- Identify at least 2 instances of suspicious, interesting, and/or unfamiliar protocols/communication.

The question asked for two instances of suspicious, interesting, or unfamiliar. I've been looking at Wireshark for about two weeks so all of this fits that bill. That being said, I went through and discussed each protocol that I saw because it was a great exercise for me to learn about this stuff and apply what I learned in class.

Protocols (organized by "most used" to "least used"

UDP - This is a transport layer protocol. It eliminates the handshake business of a similar protocol like TCP. It stands for User Datagram Protocol. It uses these things called datagrams which is simple words are just messages to perform checksums and check port numbers are consistent. This is better than TCP for live streaming. I had Outlander playing the background on the AppleTV while I worked. I imagine that UDP is incredibly useful when you want what your watching to come through more quickly to help keep everything organized. If everything went through with TCP it would most likely have a lot of lag.

TCP - Transportation Control Protocol. It's a transport layer with handshakes and such. We have discussed this so much in class and in other parts of this homework. It's TCP. We know it.

TLSv1.2 - This stands for "Transport Layer Security v1.2". It is the latest version of SSL protocol and adds a 256 hash-algorithm to boost security. SSL stands for secure sockets layer and what that entails is essentially making sure the communication between port A and B is secure and others can't mess with it. It has to continuously be updated and thus has a few different names because part of cybersecurity is being smarter than the bad guy.

SSDP - Simple Service Discovery Protocol. This network layer protocol is used to make sure network services are advertised and easily found. When you look in your WIFI menu this is what is listing your network as able to be joined.

ARP - Address Resolution Protocol is a data-link layer that associates IP addresses with MAC addresses to make sure the communication is indeed going to the correct machine.

DNS - This is an application layer protocol which is responsible for making sure that the correct internet services (server IP addresses) are associated with the correct receiver (requesting client). Considering we have many devices like my phone, the PS4, the AppleTV all on at the same time. DNS would make sure that if I request Netflix on the TV it shows up on the TV and not my phone for example. Although if it went to my phone I doubt it would work

because of the other protocols not simultaneously running because the phone was in my pocket not requesting anything.

HTTP/XML - XML is a presentation layer protocol. Google and Wikipedia are both rather scarce on explanation but from what I can understand it appears to be a protocol under development that is added onto something like HTTP (an application layer) that helps keep things in the correct order. This would be important if you are streaming so that the data came in at the correct time. It doesn't surprise me that I would see HTTP/XML alongside something like UDP. If I was streaming on the AppleTV then UDP is a faster workaround to TCP and HTTP/XML is a fast AND ORGANIZED way of getting my stuff of the screen and in the correct order.

HTTP - This is an application layer protocol that takes data from a server IP to a client IP. Part of the homework was based on this so I will just reference my car example from the other challenge. It is the car that travels on the road.

IGMP - Stands for Internet Group Management Protocol. It is a network layer protocol. It is essentially responsible for making connections and maintaining connections of many devices on a similar network. In my house, at the time I know my phone, the APPLE TV, and the PS4 were for sure connected. It wouldn't shock me to find other things connected as well.

ICMP v6 - This is the Internet Control Message Protocol. It is a network layer protocol. It is in charge of ensuring the communication happened. If the communication does not happen this is how error messages get generated. The v6 part of that name is the implementation with Internet Protocol v6.

IGMPv2 - This is IGMP but it does not need to wait for the multicast querier If I am understanding correctly; during the original IGMP days the transmission had to wait for the multicast querier to go back and forth to get all parts of the data transmission. IGMPv2 uses resources more efficiently and as a result, incorporates all of the multicasts into this version. This allows for streaming packets to transport faster.

