

Data Mining in DHCP Security

D. Radha
M.O.P. Vaishnav
College for Women
Chennai

R. Jayaparvathy
M.O.P. Vaishnav
College for Women
Chennai

M. Shanmughi
M.O.P. Vaishnav
College for Women
Chennai

V. Jothilakshmi
M.O.P. Vaishnav
College for Women
Chennai

ABSTRACT

A dynamic IP address is an IP address that is dynamically assigned to your computer by your Internet service provider (ISP). Once your computer or router is refreshed, your ISP dynamically assigns an IP address to your networking device using DHCP protocol. DHCP follows sequence in assigning an IP address to the host. If a particular host find its IP address it can easily know another host's ip address through random search. This paper concentrates on giving an insight about DHCP security.

Keywords

Data mining, data warehousing, dynamic IP address, network, DHCP, ping.

1. INTRODUCTION

1.1 Wire Shark

Applying data mining in wire shark is to categorize the traffic flows based on packets moving from one place to another place. It is also used for running multimedia-rich, real and non-real time applications. This tool is designed for line sniffing data trunk (with prior permission) that is serving a network with tens of desktops, laptops, controllers and access points (AP's).

- Wire shark is the world's most popular network analyzer with over 50,000 downloads per month.
- It's available at a free of cost.
- Created by Gerald combs under the original name Ethereal, wire shark is maintained by a dedicated group of core developers.
- It is used in many industries and educational institutions.
- It is the continuation of a project that started in 1998.

1.2 Some of the wire shark features

- Deep inspection of hundreds of protocols, with more being added all the time.
- Live capture and offline analysis.
- Standard three-pane packet browser.
- Multi-platform:- Runs on windows, Linux, OS, Solaris, free BSD, net BSD and many others.
- Captured network data can be browsed via a GUI or via the TTY-mode T shark utility.
- The most powerful display filters in the industry.
- Rich VOIP analysis.
- Reads/writes many different capture file formats:

TCP dump (lib cap), PCAP NG, catapult DC T2000, CISCO secure IDS ip log, Microsoft network monitor, Network general sniffer (compressed and uncompressed), sniffer pro and net x-ray, network instrument observer, net screen snoop Novel LAN analyzer, RADCAM WAN/LAN analyzer Shomiti / finiser surveyor, Tektronix k12xx, visual networks, visual uptime, wild packets ether peek, Token peek/ AIRO peek and many others.

- Captured files can be compressed with GZIP and also can be decompressed on the fly.
- Live data can be read from Ethernet, IEEE 802.11, PPP/HDLC, ATM, Bluetooth, USB, Token ring, frame relay, FDDI and others (depending upon your platform).
- Decryption support for many protocols, including IP sec, ISAKMP, Kerberos, SNMPv3, SSL/TLS, WEP and WPA/WPA2.
- Coloring rules can be applied to the packet list for quick, intuitive analysis.
- Output can be exported to XML, POST SCRIPT, CSV or PLAIN TEXT.

1.3 Functionality of wire shark

- Wire shark has a graphical front end plus some integrated sorting and filtering options.
- Wire shark lets the user put network interface controllers that support promiscuous, so configured address and broadcast/multicast traffic.
- However, when capturing with a packet analyzer in promiscuous mode on a port on a network switch, not all traffic through the switch is necessarily sent to the port where the capture is done, so capturing in promiscuous mode is not necessarily sufficient to see all network traffic.
- Port mirroring or various network taps extend capture to any point on the network.

1.4 Features

- Data can be captured "from the wire" from a live network connection or read from a file of already captured packets.

Wire shark is a software that "understands" the structure (encapsulation) of different networking protocols. It can parse and display the fields, along with their meanings as specified by different networking protocols. Wire shark uses PCAP to capture packets, so it can only capture packets on the types of networks that PCAP supports.

- {PCAP – packet capture

In the field of computer network administration, PCAP consists of application programming interface (API) for capturing network traffic}.

1.5 Wire shark in data mining

- There are numerous packet classification categories based on how packets are decoded and analyzed including the consideration of packet size , duration ,patterns of transmissions and bursts.
- Live data help the team leader/manager to check their co-workers whether their work going properly.
- Past data helps business peoples for business purposes and also for security purposes.

1.6 Dynamic IP address

Dynamic Host Configuration protocol, a protocol for assigning dynamic IP addresses to devices on a network. With dynamic addressing, a device can have a different IP address every time it connects to the network. In some systems, the device's IP address can even change while it is still connected.

1.7 DORA

When one installs a DHCP server into our network then DHCP server works on the basis of DORA process, first DHCP sends a hello message in to the network to discover the clients pc and when any client pc selects any IP from DHCP server provide that IP to client pc and both send acknowledgement to each other. This process is called DORA process.

2. LITERATURE REVIEW

2.1 International journal of advanced technology

Traffic engineering: Traffic engineering is the concept referring to a systematic process where traffic flows are arranged into “classified” groups for simplifying the transmission throughput networks and decreasing the change of congestions. It inherently capable of dealing with large volume of traffic through traffic aggregation. Traffic flow exhibit volatility some traffic flow specifications, such as: bandwidth and volume. Fluctuations and unsteadiness of the traffic reduce the efficiency of Traffic engineering in the process of data mining. Some reasons are link exhaustion and connection termination, bandwidth fluctuations, burst effect etc.,.

3. WIRE SHARK IN DATA MINING (PDF)

3.1 Wire shark in data mining and data warehousing

Wire shark can be used for both data mining and data warehousing. It collects the net traffic is called data warehousing (i.e. Browser history). Depending upon the data one can need browser and filtering the data based upon it is called data mining. It has a filter option which is used for mining. It contains field name, relation, value and range. Another packet like filter option is finding packets. One can easily find packets once one can have captured some packets or have read in a previously saved capture file. Find packets can be found in edit menu. It contains display filter, hex value, string, up and down.

Go to first packet:

This helps us to find the first packet.

Go to last packet.

Data can be collected if the network is not secured else one can collect it using the port numbers.

3.2 Time Display Format And Time References

While capturing packets it getting time stamped. These will be saved to the capture file, so they will be available for later analysis.

3.3 Some of the presentation formats are

- Date and time of day
- Time and day
- Seconds since beginning of capture
- Seconds since previous of capture packet
- Seconds since displayed of capture packet
- Seconds since epoch

The available precisions are automatic, seconds, decisions, cent seconds, milliseconds, microseconds and nano seconds.

4. PROPOSED APPROACH

IP address of our PC can be found simply by ‘ipconfig/all’. Through this one can find what ip address is used in our PC and by knowing another host ip address one can simply ping that ip address. If there is a proper connection then ping process will be succeed. Eg. ‘ping 192.168.16.15’. Enable wire shark before the ping process. Wire shark starts the capturing process. When ping process get succeeded one can get the mac address by applying filters. Wire shark filter the mac addresses.

One can virtually test this process by using GNS and Wire shark by connecting two PC through modem or switch and give configuration wherever necessary. Ping and capture the process as wire shark can work on both real and non real time.

In this approach one can find the MAC address for the host which one can know already by ping process.

4.1 System Model

DHCP allocates IP address dynamically to the pc's or mobiles or any wireless or wired networks.. But this is not a permanent address like MAC address or ip address used in static network. If one can watched closely the ip address follows a sequence like the 192.168.1.11, 192.168.1.12, 192.168.1.13 or 192.168.1.2, 192.168.1.4, 192.168.1.5 or 192.168.1.3, 192.168.1.6, 192.168.1.9. Dynamic networks are mostly used in local places like home, small shops. Static network is used in factories or colleges cause it is a dedicated network , it will be of more worth only when use the internet for which one paid or else it will be in vain and it is not cost effective.

