COMPUTER SYSTEMS LAB CS 558
Assignment 01 (Network Diagnostic Commands)

Vijay Purohit | 214101058

M.Tech CSE, IITG

Contents

1 *ping* Command

(a) `-c` which stands for `count`. `ping -c 2 www.google.com`

(b) `-i` switch which stands for `interval`. Wait interval seconds between sending each packet. `ping -i 2 www.google.com`

(c) `-l` switch which stand for `preload`. `ping -l 2 www.google.com`. The limit of sending such packets by normal users is **3**.

(d) The command to set the packet size is `-s` switch. `ping -s 32 www.google.com`. Packet size 32B, ICMP Header size 8B then Total Packet Size = 40 Bytes.

2 *ping* Experiment With Six Host

Table 1: Hosts AVG RTT(ms) at different hours of the day (22/03/2022) with default packet size.

Hosts	09:00	13:00	20:00	24:00
BookMyShow (104.16.123.37), USA	174.859	314.785	604.841	237.305
YouTube (142.250.64.110), USA	574.014	640.428	622.783	725.882
Twitter (104.244.42.1), USA	219.607	309.642	306.901	384.554
Spotify (35.186.224.25), USA	144.692	553.554	329.335	393.053
Instagram (2a03:2880:f212:e5:face:b00c:0:4420), Ireland	610.696	646.329	949.457	655.963
CodeForces (213.248.110.126), Russia	397.447	742.212	492.919	595.635

For the default packet size, there was no packet loss greater than 0%. But the experiment conducted with larger packet size e.g. 2048B and in some occasional cases were showing packet loss. There are many causes of packet loss: Network Congestion, Problems With Network Hardware, Software Bugs, Overloaded Devices, Security Threats, reduce throughput.
There is a *positive correlation between distance and RTT*. There can be number of reasons for this: an increased hop count. The packets have to go through more routers, at each router there may be a delay. More routers, the longer the RTT. But it is less as compared to linearized distance. So RTTs are **weakly correlated to geographic distance**.

Table 2: Single Host AVG RTT with respect to change in packet size

Time\Size	56B	64B	128B	256B	512B	1024B	2048B
09:00	397.447	438.643	533.071	483.249	468.035	492.977	538.684
20:00	492.919	540.31	506.338	579.314	870.148	631.365	779.053
23:59	595.635	563.865	867.929	648.899	656.335	640.39	771.742

Varying packet size from 64 bytes to 2048 bytes shows that for some of the hosts for higher packet size there was 100% packet loss. While for some it is showing valid RTT value. Therefore, we cannot make any strong conclusion here.

