Wire Shark Intro Lab:

When using Wireshark there are a handful of protocols that come in. Three of the first ones that I noticed were DNS, TCP, and HTTP. The total round trip time for the request to and from was relatively short, it was ~.04 seconds. The web address for the remote computer is 128.119.245.12 while my local IP address is 192.168.0.110. I printed the two packets to pdf but because they are an image I could not attach them here.

IP Lab:

My computers IP address is 192.168.0.110. The protocol fields value is 1 for ICMP. The IP header has 20 bytes. The total bytes of the entire packet is 56 so the body byte size is 56 – 20 or 36 bytes**.** The IP datagram has not been fragmented. You can tell because there is no “More Fragments” flag The fields that always change in the IP datagram are the TTL, the header checksum and the Identification. Every other field remains the same. I suspect the identification changes because every packet has it’s own id and the TTL changes because the route always adds 1 to the TTL of the packet so that it can make it 1 router further than the last packet. The routers always decrement the TTL by 1 until the value is 1 in which case it returns to sender. The value for the identification field also always increments by 1. All the values in the TTL-exceeded replies are the same. The identification is likely the same because an ID of 0 represents a TTL-exceeded packet so all packets that are of type TTL-exceeded will contain the same Id. I’m not entirely sure why the TTL values are identical because the TTL for the first hop router is always the same and it’s the last router to get the packet before it returns to my computer. For the 2000 packet size the message has been fragmented into 2 fragments. With wireshark set up to re-assemble the fragmented packets you can tell a packet is part of a fragment because it contains a “Fragments” section in the IP. This tells you how many fragments your packet was broken into, what size they, and what frame they were. If you correlate the Frame in the Fragments section to the Frame at the packet header you can tell which fragment your packet is. The Fully assembled IP packet is 1980 bytes broken up into 2 fragments of 0-1479 bytes and 1480-1979 bytes. The packets with a size of 3500 are broken up into 3 fragments. The field that always changes in each fragment is the identifier.  
  
DNS Lab:

I ran an nslookup to obtain the ip address of the Asian Institute of Technology:

PS C:\WINDOWS\system32> nslookup www.ait.ac.th/

Server: UnKnown

Address: 192.168.0.1

\*\*\* UnKnown can't find www.ait.ac.th/: Non-existent domain

PS C:\WINDOWS\system32> nslookup www.ait.ac.th

Server: UnKnown

Address: 192.168.0.1

Non-authoritative answer:

Name: ait.client.bypronto.com

Addresses: 2600:1f18:43b:9300:6b82:a21e:ab8c:7b63

52.1.32.33

Aliases: [www.ait.ac.th](http://www.ait.ac.th)

I ran an nslookup to determine oxford universities authoritative DNS servers:  
PS C:\WINDOWS\system32> nslookup -type=NS www.ox.ac.uk

Server: UnKnown

Address: 192.168.0.1

ox.ac.uk

primary name server = nighthawk.dns.ox.ac.uk

responsible mail addr = hostmaster.ox.ac.uk

serial = 2018091372

refresh = 3600 (1 hour)

retry = 1800 (30 mins)

expire = 1209600 (14 days)

default TTL = 900 (15 mins)

I wasn’t able to run nslookup so that one of the DNS servers obtained queries yahoo, I got DNS request timeouts:

PS C:\WINDOWS\system32> nslookup mail.yahoo.com www.ox.ac.uk

DNS request timed out.

timeout was 2 seconds.

Server: UnKnown

Address: 129.67.242.155

DNS request timed out.

timeout was 2 seconds.

DNS request timed out.

timeout was 2 seconds.

DNS request timed out.

timeout was 2 seconds.

DNS request timed out.

timeout was 2 seconds.

\*\*\* Request to UnKnown timed-out

The DNS query appears to be sent over a DNS protocol or a User Datagram Protocol. The source port for the query