MINI PROJECT REPORT

Subject : OPERATING SYSTEM LAB **Semester : IV**

Title : Network Management/Network Monitoring System

Abstract

A computer network is a group of computers that use a set of common communication protocols over digital interconnections for the purpose of sharing resources located on or provided by the network nodes. The interconnections between nodes are formed from a broad spectrum of telecommunication network technologies, based on physically wired, optical, and wireless radio-frequency methods that may be arranged in a variety of network topologies.

The nodes of a computer network may include personal computers, servers, networking hardware, or other specialised or general-purpose hosts. They are identified by hostnames and network addresses. Hostnames serve as memorable labels for the nodes, rarely changed after initial assignment. Network addresses serve for locating and identifying the nodes by communication protocols such as the Internet Protocol.

Computers loaded with Linux Operating Systems can also be a part of such small or large networks because of its multitasking and multiuser nature. Maintaining the system and network is a crucial task of the System/Network Administrator.

Proposed Architecture :

In this project, the bash script for the Network management provides various system related information along with the Network information. The bash script consists of the most useful Networking commands and provides the system administrator with most of the internet connection information all at once, instead of executing the individual commands one by one.

We have also implemented a feature wherein, the Bash script would scan and monitor the network using a combination of bash and ping commands. The script will scan networks for hosts attached to a specified IP address. The script will print a message if the ping command was successful or not. The project also displays various other data such as the system information, Routing tables, ARP tables, active connections, etc.

Idea :

Computer networks may be classified by many criteria, including the transmission medium used to carry signals, bandwidth, communications protocols to organize network traffic, the network size, the topology, traffic control mechanism, and organizational intent.
Computer networks support many applications and services, such as access to the World Wide Web, digital video, digital audio, shared use of application and storage servers, printers, and fax machines, and use of email and instant messaging applications.

There are several tools to monitor the network, for remote administration and to monitor the traffic flowing through network interfaces and measure the speed at which data is currently being transferred.

The System Administrator has to monitor the servers, users, logs, create backup, etc along with their other tasks. For most of the repetitive tasks, it is instrumental to write a script to automate the process. In this project, we have implemented a Shell Script that aims to automate the task of getting most of the information about the System, Network, host, Internal IP, External IP, Uptime, etc. Additionally, we have taken care of formatting the output. So as to make it understandable for any user.

Scope :

Every sector requires networking in some or the other way. In the business sector, networking is applicable from manufacturing to business processing. As organizations and institutions invest in domains like technology, cloud computing, big data, etc they all depend on a workforce with networking skills to make the most of this technology. Today the supply of networking workforce does not match with the demand, due to which in the future these professionals will discover opportunities for the growing economy.

The networking domain is huge with sub-domains like routing and switching, security, service provider, collaboration, etc. The professionals working in this domain handle the basics of networking. Every company will require routing and switching professionals to handle their networks. Same is applicable in the security domain. Every network set up in companies will need security to protect their sensitive data. Hence the job opportunities in this domain multiply day by day.

Shell Script Code :

```
#! /bin/bash
# unset any variable which system may be using

unset resetattr os architecture kernelrelease internalip externalip
nameserver

if [[ $# -eq 0 ]]
```

```
then
{

# Define Variable resetattr
resetattr=$(tput sgr0)

# Check if connected to Internet or not
ping www.google.com &> /dev/null && echo -e '\E[32m'"Internet: $resetattr
Connected" || echo -e '\E[32m'"Internet: $resetattr Disconnected"

# Check OS Type
os=$(uname -o)
echo -e '\E[32m'"Operating System Type :" $resetattr $os

# Check OS Version
OSSTR=$(uname -s)
echo -e '\E[32m'"OS Version :" $resetattr $OSSTR

# Check Architecture
architecture=$(uname -m)
echo -e '\E[32m'"Architecture :" $resetattr $architecture

# Check Kernel Release
kernelrelease=$(uname -r)
echo -e '\E[32m'"Kernel Release :" $resetattr $kernelrelease

# Check hostname
echo -e '\E[32m'"Hostname :" $resetattr $HOSTNAME

# Check Internal IP
internalip=$(ipconfig | awk '/IPv4/ {print}' | cut -d ':' -f 2)
echo -e '\E[32m'"Internal IP :" $resetattr $internalip

# Check External IP
externalip=$(curl -s ipecho.net/plain;echo)
echo -e '\E[32m'"External IP : $resetattr "$externalip
echo

# Check DNS
nameservers=$(nslookup google.com | awk '{print}')
```

```
echo -e '\E[32m'"Name Servers :" $resetattr $nameservers
echo

echo -e '\E[32m'"Checking if the host is up.." $resetattr
declare -a name
read -p "Enter the Hostnames : " name
for i in ${name[@]}
do
ping -c 1 $i &> /dev/null

if [ $? -ne 0 ]; then
  echo "`date`: ping failed, $i host is down!"
else
  echo "`date`: ping successful, $i host is up!"
fi
done
echo
sleep 5

# Displaying IP Routing tables
echo -e '\E[32m'"Displaying IPV4 Route Table :" $resetattr
route PRINT -4
echo
sleep 5

# Displaying IP Routing tables
echo -e '\E[32m'"Displaying IPV6 Route Table :" $resetattr
route PRINT -6
echo
sleep 5

# Displaying Connection Information
echo -e '\E[32m'"Displaying Connection Information :" $resetattr
netstat -a
echo
sleep 5

# Displaying ARP tables
echo -e '\E[32m'"Displaying ARP Tables :" $resetattr
arp -a
```