Dynamic network is not a dedicated network but it will supply network based upon our budget. As the DHCP allocates the IP address automatically while rebooting and there is less security. Here data mining is applied using Wire shark.

Enable the wire shark before ping. First find our ip address then ping the ip address randomly like in a sequential order or with the difference of 2,3,4,5,.....10 etc.,. If our wireless

network can connect 5 systems then one can find at least 3 out of 5 devices MAC address.

The top screenshot shows a web browser window displaying the Reliance Wi-Pod status page. The page includes a navigation bar with 'Status', 'Activate', 'Messages', 'Settings', and 'Help'. The 'Status' section is active, showing 'Basic Status' and 'System Information'. The 'System Information' section displays the following details:

- Software Version: AC3633_V2_Reliance_YR9AD914_R1
- MEID: A000004E5AE64F
- Hardware Version: AC3633_V2MB_A

The 'Wi-Fi Status' section displays the following details:

- Current Profile: WPA2-PSK
- SSID: Reliance Wi-Pod-A81
- IP Address: 192.168.0.1
- Subnet Mask: 255.255.255.0
- Number of Users: 5 / 5
- Connection Speed: Up to 14.7Mbps

The 'Terminals' section displays a table with the following data:

IP Address	Device Name	MAC Address	Time
192.168.0.101	Janu	9C:35:EB:68:DE:6E	Expired
192.168.0.102	android-4a41114609dbe45e	80:01:84:3C:06:F2	Expired
192.168.0.103	android-ada098f549f2de26	90:68:C3:6A:D2:20	23:11:17
192.168.0.104	android-a512-f0321edbd47	8C:3A:EA:80:21:6C	23:11:18

The bottom screenshot shows a command prompt window displaying the output of the 'ipconfig /all' command. The output shows the following details for the Ethernet adapter Local Area Connection 2:

- Host Name: Admin-PC
- Primary Dns Suffix: .
- Node Type: Hybrid
- IP Routing Enabled: No
- WINS Proxy Enabled: No
- Connection-specific DNS Suffix: .
- Description: Remote NDIS based Internet Sharing Device
- Physical Address: 34-4B-50-B7-EF-62
- DHCP Enabled: Yes
- Autoconfiguration Enabled: Yes
- Link-local IPv6 Address: fe80::31d2:b01d:46b2:41b4%13(Preferred)
- IPv4 Address: 192.168.0.100(Preferred)
- Subnet Mask: 255.255.255.0
- Lease Obtained: Monday, February 08, 2016 10:37:08 PM
- Lease Expires: Tuesday, February 09, 2016 10:37:07 PM
- Default Gateway: 192.168.0.1
- DHCP Server: 192.168.0.1
- DHCPv6 IAID: 305417040
- DHCPv6 Client DUID: 00-01-00-01-1D-58-03-68-F0-DE-F1-DB-E1-33
- DNS Servers: 192.168.0.1
- NetBIOS over Tcpip: Enabled

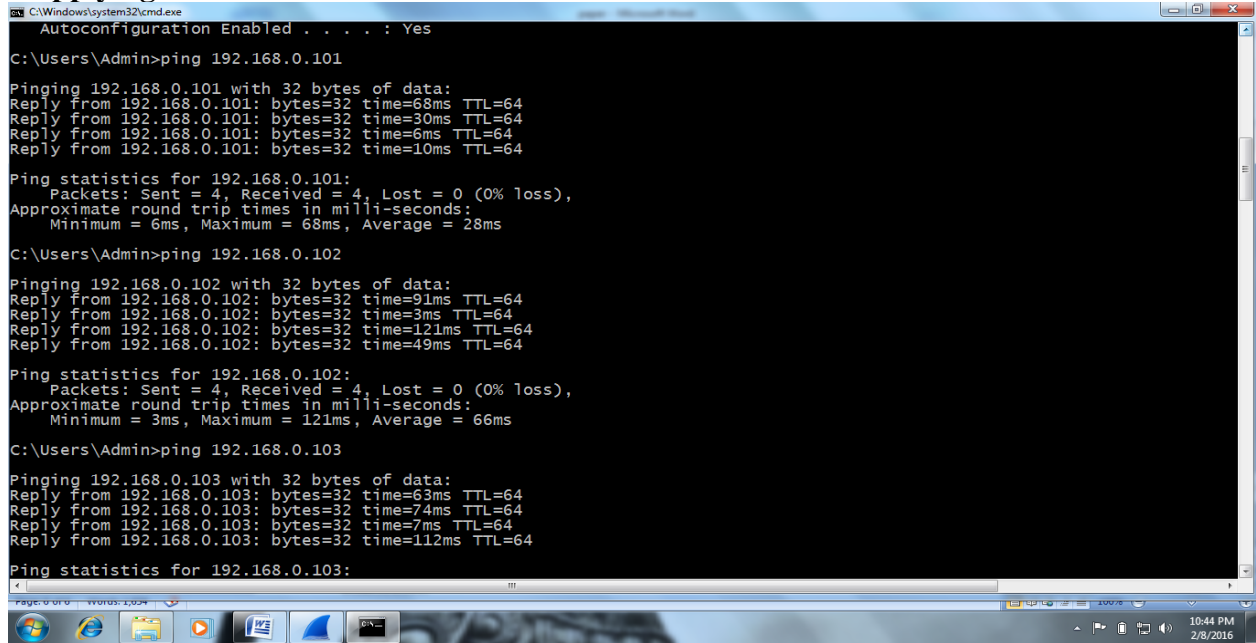
The output also shows details for the Ethernet adapter Local Area Connection:

- Media State: Media disconnected
- Connection-specific DNS Suffix: .
- Description: Realtek PCIe GBE Family Controller
- Physical Address: F0-DE-F1-DB-E1-33
- DHCP Enabled: Yes
- Autoconfiguration Enabled: Yes

First I'm knowing my systems IP address. From this I'm applying a random approach like dhcp first allocates IP address where it is directly connected. This network can

support 5 devices and one PC or laptop. So from this I can guess how devices I can connect.