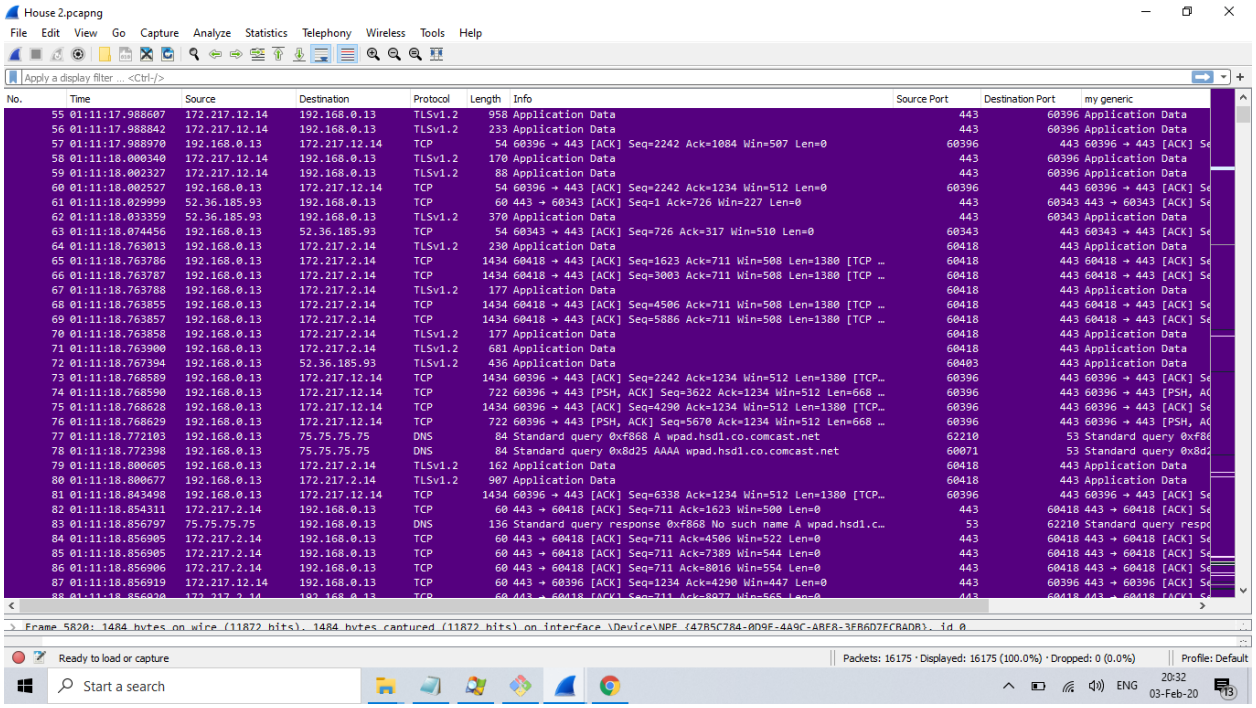
IGMPv3 -This does everything v2 did but it adds and supports "snooping features" which forces all traffic for that specific communique to go through specific ports. Think about it. If you can organize one specific communique to a specific port structure it would logically make it more efficient because in the delivery stage it is a simple triage maneuver.

MDNS - This is similar to the DNS protocol and that said it is an application layer protocol. The "m" stands for "multicast". It is used on smaller networks without a local name server and is a zero-configuration service. If I am understanding what I read correctly then that means it is

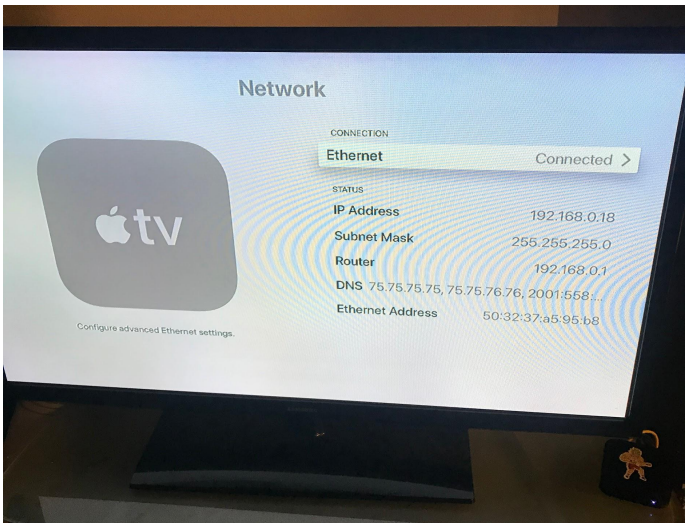
usually used in conjunction with normal DNS because once normal DNS configures the connecting then multicasting and mDNS can keep it alive because that's a less "thinking" type job.