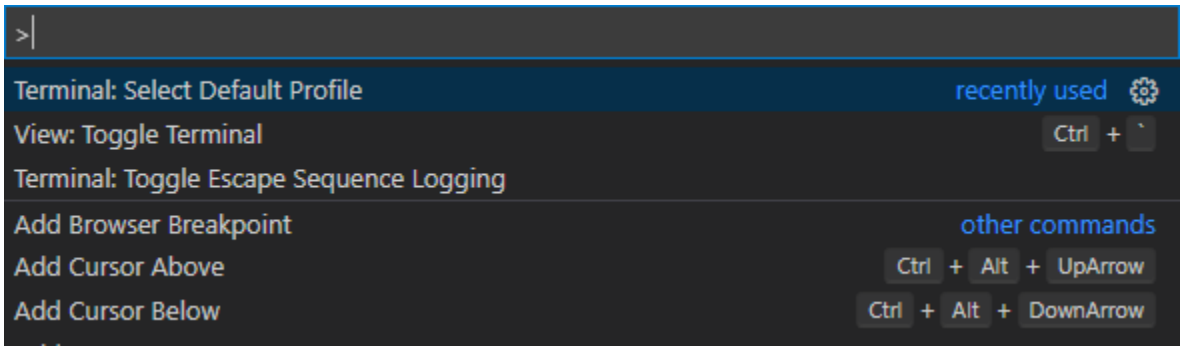
```
echo
sleep 5

# Check System Uptime
sysuptime=$(systeminfo | awk '/Boot Time/ {print}')
echo -e '\E[32m'"System Uptime Days/(HH:MM) :" $resetattr $sysuptime
echo

echo "Commands unavailable in Git bash :"
echo
echo -e '\E[32m'"ifconfig" : $resetattr "Display and manipulate route and
network interfaces." #ipconfig
echo -e '\E[32m'"ip" : $resetattr "It is a replacement of ifconfig
command." #ipconfig
echo -e '\E[32m'"traceroute" : $resetattr "Network troubleshooting
utility."
echo -e '\E[32m'"tracepath" : $resetattr "Similar to traceroute but
doesn't require root privileges."
echo -e '\E[32m'"ss" : $resetattr "It is a replacement of netstat."
#netstat
echo -e '\E[32m'"dig" : $resetattr "Query DNS related information." #curl
echo -e '\E[32m'"host" : $resetattr "Performs DNS lookups." #nslookup
echo -e '\E[32m'"iwconfig" : $resetattr "Used to configure wireless
network interface." #ipconfig
echo -e '\E[32m'"wget" : $resetattr "To download a file from internet."
#curl
echo -e '\E[32m'"mtr" : $resetattr "Combines ping and tracepath into a
single command."
echo -e '\E[32m'"whois" : $resetattr "Will tell you about the website's
whois."
echo -e '\E[32m'"ifplugstatus" : $resetattr "Tells whether a cable is
plugged in or not."
echo

echo -e '\E[32m'"Scanning Network Subnet.." $resetattr
is_alive_ping()
{
  ping -c 1 $1 &> /dev/null
  [ $? -eq 0 ] && echo Node with IP: $i is up!
}
```

```
for i in 192.168.1.{1..255}
do
is_alive_ping $i & disown
done

# Unset Variables
unset resetattr os architecture kernelrelease internalip externalip
nameserver

}
fi
```

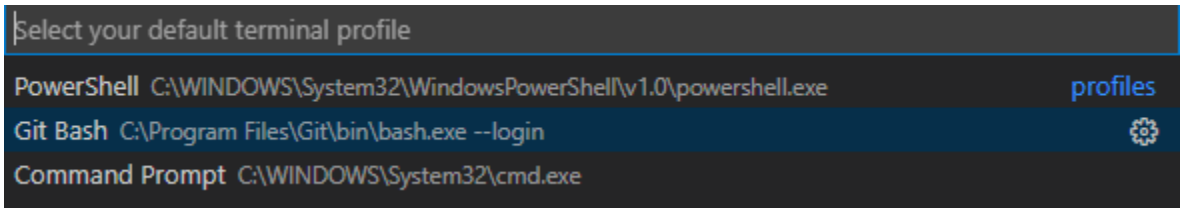
P.T.O.