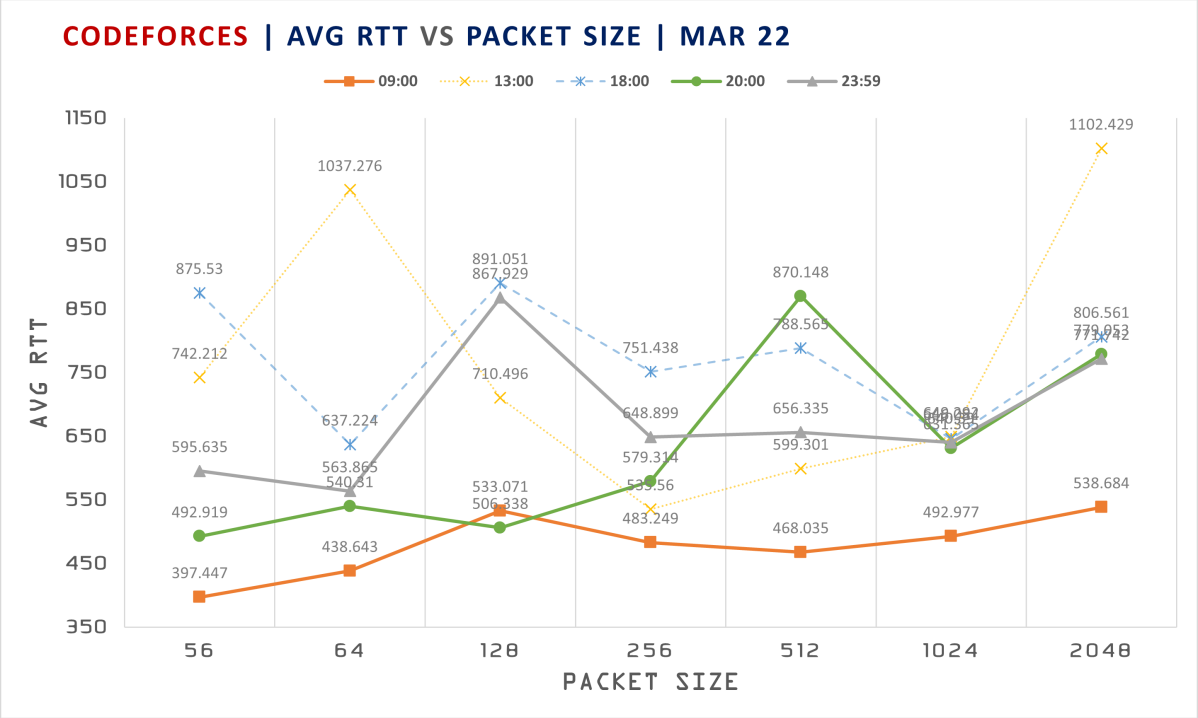


Figure 1: Avg RTT vs Packet Size

3 *ifconfig* and *route* Commands

(a) `ifconfig` (interface configuration) is used to view and configure the network interfaces. It is used at boot time to set up interfaces as necessary. If no arguments are given, `ifconfig` displays the status of the currently active interfaces. **eno1, lo and wlp2s0** are the names of the my system active network interfaces.

Interface Details:

- `UP`:kernel modules related to the interface have been loaded and interface is activated.
- `BROADCAST`: interface is configured to handle broadcast packets.
- `RUNNING` indicates that interface is ready to accept data.
- `MULTICAST` indicates that interface supports multicasting.
- `mtu` is maximum transmission unit.
- `inet addr` is IPv4 address assigned to the interface. `inet6 addr` is IPv6 address assigned to the interface.
- `broadcast` is broadcast address for the interface.
- `netmask` is network mask for the interface.
- `ether` is hardware address of the ethernet interface (also known as MAC address).
- `scopeid` is scope of IPv6 address. It can be link-local or global.

Interface stats:

- `txqueuelen` is length of transmission queue.
- `RX packets, TX packets` is a total number of packets received/ transmitted.

- `RX bytes/TX bytes` is a total number of bytes received/transmitted over interface.
- `RX dropped` is a number of dropped packets due to unintended VLAN tags.
- `RX overruns` is a number of received packets that experienced fifo overruns.
- `RX frame` is a number of misaligned frames, i.e. frames with length not divisible by 8.
- `TX carriers` is a number of packets that experienced loss of carriers.
- `TX collisions` is a number of transmitted packets that experienced Ethernet collisions.
- `interrupt` it is the interrupt number used by NIC.

(b) Options that can be provided with `ifconfig` are `-a, -s, -v, interface_name, up, down, mtu N, dstaddr addr, netmask addr, add addr, del addr, media type, address`.

- `-a`: display all interfaces which are currently available, even if down. `ifconfig -a`.
- `-s`: display a short list. `ifconfig -s`.
- `-v`: be more verbose for some error conditions. `ifconfig -v`.
- `up`: This flag causes the interface to be activated. `ifconfig interface up`.

(c) The output of the kernel routing table is organized in the following columns

- `Destination`: The destination network or destination host.
- `Gateway`: The gateway address or '*' if none set.
- `Genmask`: The netmask for the destination net; '255.255.255.255' for a host destination and '0.0.0.0' for the default route.
- `Flags`: Possible flags include: U (route is up), H (target is a host), G (use gateway).
- `Metric`: The 'distance' to the target (usually counted in hops).
- `Ref`: Number of references to this route. (Not used in the Linux kernel.)
- `Use`: Count of lookups for the route.
- `Iface`: Interface to which packets for this route will be sent.
- `MSS`: Default maximum segment size for TCP connections over this route.

