0  
 libgtksourceview-4-common libido3-0.1-0 libindicator3-7 libisofs6t64 libjack-jackd2-0 libkeybinder-3.0-0  
 liblightdm-gobject-1-0 libmessaging-menu0 libmousepad0 libpangomm-1.4-1v5 libqrencode4 libsigc++-2.0-0v5 libtagc0  
 libthunarx-3-0 libtumbler-1-0t64 libunity-gtk2-parser0 libunity-gtk3-parser0 libunity-settings-daemon1 libutempter0  
 libxfce4panel-2.0-4 libxfce4ui-2-0 libxfce4ui-common libxfce4ui-utils libxfce4util-bin libxfce4util-common  
 libxfce4util7 libxfconf-0-3 libxnvctrl0 libxpresent1 light-locker light-locker-settings lightdm lm-sensors mousepad  
 p7zip-full pavucontrol policykit-1-gnome pulseaudio-utils python3-psutil ristretto system-config-printer  
 tango-icon-theme thunar thunar-archive-plugin thunar-data thunar-media-tags-plugin thunar-volman tumbler  
 tumbler-common ubuntu-touch-sounds unity-greeter unity-gtk-module-common unity-gtk2-module unity-gtk3-module  
 unity-settings-daemon unity-settings-schemas xarchiver xfburn xfce4-appfinder xfce4-battery-plugin
```

create non-root user if needed (won't fail if exists)

```
sudo useradd -m frienduser || true
```

```
ujjwaltyagi@ujjwaltyagi:~$ sudo adduser vaibahvg
info: Adding user `vaibahvg' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `vaibahvg' (1002) ...
info: Adding new user `vaibahvg' (1002) with group `vaibahvg (1002)' ...
info: Creating home directory `/home/vaibahvg' ...
info: Copying files from `/etc/skel' ...
New password:
BAD PASSWORD: The password fails the dictionary check - it is too simplistic/systematic
Retype new password:
passwd: password updated successfully
Changing the user information for vaibahvg
Enter the new value, or press ENTER for the default
      Full Name []: vaibhavg
      Room Number []: 34
      Work Phone []:
      Home Phone []:
      Other []:
Is the information correct? [Y/n] y
info: Adding new user `vaibahvg' to supplemental / extra groups `users' ...
info: Adding user `vaibahvg' to group `users' ...
info: Processing group `users': reading list of users ...
info: User `vaibahvg' added to group `users'
```

enable and start SSH

```
sudo systemctl enable --now ssh
```

```
ujjwaltyagi@ujjwaltyagi:~$ sudo systemctl status ssh
[sudo] password for ujjwaltyagi:
● ssh.service - OpenBSD Secure Shell server
    Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: enabled)
    Active: inactive (dead)
  TriggeredBy: ● ssh.socket
    Docs: man:sshd(8)
           man:sshd_config(5)
ujjwaltyagi@ujjwaltyagi:~$
```

add your public key (replace) and set safe permissions

```
sudo -u frienduser mkdir -p /home/frienduser/.ssh
echo "<your-pubkey>" | sudo tee -a /home/frienduser/.ssh/authorized_keys
sudo chmod 700 /home/frienduser/.ssh
sudo chmod 600 /home/frienduser/.ssh/authorized_keys
sudo chown -R frienduser:frienduser /home/frienduser/.ssh
```

```
ujjwaltyagi@ujjwaltyagi:~$ mkdir -p ~/.ssh
ujjwaltyagi@ujjwaltyagi:~$ chmod 700 ~/.ssh
ujjwaltyagi@ujjwaltyagi:~$ touch ~/.ssh/authorized_keys
ujjwaltyagi@ujjwaltyagi:~$ chmod 600 ~/.ssh/authorized_keys
chmod: cannot access '/home/ujjwaltyagi/.ssh/authorized_keys': No such file or directory
ujjwaltyagi@ujjwaltyagi:~$ chmod 600 ~/.ssh/authorized_keys
```

Verify sshd is running:

