

Detailed Report on Network Cable Preparation, IP Configuration, and Network Tools

**Bachelor of Technology
Computer Science and Engineering**

Submitted By

NAME – ABHIRUP BAG
ROLL NUMBER – 13000122082
DEPARTMENT – CSE(B1)
SEMESTER – 6
PAPER – Computer Networks Lab
(PCC-CS692)

February 2025



**Techno Main
EM-4/1, Sector-V, Salt Lake
Kolkata- 700091
West Bengal
India**

Table of Contents

1. Introduction

2. CAT-5/CAT-6 Cable Preparation with RJ-45 Connector

2.1 Tools Required

2.2 Wiring Standards (TIA/EIA-568A & TIA/EIA-568B)

2.3 Straight-Through vs. Crossover Cabling

2.4 Steps to Crimp an RJ-45 Connector

3. IP Address Configuration on Windows and Linux

3.1 Static IP Configuration

3.2 DHCP Configuration

4. Network Tools and Commands

4.1 ipconfig (Windows)

4.2 ifconfig (Linux)

4.3 ip

4.4 hostname

4.5 ping

4.6 netstat

4.7 route

4.8 traceroute / tracert

4.9 tcpdump

4.10 Wireshark

5. Conclusion

6. References

Detailed Report on Networking Concepts and Tools

1. CAT-5/CAT-6 Cable Preparation with RJ-45 Connector

A. Introduction

CAT-5 and CAT-6 cables are twisted pair cables used in networking for connecting devices like computers, routers, and switches. These cables use RJ-45 connectors for termination.

B. Tools Required

- i) CAT-5/CAT-6 cable
- ii) RJ-45 connectors
- iii) Crimping tool
- iv) Cable stripper
- v) Network tester

C. Wiring Standards

There are two standards for Ethernet cabling:

- (a) TIA/EIA-568A (T568A)
- (b) TIA/EIA-568B (T568B)

D. Used for connecting different devices (PC to switch, switch to router). The wire order remains the same on both ends.

- i) **Straight-through cable:** Used to connect different types of devices (e.g., PC to switch, switch to router).
- ii) **Crossover cable:** Used to connect similar devices (e.g., PC to PC, switch to switch).

Straight-Through Wiring (T568B Standard)

Pin	Colour
1	Orange-White

2	Orange
3	Green-White
4	Blue
5	Blue-White
6	Green
7	Brown-White
8	Brown

Crossover Wiring (T568A on One End, T568B on Other)

Pin (End 1 - T568A)	Pin (End 2 - T568B)
1 (Green-White)	1 (Orange-White)
2 (Green)	2 (Orange)
3 (Orange-White)	3 (Green-White)
4 (Blue)	4 (Blue)
5 (Blue-White)	5 (Blue-White)
6 (Orange)	6 (Green)
7 (Brown-White)	7 (Brown-White)
8 (Brown)	8 (Brown)

E. Steps to Crimp an RJ-45 Connector

- i) Strip about 1 inch of the cable's outer sheath.
 - ii) Untwist and arrange the wires according to the desired standard.
 - iii) Insert the wires into the RJ-45 connector.
 - iv) Use the crimping tool to secure the connector.
 - v) Test the cable using a network tester.
-

2. IP Address Configuration (Static and DHCP) on Linux and Windows

A. Windows Configuration

i) Static IP Configuration

- (a) Open Control Panel → Network and Sharing Centre → Change adapter settings.
- (b) Right-click on the network adapter → Properties.
- (c) Select "Internet Protocol Version 4 (TCP/IPv4)" → Properties.
- (d) Select "Use the following IP address" and enter details.
- (e) Click "OK" to apply changes.

ii) DHCP Configuration

- (a) Follow steps 1-3 above.
- (b) Select "Obtain an IP address automatically".
- (c) Click "OK" to apply changes.

iii) Linux IP Configuration

(a) Static IP Configuration (Ubuntu/Debian)

- Open Terminal.

- Edit network configuration:

```
sudo nano /etc/netplan/00-installer-config.yaml
```

- Add the following:

```
network:
  ethernets:
    eth0:
      dhcp4: no
      addresses: [192.168.1.100/24]
      gateway4: 192.168.1.1
      nameservers:
        addresses: [8.8.8.8, 8.8.4.4]
  version: 2
```

- Save and exit. Restart the network service:

```
sudo netplan apply
```

(b)DHCP Configuration

- Modify /etc/netplan/00-installer-config.yaml

```
network:
  ethernets:
    eth0:
      dhcp4: true
  version: 2
```

- Apply changes:

```
sudo netplan apply
```

3. Important Network Tools and Commands

A. ipconfig (Windows)

Displays network configuration details.

```
ipconfig /all
```

Example output:

```
IPv4 Address: 192.168.1.100  
Subnet Mask: 255.255.255.0  
Default Gateway: 192.168.1.1
```

