Basic Packet Sniffer

By Cameron Bell

What are Packet Sniffers?

A tool designed to monitor a network by recording all packets sent across it.

How do they work?

- **Computers send data across networks in the form of packets.**
- **Small units of data that can be constructed to form a file.**
- **Typically these are sent to each machine on the network.**
- **Machines that don't require it simply drop them.**
- **A packet sniffer sets it's host machine too receive all packets (promiscuous mode)**
- **The sniffer program then records each packet sent to/from the host machine.**
- **This can be seen in figure 1, where any data packets sent between User and Router to Sniffing device, which is running a packet sniffer and recording every packet.**

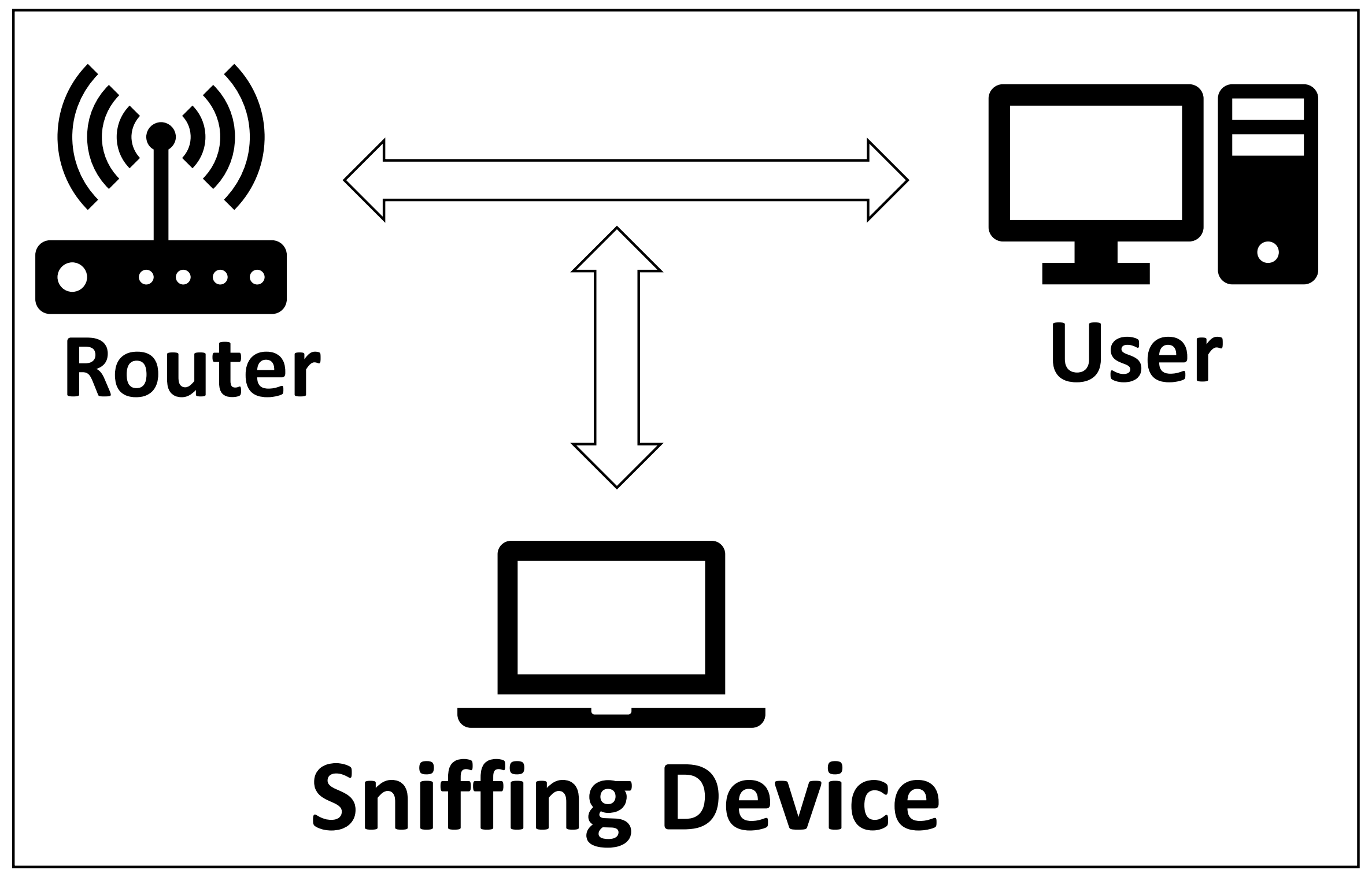
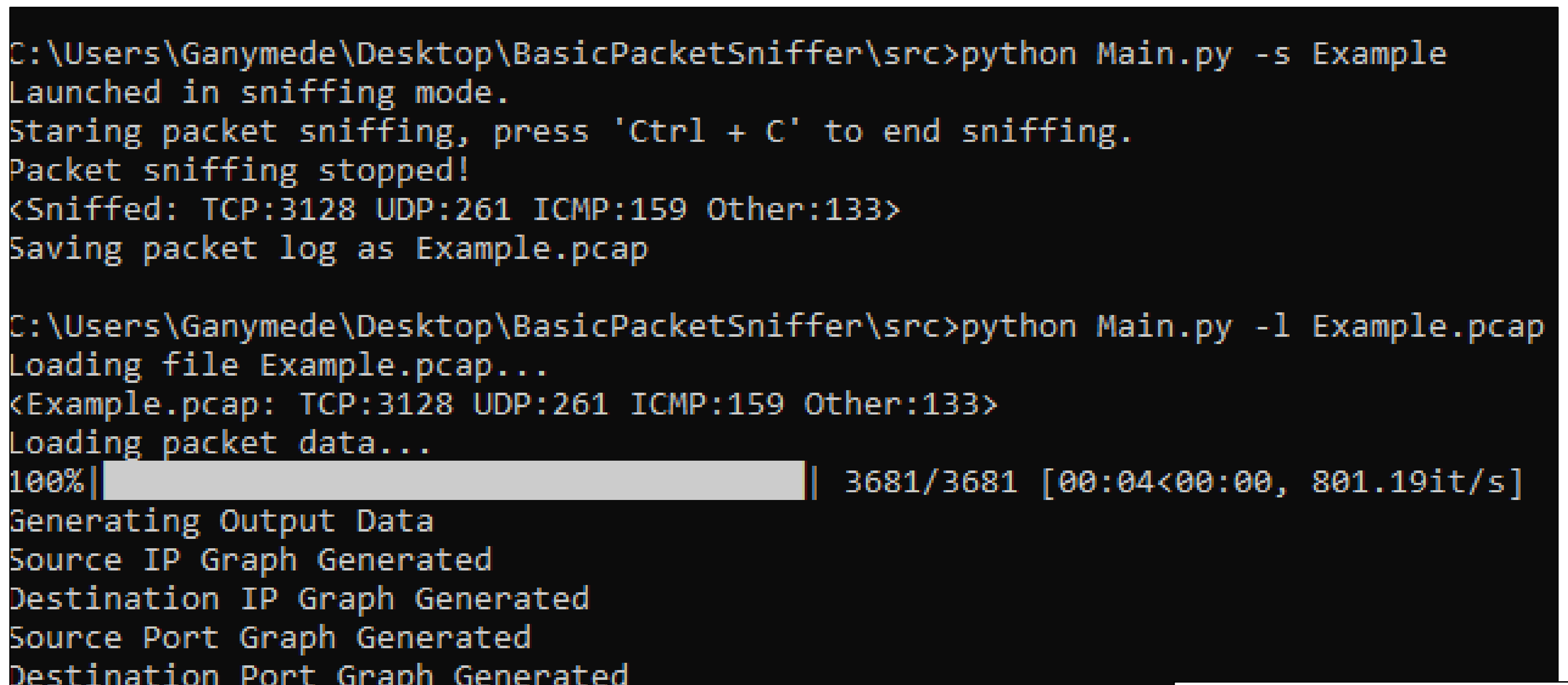