4.2 Applying in Reliance



```
C:\Windows\system32\cmd.exe
Autoconfiguration Enabled . . . : Yes
C:\Users\Admin>ping 192.168.0.101

Pinging 192.168.0.101 with 32 bytes of data:
Reply from 192.168.0.101: bytes=32 time=68ms TTL=64
Reply from 192.168.0.101: bytes=32 time=30ms TTL=64
Reply from 192.168.0.101: bytes=32 time=6ms TTL=64
Reply from 192.168.0.101: bytes=32 time=10ms TTL=64

Ping statistics for 192.168.0.101:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 6ms, Maximum = 68ms, Average = 28ms

C:\Users\Admin>ping 192.168.0.102

Pinging 192.168.0.102 with 32 bytes of data:
Reply from 192.168.0.102: bytes=32 time=91ms TTL=64
Reply from 192.168.0.102: bytes=32 time=3ms TTL=64
Reply from 192.168.0.102: bytes=32 time=121ms TTL=64
Reply from 192.168.0.102: bytes=32 time=49ms TTL=64

Ping statistics for 192.168.0.102:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 3ms, Maximum = 121ms, Average = 66ms

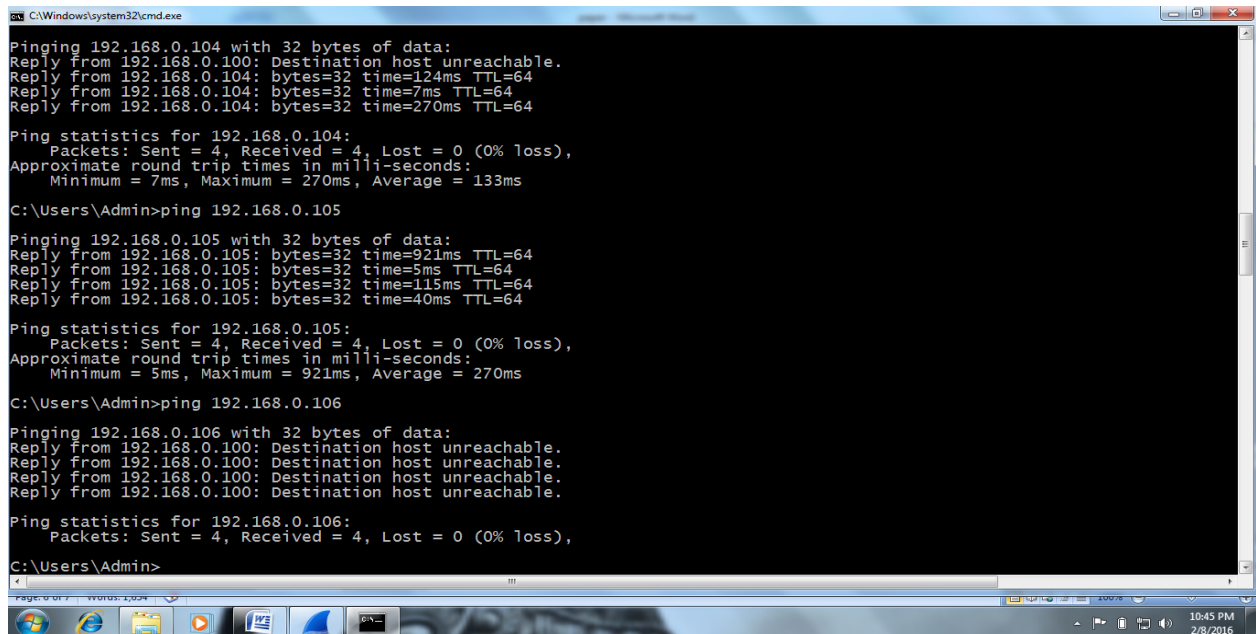
C:\Users\Admin>ping 192.168.0.103

Pinging 192.168.0.103 with 32 bytes of data:
Reply from 192.168.0.103: bytes=32 time=63ms TTL=64
Reply from 192.168.0.103: bytes=32 time=74ms TTL=64
Reply from 192.168.0.103: bytes=32 time=7ms TTL=64
Reply from 192.168.0.103: bytes=32 time=112ms TTL=64

Ping statistics for 192.168.0.103:
```

My system is the first device so it automatically allocates first ip address to my system. So first I am applying sequential order and it works so I started pingging 5 ip addresses