B. ifconfig (Linux)

Displays network interface information.

```
ifconfig
```

Example output:

```
abhirup ~ via C v13.3.0-gcc took 1s  
● X ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1400  
    inet 172.31.240.53 netmask 255.255.240.0 broadcast 172.31.255.255  
    inet6 fe80::215:5dff:feaa:b4c4 prefixlen 64 scopeid 0x20<link>  
    ether 00:15:5d:aa:b4:c4 txqueuelen 1000 (Ethernet)  
    RX packets 822 bytes 540053 (540.0 KB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 443 bytes 49210 (49.2 KB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 40069 bytes 53568550 (53.5 MB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 40069 bytes 53568550 (53.5 MB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

C. ip (Linux)

Displays and configures IP addresses.

```

abhirup ~ via C v13.3.0-gcc took 0s
● > ip a
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
    inet 10.255.255.254/32 brd 10.255.255.254 scope global lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1400 qdisc mq state UP group default qlen 1000
    link/ether 00:15:5d:aa:b4:c4 brd ff:ff:ff:ff:ff:ff
    inet 172.31.240.53/20 brd 172.31.255.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::215:5dff:feaa:b4c4/64 scope link
        valid_lft forever preferred_lft forever

```

D. hostname

Displays or sets the system's hostname.

```

abhirup ~ via C v13.3.0-gcc took 0s
● > hostnamectl
Static hostname: LAPTOP-KIQ8R4C3
    Icon name: computer-container
    Chassis: container □
    Machine ID: 43fa5164d3e345a094d99daf4d495ff8
    Boot ID: 40eb6247b3d44960a18bb92917b5fd26
    Virtualization: wsl
    Operating System: Ubuntu 24.04.1 LTS
    Kernel: Linux 5.15.167.4-microsoft-standard-WSL2
    Architecture: x86_64

```

E. ping

Tests network connectivity.

Example:

ping 10.10.101.135

```

abhirup ~ via C v13.3.0-gcc took 0s
● > ping 10.10.101.135
PING 10.10.101.135 (10.10.101.135) 56(84) bytes of data.

--- 10.10.101.135 ping statistics ---
10 packets transmitted, 0 received, 100% packet loss, time 9791ms

```

F. netstat

Displays network connections and statistics.


```
abhirup ~ via C v13.3.0-gcc took 0s
> 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      0 127.0.0.1:43433         0.0.0.0:*               LISTEN      22172/node
tcp      0      0 127.0.0.54:53          0.0.0.0:*               LISTEN      -
tcp      0      0 127.0.0.53:53          0.0.0.0:*               LISTEN      -
tcp      0      0 10.255.255.254:53      0.0.0.0:*               LISTEN      -
udp      0      0 127.0.0.54:53          0.0.0.0:*               -           -
udp      0      0 127.0.0.53:53          0.0.0.0:*               -           -
udp      0      0 10.255.255.254:53      0.0.0.0:*               -           -
udp      0      0 127.0.0.1:323          0.0.0.0:*               -           -
udp6     0      0 :::1:323               :::*                    -           -
```

G.route

Displays/manages routing table.

```
abhirup ~ via C v13.3.0-gcc took 0s
> route -n
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
0.0.0.0          172.31.240.1    0.0.0.0          UG    0      0      0 eth0
172.31.240.0     0.0.0.0         255.255.240.0    U     0      0      0 eth0
```

H.traceroute (Linux) / tracert (Windows)

Traces the route packets take to a destination.

```

abhirup ~ via C v13.3.0-gcc took 0s
⊗ > traceroute google.com
traceroute to google.com (142.250.194.110), 64 hops max
 1  172.31.240.1  0.538ms  0.545ms  0.365ms
 2  192.168.129.135  7.558ms  4.332ms  3.566ms
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * ^C

```

I. tcpdump

Captures and analyzes network traffic (Linux).

```

abhirup ~ via C v13.3.0-gcc took 0s
⊗ > tcpdump -i eth0
tcpdump: eth0: You don't have permission to perform this capture on that device
(socket: Operation not permitted)

```

J. Wireshark

Graphical tool for analyzing network packets.

4. Open Wireshark.

A GUI-based network packet analyzer.

- Open Wireshark.
- Select the network interface.
- Start packet capture.
- Apply filters (e.g., tcp.port == 80).

5. Conclusion

This report covered the preparation of network cables, IP address configuration, and essential networking tools. Understanding these basics is crucial for network administration and troubleshooting.

6. References

- ANSI/TIA-568.2-D (TIA)
- IEEE 802.3 (IEEE)
- RFC 791 (IETF)
- RFC 2131 (IETF)
- RFC 792 (IETF)
- Microsoft Official Documentation ([Microsoft Docs](#))
- Linux Man Pages
- Wireshark User Guide (Wireshark Docs)