(d)

- `-n`: show numerical addresses instead of trying to determine symbolic host names. This is useful if you are trying to determine why the route to your name-server has vanished.

```
vijayp@d-vin-u:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         172.16.112.1    0.0.0.0         UG    100    0        0 eno1
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 eno1
172.16.112.1    0.0.0.0         255.255.255.255 UH    20100  0        0 eno1
172.16.115.128  0.0.0.0         255.255.255.128 U     100    0        0 eno1
```

Figure 2: route -n output

- `-C`: operate on the kernel's routing cache.

```
vijayp@d-vin-u:~$ route -Cn
Kernel IP routing cache
Source          Destination     Gateway         Flags Metric Ref    Use Iface
```

Figure 3: route -C output

- `-ee`: will generate a very long line with all parameters from the routing table.

```
vijayp@d-vin-u:~$ route -ee
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface    MSS   Window irtt
default         _gateway        0.0.0.0         UG    100    0        0 eno1     0     0      0
link-local      0.0.0.0         255.255.0.0     U     1000   0        0 eno1     0     0      0
_gateway        0.0.0.0         255.255.255.255 UH    20100  0        0 eno1     0     0      0
172.16.115.128  0.0.0.0         255.255.255.128 U     100    0        0 eno1     0     0      0
```

Figure 4: route -ee output

- `add (del)`: add a new route. (delete a route)

```
vijayp@d-vin-u:~$ sudo route add default gw 172.16.112.1
vijayp@d-vin-u:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         172.16.112.1    0.0.0.0         UG    0      0        0 eno1
0.0.0.0         172.16.112.1    0.0.0.0         UG    100    0        0 eno1
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 eno1
172.16.112.1    0.0.0.0         255.255.255.255 UH    20100  0        0 eno1
172.16.115.128  0.0.0.0         255.255.255.128 U     100    0        0 eno1
vijayp@d-vin-u:~$ sudo route del default gw 172.16.112.1
vijayp@d-vin-u:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         172.16.112.1    0.0.0.0         UG    100    0        0 eno1
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 eno1
172.16.112.1    0.0.0.0         255.255.255.255 UH    20100  0        0 eno1
172.16.115.128  0.0.0.0         255.255.255.128 U     100    0        0 eno1
```

Figure 5: route add and delete output

4 *netstat* Command

(a) By default, netstat displays a list of open sockets (tcp, udp). If we don't specify any address families, then the active sockets of all configured address families will be printed.

(b) Option `-at` where it will list all listening and non listening tcp ports.

```
vijayp@d-vin-u:~$ netstat -at
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN
tcp        0      0 localhost:ipp           0.0.0.0:*               LISTEN
tcp        0      0 d-vin-u:46420           ec2-44-236-127-11:https ESTABLISHED
tcp        0      0 d-vin-u:39908           ec2-52-33-84-190.:https ESTABLISHED
tcp        0      0 d-vin-u:36560           93.186.120.34.bc.:https ESTABLISHED
tcp        0      0 d-vin-u:36558           93.186.120.34.bc.:https ESTABLISHED
tcp        0      0 d-vin-u:60566           si-in-f188.1e100.n:5228 ESTABLISHED
tcp        0      0 d-vin-u:32894           server-13-227-141:https ESTABLISHED
tcp        0      0 d-vin-u:36902           140.227.186.35.bc:https ESTABLISHED
tcp6       0      0 ip6-localhost:ipp       [::]:*                  LISTEN
```

Figure 6: netstat -at command output

- `Proto`: The protocol (tcp, udp, udpl, raw) used by the socket.
- `Recv-Q`: The count of bytes not copied by the user program connected to this socket.
- `Send-Q` The count of bytes not acknowledged by the remote host.
- `Local Address` Address and port number of the local end of the socket.
- `Foreign Address` Address and port number of the remote end of the socket. Analogous to "Local Address".
- `State` The state of the socket. Normally this can be one of several values: ESTABLISHED (established connection). LISTEN (listening for incoming connections).

