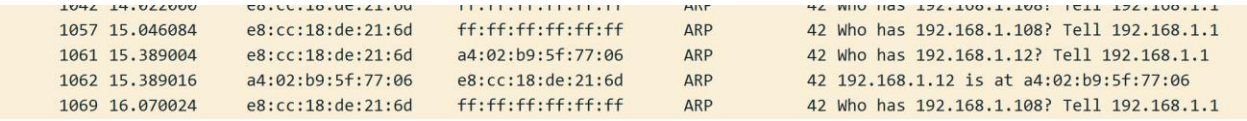
Part A



1057	15.046084	e8:cc:18:de:21:6d	ff:ff:ff:ff:ff:ff	ARP	42 Who has 192.168.1.108? Tell 192.168.1.1
1061	15.389004	e8:cc:18:de:21:6d	a4:02:b9:5f:77:06	ARP	42 Who has 192.168.1.12? Tell 192.168.1.1
1062	15.389016	a4:02:b9:5f:77:06	e8:cc:18:de:21:6d	ARP	42 192.168.1.12 is at a4:02:b9:5f:77:06
1069	16.070024	e8:cc:18:de:21:6d	ff:ff:ff:ff:ff:ff	ARP	42 Who has 192.168.1.108? Tell 192.168.1.1

The Wireshark capture contains these ARP packets, which the analysis_pcap_arp.py can parse and extract field data the from bytes. Tested for Python 3.

Part B

Relevant Exchange:

Time Stamp:1556481203.538317----------
Hardware type:1
Protocol type:2048
Hardware size:6
Protocol size:4
Opcode:1
Sender MAC address:e8:cc:18:de:21:6d
Sender IP address:192.168.1.1
Target MAC address:00:00:00:00:00:00
Target IP address:192.168.1.12

Time Stamp:1556481203.538329----------
Hardware type:1
Protocol type:2048
Hardware size:6
Protocol size:4
Opcode:2
Sender MAC address:a4:02:b9:5f:77:06
Sender IP address:192.168.1.12
Target MAC address:e8:cc:18:de:21:6d
Target IP address:192.168.1.1

The router which is making the request (opcode 1) has MAC: e8:cc:18:de:21:6d and IP: 192.168.1.1.

My laptop which replied (opcode 2) to the request directed towards it has MAC: a4:02:b9:5f:77:06 and IP: 192.168.1.12.

Bonus

Gratuitous requests were also captured, since they are broadcasted by the router and thus noticed by any device on the LAN. An example would be the gratuitous requests for the IP of 192.168.1.108. This IP is not my own, which is 192.168.1.12.