randomly. I not checked those devices whether it really have that ip address or not.



```
C:\Windows\system32\cmd.exe
Pinging 192.168.0.104 with 32 bytes of data:
Reply from 192.168.0.100: Destination host unreachable.
Reply from 192.168.0.104: bytes=32 time=124ms TTL=64
Reply from 192.168.0.104: bytes=32 time=7ms TTL=64
Reply from 192.168.0.104: bytes=32 time=270ms TTL=64

Ping statistics for 192.168.0.104:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 7ms, Maximum = 270ms, Average = 133ms

C:\Users\Admin>ping 192.168.0.105

Pinging 192.168.0.105 with 32 bytes of data:
Reply from 192.168.0.105: bytes=32 time=921ms TTL=64
Reply from 192.168.0.105: bytes=32 time=5ms TTL=64
Reply from 192.168.0.105: bytes=32 time=115ms TTL=64
Reply from 192.168.0.105: bytes=32 time=40ms TTL=64

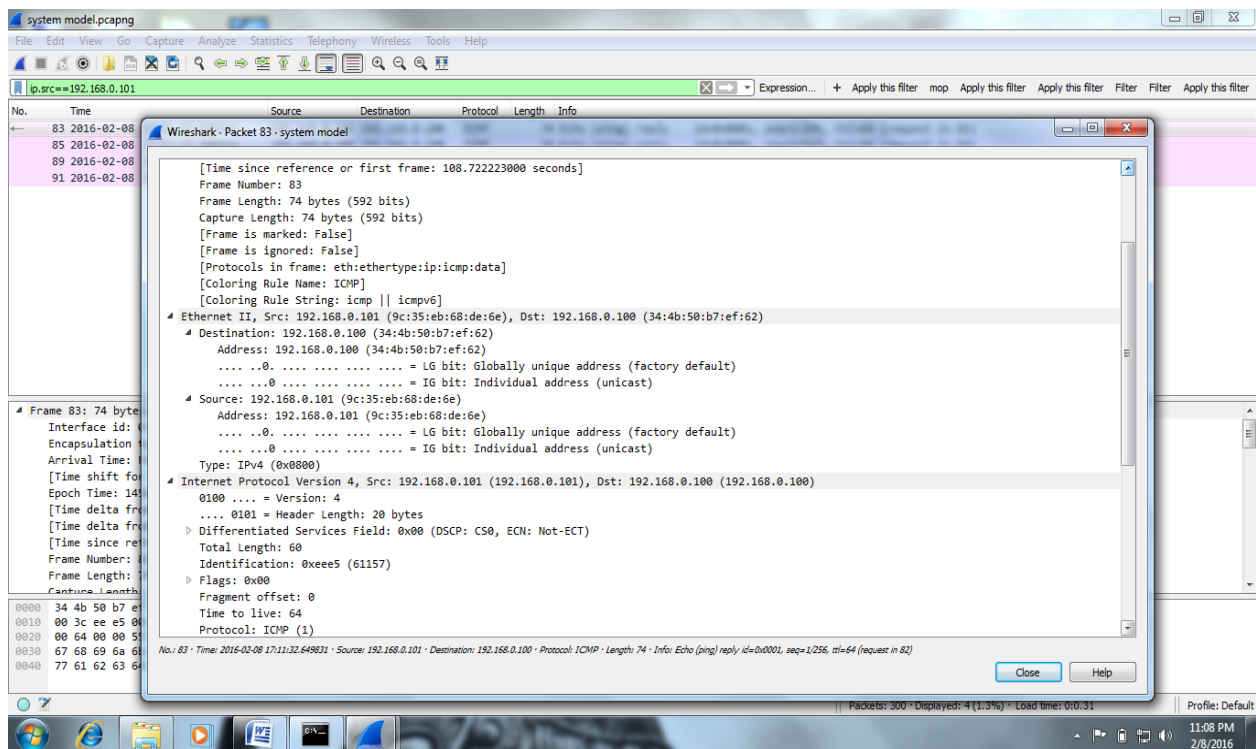
Ping statistics for 192.168.0.105:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 5ms, Maximum = 921ms, Average = 270ms

C:\Users\Admin>ping 192.168.0.106

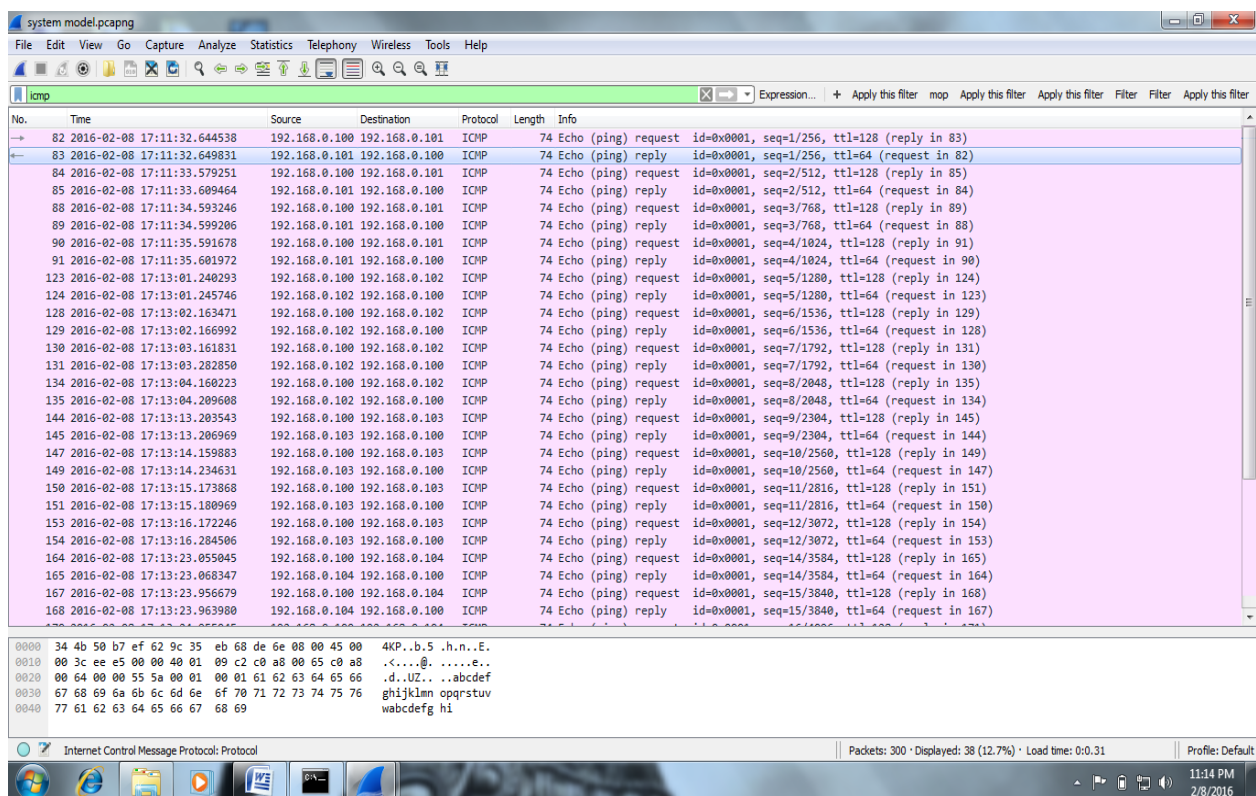
Pinging 192.168.0.106 with 32 bytes of data:
Reply from 192.168.0.100: Destination host unreachable.
Reply from 192.168.0.100: Destination host unreachable.
Reply from 192.168.0.100: Destination host unreachable.
Reply from 192.168.0.100: Destination host unreachable.

Ping statistics for 192.168.0.106:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

In the above picture after successful ping of 5 ip address and I'm trying to ping the sixth device where it not exist. So I'm getting a message like destination host unreachable.



The above picture is the result of the filtered result from wire shark.



The image displays two screenshots of the Wireshark network protocol analyzer interface, showing ICMP Echo (ping) traffic. The top screenshot shows a list of 183 packets, and the bottom screenshot shows a list of 189 packets. Both screenshots show a sequence of requests and replies between 192.168.0.101 and 192.168.0.102.

Top Screenshot (Packets 89-183):

No.	Time	Source	Destination	Protocol	Length	Info
89	2016-02-08 17:11:34.599206	192.168.0.101	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=3/768, ttl=64 (request in 88)
90	2016-02-08 17:11:35.591678	192.168.0.100	192.168.0.101	ICMP	74	Echo (ping) request id=0x0001, seq=4/1024, ttl=128 (reply in 91)
91	2016-02-08 17:11:35.601972	192.168.0.101	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=4/1024, ttl=64 (request in 90)
123	2016-02-08 17:13:01.240293	192.168.0.100	192.168.0.102	ICMP	74	Echo (ping) request id=0x0001, seq=5/1280, ttl=128 (reply in 124)
124	2016-02-08 17:13:01.245746	192.168.0.102	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=5/1280, ttl=64 (request in 123)
128	2016-02-08 17:13:02.163471	192.168.0.100	192.168.0.102	ICMP	74	Echo (ping) request id=0x0001, seq=6/1536, ttl=128 (reply in 129)
129	2016-02-08 17:13:02.166992	192.168.0.102	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=6/1536, ttl=64 (request in 128)
130	2016-02-08 17:13:03.161831	192.168.0.100	192.168.0.102	ICMP	74	Echo (ping) request id=0x0001, seq=7/1792, ttl=128 (reply in 131)
131	2016-02-08 17:13:03.282850	192.168.0.102	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=7/1792, ttl=64 (request in 130)
134	2016-02-08 17:13:04.160223	192.168.0.100	192.168.0.102	ICMP	74	Echo (ping) request id=0x0001, seq=8/2048, ttl=128 (reply in 135)
135	2016-02-08 17:13:04.209608	192.168.0.102	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=8/2048, ttl=64 (request in 134)
144	2016-02-08 17:13:13.203543	192.168.0.100	192.168.0.103	ICMP	74	Echo (ping) request id=0x0001, seq=9/2304, ttl=128 (reply in 145)
145	2016-02-08 17:13:13.206969	192.168.0.103	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=9/2304, ttl=64 (request in 144)
147	2016-02-08 17:13:14.159883	192.168.0.100	192.168.0.103	ICMP	74	Echo (ping) request id=0x0001, seq=10/2560, ttl=128 (reply in 149)
149	2016-02-08 17:13:14.234631	192.168.0.103	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=10/2560, ttl=64 (request in 147)
150	2016-02-08 17:13:15.173868	192.168.0.100	192.168.0.103	ICMP	74	Echo (ping) request id=0x0001, seq=11/2816, ttl=128 (reply in 151)
151	2016-02-08 17:13:15.180969	192.168.0.103	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=11/2816, ttl=64 (request in 150)
153	2016-02-08 17:13:16.172246	192.168.0.100	192.168.0.103	ICMP	74	Echo (ping) request id=0x0001, seq=12/3072, ttl=128 (reply in 154)
154	2016-02-08 17:13:16.284506	192.168.0.103	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=12/3072, ttl=64 (request in 153)
164	2016-02-08 17:13:23.055045	192.168.0.100	192.168.0.104	ICMP	74	Echo (ping) request id=0x0001, seq=14/3584, ttl=128 (reply in 165)
165	2016-02-08 17:13:23.068347	192.168.0.104	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=14/3584, ttl=64 (request in 164)
167	2016-02-08 17:13:23.956679	192.168.0.100	192.168.0.104	ICMP	74	Echo (ping) request id=0x0001, seq=15/3840, ttl=128 (reply in 168)
168	2016-02-08 17:13:23.963980	192.168.0.104	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=15/3840, ttl=64 (request in 167)
170	2016-02-08 17:13:24.955045	192.168.0.100	192.168.0.104	ICMP	74	Echo (ping) request id=0x0001, seq=16/4096, ttl=128 (reply in 171)
171	2016-02-08 17:13:25.225257	192.168.0.104	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=16/4096, ttl=64 (request in 170)
179	2016-02-08 17:13:29.485286	192.168.0.100	192.168.0.105	ICMP	74	Echo (ping) request id=0x0001, seq=17/4352, ttl=128 (reply in 182)
182	2016-02-08 17:13:29.529474	192.168.0.105	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=17/4352, ttl=64 (request in 179)
183	2016-02-08 17:13:29.619483	192.168.0.100	192.168.0.105	ICMP	74	Echo (ping) request id=0x0001, seq=18/4608, ttl=128 (reply in 184)

Bottom Screenshot (Packets 128-189):

No.	Time	Source	Destination	Protocol	Length	Info
128	2016-02-08 17:13:02.163471	192.168.0.100	192.168.0.102	ICMP	74	Echo (ping) request id=0x0001, seq=6/1536, ttl=128 (reply in 129)
129	2016-02-08 17:13:02.166992	192.168.0.102	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=6/1536, ttl=64 (request in 128)
130	2016-02-08 17:13:03.161831	192.168.0.100	192.168.0.102	ICMP	74	Echo (ping) request id=0x0001, seq=7/1792, ttl=128 (reply in 131)
131	2016-02-08 17:13:03.282850	192.168.0.102	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=7/1792, ttl=64 (request in 130)
134	2016-02-08 17:13:04.160223	192.168.0.100	192.168.0.102	ICMP	74	Echo (ping) request id=0x0001, seq=8/2048, ttl=128 (reply in 135)
135	2016-02-08 17:13:04.209608	192.168.0.102	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=8/2048, ttl=64 (request in 134)
144	2016-02-08 17:13:13.203543	192.168.0.100	192.168.0.103	ICMP	74	Echo (ping) request id=0x0001, seq=9/2304, ttl=128 (reply in 145)
145	2016-02-08 17:13:13.206969	192.168.0.103	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=9/2304, ttl=64 (request in 144)