(c) *netstat -r* display the kernel routing tables. `netstat -r and route -e` produce the same output.

```
vijayp@d-vin-u:~$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
default         _gateway        0.0.0.0         UG        0 0          0 eno1
link-local      0.0.0.0         255.255.0.0     U         0 0          0 eno1
_gateway        0.0.0.0         255.255.255.255 UH        0 0          0 eno1
172.16.115.128  0.0.0.0         255.255.255.128 U         0 0          0 eno1
```

Figure 7: netstat -r command output

- `Destination`: The destination network or destination host.

- `Gateway`: The gateway address or '*' if none set.
- `Genmask`: The netmask for the destination net; '255.255.255.255' for a host destination and '0.0.0.0' for the default route.
- `Flags`: Possible flags include: U (route is up), H (target is a host)
- `MSS`: Default maximum segment size for TCP connections over this route.
- `Window`: Window Default window size for TCP connections over this route.
- `irtt`: Initial RTT (Round Trip Time).
- `Iface`: Interface to which packets for this route will be sent.

(d) Option `-i` or `netstat -ie` can be used to display extended information on the interfaces. Currently, there are 3 network interfaces running in my computer. **eno1, lo and wlp2s0** are the names of the my system active network interfaces.

(e) `netstat` with option `-su` can be used to show the statistics of all UDP connections.

```
vijayp@d-vin-u:~$ netstat -su
IcmpMsg:
    InType3: 43
    InType8: 3
    OutType0: 3
    OutType3: 70
Udp:
    37258 packets received
    67 packets to unknown port received
    504 packet receive errors
    6488 packets sent
    504 receive buffer errors
    0 send buffer errors
    IgnoredMulti: 75
UdpLite:
IpExt:
    InMcastPkts: 61
    OutMcastPkts: 83
    InBcastPkts: 75
    InOctets: 164278759
    OutOctets: 1885734
    InMcastOctets: 6070
    OutMcastOctets: 10044
    InBcastOctets: 6274
    InNoECTPkts: 124657
MPTcpExt:
```