```
sudo systemctl status ssh --no-pager
# or
ss -tlnp | grep :22
```

```
ujjwaltyagi@ujjwaltyagi:~$ sudo systemctl enable --now ssh
Synchronizing state of ssh.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable ssh
ujjwaltyagi@ujjwaltyagi:~$ sudo systemctl status ssh --no-pager
● ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
  Active: active (running) since Mon 2025-11-03 15:33:37 IST; 1h 29min ago
TriggeredBy: ● ssh.socket
    Docs: man:sshd(8)
          man:sshd_config(5)
  Main PID: 1161 (sshd)
    Tasks: 1 (limit: 9659)
   Memory: 4.1M (peak: 20.5M)
      CPU: 828ms
     CGroup: /system.slice/ssh.service
             └─1161 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Nov 03 16:52:44 ujjwaltyagi sshd[5669]: pam_unix(sshd:auth): authentication...2.15
Nov 03 16:52:47 ujjwaltyagi sshd[5669]: Failed password for invalid user va...ssh2
Nov 03 16:52:57 ujjwaltyagi sshd[5669]: pam_unix(sshd:auth): check pass; us...nown
Nov 03 16:53:00 ujjwaltyagi sshd[5669]: Failed password for invalid user va...ssh2
Nov 03 16:53:10 ujjwaltyagi sshd[5669]: Failed password for invalid user va...ssh2
Nov 03 16:53:10 ujjwaltyagi sshd[5669]: Connection closed by invalid user v...uth]
Nov 03 16:53:10 ujjwaltyagi sshd[5669]: PAM 1 more authentication failure; ...2.15
Nov 03 16:58:01 ujjwaltyagi sshd[5683]: Failed password for ujjwaltyagi fro...ssh2
Nov 03 16:58:05 ujjwaltyagi sshd[5683]: Accepted password for ujjwaltyagi f...ssh2
Nov 03 16:58:05 ujjwaltyagi sshd[5683]: pam_unix(sshd:session): session ope...d=0
Hint: Some lines were ellipsized, use -l to show in full.
```

2) Test X11 forwarding (single GUI app) — from your laptop

Preflight on your laptop:

Linux: usually ready.

macOS: install & run XQuartz and start it before SSH.

Windows: install an X server like VcXsrv or Xming and run it.

Then on your laptop run:

```
ssh -p 22 -X frienduser@FRIEND_IP
``
```

```
# once connected, run a simple GUI test:  
```bash  
xeyes & # or gedit & or xclock &
```

```
vaibahvg@ujjwaltyagi:~$ exit
logout
ujjwaltyagi@ujjwaltyagi:~$ echo "Hello from ujjwal" >test2.txt
ujjwaltyagi@ujjwaltyagi:~$ scp test2.txt vaibahvg@10.0.2.15:/home/vaibahvg
sign_and_send_pubkey: signing failed for ED25519 "/home/ujjwaltyagi/.ssh/vaibahvg@10.0.2.15's password:
test2.txt 100% 18 3.0KB/s 0
ujjwaltyagi@ujjwaltyagi:~$ sudo systemctl enable --now ssh
[sudo] password for ujjwaltyagi:
Synchronizing state of ssh.service with SysV service script with /usr/lib/systemd/system-sysv-install enable ssh
ujjwaltyagi@ujjwaltyagi:~$ ssh -p 22 -X vaibahvg@10.0.2.15
sign_and_send_pubkey: signing failed for ED25519 "/home/ujjwaltyagi/.ssh/vaibahvg@10.0.2.15's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-29-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro
```

Expanded Security Maintenance for Applications is enabled.

132 updates can be applied immediately.  
76 of these updates are standard security updates.  
To see these additional updates run: apt list --upgradable

Last login: Mon Nov 3 17:38:42 2025 from 10.0.2.15  
/usr/bin/xauth: file /home/vaibahvg/.Xauthority does not exist  
vaibahvg@ujjwaltyagi:~\$ xeyes

Success criteria: the GUI app window appears on your laptop and is responsive.

Troubleshooting quick checks:

If you see Warning: No xauth data; using fake or X apps fail: try -Y (trusted) instead of -X: ssh -Y -p 22 frienduser@FRIEND\_IP.

Ensure DISPLAY is set on the SSH session (run echo \$DISPLAY — should show something like localhost:10.0).

Ensure client X server (XQuartz/VcXsrv) is running and allowing connections.