147	2016-02-08 17:13:14.159883	192.168.0.100	192.168.0.103	ICMP	74	Echo (ping) request id=0x0001, seq=10/2560, ttl=128 (reply in 149)
149	2016-02-08 17:13:14.234631	192.168.0.103	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=10/2560, ttl=64 (request in 147)
150	2016-02-08 17:13:15.173868	192.168.0.100	192.168.0.103	ICMP	74	Echo (ping) request id=0x0001, seq=11/2816, ttl=128 (reply in 151)
151	2016-02-08 17:13:15.180969	192.168.0.103	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=11/2816, ttl=64 (request in 150)
153	2016-02-08 17:13:16.172246	192.168.0.100	192.168.0.103	ICMP	74	Echo (ping) request id=0x0001, seq=12/3072, ttl=128 (reply in 154)
154	2016-02-08 17:13:16.284506	192.168.0.103	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=12/3072, ttl=64 (request in 153)
164	2016-02-08 17:13:23.055045	192.168.0.100	192.168.0.104	ICMP	74	Echo (ping) request id=0x0001, seq=14/3584, ttl=128 (reply in 165)
165	2016-02-08 17:13:23.068347	192.168.0.104	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=14/3584, ttl=64 (request in 164)
167	2016-02-08 17:13:23.956679	192.168.0.100	192.168.0.104	ICMP	74	Echo (ping) request id=0x0001, seq=15/3840, ttl=128 (reply in 168)
168	2016-02-08 17:13:23.963980	192.168.0.104	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=15/3840, ttl=64 (request in 167)
170	2016-02-08 17:13:24.955045	192.168.0.100	192.168.0.104	ICMP	74	Echo (ping) request id=0x0001, seq=16/4096, ttl=128 (reply in 171)
171	2016-02-08 17:13:25.225257	192.168.0.104	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=16/4096, ttl=64 (request in 170)
179	2016-02-08 17:13:29.485286	192.168.0.100	192.168.0.105	ICMP	74	Echo (ping) request id=0x0001, seq=17/4352, ttl=128 (reply in 182)
182	2016-02-08 17:13:29.529474	192.168.0.105	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=17/4352, ttl=64 (request in 179)
183	2016-02-08 17:13:29.619483	192.168.0.100	192.168.0.105	ICMP	74	Echo (ping) request id=0x0001, seq=18/4608, ttl=128 (reply in 184)
184	2016-02-08 17:13:29.624725	192.168.0.105	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=18/4608, ttl=64 (request in 183)
186	2016-02-08 17:13:30.617888	192.168.0.100	192.168.0.105	ICMP	74	Echo (ping) request id=0x0001, seq=19/4864, ttl=128 (reply in 187)
187	2016-02-08 17:13:30.732773	192.168.0.105	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=19/4864, ttl=64 (request in 186)
188	2016-02-08 17:13:31.616255	192.168.0.100	192.168.0.105	ICMP	74	Echo (ping) request id=0x0001, seq=20/5120, ttl=128 (reply in 189)
189	2016-02-08 17:13:31.657115	192.168.0.105	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=20/5120, ttl=64 (request in 188)

The filtered process can be get by giving the command 'ICMP'. This will be helpful to get the information of both source and destination.

First think of how many devices it can accept. Then how many devices already got connected. Then one have to apply

guess based upon it. For example my device is connected as fifth and already some devices are working means then I'll be guessing reversely. If three devices working means both forward and reverse approach.

4.3 Applying in BSNL

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\User>ipconfig/all

Windows IP Configuration

Host Name . . . . . : hp
Primary Dns Suffix . . . . . :
Node Type . . . . . : Hybrid
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No

Wireless LAN adapter Local Area Connection* 2:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :
Description . . . . . : Microsoft Wi-Fi Direct Virtual Adapter
Physical Address. . . . . : 1C-3E-84-24-53-3B
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes

Ethernet adapter Ethernet:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :
Description . . . . . : Realtek PCIe FE Family Controller
Physical Address. . . . . : 74-46-A0-88-4F-F9
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes

Wireless LAN adapter Wi-Fi:

Connection-specific DNS Suffix . :
Description . . . . . : Ralink RT3290 802.11bgn Wi-Fi Adapter
Physical Address. . . . . : 1C-3E-84-24-53-39
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::9502:508f:94a3:9546x3(Preferred)
IPv4 Address. . . . . : 192.168.1.6(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Lease Obtained. . . . . : Saturday, February 13, 2016 5:42:26 PM
Lease Expires . . . . . : Sunday, February 14, 2016 5:42:26 AM
Default Gateway . . . . . : fe80::1ea5:32ff:fe25:b900x3
192.168.1.1
DHCP Server . . . . . : 192.168.1.1
DHCPv6 Iaid . . . . . : 52182660
DHCPv6 Client DUID. . . . . : 00-01-00-01-E-3D-EE-E6-74-46-A0-88-4F-F9

DNS Servers . . . . . : 192.168.1.1
NetBIOS over Tcpip. . . . . : Enabled

Tunnel adapter isatap.{50698D27-D9C6-4E32-8AF7-280CA21FEEC1}:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :
Description . . . . . : Microsoft ISATAP Adapter
```

```
Connection-specific DNS Suffix . :
Description . . . . . : Ralink RT3290 802.11bgn Wi-Fi Adapter
Physical Address. . . . . : 1C-3E-84-24-53-39
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::9502:508f:94a3:9546x3(Preferred)
IPv4 Address. . . . . : 192.168.1.6(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Lease Obtained. . . . . : Saturday, February 13, 2016 5:42:26 PM
Lease Expires . . . . . : Sunday, February 14, 2016 5:42:26 AM
Default Gateway . . . . . : fe80::1ea5:32ff:fe25:b900x3
192.168.1.1
DHCP Server . . . . . : 192.168.1.1
DHCPv6 Iaid . . . . . : 52182660
DHCPv6 Client DUID. . . . . : 00-01-00-01-E-3D-EE-E6-74-46-A0-88-4F-F9

DNS Servers . . . . . : 192.168.1.1
NetBIOS over Tcpip. . . . . : Enabled

Tunnel adapter isatap.{50698D27-D9C6-4E32-8AF7-280CA21FEEC1}:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :
Description . . . . . : Microsoft ISATAP Adapter
Physical Address. . . . . : 00-00-00-00-00-00-E0
DHCP Enabled. . . . . : No
Autoconfiguration Enabled . . . . : Yes

Tunnel adapter Teredo Tunneling Pseudo-Interface:

Connection-specific DNS Suffix . :
Description . . . . . : Microsoft Teredo Tunneling Adapter
Physical Address. . . . . : 00-00-00-00-00-00-E0
DHCP Enabled. . . . . : No
Autoconfiguration Enabled . . . . : Yes
IPv6 Address . . . . . : 2001:0:5ef5:79fb:341d:2379:3f57:fef9(Pref
erred)
Link-Local IPv6 Address . . . . . : fe80::341d:2379:3f57:fef9x19(Preferred)
Default Gateway . . . . . : ::
DHCPv6 Iaid . . . . . : 218767104
DHCPv6 Client DUID. . . . . : 00-01-00-01-E-3D-EE-E6-74-46-A0-88-4F-F9

NetBIOS over Tcpip. . . . . : Disabled

C:\Users\User>ping 192.168.1.7

Pinging 192.168.1.7 with 32 bytes of data:
Reply from 192.168.1.6: Destination host unreachable.
Reply from 192.168.1.6: Destination host unreachable.
Reply from 192.168.1.6: Destination host unreachable.
Reply from 192.168.1.6: Destination host unreachable.

Ping statistics for 192.168.1.7:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

