Traffic Capture

First Test

No packets are captured because there's no pinging between the interfaces (no connection, so no packets sent).

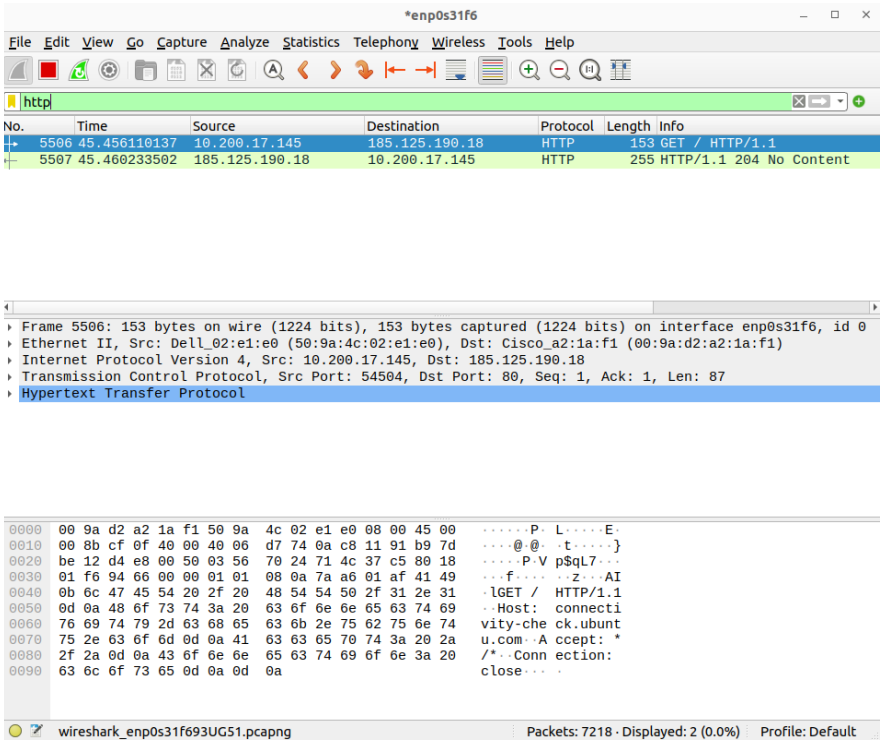
However, if we use P4Pi-9 to ping the ethernet interface (192.168.10.1), we can see multiple packets captured on wireshark:

SSH - Client: Encrypted packet
SSH - Server: Encrypted packet
TCP - 49924 -> 22 [ACK]
ICMP - Echo (ping) request
ICMP - Echo (ping) reply
ARP - Who has 192.168.10.2? Tell 192.168.10.1
ARP - 192.168.10.2 is at e4:5f:01:87:60:14

Due to remote operation using the Lab computer, we can see the SSH and TCP Protocol packets very often, almost like a background constant.
The ICMP and ARP Protocol packets exist due to the pinging that is being performed.

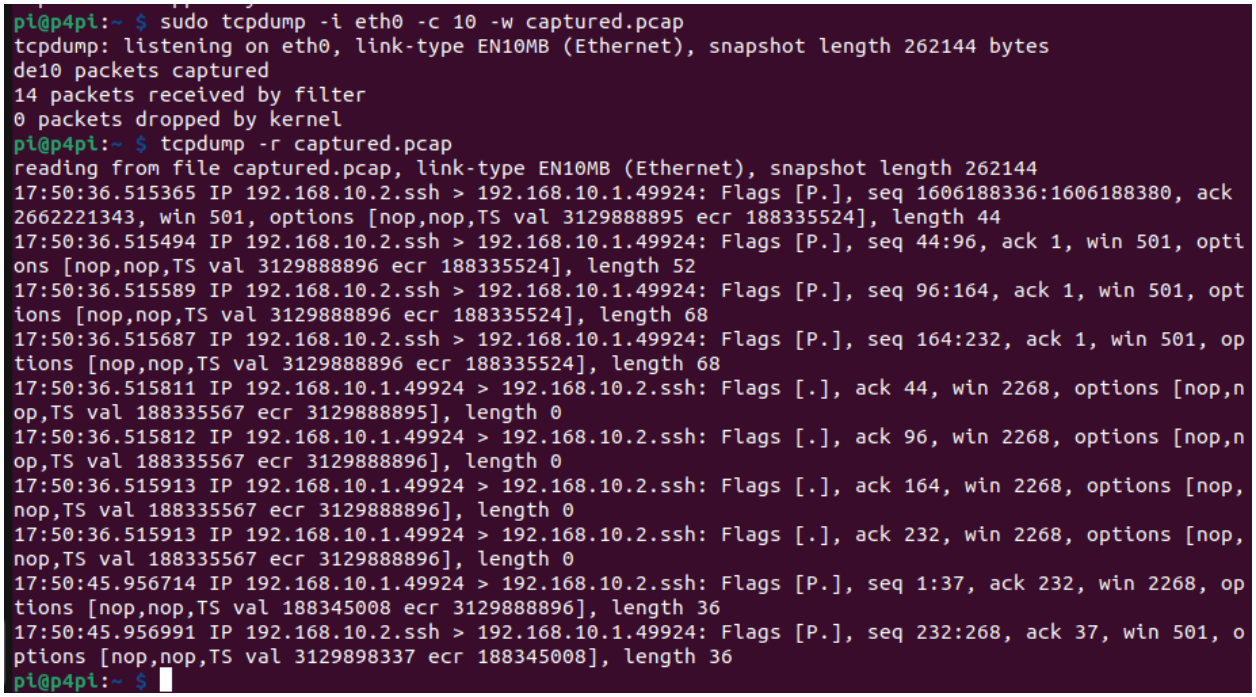
Second Test

By filtering for HTTP and using Firefox on the Lab computer, we are able to detect HTTP Protocol packets.