Stepwise tutorial with screenshot of demonstration :

1. Install ‘Visual Studio code’ and ‘Git Bash’
2. Open Visual Studio Code in the administrator mode
3. Click Ctrl+Shift+P to open the search bar and search for “Select Default Profile”



4. Hit Enter and Select “Git Bash”



5. Press Ctrl+` to open the terminal

6. Navigate to the folder in which the code to be run is stored using the “cd” command
7. Run the file by typing “./net_monitor.sh” in the terminal
8. The Output will be displayed as follows :

```
Jessica Dsouza@Jessica MINGW64 ~/Documents/OS_project (master)
$ ./net_monitor.sh
Internet:  Connected
Operating System Type :  Msys
OS Version :  MINGW64_NT-10.0-18363
Architecture :  x86_64
Kernel Release :  3.1.7-340.x86_64
Hostname :  Jessica
Internal IP :  192.168.56.1 192.168.1.204
External IP :  45.119.15.222

Name Servers :  Server: UnKnown Address: 192.168.1.1 Name: google.com Addresses: 2404:6800:4009:809::200e 216.58.196.78
```

The above output displays the System and Hardware details along with the internal and external IP addresses. It also displays the Names of the servers for the domain "google.com", thus, maps its IP addresses with the help of the domain name.

9. The following output snapshot displays the result of the ping command for the specified hostnames :

```
Checking if the host is up..
Enter the Hostnames : google.com yahoo.com 192.168.1.204 198.162.1.1 Codeblocks
Thu, May  6, 2021 12:40:40 PM: ping successful, google.com host is up!
Thu, May  6, 2021 12:40:43 PM: ping successful, yahoo.com host is up!
Thu, May  6, 2021 12:40:46 PM: ping successful, 192.168.1.204 host is up!
Thu, May  6, 2021 12:41:05 PM: ping failed, 198.162.1.1 host is down!
Thu, May  6, 2021 12:41:08 PM: ping failed, Codeblocks host is down!
```

It takes the input from the user as hostnames. Then the ping command sends packets of data to the specified internet destinations and returns the result telling the user whether the ping command was successful or not. Hence, depicts whether the particular host is up or down.

P.T.O.

10. Following the output of the ‘route PRINT -4’ command which displays the Routing Table for IPv4 addresses :

```
Displaying IPV4 Route Table :
===========================================================================
Interface List
  9...b8 70 f4 dd 73 8a ......Broadcom NetLink (TM) Gigabit Ethernet
 41...0a 00 27 00 00 29 ......VirtualBox Host-Only Ethernet Adapter
 14...1e 55 f9 bd 5f 3d ......Microsoft Wi-Fi Direct Virtual Adapter
  3...2e 55 f9 bd 5f 3d ......Microsoft Wi-Fi Direct Virtual Adapter #2
 10...ec 55 f9 bd 5f 3d ......Qualcomm Atheros AR5B97 Wireless Network Adapter
  1...........................Software Loopback Interface 1
===========================================================================

IPv4 Route Table
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0      192.168.1.1    192.168.1.204     40
        127.0.0.0        255.0.0.0         On-link         127.0.0.1    331
        127.0.0.1  255.255.255.255         On-link         127.0.0.1    331
  127.255.255.255  255.255.255.255         On-link         127.0.0.1    331
      192.168.1.0    255.255.255.0         On-link     192.168.1.204    296
    192.168.1.204  255.255.255.255         On-link     192.168.1.204    296
    192.168.1.255  255.255.255.255         On-link     192.168.1.204    296
     192.168.56.0    255.255.255.0         On-link      192.168.56.1    281
     192.168.56.1  255.255.255.255         On-link      192.168.56.1    281
   192.168.56.255  255.255.255.255         On-link      192.168.56.1    281
        224.0.0.0        240.0.0.0         On-link         127.0.0.1    331
        224.0.0.0        240.0.0.0         On-link     192.168.1.204    296
        224.0.0.0        240.0.0.0         On-link      192.168.56.1    281
  255.255.255.255  255.255.255.255         On-link         127.0.0.1    331
  255.255.255.255  255.255.255.255         On-link     192.168.1.204    296
  255.255.255.255  255.255.255.255         On-link      192.168.56.1    281
===========================================================================
Persistent Routes:
  None
```

