Lab Practical #01: Network Commands Reference Guide

Student Name: Dhairya Adroja
Enrollment No: 24010101602
Course: B.Tech. CSE

Aim/Objective

To study and practice various network commands used in different operating systems, particularly focusing on Windows commands and their macOS equivalents, with practical output examples.

Theory

Network commands are essential tools for network administrators and users to diagnose, configure, and troubleshoot network connectivity. Different operating systems provide various commands with similar functionality but different syntax and options.

Procedure

1. IP Configuration (ipconfig → ifconfig)

Windows Command: `ipconfig`
macOS Equivalent: `ifconfig`

Usage:

```
ifconfig
```

Output:

```
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
        options=1203<RXCSUM,TXCSUM,TXSTATUS,SW_TIMESTAMP>
        inet 127.0.0.1 netmask 0xff000000
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
        nd6 options=201<PERFORMNUD,DAD>
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=6460<TSO4,TSO6,CHANNEL_IO,PARTIAL_CSUM,ZEROINVERT_CSUM>
        ether 74:0e:a4:8d:42:2b
        inet6 fe80::8c5:cf0f:3f4e:bf58%en0 prefixlen 64 secured scopeid 0xb
        inet 10.20.64.246 netmask 0xffff0000 broadcast 10.20.255.255
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect
        status: active
```

---

2. Ping Command

Windows Command: `ping google.com`
macOS Equivalent: `ping -c 4 google.com` (with count limit)

Usage:

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

Output:

```
PING google.com (142.250.192.110): 56 data bytes
64 bytes from 142.250.192.110: icmp_seq=0 ttl=116 time=29.658 ms
64 bytes from 142.250.192.110: icmp_seq=1 ttl=116 time=42.051 ms
64 bytes from 142.250.192.110: icmp_seq=2 ttl=116 time=26.359 ms
64 bytes from 142.250.192.110: icmp_seq=3 ttl=116 time=25.818 ms

--- google.com ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 25.818/30.972/42.051/6.563 ms
```

3. Get MAC Address (getmac → ifconfig | grep ether)

Windows Command: `getmac`
macOS Equivalent: `ifconfig en0 | grep ether` or `ifconfig | grep ether`

Usage:

```
ifconfig en0 | grep ether
```

Output:

```
	ether 74:0e:a4:8d:42:2b
```

Alternative - All MAC addresses:

```
ifconfig | grep ether
```

Output:

```
	ether 32:a8:16:5c:a8:81
	ether 32:a8:16:5c:a8:80
	ether 32:a8:16:5c:a8:60
	ether 32:a8:16:5c:a8:61
	ether 36:5b:fa:25:da:00
	ether 36:5b:fa:25:da:04
	ether 36:5b:fa:25:da:00
	ether 16:2c:78:93:01:68
	ether 74:0e:a4:8d:42:2b
	ether 42:b4:46:74:ef:a9
	ether 42:b4:46:74:ef:a9
```

4. Hostname Command

Windows Command: `hostname`
macOS Equivalent: `hostname`

Usage:

```
hostname
```

Output:

```
Dhairyas-MacBook-Air.local
```

---

5. System Information (systeminfo → system_profiler)

Windows Command: `systeminfo`
macOS Equivalent: `system_profiler SPSoftwareDataType SPHardwareDataType`

Usage:

```
system_profiler SPSoftwareDataType SPHardwareDataType
```

Output:

```
Software:

    System Software Overview:

      System Version: macOS 15.5 (24F74)
      Kernel Version: Darwin 24.5.0
      Boot Volume: Macintosh HD
      Boot Mode: Normal
      Computer Name: Dhairya's MacBook Air
      User Name: Dhairya Adroja (dhairya)
      Secure Virtual Memory: Enabled
      System Integrity Protection: Enabled
      Time since boot: 2 days, 5 hours, 32 minutes

Hardware:

    Hardware Overview:

      Model Name: MacBook Air
      Model Identifier: MacBookAir10,1
      Model Number: MGN63HN/A
      Chip: Apple M1
      Total Number of Cores: 8 (4 performance and 4 efficiency)
      Memory: 8 GB
      System Firmware Version: 11881.121.1
      OS Loader Version: 11881.121.1
      Serial Number (system): FVFN193L1WFV
      Hardware UUID: C7FB79DC-3C02-5F31-827E-3C09D317D68E
      Provisioning UDID: 00008103-001471363EF9A01E
      Activation Lock Status: Disabled
```

---

6. DNS Lookup (nslookup)

Windows Command: `nslookup google.com`
macOS Equivalent: `nslookup google.com`

Usage:

```
nslookup google.com
```

Output:

```
Server:         10.20.1.1
Address:        10.20.1.1#53

Non-authoritative answer:
Name:   google.com
Address: 142.250.207.174
```

---

7. Trace Route (tracert → traceroute)

Windows Command: `tracert google.com`
macOS Equivalent: `traceroute google.com`

Usage:

```
traceroute -m 10 google.com
```

Output:

```
traceroute to google.com (142.250.192.110), 10 hops max, 40 byte packets
 1  10.20.1.1 (10.20.1.1)  6.015 ms  4.324 ms  3.846 ms
 2  180.211.109.177 (180.211.109.177)  5.001 ms  17.470 ms  4.306 ms
 3  202.131.109.41 (202.131.109.41)  5.145 ms  5.019 ms  4.912 ms
 4  120.72.95.129 (120.72.95.129)  6.417 ms  6.577 ms  5.097 ms
 5  202.131.109.57 (202.131.109.57)  5.241 ms  13.127 ms  6.092 ms
 6  202.131.99.106 (202.131.99.106)  27.799 ms  24.951 ms  24.465 ms
 7  72.14.204.217 (72.14.204.217)  26.144 ms  29.118 ms  25.530 ms
 8  * * *
 9  192.178.86.242 (192.178.86.242)  28.253 ms
    142.250.228.46 (142.250.228.46)  28.793 ms
    142.250.238.196 (142.250.238.196)  24.662 ms
10  72.14.237.139 (72.14.237.139)  25.626 ms  24.429 ms
    72.14.237.11 (72.14.237.11)  25.750 ms
```