Figure 1



```
C:\Users\Ganymede\Desktop\BasicPacketSniffer\src>python Main.py -s Example
Launched in sniffing mode.
Staring packet sniffing, press 'Ctrl + C' to end sniffing.
Packet sniffing stopped!
<Sniffed: TCP:3128 UDP:261 ICMP:159 Other:133>
Saving packet log as Example.pcap

C:\Users\Ganymede\Desktop\BasicPacketSniffer\src>python Main.py -l Example.pcap
Loading file Example.pcap...
<Example.pcap: TCP:3128 UDP:261 ICMP:159 Other:133>
Loading packet data...
100%|██████████████████████████████| 3681/3681 [00:04<00:00, 801.19it/s]
Generating Output Data
Source IP Graph Generated
Destination IP Graph Generated
Source Port Graph Generated
Destination Port Graph Generated
IPinfo connection successful
Source IP map Generated
Destination IP map Generated
Document Generated
Outputs saved in the "./Example.pcap Output" folder
```

Figure 2

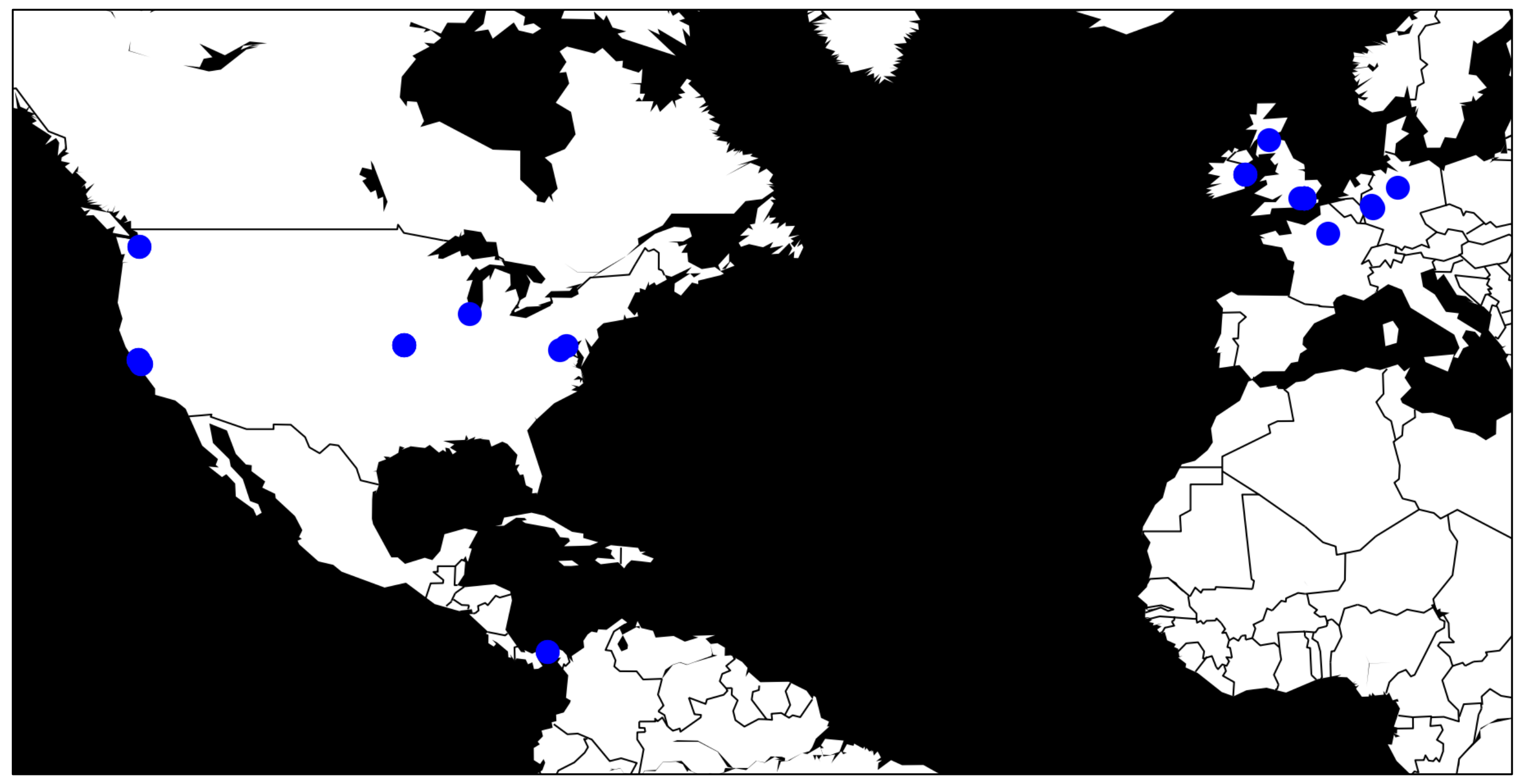
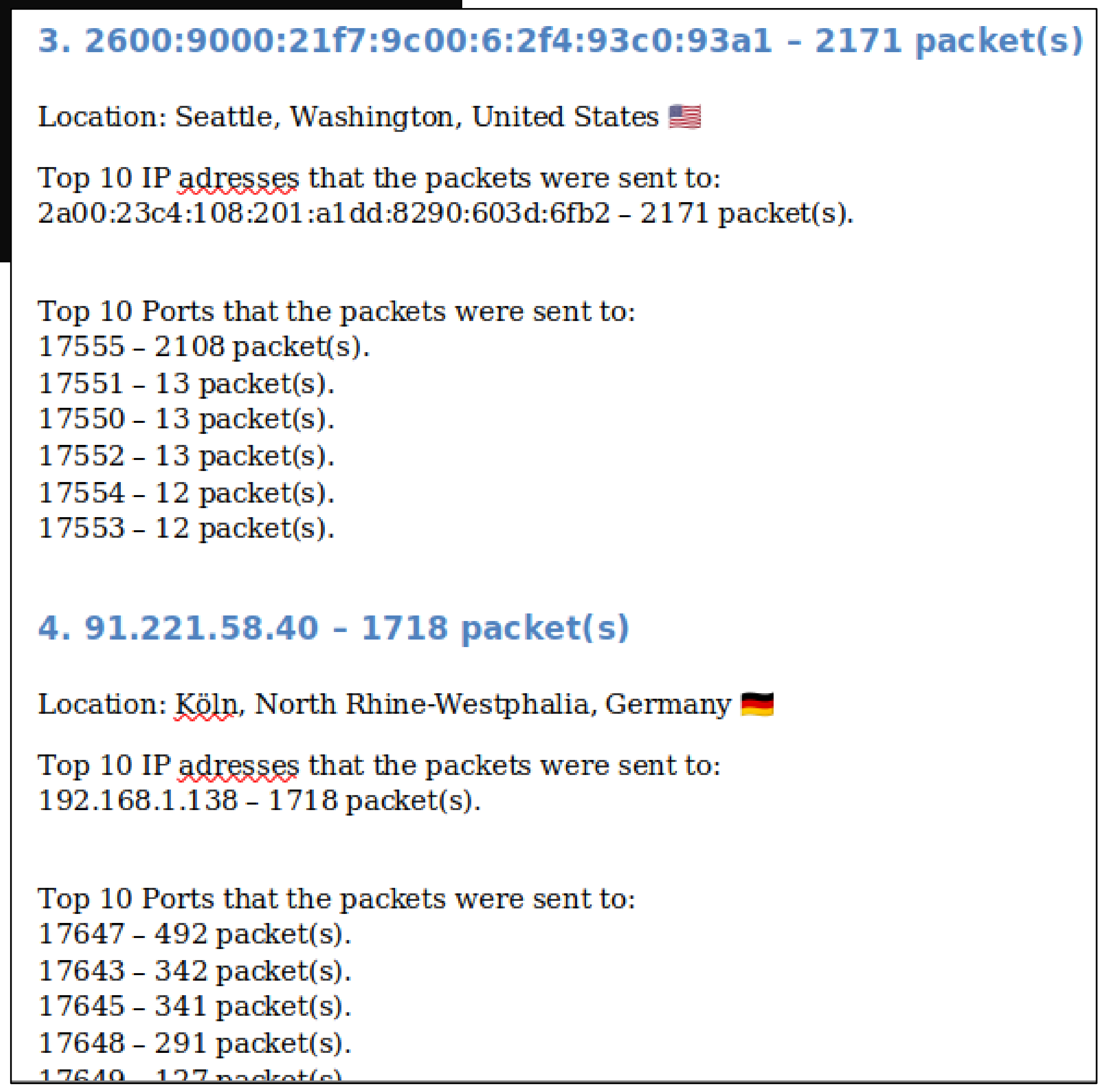