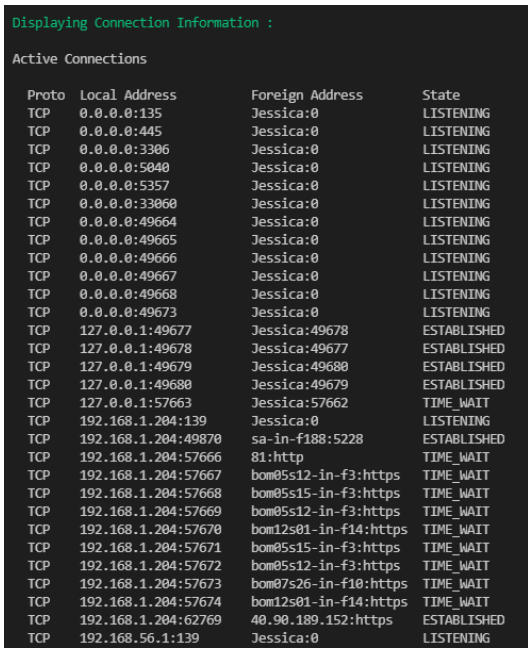
P.T.O.

11. Following the output of the ‘route PRINT -6’ command which displays the Routing Table for IPv6 addresses :

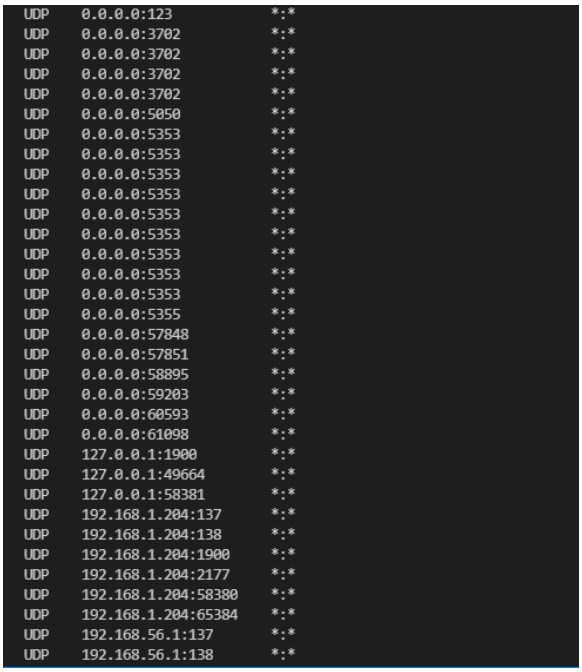
```
Displaying IPV6 Route Table :
===========================================================================
Interface List
  9...b8 70 f4 dd 73 8a ......Broadcom NetLink (TM) Gigabit Ethernet
 41...0a 00 27 00 00 29 ......VirtualBox Host-Only Ethernet Adapter
 14...1e 55 f9 bd 5f 3d ......Microsoft Wi-Fi Direct Virtual Adapter
  3...2e 55 f9 bd 5f 3d ......Microsoft Wi-Fi Direct Virtual Adapter #2
 10...ec 55 f9 bd 5f 3d ......Qualcomm Atheros AR5B97 Wireless Network Adapter
  1...........................Software Loopback Interface 1
===========================================================================

IPv6 Route Table
===========================================================================
Active Routes:
 If Metric Network Destination      Gateway
  1    331 ::1/128                  On-link
 10    296 fe80::/64                On-link
 41    281 fe80::/64                On-link
 41    281 fe80::4fa:1355:a0f5:c0c7/128
                                    On-link
 10    296 fe80::f8c3:7d5a:6c18:655d/128
                                    On-link
  1    331 ff00::/8                 On-link
 10    296 ff00::/8                 On-link
 41    281 ff00::/8                 On-link
===========================================================================
Persistent Routes:
  None
```

P.T.O.