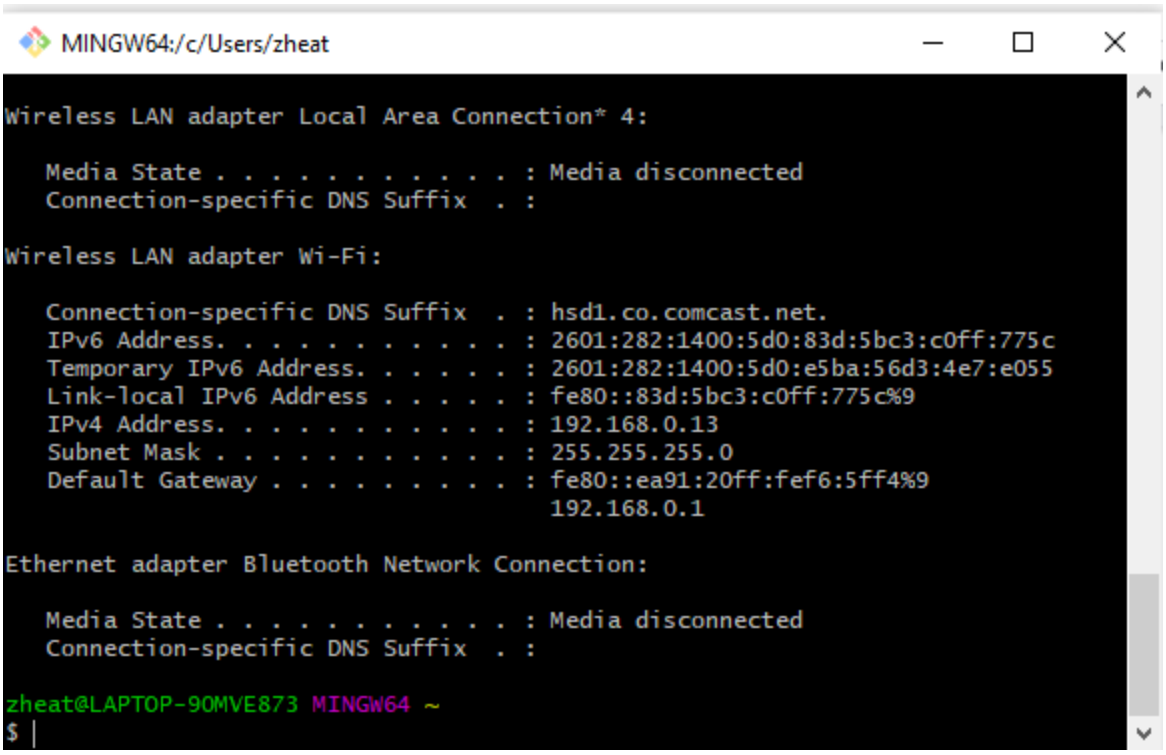
When I sat down to start thinking about how to report on two hours of internet traffic that contains close to 83,000 packets to go through I had to come up with a logical method to attack it. I started by looking at by first attempting to see what protocols were being utilized. I did that by filtering in wireshark the appropriate column. That is how I gathered the information to generate the list above. I then noticed something really strange which was how much UTP protocol there was. It really shocked me that in just two hours that 83,000 packets had come and gone.

The next step was to narrow down which IP addresses were being utilized and this gave me another huge shock. I noticed a large frequency of the IP address 192.168.0.13. I started by applying a new filter and made that IP address out to be dark purple using Wireshark's tools. This is what it came out to be:



Obviously a screenshot cannot show this but if you scroll through the capture; it appears more than 75% of the traffic was from that IP address! I thought to myself what would cause that much traffic and my first thought was the AppleTV as it is always streaming. I was wrong on that because the IP address while it was wired did not match. I thought I would try connecting it to the WIFI and the IP remained incorrect. I then did IPCONFIG on my laptop and noticed that the majority of the traffic was coming from my personal laptop. Some screenshots of this are below.