Figure 3



3. 2600:9000:21f7:9c00:6:2f4:93c0:93a1 - 2171 packet(s)

Location: Seattle, Washington, United States 🇺🇸

Top 10 IP adresses that the packets were sent to:
2a00:23c4:108:201:a1dd:8290:603d:6fb2 – 2171 packet(s).

Top 10 Ports that the packets were sent to:
17555 – 2108 packet(s).
17551 – 13 packet(s).
17550 – 13 packet(s).
17552 – 13 packet(s).
17554 – 12 packet(s).
17553 – 12 packet(s).

4. 91.221.58.40 - 1718 packet(s)

Location: Köln, North Rhine-Westphalia, Germany 🇩🇪

Top 10 IP adresses that the packets were sent to:
192.168.1.138 – 1718 packet(s).

Top 10 Ports that the packets were sent to:
17647 – 492 packet(s).
17643 – 342 packet(s).
17645 – 341 packet(s).
17648 – 291 packet(s).
17649 – 127 packet(s).

Figure 4

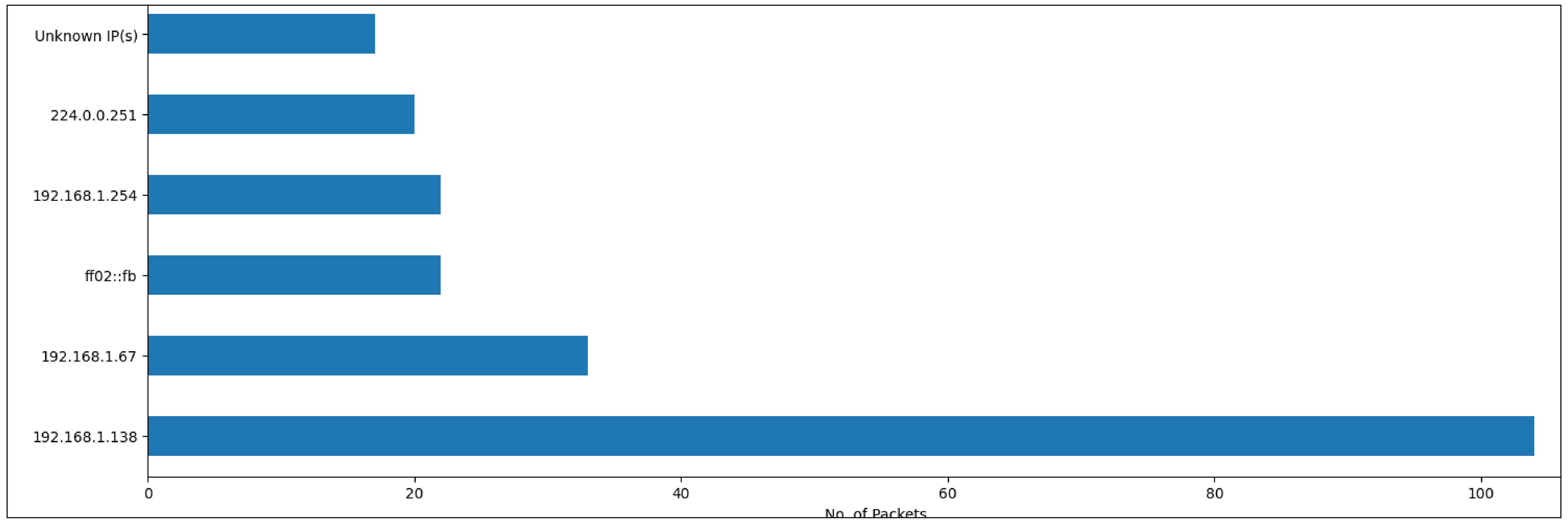


Figure 5

Tool Outcome

The tool met most of the outcomes set out at the beginning, with the tool able to:

- **Sniff and process the data (figure 3).**
- **Generate charts for the source and destination IP addresses and ports (figure 5).**
- **Generate maps showing the locations the packets were sent from and to (figure 3).**
- **Generate an analysis document containing a basic write up about the packet log (figure 4).**
- **Specify an IP address to use only packets relating to that when generating charts.**

There were some features such as a GUI and packet generating function that were planned but cut.

Designing the Tool

- **The initial part of this project was investigating existing packet sniffing tools to design one that didn't exist.**
- **It was decided to develop a simple to use tool with a focus visually analysing the recorded packet data.**
- **The tool should be able to sniff a network and save the recorded packet log as a pcap file, with options which packets are recorded.**
- **Process the recorded data and produce charts and a document based on the data, to allow the data to be easily analysed. With options to narrow the data used in the chart/document generation.**

Developing it

The tool was developed in python, using the packages:

- **Scapy for the packet capture.**
- **Pandas for the processing and storage of the data.**
- **Matplotlib to generate the charts from the processed data.**
- **Ipinfo used to look up information on IP address to get packet source/destination locations.**
- **Python-docx to generate a docx file to hold the file analysis.**