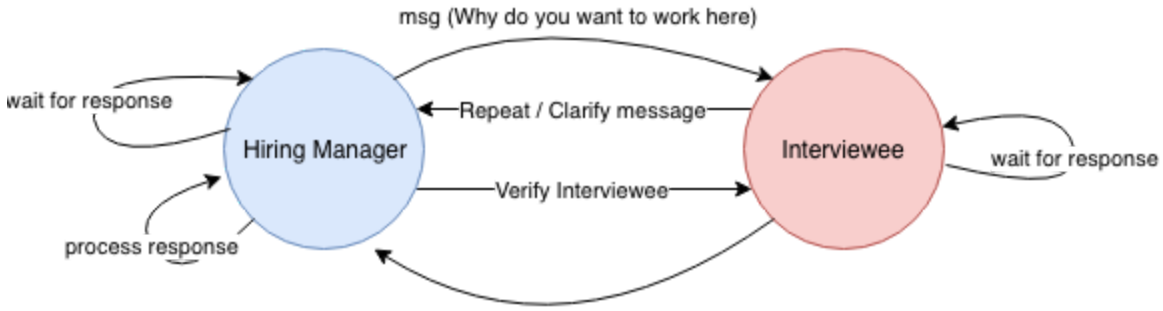
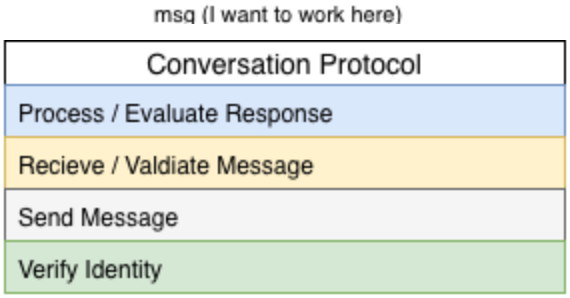
Joseph Mulray
CS472
Homework 1
October 1 2018

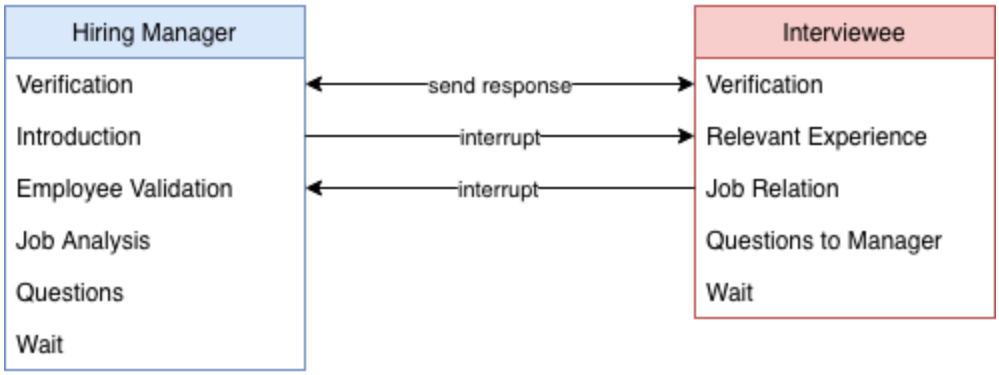
1. [25 points] The goal is to abstract a conversation into formal components. Think about a co-op interview. Imagine the conversation between you and the hiring manager and answer the following questions:

The hiring manager is going to "send" a message to the to the interviewee and wait for a response back. It does not matter in the sense that these roles can be reversed. It makes me think of the "well do you have any questions for me" and in this instance the interviewee will wait for a response from the hiring manager. Messages can be sent and states can be waiting for a response on both sides. Yes there are standard questions and answers. In the case the hiring manager is asking for a specific response from the interviewee, he is looking for a yes or no answer, not looking for something that is unrelated. Each side knows when each other is done when there is no more responses coming through or an interruption is sent. When no more messages are being received that can be considered an ending to a response. Yes there are several different states of conversation. There are several assumptions made in the protocol, one is definitely language. Another one is the ability to hear the response, or the amount of noise that interferes with the message. There are definitely security present in this protocol. There needs to be a way to verify who you are actually interviewing with is that person and not someone else. Also that the messages coming from that interviewee are from him and not someone else and vise versa. There needs to a protocol added for this. If this conversation was on the phone vs on the internet would absolutely change the protocol. There needs to be more verification and security on verifying the interviewee. The same structure of questions and conversation would not change but the layer involving security and verification would change in the type of conversation.