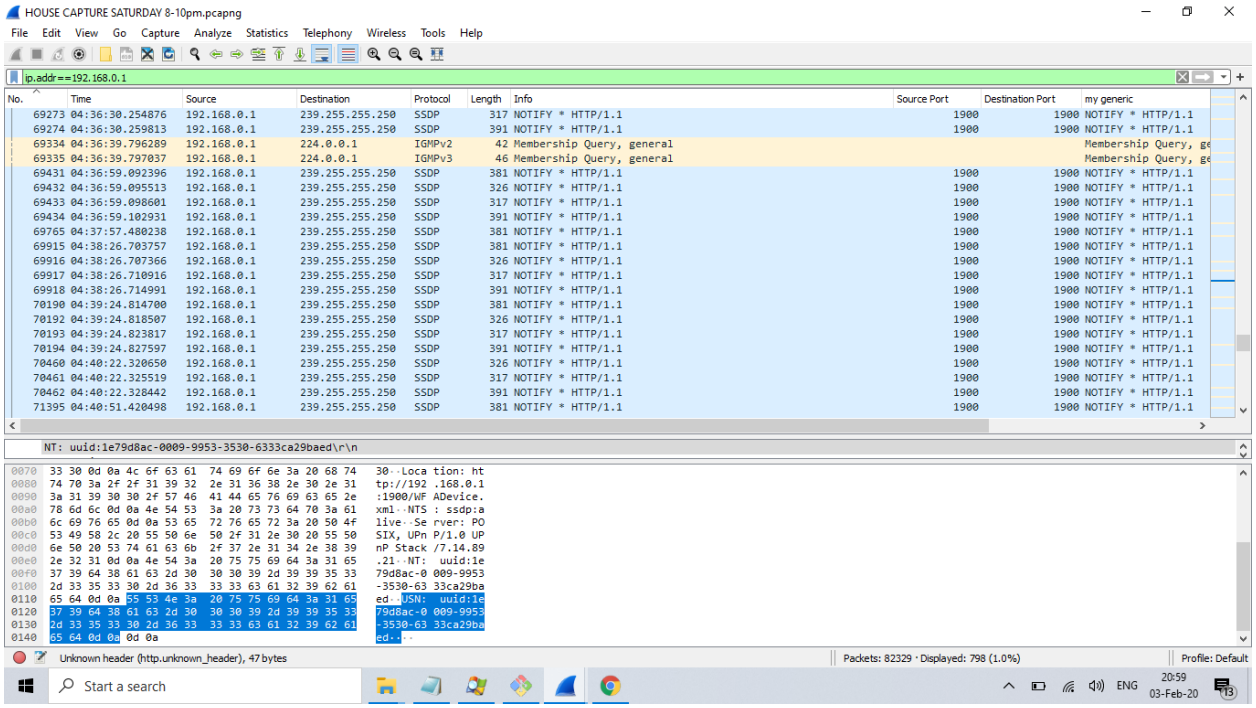




Once I discovered that it was not the AppleTV creating all the packets I had to figure out why my laptop had so much traffic. I also had to figure out why UDP protocol was more popular than a beer at a rugby match. I ran another capture on Monday after work and this time I did all sorts of experiments. I started by messing with the AppleTV and then I started going to different websites. When I went to Youtube there was a MASSIVE spike. I went from 4,000 to 16,000 in

the matter of five minutes. It turns out that Youtube streaming using UDP protocol creates multiple packets.

I then thought to myself; why is it that Youtube streaming makes all these packets while Outlander has been on the AppleTV and I'm not seeing a ton of traffic as a result of that? I started to analyze this question by typing in the AppleTV's IP address as a filter 192.168.0.1.



I believe that this made me understand a few of the protocols more that I did just by defining them. If I am correct UDP is able to transport a packet through any port it pleases. This would make sense because Youtube has been around a lot longer than the AppleTV. It would make sense in terms of development that UDP would go through any port. When I was researching the protocol IGMP and all its versions I understood what it meant when it said that it would all go to a similar port. I didn't understand what it would look like in real time. The only other protocol active on the AppleTV's IP address was SSDP which is essentially a check that the WIFI is still connected. This makes perfect sense on the surface. All of the streaming data entered in the IGMP protocols and the SSDP is continuously checking for connection. This is why you get an error message when the AppleTV disconnects!

There were are few other conversations which were mostly my research while doing the homework. There were some encrypted sites that were using certificates and such that showed up in the traffic. In order to discuss them it would be another page and a half and the homework is mandating 2-3 paragraphs which I think I may have already pushed. I am happy to analyze it again if needed.