```
Command Prompt

C:\Users\User>ping 192.168.1.5

Pinging 192.168.1.5 with 32 bytes of data:
Reply from 192.168.1.5: bytes=32 time=3235ms TTL=64
Reply from 192.168.1.5: bytes=32 time=5ms TTL=64
Reply from 192.168.1.5: bytes=32 time=1023ms TTL=64
Reply from 192.168.1.5: bytes=32 time=6ms TTL=64

Ping statistics for 192.168.1.5:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 5ms, Maximum = 3235ms, Average = 1067ms

C:\Users\User>ping 192.168.1.4

Pinging 192.168.1.4 with 32 bytes of data:
Reply from 192.168.1.4: bytes=32 time=926ms TTL=64
Reply from 192.168.1.4: bytes=32 time=3ms TTL=64
Reply from 192.168.1.4: bytes=32 time=87ms TTL=64
Reply from 192.168.1.4: bytes=32 time=44ms TTL=64

Ping statistics for 192.168.1.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 3ms, Maximum = 926ms, Average = 265ms

C:\Users\User>ping 192.168.1.3

Pinging 192.168.1.3 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.1.3:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\Users\User>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\User>ping 192.168.1.1

Pinging 192.168.1.1 with 32 bytes of data:
Reply from 192.168.1.1: bytes=32 time<1ms TTL=254
Reply from 192.168.1.1: bytes=32 time=1ms TTL=254
Reply from 192.168.1.1: bytes=32 time=33ms TTL=254
Reply from 192.168.1.1: bytes=32 time=33ms TTL=254

Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 33ms, Average = 0ms
```

```
Command Prompt

Approximate round trip times in milli-seconds:
    Minimum = 5ms, Maximum = 3235ms, Average = 1067ms

C:\Users\User>ping 192.168.1.4

Pinging 192.168.1.4 with 32 bytes of data:
Reply from 192.168.1.4: bytes=32 time=926ms TTL=64
Reply from 192.168.1.4: bytes=32 time=3ms TTL=64
Reply from 192.168.1.4: bytes=32 time=87ms TTL=64
Reply from 192.168.1.4: bytes=32 time=44ms TTL=64

Ping statistics for 192.168.1.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 3ms, Maximum = 926ms, Average = 265ms

C:\Users\User>ping 192.168.1.3

Pinging 192.168.1.3 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.1.3:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\Users\User>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\User>ping 192.168.1.1

Pinging 192.168.1.1 with 32 bytes of data:
Reply from 192.168.1.1: bytes=32 time<1ms TTL=254
Reply from 192.168.1.1: bytes=32 time=1ms TTL=254
Reply from 192.168.1.1: bytes=32 time=33ms TTL=254
Reply from 192.168.1.1: bytes=32 time=33ms TTL=254

Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 33ms, Average = 0ms

C:\Users\User>ping 192.168.1.0

Pinging 192.168.1.0 with 32 bytes of data:
Reply from 192.168.1.6: Destination host unreachable.
```

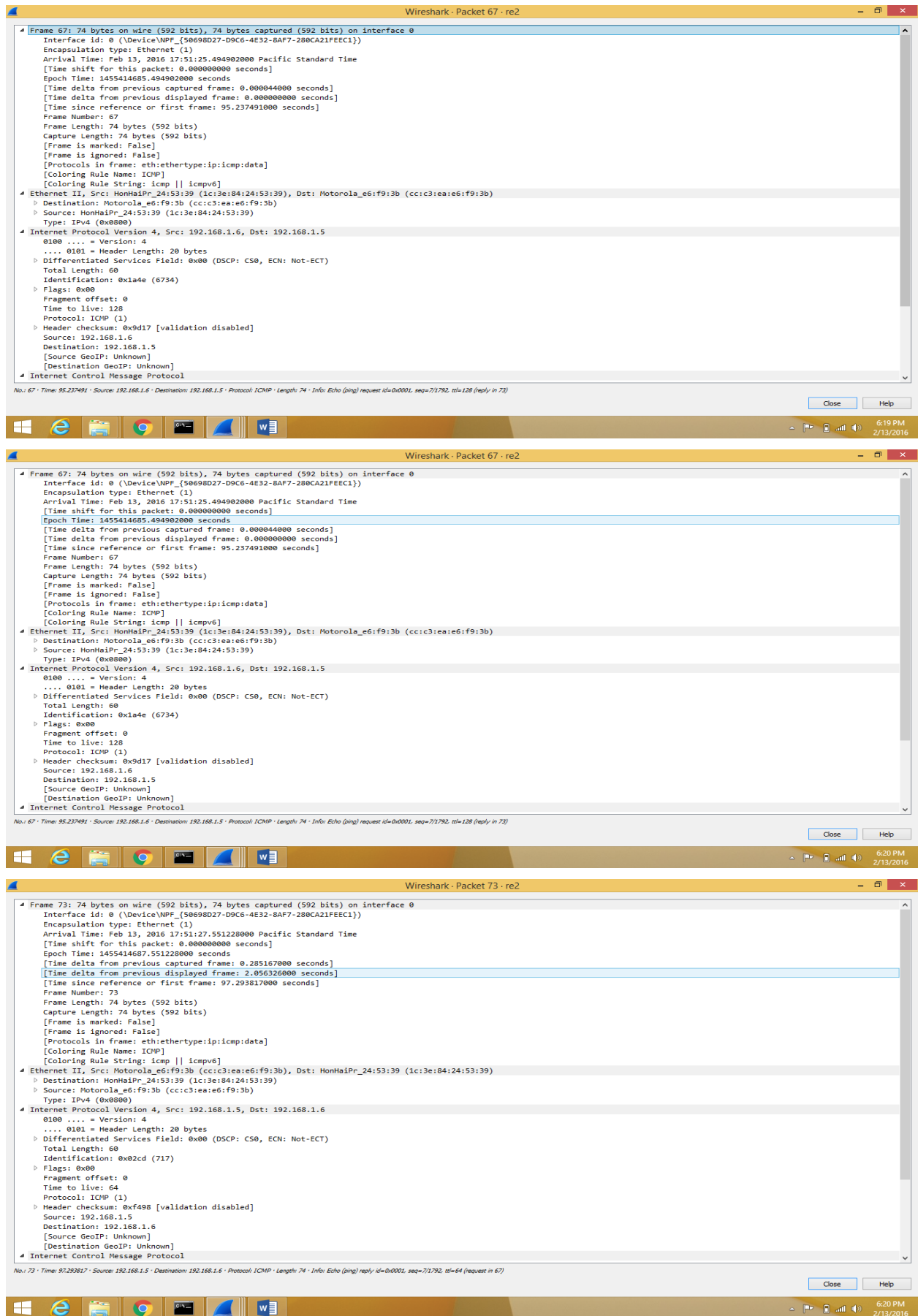

The screenshot displays a Windows desktop environment. In the foreground, a Command Prompt window is open, showing the results of several ping commands. The first ping is to 192.168.1.3, which fails with 100% loss. Subsequent pings to 192.168.1.2, 192.168.1.1, and 192.168.1.0 succeed. Pings to 192.168.1.6 and 192.168.1.7 fail with 'Destination host unreachable'.

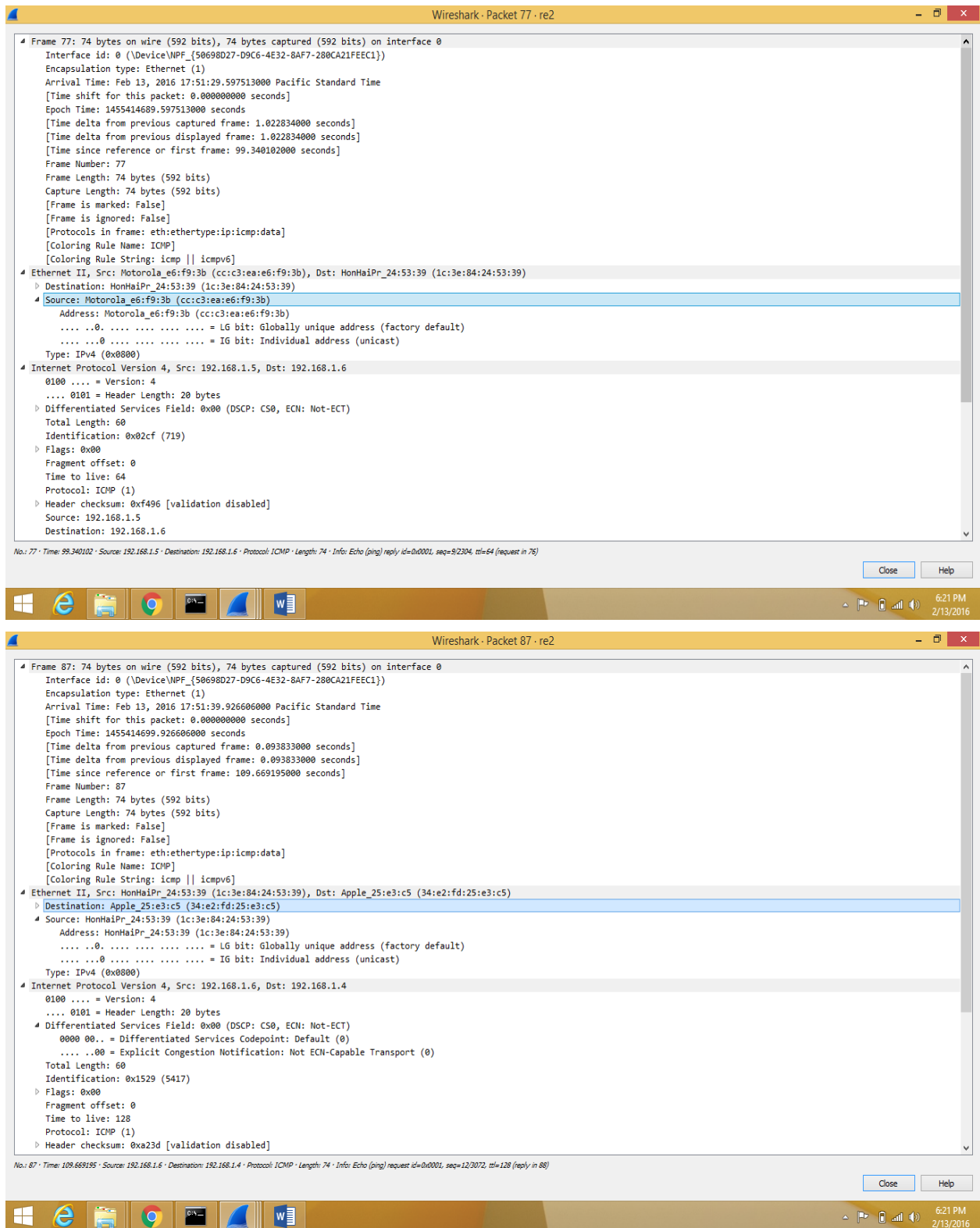
Behind the Command Prompt, a Microsoft Word document is partially visible. To the right, a packet capture analysis window titled 're2.pcapng' is open, displaying a list of captured packets. The packets are primarily ICMP Echo (ping) requests and replies between 192.168.1.1 and 192.168.1.6. The bottom of the window shows a hex dump of the selected packet (No. 111).

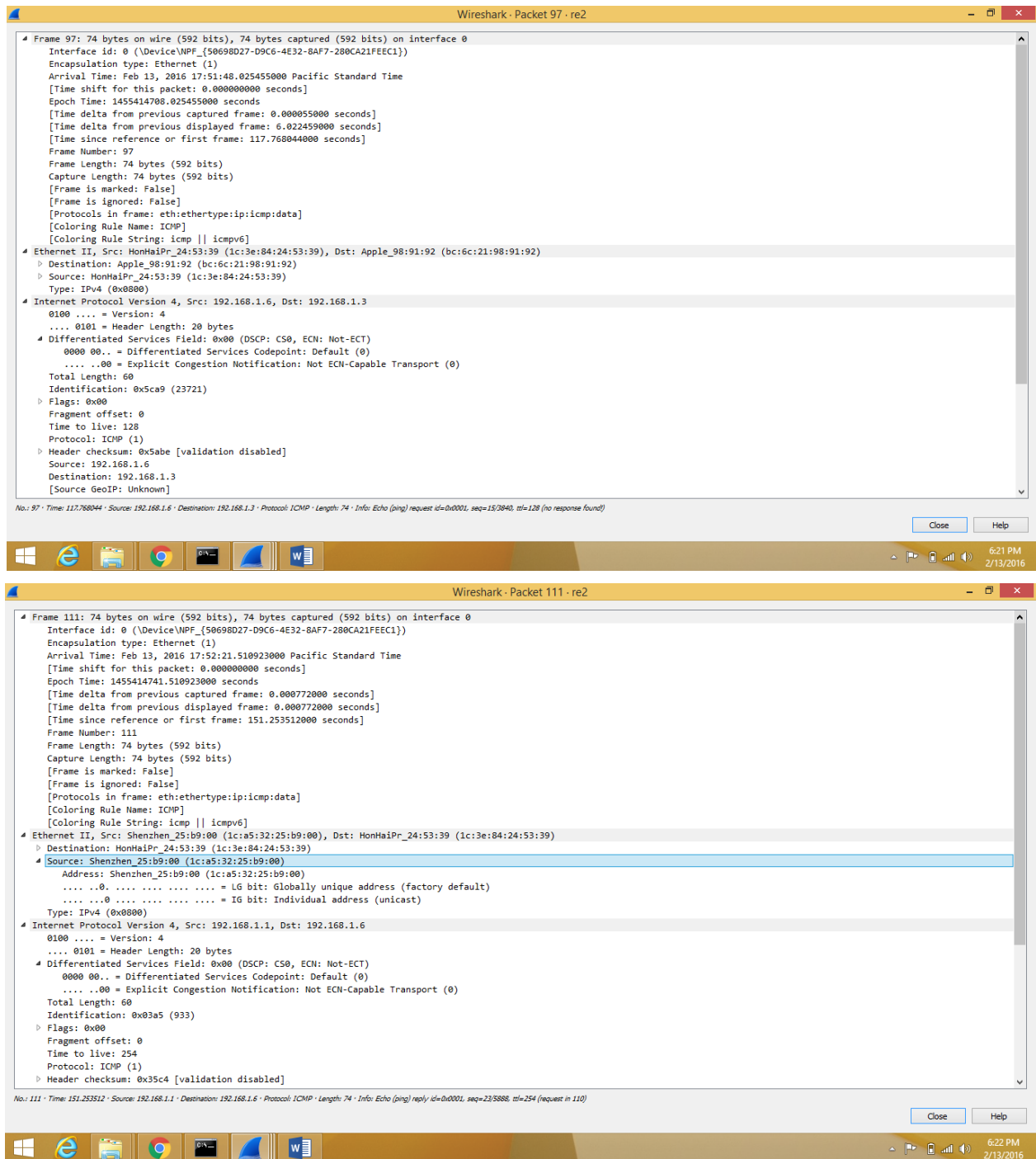
No.	Time	Source	Destination	Protocol	Length	Info
67	95.237491	192.168.1.6	192.168.1.5	ICMP	74	Echo (ping) request id=0x0001, seq=7/1792, ttl=128 (reply in 73)
73	97.293817	192.168.1.5	192.168.1.6	ICMP	74	Echo (ping) reply id=0x0001, seq=7/1792, ttl=64 (request in 67)
74	97.304540	192.168.1.6	192.168.1.5	ICMP	74	Echo (ping) request id=0x0001, seq=8/2048, ttl=128 (reply in 75)
75	97.310046	192.168.1.5	192.168.1.6	ICMP	74	Echo (ping) reply id=0x0001, seq=8/2048, ttl=64 (request in 74)
76	98.317268	192.168.1.6	192.168.1.5	ICMP	74	Echo (ping) request id=0x0001, seq=9/2304, ttl=128 (reply in 77)
77	99.340182	192.168.1.5	192.168.1.6	ICMP	74	Echo (ping) reply id=0x0001, seq=9/2304, ttl=64 (request in 76)
78	99.350933	192.168.1.6	192.168.1.5	ICMP	74	Echo (ping) request id=0x0001, seq=10/2560, ttl=128 (reply in 79)
79	99.356983	192.168.1.5	192.168.1.6	ICMP	74	Echo (ping) reply id=0x0001, seq=10/2560, ttl=64 (request in 78)
85	109.572967	192.168.1.6	192.168.1.4	ICMP	74	Echo (ping) request id=0x0001, seq=11/2816, ttl=128 (reply in 86)
86	109.575362	192.168.1.4	192.168.1.6	ICMP	74	Echo (ping) reply id=0x0001, seq=11/2816, ttl=64 (request in 85)
87	109.669195	192.168.1.6	192.168.1.4	ICMP	74	Echo (ping) request id=0x0001, seq=12/3072, ttl=128 (reply in 88)
88	109.672171	192.168.1.4	192.168.1.6	ICMP	74	Echo (ping) reply id=0x0001, seq=12/3072, ttl=64 (request in 87)
90	110.684722	192.168.1.6	192.168.1.4	ICMP	74	Echo (ping) request id=0x0001, seq=13/3328, ttl=128 (reply in 91)
91	110.771825	192.168.1.4	192.168.1.6	ICMP	74	Echo (ping) reply id=0x0001, seq=13/3328, ttl=64 (request in 90)