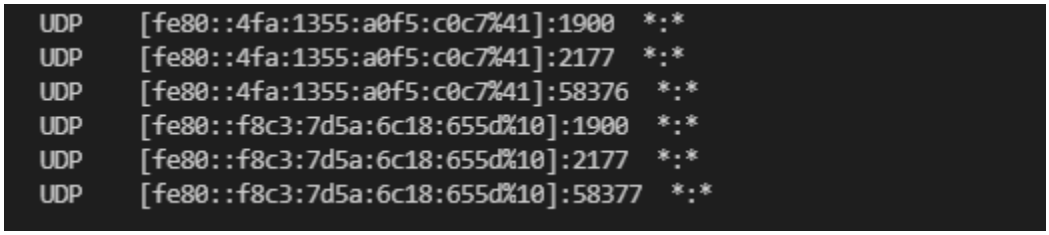
12. Following is the output of the ‘netstat -a’ command which displays the information about the active connections for TCP and UDP networks :



```
Displaying Connection Information :

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:135            Jessica:0              LISTENING
  TCP    0.0.0.0:445            Jessica:0              LISTENING
  TCP    0.0.0.0:3306           Jessica:0              LISTENING
  TCP    0.0.0.0:5040           Jessica:0              LISTENING
  TCP    0.0.0.0:5357           Jessica:0              LISTENING
  TCP    0.0.0.0:33060          Jessica:0              LISTENING
  TCP    0.0.0.0:49664          Jessica:0              LISTENING
  TCP    0.0.0.0:49665          Jessica:0              LISTENING
  TCP    0.0.0.0:49666          Jessica:0              LISTENING
  TCP    0.0.0.0:49667          Jessica:0              LISTENING
  TCP    0.0.0.0:49668          Jessica:0              LISTENING
  TCP    0.0.0.0:49673          Jessica:0              LISTENING
  TCP    127.0.0.1:49677        Jessica:49678          ESTABLISHED
  TCP    127.0.0.1:49678        Jessica:49677          ESTABLISHED
  TCP    127.0.0.1:49679        Jessica:49680          ESTABLISHED
  TCP    127.0.0.1:49680        Jessica:49679          ESTABLISHED
  TCP    127.0.0.1:57663        Jessica:57662          TIME_WAIT
  TCP    192.168.1.204:139      Jessica:0              LISTENING
  TCP    192.168.1.204:49870    sa-in-f188:5228        ESTABLISHED
  TCP    192.168.1.204:57666    81:http                TIME_WAIT
  TCP    192.168.1.204:57667    bom05s12-in-f3:https   TIME_WAIT
  TCP    192.168.1.204:57668    bom05s15-in-f3:https   TIME_WAIT
  TCP    192.168.1.204:57669    bom05s12-in-f3:https   TIME_WAIT
  TCP    192.168.1.204:57670    bom12s01-in-f14:https  TIME_WAIT
  TCP    192.168.1.204:57671    bom05s15-in-f3:https   TIME_WAIT
  TCP    192.168.1.204:57672    bom05s12-in-f3:https   TIME_WAIT
  TCP    192.168.1.204:57673    bom07s26-in-f10:https  TIME_WAIT
  TCP    192.168.1.204:57674    bom12s01-in-f14:https  TIME_WAIT
  TCP    192.168.1.204:62769    40.90.189.152:https    ESTABLISHED
  TCP    192.168.56.1:139       Jessica:0              LISTENING
```



```
  UDP    0.0.0.0:123            *:*
  UDP    0.0.0.0:3702           *:*
  UDP    0.0.0.0:3702           *:*
  UDP    0.0.0.0:3702           *:*
  UDP    0.0.0.0:3702           *:*
  UDP    0.0.0.0:5050           *:*
  UDP    0.0.0.0:5353           *:*
  UDP    0.0.0.0:5353           *:*
  UDP    0.0.0.0:5353           *:*
  UDP    0.0.0.0:5353           *:*
  UDP    0.0.0.0:5353           *:*
  UDP    0.0.0.0:5353           *:*
  UDP    0.0.0.0:5353           *:*
  UDP    0.0.0.0:5353           *:*
  UDP    0.0.0.0:5353           *:*
  UDP    0.0.0.0:5355           *:*
  UDP    0.0.0.0:57848          *:*
  UDP    0.0.0.0:57851          *:*
  UDP    0.0.0.0:58895          *:*
  UDP    0.0.0.0:59203          *:*
  UDP    0.0.0.0:60593          *:*
  UDP    0.0.0.0:61098          *:*
  UDP    127.0.0.1:1900         *:*
  UDP    127.0.0.1:49664        *:*
  UDP    127.0.0.1:58381        *:*
  UDP    192.168.1.204:137      *:*
  UDP    192.168.1.204:138      *:*
  UDP    192.168.1.204:1900     *:*
  UDP    192.168.1.204:2177     *:*
  UDP    192.168.1.204:58380    *:*
  UDP    192.168.1.204:65384    *:*
  UDP    192.168.56.1:137       *:*
  UDP    192.168.56.1:138       *:*
```



```
  UDP    [fe80::4fa:1355:a0f5:c0c7%41]:1900  *:*
  UDP    [fe80::4fa:1355:a0f5:c0c7%41]:2177  *:*
  UDP    [fe80::4fa:1355:a0f5:c0c7%41]:58376  *:*
  UDP    [fe80::f8c3:7d5a:6c18:655d%10]:1900  *:*
  UDP    [fe80::f8c3:7d5a:6c18:655d%10]:2177  *:*
  UDP    [fe80::f8c3:7d5a:6c18:655d%10]:58377  *:*
```

P.T.O.

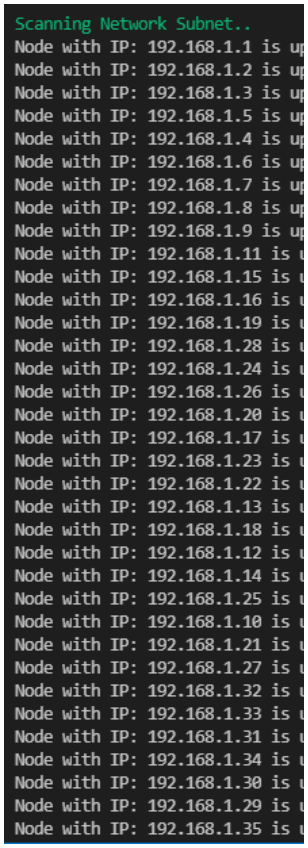
13. Following is the output of ‘arp -a’ command which displays the ARP tables and ‘systeminfo’ command which specifies the system uptime :

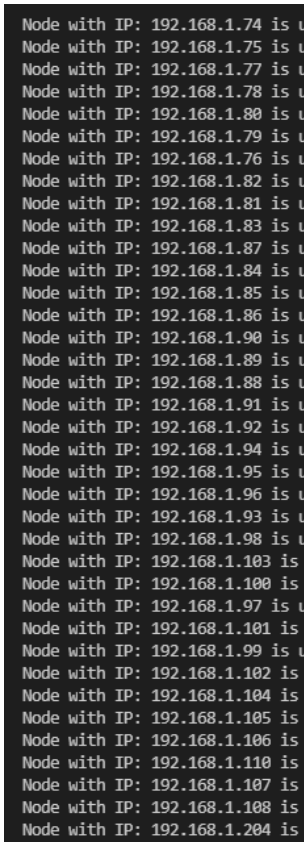
```
Displaying ARP Tables :

Interface: 192.168.1.204 --- 0xa
  Internet Address      Physical Address      Type
  192.168.1.1           a0-47-d7-21-2f-68     dynamic
  192.168.1.202         10-77-17-c8-14-f8     dynamic
  192.168.1.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

Interface: 192.168.56.1 --- 0x29
  Internet Address      Physical Address      Type
  192.168.56.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

System Uptime Days/(HH:MM) :  System Boot Time: 3/22/2021, 10:39:34 AM
```

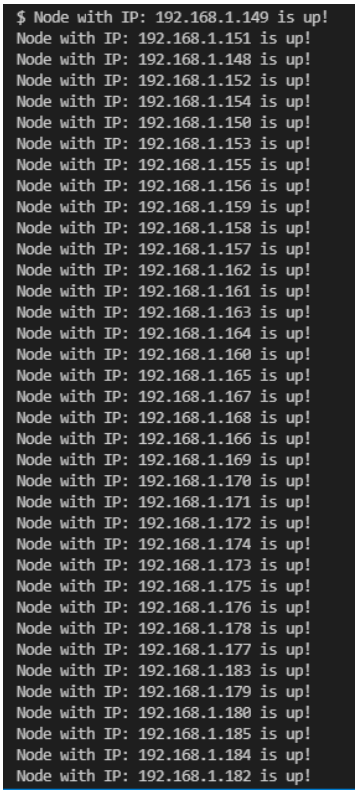
14. Following are the Network management/monitoring commands in linux which are not supported by Git Bash :

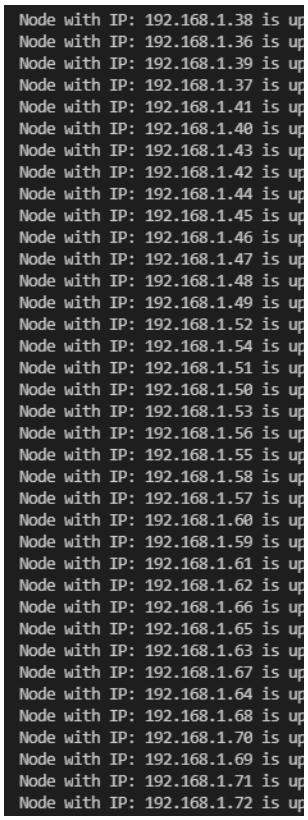
```
Commands unavailable in Git bash :

ifconfig :  Display and manipulate route and network interfaces.
ip :  It is a replacement of ifconfig command.
traceroute :  Network troubleshooting utility.
tracepath :  Similar to traceroute but doesn't require root privileges.
ss :  It is a replacement of netstat.
dig :  Query DNS related information.
host :  Performs DNS lookups.
iwconfig :  Used to configure wireless network interface.
wget :  To download a file from internet.
mtr :  Combines ping and tracepath into a single command.
whois :  Will tell you about the website's whois.
ifplugstatus :  Tells whether a cable is plugged in or not.
```

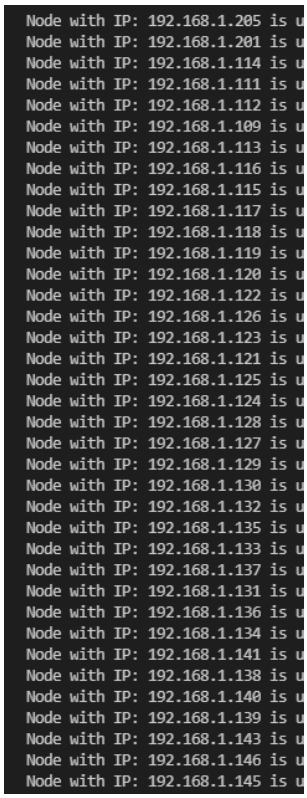
15. Following is the output of the ‘ping’ command that scans for all the nodes with IP addresses 192.168.1.{1..255} that are up on the network :


Scanning Network Subnet..
Node with IP: 192.168.1.1 is up
Node with IP: 192.168.1.2 is up
Node with IP: 192.168.1.3 is up
Node with IP: 192.168.1.5 is up
Node with IP: 192.168.1.4 is up
Node with IP: 192.168.1.6 is up
Node with IP: 192.168.1.7 is up
Node with IP: 192.168.1.8 is up
Node with IP: 192.168.1.9 is up
Node with IP: 192.168.1.11 is u
Node with IP: 192.168.1.15 is u
Node with IP: 192.168.1.16 is u
Node with IP: 192.168.1.19 is u
Node with IP: 192.168.1.28 is u
Node with IP: 192.168.1.24 is u
Node with IP: 192.168.1.26 is u
Node with IP: 192.168.1.20 is u
Node with IP: 192.168.1.17 is u
Node with IP: 192.168.1.23 is u
Node with IP: 192.168.1.22 is u
Node with IP: 192.168.1.13 is u
Node with IP: 192.168.1.18 is u
Node with IP: 192.168.1.12 is u
Node with IP: 192.168.1.14 is u
Node with IP: 192.168.1.25 is u
Node with IP: 192.168.1.10 is u
Node with IP: 192.168.1.21 is u
Node with IP: 192.168.1.27 is u
Node with IP: 192.168.1.32 is u
Node with IP: 192.168.1.33 is u
Node with IP: 192.168.1.31 is u
Node with IP: 192.168.1.34 is u
Node with IP: 192.168.1.30 is u
Node with IP: 192.168.1.29 is u
Node with IP: 192.168.1.35 is u


Node with IP: 192.168.1.74 is u
Node with IP: 192.168.1.75 is u
Node with IP: 192.168.1.77 is u
Node with IP: 192.168.1.78 is u
Node with IP: 192.168.1.80 is u
Node with IP: 192.168.1.79 is u
Node with IP: 192.168.1.76 is u
Node with IP: 192.168.1.82 is u
Node with IP: 192.168.1.81 is u
Node with IP: 192.168.1.83 is u
Node with IP: 192.168.1.87 is u
Node with IP: 192.168.1.84 is u
Node with IP: 192.168.1.85 is u
Node with IP: 192.168.1.86 is u
Node with IP: 192.168.1.90 is u
Node with IP: 192.168.1.89 is u
Node with IP: 192.168.1.88 is u
Node with IP: 192.168.1.91 is u
Node with IP: 192.168.1.92 is u
Node with IP: 192.168.1.94 is u
Node with IP: 192.168.1.95 is u
Node with IP: 192.168.1.96 is u
Node with IP: 192.168.1.93 is u
Node with IP: 192.168.1.98 is u
Node with IP: 192.168.1.103 is
Node with IP: 192.168.1.100 is
Node with IP: 192.168.1.97 is u
Node with IP: 192.168.1.101 is
Node with IP: 192.168.1.99 is u
Node with IP: 192.168.1.102 is
Node with IP: 192.168.1.104 is
Node with IP: 192.168.1.105 is
Node with IP: 192.168.1.106 is
Node with IP: 192.168.1.110 is
Node with IP: 192.168.1.107 is
Node with IP: 192.168.1.108 is
Node with IP: 192.168.1.204 is


$ Node with IP: 192.168.1.149 is up!
Node with IP: 192.168.1.151 is up!
Node with IP: 192.168.1.148 is up!
Node with IP: 192.168.1.152 is up!
Node with IP: 192.168.1.154 is up!
Node with IP: 192.168.1.150 is up!
Node with IP: 192.168.1.153 is up!
Node with IP: 192.168.1.155 is up!
Node with IP: 192.168.1.156 is up!
Node with IP: 192.168.1.159 is up!
Node with IP: 192.168.1.158 is up!
Node with IP: 192.168.1.157 is up!
Node with IP: 192.168.1.162 is up!
Node with IP: 192.168.1.161 is up!
Node with IP: 192.168.1.163 is up!
Node with IP: 192.168.1.164 is up!
Node with IP: 192.168.1.160 is up!
Node with IP: 192.168.1.165 is up!
Node with IP: 192.168.1.167 is up!
Node with IP: 192.168.1.168 is up!
Node with IP: 192.168.1.166 is up!
Node with IP: 192.168.1.169 is up!
Node with IP: 192.168.1.170 is up!
Node with IP: 192.168.1.171 is up!
Node with IP: 192.168.1.172 is up!
Node with IP: 192.168.1.174 is up!
Node with IP: 192.168.1.173 is up!
Node with IP: 192.168.1.175 is up!
Node with IP: 192.168.1.176 is up!
Node with IP: 192.168.1.178 is up!
Node with IP: 192.168.1.177 is up!
Node with IP: 192.168.1.183 is up!
Node with IP: 192.168.1.179 is up!
Node with IP: 192.168.1.180 is up!
Node with IP: 192.168.1.185 is up!
Node with IP: 192.168.1.184 is up!
Node with IP: 192.168.1.182 is up!