Figure 8: netstat -su command output

(f) `lo` is the loopback interface. This is a special network interface that the system uses to communicate with itself and to represent the loopback facility. Any traffic that a computer program sends on the loopback network is addressed to the same computer. The most commonly used IP address on the loopback network is 127.0.0.1/8 for IPv4 and ::1 for IPv6. The standard domain name for the address is localhost. It is used mainly for diagnostics and troubleshooting, and to connect to servers running on the local machine

```
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 5262  bytes 608805 (608.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 5262  bytes 608805 (608.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 9: loopback interface

5 *traceroute* Tool

traceroute tracks the route packets taken from an IP network on their way to a given host. It utilizes the IP protocol's time to live (TTL) field and attempts to elicit an ICMP TIME_EXCEEDED response from each gateway along the path to the host.

Table 3: Hosts vs Hope Count in each time slot (22-March-2022)

Hosts	HC_1 (13:00)	HC_2 (18:00)	HC_3 (23:00)
BookMyShow	10	10	10
YouTube	19	19	19
Twitter	15	15	15
Spotify	11	11	11
Instagram	14	13	11
CodeForces	19	16	16

(a) Most common hops between two routes are: _gateway (192.168.244.90), 192.168.25.231 (192.168.25.231), dsl-tn-dynamic-221.222.22.125.airtelbroadband.in.

(b) The reason for different route every time can be: As `traceroute` to a router address uses process switching (checking the table and making a decision) for every packet, that is influenced by load balancing (distribute traffic among all its connections), and we have every time a different path. There can be case also where destination router has two or more connections to different ISPs and the connections are doing load balancing. Another possible reason can be a roter might just be misconfigured.

(c) There are different cases why we don't see a complete paths: Firewall might have blocked the ICMP protocol, Firewall might have blocked the UDP Ports for tracing hosts, Router might be busy routing packets and not have the resources to send out the ICMP packets, The interface receiving the ICMP TTL exceeded messages does not have routing back to that address then its not displayed in the traceroute results.

(d) Yes, it is possible to find the route to certain hosts by using the option `-T`. If some filters are present in the network path, then most probably any "unlikely" udp ports (as for default method) or even icmp echoes (as for icmp) are filtered, and whole tracerouting will just stop at such a firewall.

6 Network Addresses

(a) `arp`: arp with no mode specifier will print the current content of the table. `arp -a`: Use alternate BSD style output format (with no fixed columns). `arp -e`: Use default Linux style output format (with fixed columns).
Table Column Description:

- `Address`: The IPv4 address of the machine.
- `HWtype`: The type of connection. In my system it is through ethernet

- `HWaddress`: The MAC address of the machine.
- `Flags Mask`: It tells the address is manually set, learned, published (announced by another node than the requested) or is incomplete. **C**: It tells that entries are dynamically learned by `arp` protocol. Each complete entry in the ARP cache will be marked with the C flag. **M**: It tells that entries have been manually entered/added in the memory instead of dynamically learned from `arp` protocol. Permanent entries are marked with M. **P**: (Published) It tells the host to respond to packets which are ARP request and ARP response.
- `Iface`: It is the interface name.

(b) If network is reachable, arp can be used to make a static entry into the ARP table, otherwise it will show network is unreachable. Similarly, deletion of entry will remove it from arp table. `add entry`: To add entry into ARP table, we have to use option `arp -s address hw_addr`. The format of the hw_addr parameter is dependent on the hardware class, but for most classes one can assume that the usual presentation can be used. `delete entry`: To delete entry from ARP table, we have to use option `arp -d address`. Root or net-admin privilege is required to do this. (Fig 10).