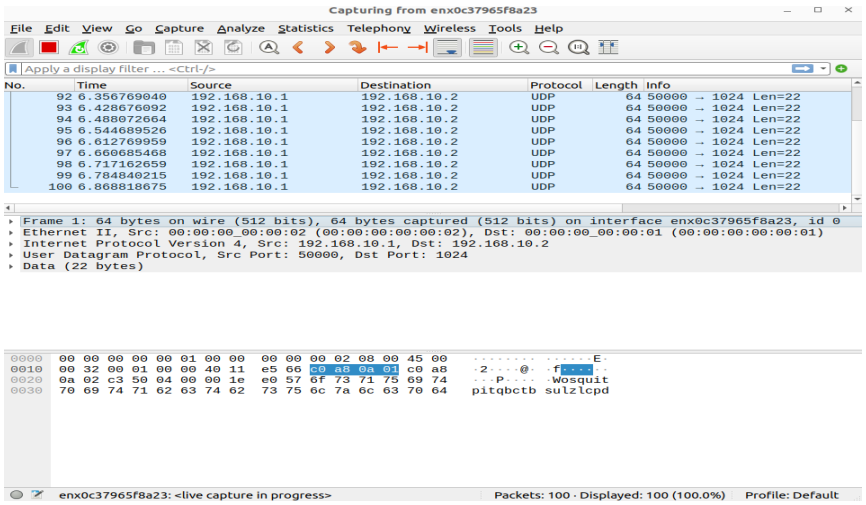
Third Test

After waiting a short while, the interfaces began communicating, and the tcpdump output shows that below:

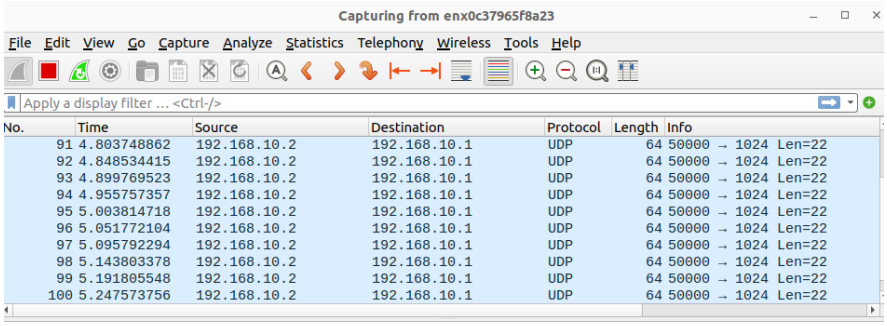


Sending Traffic

Sending packets from the Lab machine to P4Pi-9, we see 100 UDP Protocol packets being detected on Wireshark.



The opposite gives the same result: 100 UDP Protocol packets being detected on Wireshark.



If a filter is necessary, it would be the UDP Protocol filter. In this case, there was no need to apply a filter since all the packets detected were UDP Protocol ones. Each packet is 64 bytes with 22 bytes worth dedicated to the payload. Based on what is observed, the protocol used is UDP.

Script Modification

The following is a screenshot of the modified python script:

```
#!/usr/bin/python

from scapy.all import Ether, IP, sendp, get_if_hwaddr, get_if_list, TCP, Raw, UDP
import sys
import random, string


def randomword(length):
    return ''.join(random.choice(string.ascii_lowercase) for i in range(length))

def send_random_traffic(num_packets, interface, src_ip, dst_ip):
    dst_mac = "00:00:00:00:00:01"
    src_mac= "0c:37:96:5f:8a:23"
    total_pkts = 0
    port = 1024
    for i in range(num_packets):
            data = randomword(458)
            p = Ether(dst=dst_mac,src=src_mac)/IP(dst=dst_ip,src=src_ip)
            p = p/TCP(sport= 5555, dport=port)/Raw(load=data)
            sendp(p, iface = interface, inter = 0.01)
            # If you want to see the contents of the packet, uncomment the line below
            # print(p.show())
            total_pkts += 1
    print("Sent %s packets in total" % total_pkts)

if __name__ == '__main__':
    if len(sys.argv) < 5:
        print("Usage: python send.py number_of_packets interface_name src_ip_address dst_ip_address")
        sys.exit(1)
    else:
        num_packets = sys.argv[1]
        interface = sys.argv[2]
        src_ip = sys.argv[3]
        dst_ip = sys.argv[4]
        send_random_traffic(int(num_packets), interface, src_ip, dst_ip)
~
```

This yields the following result: