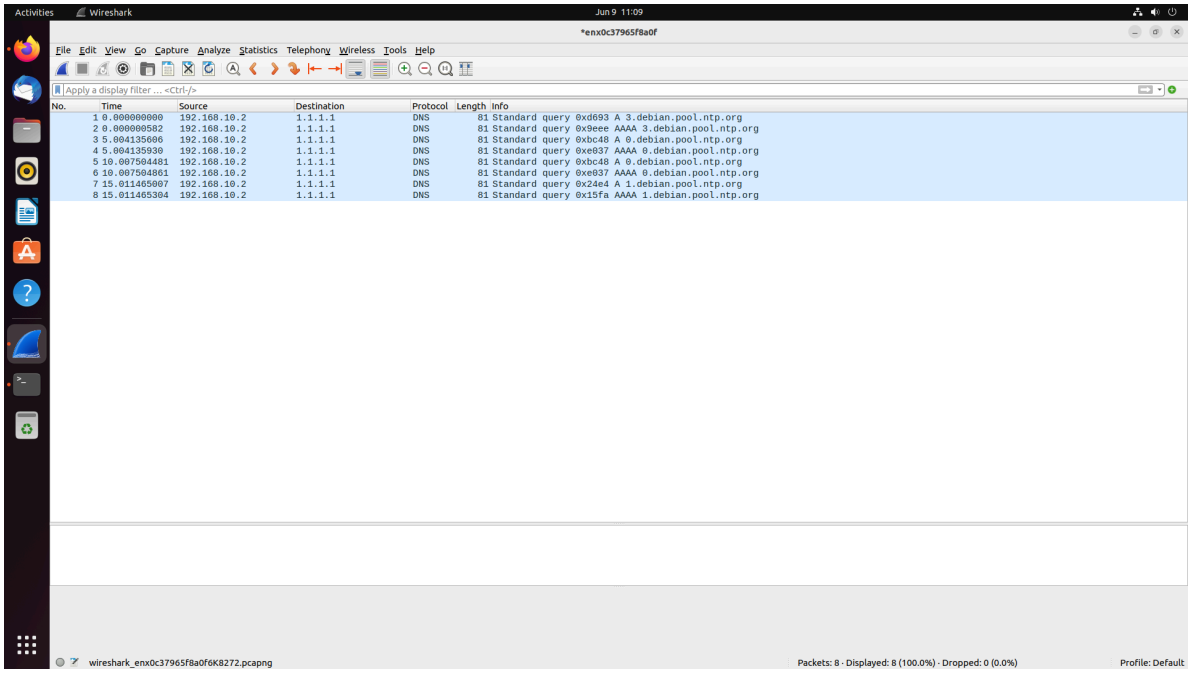
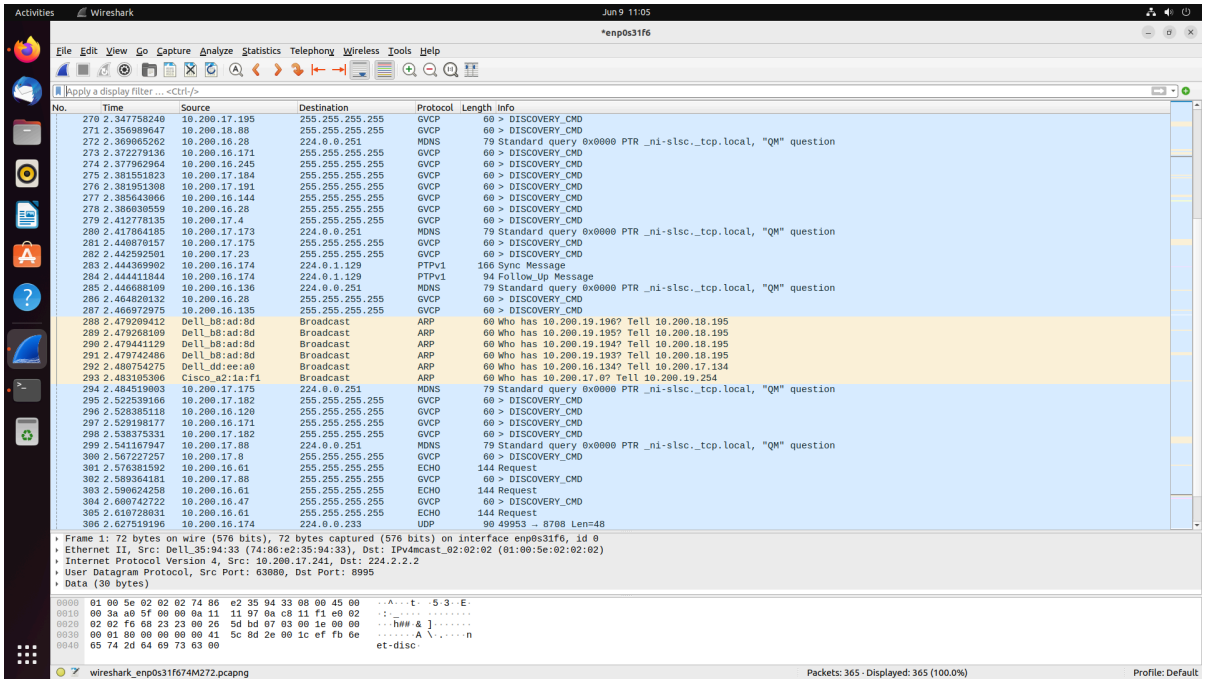
Capture on ethernet interface




Every 5s, two standard queries are made from Pi onto a destination described as 1.1.1.1. This is a DNS (Domain Name System) protocol. We will see later when we apply the UDP (User Datagram Protocol) filter onto the Pi that the DNS still appears. This makes me think DNS is a type of UDP.

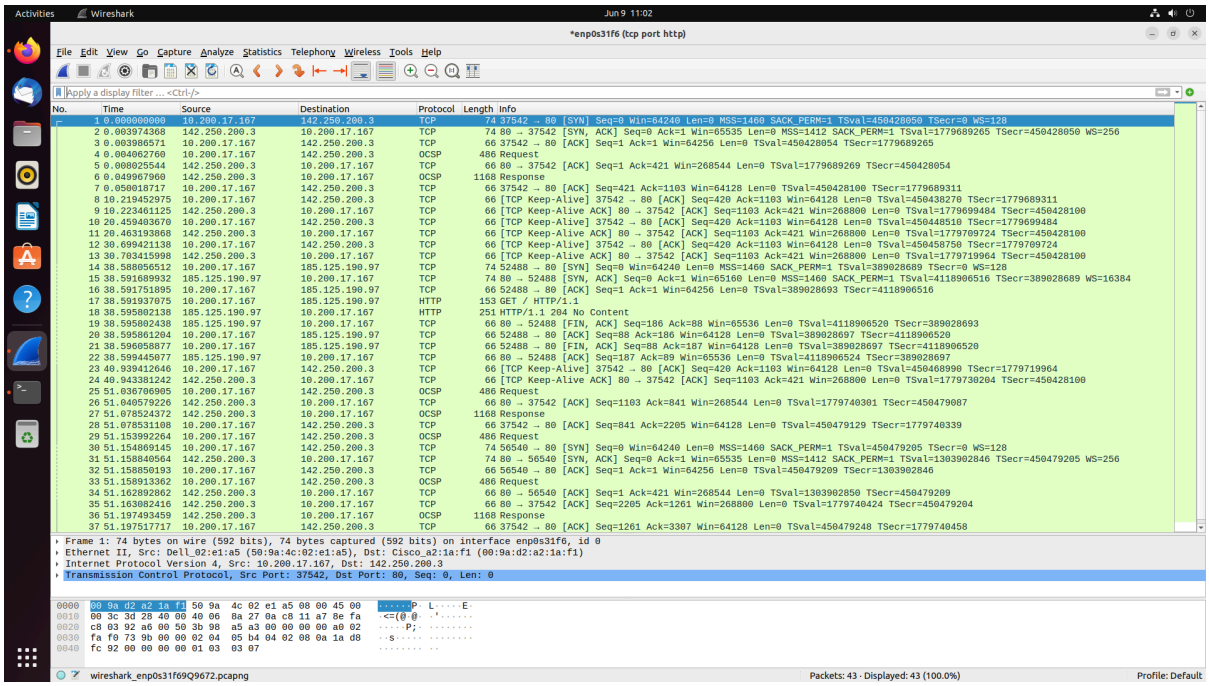
From my understanding, a quick Google search reveals DNS requests are just a way of querying the “names” of the destination computers.

Capture on network interface, no filter



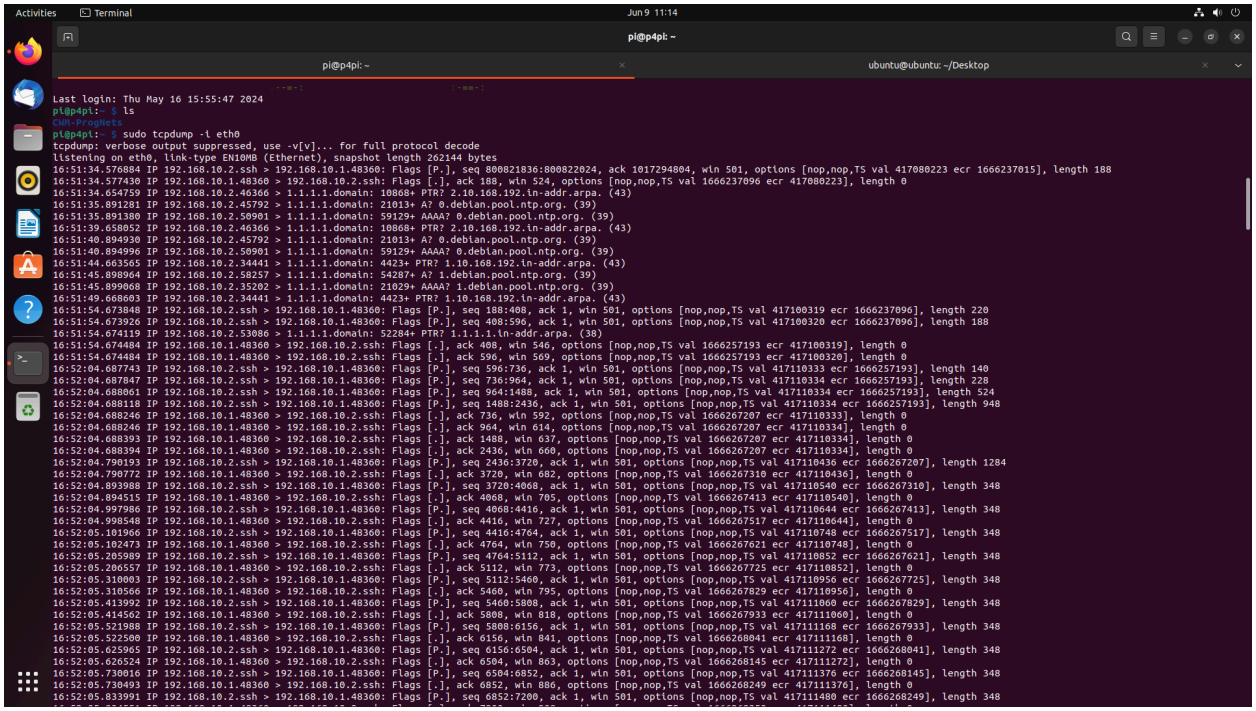
A lot of traffic happening on the departmental network. We can see various types of protocols (GVCP, MDNS, PTPv1, ECHO). It is interesting to see that the MDNS ones are also talking about standard queries so this makes me think this is related to regular DNS. A Google search reveals MDNS is like DNS but for smaller networks. ECHO (request) is used regularly to check that the device is still connected to the network. Also interesting that for the most part, packets of the same protocol have the same lengths.  Can also see ARP broadcasts for when some computers (Dell) don't have some IP addresses in their ARP table.

Capture on network interface, HTTP traffic

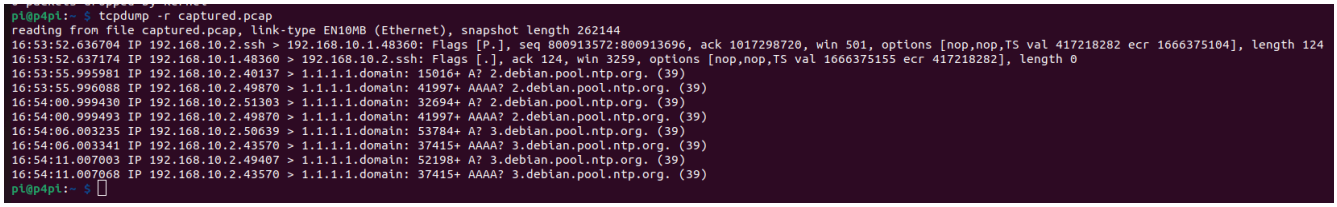



In order to capture HTTP traffic, I needed to access the web in some way (e.g. doing a Google search). We can see even a 1-minute browser search requires a lot of packets to be sent.

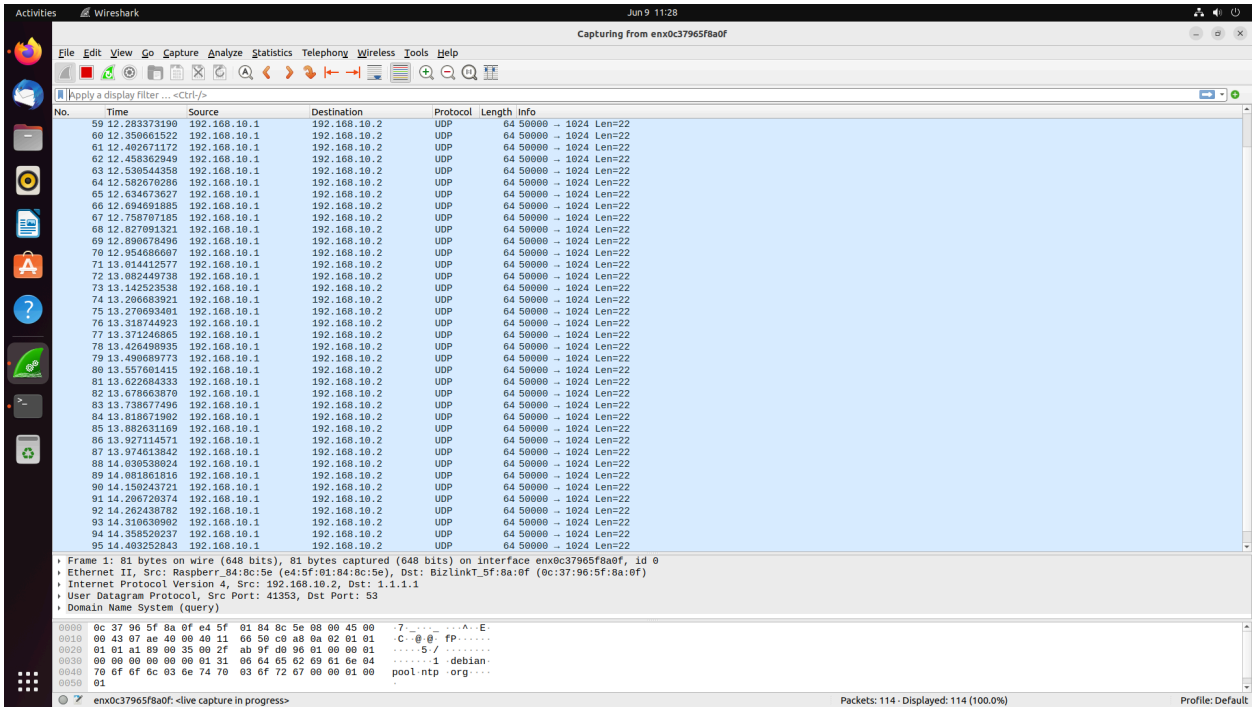
Capture on Pi using tcpdump (manual termination)




Captured on Pi using tcpdump (limit to 10 packets)

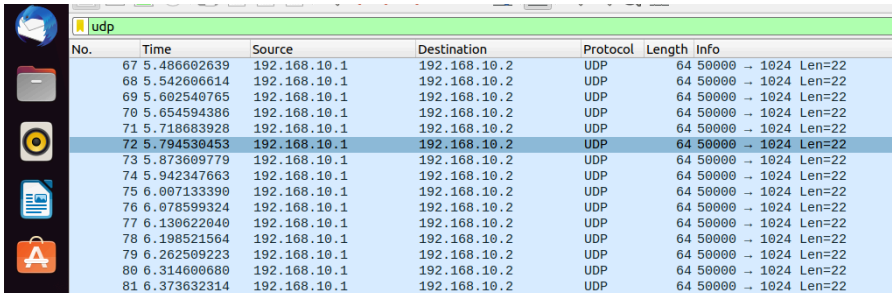


Send 100 packets from Pi to 192.168.10.2

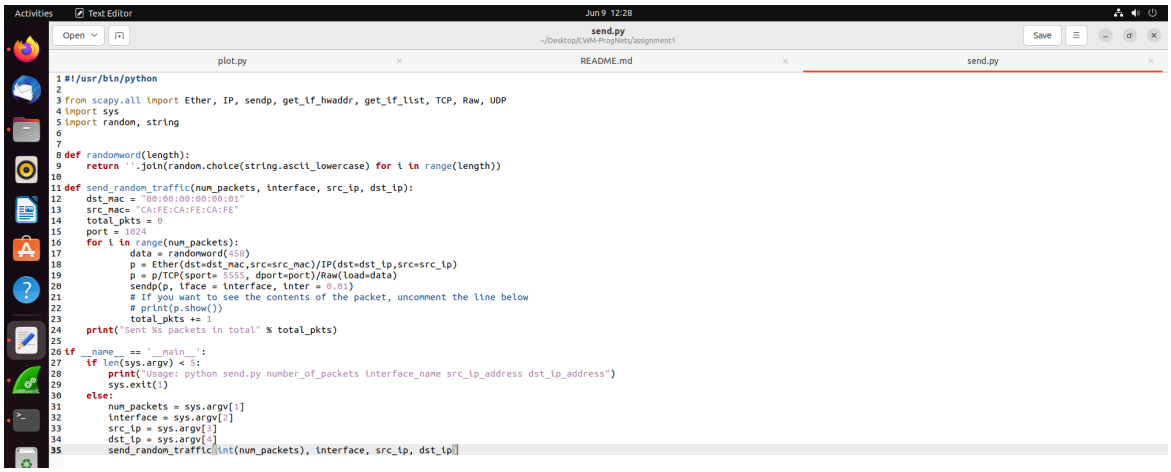



Transport protocol used is UDP. The payload is 22 bytes, but the total length is 81 bytes. For reference for the next section, source port is 50000

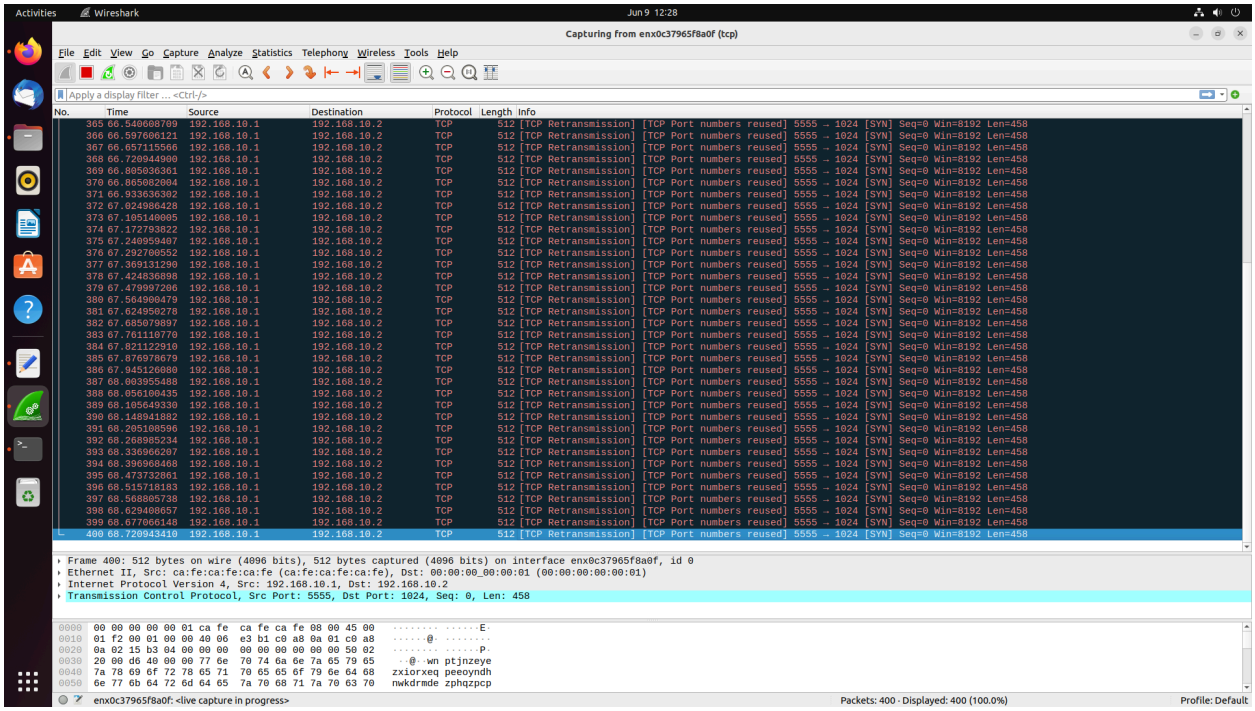
Can apply a UDP filter to only capture the packets we send




Modified Python file:



After changing to TCP, and applying TCP filter:




Can see the protocol is TCP, source port changed from 50000 to 5555, payload changed from 22 to 458 bytes, total length changed from 81 to 512 bytes.