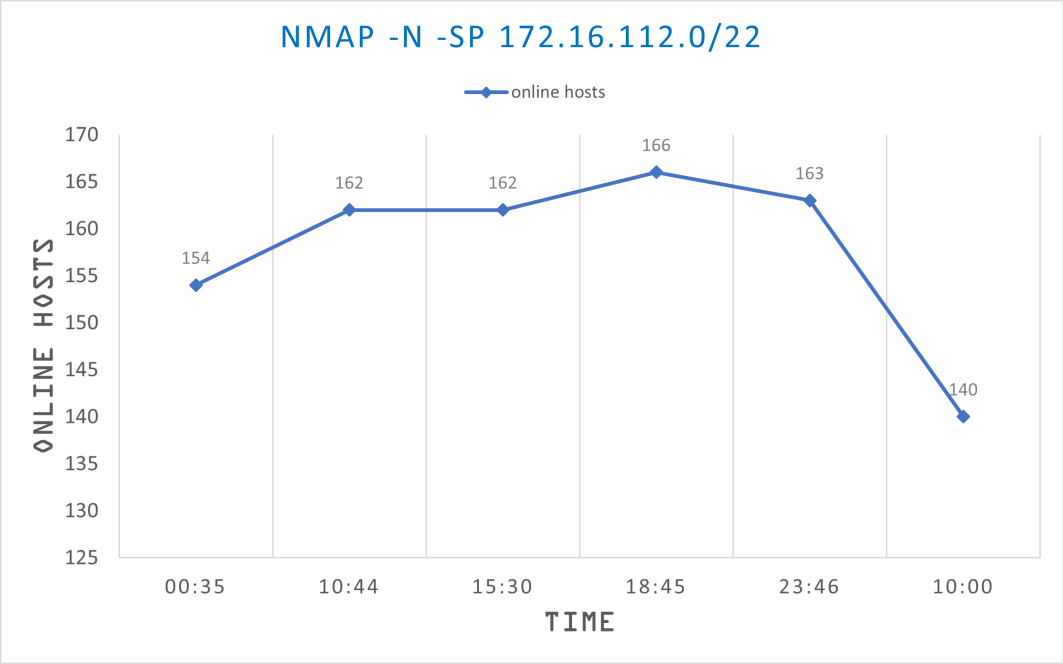
```
vijayp@d-vin-u:~$ sudo arp -s 172.16.115.170 00:03:0f:1d:ab:40
vijayp@d-vin-u:~$ sudo arp -s 172.16.115.171 00:03:0f:1d:ab:41
vijayp@d-vin-u:~$ sudo arp -s 172.16.115.172 00:03:0f:1d:ab:42
vijayp@d-vin-u:~$ sudo arp -s 172.16.115.173 00:03:0f:1d:ab:43
vijayp@d-vin-u:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
172.16.112.33            ether   00:03:0f:1d:ab:58   C                     eno1
172.16.115.171           ether   00:03:0f:1d:ab:41   CM                    eno1
172.16.112.66            ether   00:03:0f:1a:fc:38   C                     eno1
172.16.112.43            ether   00:03:0f:1a:fd:06   C                     eno1
172.16.112.31            ether   00:03:0f:1a:fc:ea   C                     eno1
172.16.112.48            ether   00:03:0f:1b:60:fe   C                     eno1
172.16.112.58            ether   00:03:0f:1a:fd:00   C                     eno1
172.16.112.36            ether   00:03:0f:1d:ab:26   C                     eno1
172.16.112.46            ether   00:03:0f:1d:ac:10   C                     eno1
172.16.112.29            ether   00:03:0f:1d:aa:d4   C                     eno1
172.16.112.51            ether   00:03:0f:1b:60:bc   C                     eno1
172.16.112.34            ether   00:03:0f:1d:ab:fe   C                     eno1
172.16.112.39            ether   00:03:0f:1a:ff:c4   C                     eno1
172.16.115.172           ether   00:03:0f:1d:ab:42   CM                    eno1
172.16.112.44            ether   00:03:0f:1d:ac:0c   C                     eno1
172.16.112.27            ether   00:03:0f:1d:ab:c4   C                     eno1
172.16.112.49            ether   00:03:0f:1b:61:22   C                     eno1
172.16.112.54            ether   00:03:0f:1a:ff:70   C                     eno1
172.16.115.170           ether   00:03:0f:1d:ab:40   CM                    eno1
172.16.112.37            ether   00:03:0f:1d:ab:32   C                     eno1
172.16.112.42            ether   00:03:0f:1d:ab:f6   C                     eno1
172.16.112.47            ether   00:03:0f:1d:ab:36   C                     eno1
172.16.112.30            ether   00:03:0f:1d:ab:f0   C                     eno1
172.16.112.35            ether   00:03:0f:1d:ab:1a   C                     eno1
172.16.115.173           ether   00:03:0f:1d:ab:43   CM                    eno1
```

Figure 10: arp add command output

(c) ARP supports a range of `/proc` interfaces to configure parameters on a global or per-interface basis (*/proc/sys/net/ipv4/neigh/*/** files). At some point between `base_reachable_time/2 and 3base_reachable_time/2`, the entry will still be in the cache, but it will be marked with a state of STALE. If the entry hasn't been used and is stale for `gc_stale_time` seconds, it should be eligible to be removed. If `gc_stale_time` passed and marked the entry as okay to be removed, it will be removed when the garbage collector runs (usually after `gc_interval` seconds). The garbage collector for the route cache only expires entries every 5 minutes (*gc_timeout* sec) on a lot of kernels.
This means the neighbor entry will have to be marked as stale (maybe 30 seconds, depending on `base_reachable_time`), then 5 minutes will have to go by before the route cache stops referencing the entry, followed by some combination of `gc_stale_time` and `gc_interval` passing before it actually gets cleaned up.

(d) In order for a network device to be able to communicate, the MAC Address it is using must be unique. No other device on that local network subnet can use that MAC Address. If two devices have the same MAC Address neither computer can communicate properly. It then depends on how the routers and systems on the network are configured. On an Ethernet LAN, this will cause a high number of collisions. We can start seeing unreachable host errors due to the collisions as well. The overall results really do vary depending on when the duplicate machine is coming online and how the current infrastructure is setup to handle such items.

7 Local Network Analysis



Time	Hosts
00:35	154
10:44	162
15:30	162
18:45	166
23:46	163
10:00	140

Figure 11: Local Network Analysis. SubNet: **172.16.112.0/22** and Hosts Online