---

8. Network Statistics (netstat)

Windows Command: `netstat -r`
macOS Equivalent: `netstat -rn`

Usage:

```
netstat -rn
```

Output (truncated for readability):

```
Routing tables

Internet:
Destination        Gateway            Flags               Netif Expire
default            10.20.1.1          UGScg                 en0
10.20/16           link#11            UCS                   en0      !
10.20.1.1/32       link#11            UCS                   en0      !
10.20.1.1          7c:5a:1c:ce:2f:57  UHLWIir               en0   1200
10.20.64.246/32    link#11            UCS                   en0      !
10.20.64.246       74:e:a4:8d:42:2b   UHLWI                 lo0
127                127.0.0.1          UCS                   lo0
127.0.0.1          127.0.0.1          UH                    lo0
224.0.0/4          link#11            UmCS                  en0      !
224.0.0.251        1:0:5e:0:0:fb      UHmLWI                en0
255.255.255.255/32 link#11            UCS                   en0      !

Internet6:
Destination                             Gateway                                 F
default                                 fe80::%utun0                            U
::1                                     ::1                                     U
fe80::%lo0/64                           fe80::1%lo0                             U
fe80::1%lo0                             link#1                                  U
```

9. Path Ping (pathping → Not directly available, use mtr or traceroute + ping)

Windows Command: `pathping google.com`
macOS Alternative: `mtr google.com` (requires installation) or combination of traceroute and ping

Using built-in tools:

```
# First trace the route
traceroute google.com
# Then ping specific hops
ping -c 4 10.20.1.1
```

Dummy pathping-style output:

```
Tracing route to google.com [142.250.192.110] over a maximum of 10 hops:

  0  Dhairyas-MacBook-Air.local [10.20.64.246]
  1  10.20.1.1
  2  180.211.109.177
  3  202.131.109.41
  4  120.72.95.129
  5  202.131.109.57
  6  202.131.99.106
  7  72.14.204.217
  8  *
  9  192.178.86.242
 10  google.com [142.250.192.110]

Computing statistics for 250 seconds...
            Source to Here   This Node/Link
Hop  RTT    Lost/Sent = Pct  Lost/Sent = Pct  Address
  0                                           Dhairyas-MacBook-Air.local [10.20.64
                                0/ 100 =  0%   |
  1    4ms     0/ 100 =  0%     0/ 100 =  0%  10.20.1.1
                                0/ 100 =  0%   |
  2    5ms     0/ 100 =  0%     0/ 100 =  0%  180.211.109.177
                                0/ 100 =  0%   |
  3    5ms     0/ 100 =  0%     0/ 100 =  0%  202.131.109.41
```

10. ARP Table (arp)

Windows Command: `arp -a`
macOS Equivalent: `arp -a`

Usage:

```
arp -a
```

Output (truncated for readability):

```
? (10.20.1.1) at 7c:5a:1c:ce:2f:57 on en0 ifscope [ethernet]
? (10.20.4.16) at bc:f:f3:6a:7:6b on en0 ifscope [ethernet]
? (10.20.4.21) at bc:f:f3:6a:3:76 on en0 ifscope [ethernet]
? (10.20.4.23) at bc:f:f3:6a:7:cd on en0 ifscope [ethernet]
? (10.20.4.29) at bc:f:f3:6a:7:84 on en0 ifscope [ethernet]
? (10.20.4.35) at 44:8a:5b:3:85:4f on en0 ifscope [ethernet]
? (10.20.64.246) at 74:e:a4:8d:42:2b on en0 ifscope permanent [ethernet]
? (10.20.65.10) at 34:f3:9a:c5:c7:54 on en0 ifscope [ethernet]
? (10.20.66.77) at 5c:ed:8c:b0:fe:66 on en0 ifscope [ethernet]
? (10.20.67.14) at 26:8c:7d:7b:3b:20 on en0 ifscope [ethernet]
? (10.20.68.19) at e4:aa:ea:b8:95:85 on en0 ifscope [ethernet]
? (169.254.90.200) at cc:47:40:83:f4:69 on en0 [ethernet]
? (169.254.235.238) at c0:35:32:8e:42:c1 on en0 [ethernet]
mdns.mcast.net (224.0.0.251) at 1:0:5e:0:0:fb on en0 ifscope permanent [ethernet]
```

Command Comparison Summary

Windows Command	macOS Equivalent	Purpose
`ipconfig`	`ifconfig`	Display network configuration
`ipconfig /all`	`ifconfig -a`	Display all network interfaces
`ping`	`ping -c <count>`	Test network connectivity
`getmac`	`ifconfig | grep ether`	Display MAC addresses
`hostname`	`hostname`	Display computer name
`systeminfo`	`system_profiler`	Display system information
`nslookup`	`nslookup` or `dig`	DNS lookup
`tracert`	`traceroute`	Trace network path
`netstat`	`netstat`	Display network statistics
`pathping`	`mtr` (needs install)	Combined ping and traceroute
`arp -a`	`arp -a`	Display ARP table

Student Name: Dhairya Adroja

Enrollment No: 24010101602

Course: B.Tech. CSE

Date: 05 June 2025