Security note: X11 forwards every GUI app over SSH, but X11 is older and can let remote apps access local X resources — use only with trusted machines. For stronger isolation, prefer the VNC-over-SSH approach below.

## 3) Full desktop: VNC server on friend's laptop + SSH tunnel from your laptop

---

On friend's laptop (as frienduser):

**start a VNC server on display :1 (creates :1 -> TCP 5901)**

---

```
vncserver :1
optionally stop with: vncserver -kill :1
```

(Configure desktop environment in `~/.vnc/xstartup` if needed — many distros auto-configure.)

On your laptop: create a local SSH tunnel (keeps VNC server bound to localhost on remote; only SSH port open externally)

**create an SSH tunnel mapping your localhost:5901 to friend's localhost:5901**

---

```
ssh -L 5901:localhost:5901 -p 22 frienduser@FRIEND_IP -N &
-N = no remote command; & runs in background (adjust as preferred)
```

```
ujjwaltyagi@ujjwaltyagi:~$ ssh -L 5901:localhost:5901 -p 22 vaibahvg@10.0.2.15 -N &
[1] 9487
exit
logout
There are stopped jobs.
[1]+ Stopped ssh -L 5901:localhost:5901 -p 22 vaibahvg@10.0.2.15 -N
```

Then open your VNC viewer and connect to:

```
localhost:5901
```

```
ujjwaltyagi@ujjwaltyagi:~$ vncserver :1
New 'X' desktop is ujjwaltyagi:1
Starting applications specified in /home/ujjwaltyagi/.vnc/xstartup
Log file is /home/ujjwaltyagi/.vnc/ujjwaltyagi:1.log

ujjwaltyagi@ujjwaltyagi:~$ vncviewer

TigerVNC Viewer v1.13.1
Built on: 2024-04-01 08:26
Copyright (C) 1999-2022 TigerVNC Team and many others (see README.rst)
See https://www.tigervnc.org for information on TigerVNC.

Mon Nov 3 18:51:47 2025
DecodeManager: Detected 5 CPU core(s)
DecodeManager: Creating 4 decoder thread(s)
CConn: Connected to host localhost port 5901

Mon Nov 3 18:51:52 2025
CConn: End of stream
CConn: The connection was dropped by the server before the session could
 be established.
DecodeManager: Total: 0 rects, 0 pixels
DecodeManager: 0 B (1:-nan ratio)
```

## Output / Observations

Command	Result
ip addr	Lists network interfaces and IP addresses
ping	Replies received indicate connectivity
traceroute	Displays the route path to the destination
nslookup	Shows DNS IP information
ssh	Connects to a remote machine securely
scp	Transfers files securely over SSH

## Result

Basic Linux networking commands were successfully executed and network connectivity and configuration were verified.

## Conclusion

Linux provides powerful built-in commands for networking tasks such as configuration, troubleshooting, monitoring, and secure communication between systems.

## Viva Questions

? What is the purpose of the **ping** command?

❖ The **ping** command is used to test network connectivity between the source and a destination host. It sends ICMP Echo Request packets and waits for Echo Reply packets to verify whether the destination is reachable and to measure round-trip time.

---

? What is the difference between SSH and Telnet?

❖ | Feature | SSH (Secure Shell) | Telnet |

----- ----- -----	Security   Encrypted communication	No encryption	Default Port   22
23	Usage   Secure remote login	Unsecure remote login	Current Status   Widely used   Mostly outdated

*SSH is preferred over Telnet because it provides secure communication.*

---

? How does **traceroute** help in network troubleshooting?

❖ **traceroute** displays the path taken by packets from the local system to a remote host. It shows each intermediate router (hop) along the path and the time taken. It helps identify:

- Network delays
  - Connection failures
  - Routing issues
- 

? Why is DNS used in networking?

❖ DNS (Domain Name System) translates human-friendly domain names like [www.google.com](http://www.google.com) into machine-readable IP addresses like **142.250.182.14**. This is necessary because computers communicate using IP addresses, not domain names.

---

? How can you transfer a file securely in Linux?

❖ Files can be transferred securely in Linux using the **scp** (Secure Copy) command, which uses SSH encryption.

Example:

```
scp file.txt user@192.168.1.10:/home/user/
```