Node with IP: 192.168.1.38 is up
Node with IP: 192.168.1.36 is up
Node with IP: 192.168.1.39 is up
Node with IP: 192.168.1.37 is up
Node with IP: 192.168.1.41 is up
Node with IP: 192.168.1.40 is up
Node with IP: 192.168.1.43 is up
Node with IP: 192.168.1.42 is up
Node with IP: 192.168.1.44 is up
Node with IP: 192.168.1.45 is up
Node with IP: 192.168.1.46 is up
Node with IP: 192.168.1.47 is up
Node with IP: 192.168.1.48 is up
Node with IP: 192.168.1.49 is up
Node with IP: 192.168.1.52 is up
Node with IP: 192.168.1.54 is up
Node with IP: 192.168.1.51 is up
Node with IP: 192.168.1.50 is up
Node with IP: 192.168.1.53 is up
Node with IP: 192.168.1.56 is up
Node with IP: 192.168.1.55 is up
Node with IP: 192.168.1.58 is up
Node with IP: 192.168.1.57 is up
Node with IP: 192.168.1.60 is up
Node with IP: 192.168.1.59 is up
Node with IP: 192.168.1.61 is up
Node with IP: 192.168.1.62 is up
Node with IP: 192.168.1.66 is up
Node with IP: 192.168.1.65 is up
Node with IP: 192.168.1.63 is up
Node with IP: 192.168.1.67 is up
Node with IP: 192.168.1.64 is up
Node with IP: 192.168.1.68 is up
Node with IP: 192.168.1.70 is up
Node with IP: 192.168.1.69 is up
Node with IP: 192.168.1.71 is up
Node with IP: 192.168.1.72 is up


Node with IP: 192.168.1.205 is u
Node with IP: 192.168.1.201 is u
Node with IP: 192.168.1.114 is u
Node with IP: 192.168.1.111 is u
Node with IP: 192.168.1.112 is u
Node with IP: 192.168.1.109 is u
Node with IP: 192.168.1.113 is u
Node with IP: 192.168.1.116 is u
Node with IP: 192.168.1.115 is u
Node with IP: 192.168.1.117 is u
Node with IP: 192.168.1.118 is u
Node with IP: 192.168.1.119 is u
Node with IP: 192.168.1.120 is u
Node with IP: 192.168.1.122 is u
Node with IP: 192.168.1.126 is u
Node with IP: 192.168.1.123 is u
Node with IP: 192.168.1.121 is u
Node with IP: 192.168.1.125 is u
Node with IP: 192.168.1.124 is u
Node with IP: 192.168.1.128 is u
Node with IP: 192.168.1.127 is u
Node with IP: 192.168.1.129 is u
Node with IP: 192.168.1.130 is u
Node with IP: 192.168.1.132 is u
Node with IP: 192.168.1.135 is u
Node with IP: 192.168.1.133 is u
Node with IP: 192.168.1.137 is u
Node with IP: 192.168.1.131 is u
Node with IP: 192.168.1.136 is u
Node with IP: 192.168.1.134 is u
Node with IP: 192.168.1.141 is u
Node with IP: 192.168.1.138 is u
Node with IP: 192.168.1.140 is u
Node with IP: 192.168.1.139 is u
Node with IP: 192.168.1.143 is u
Node with IP: 192.168.1.146 is u
Node with IP: 192.168.1.145 is u

Conclusion :

Thus, in this project, the bash script for the network management provides various system related information along with the Network information. The bash script consists of the most useful Networking commands and provides the system administrator with most of the internet connection information all at once, instead of executing the individual commands one by one.

We have also implemented a feature to check whether the particular host i.e any domain, IP address, etc. is connected to the internet or not. This has been achieved with help of the ping command. The project also displays various other data such as the system information, Routing tables, ARP tables, active connections, etc.

References :

www.tecmint.com
www.google.com
linuxcommand.org
linuxconfig.org
www.geeksforgeeks.org
bash.cyberciti.biz
www.tutorialspoint.com