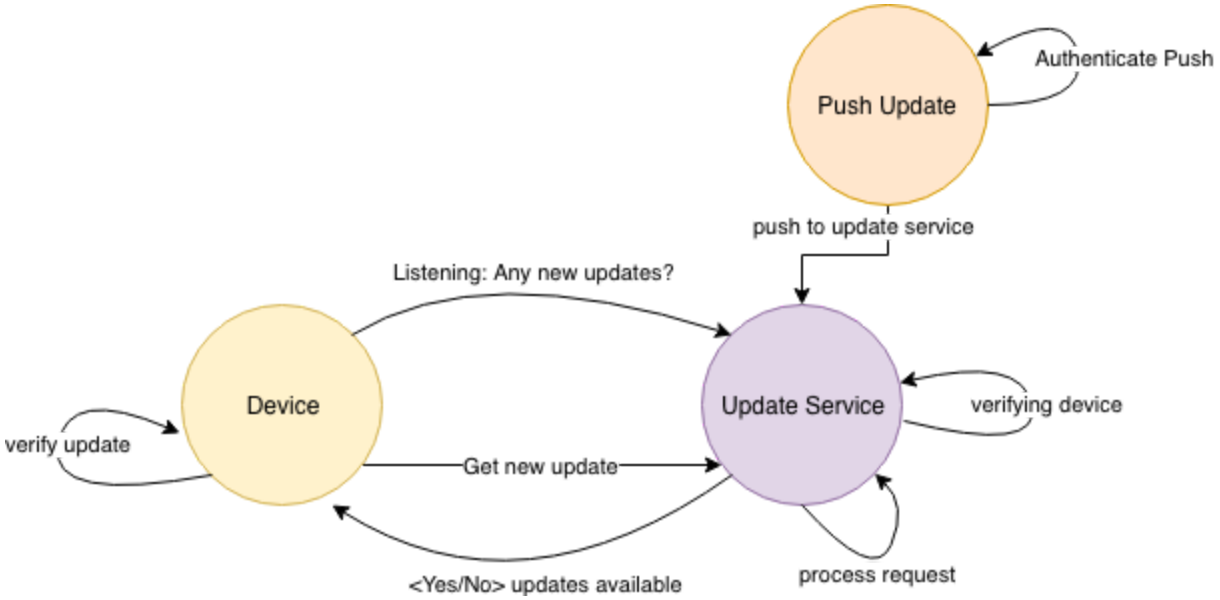
Assumptions:
"Understand / Process Response"
"Receive / Hear Response"
"Validate Response"



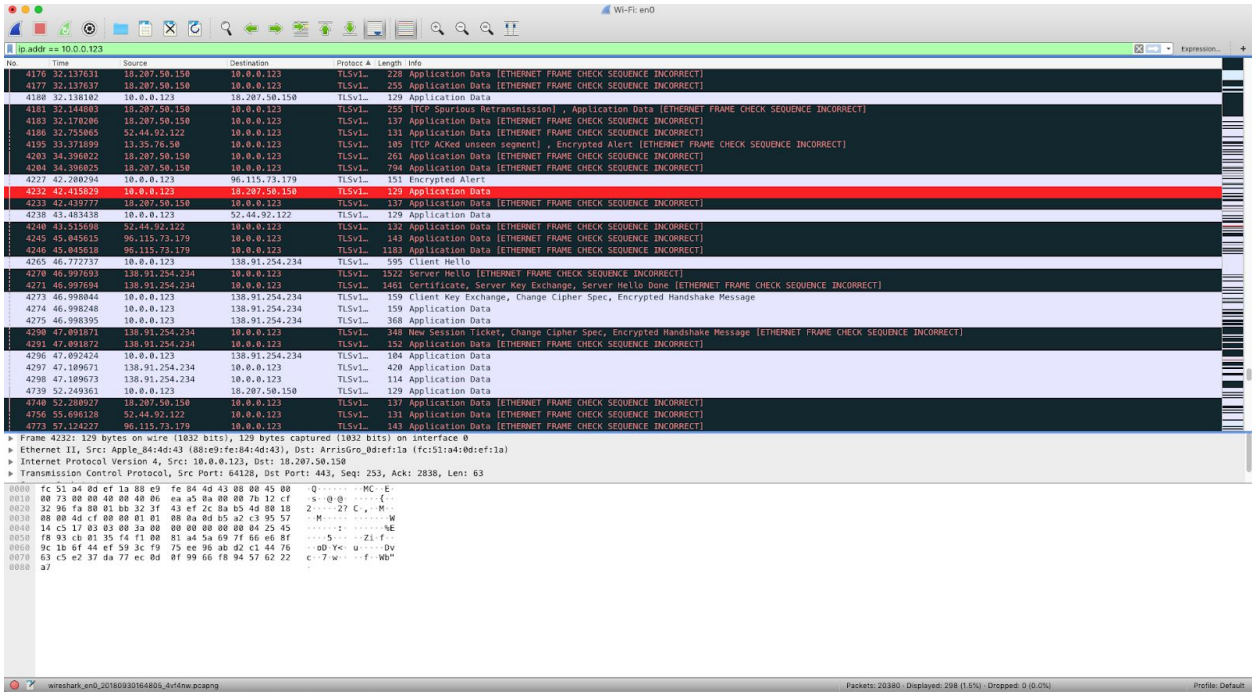
[25 points] Now abstract what you think exists in a protocol. Think of when your computer connects to an update service (Apple iTunes Store, Adobe, Windows Update).
Answer the same questions in question #1.