92	111.700766	192.168.1.6	192.168.1.4	ICMP	74	Echo (ping) request id=0x0001, seq=14/3584, ttl=128 (reply in 93)
93	111.745585	192.168.1.4	192.168.1.6	ICMP	74	Echo (ping) reply id=0x0001, seq=14/3584, ttl=64 (request in 92)
97	117.768044	192.168.1.6	192.168.1.3	ICMP	74	Echo (ping) request id=0x0001, seq=15/3840, ttl=128 (no response found!)
98	121.957240	192.168.1.6	192.168.1.3	ICMP	74	Echo (ping) request id=0x0001, seq=16/4096, ttl=128 (no response found!)
100	126.955236	192.168.1.6	192.168.1.3	ICMP	74	Echo (ping) request id=0x0001, seq=17/4352, ttl=128 (no response found!)
101	131.968561	192.168.1.6	192.168.1.3	ICMP	74	Echo (ping) request id=0x0001, seq=18/4608, ttl=128 (no response found!)
110	151.252740	192.168.1.6	192.168.1.1	ICMP	74	Echo (ping) request id=0x0001, seq=23/5888, ttl=128 (reply in 111)
111	151.253512	192.168.1.1	192.168.1.6	ICMP	74	Echo (ping) reply id=0x0001, seq=23/5888, ttl=254 (request in 110)
118	152.272585	192.168.1.6	192.168.1.1	ICMP	74	Echo (ping) request id=0x0001, seq=24/6144, ttl=128 (reply in 119)
119	152.273565	192.168.1.1	192.168.1.6	ICMP	74	Echo (ping) reply id=0x0001, seq=24/6144, ttl=254 (request in 118)
120	153.288302	192.168.1.6	192.168.1.1	ICMP	74	Echo (ping) request id=0x0001, seq=25/6400, ttl=128 (reply in 121)
121	153.321713	192.168.1.1	192.168.1.6	ICMP	74	Echo (ping) reply id=0x0001, seq=25/6400, ttl=254 (request in 120)
123	154.303473	192.168.1.6	192.168.1.1	ICMP	74	Echo (ping) request id=0x0001, seq=26/6656, ttl=128 (reply in 124)
124	154.304406	192.168.1.1	192.168.1.6	ICMP	74	Echo (ping) reply id=0x0001, seq=26/6656, ttl=254 (request in 123)

Below the packet list, a hex dump is shown for packet 111:

```
0000 1c 3e 84 24 53 39 1c a5 32 25 b9 00 00 00 45 00 ->.$59.. 2%....E.
0010 00 3c 03 a5 00 00 fe 01 35 c4 c0 a8 01 01 c0 a8 -<..... S.....
0020 01 06 00 00 55 44 00 01 00 17 61 62 63 64 65 66 ....UD.. ..abcdef
0030 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 ghijklmn opqrstuv
0040 77 61 62 63 64 65 66 67 68 69 wabcedfg hi
```







5. CONCLUSION

Data mining has been discussed in this paper. IP address plays a vital role in today's network. Hence it also needs protection like http turned into https for security needed websites or it needs some encryption for the PC's which has some important files. This random approach can be done by anyone who has little knowledge about ip address hence protection is needed for PC's. Wire shark tool is an ethical tool. An ethical tool itself finds mac address so easily means naturally one has to worry about the security of IP address.

6. REFERENCES

- [1] <http://www.wireshark.org/>
- [2] <http://www.packets.com/>
- [3] <http://www.network.com/>
- [4] <http://www.linkage.rockefeller.edu/wli/zipf/org/>