The computer or device is going to listen for a message from the update service. When an update is needed to be pushed out the update service is going to send a message to a device to pull down the update. There are standard questions and answers to this protocol like sending the version of software in the header maybe one update has to be applied sequentially on some computers, windows 8.0

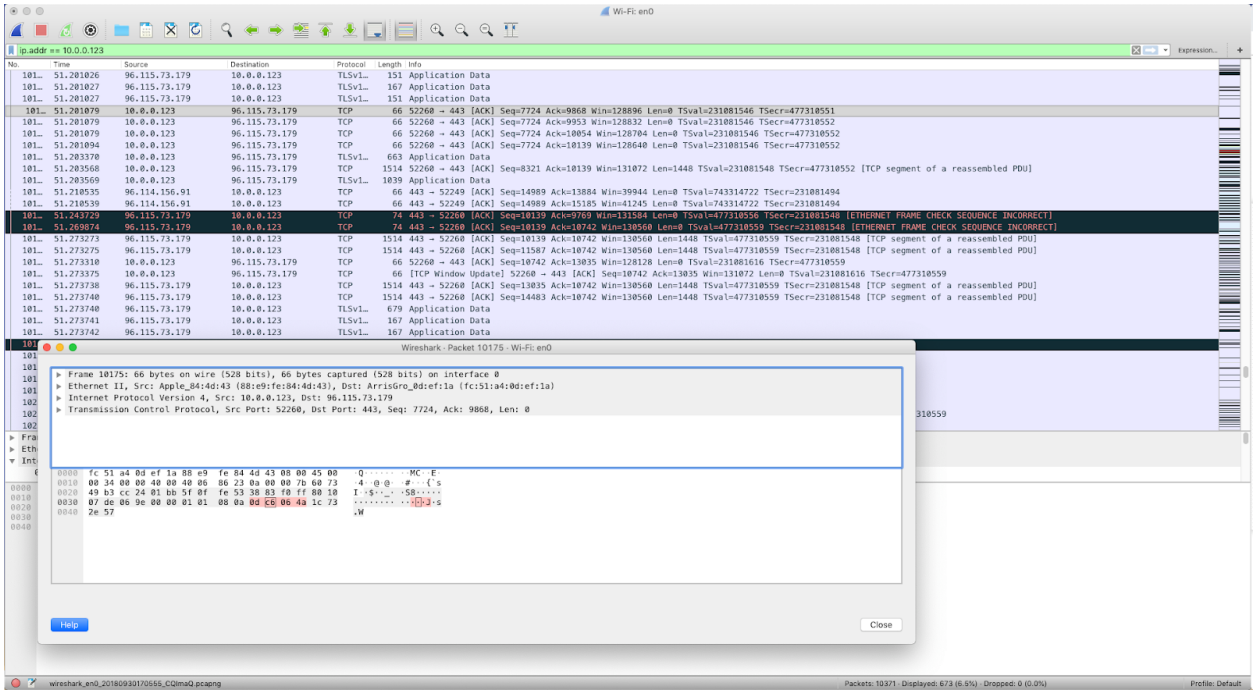
-> 8.2 -> 10. Each side knows when one is done when a status message is sent essentially saying the update is complete, after that the device will continue to listen for updates. There are several assumptions on this protocol, stable connection, verifying the update image is actually the correct image and not a modified one, verifying upgrading one of your devices, not a spoofed device. This ties into the idea of security in this protocols by having verification of the software and device, verifying the update service is actually the update service, there are several layers that can be added to improve security risk for an update service. I think there is a baseline protocol that can be used and extra security can be added for updates that involve the operating system vs application updates, but the listening and waiting for a push will be the same. For updating in person vs online, vs the web protocol on verifying the update needs be modified but the other protocols between the device and listening should not need any changes.



[25 points] Download the free packet analyzer WireShark from http://www.wireshark.org/ and install it on your PC/workstation. Capture packets between your PC and the network (for example, when browsing to a website and one other activity). Analyze the packets (submit screenshot of a web access (HTTP GET) and two other activities, like sending an email, IM and/or BitTorrent, video streaming) and show all of the layers decoded and match them to the TCP/IP model. How is the computer addressed at each layer? How does one layer know which layer to give the information to next? Is there more than one protocol at any layer?



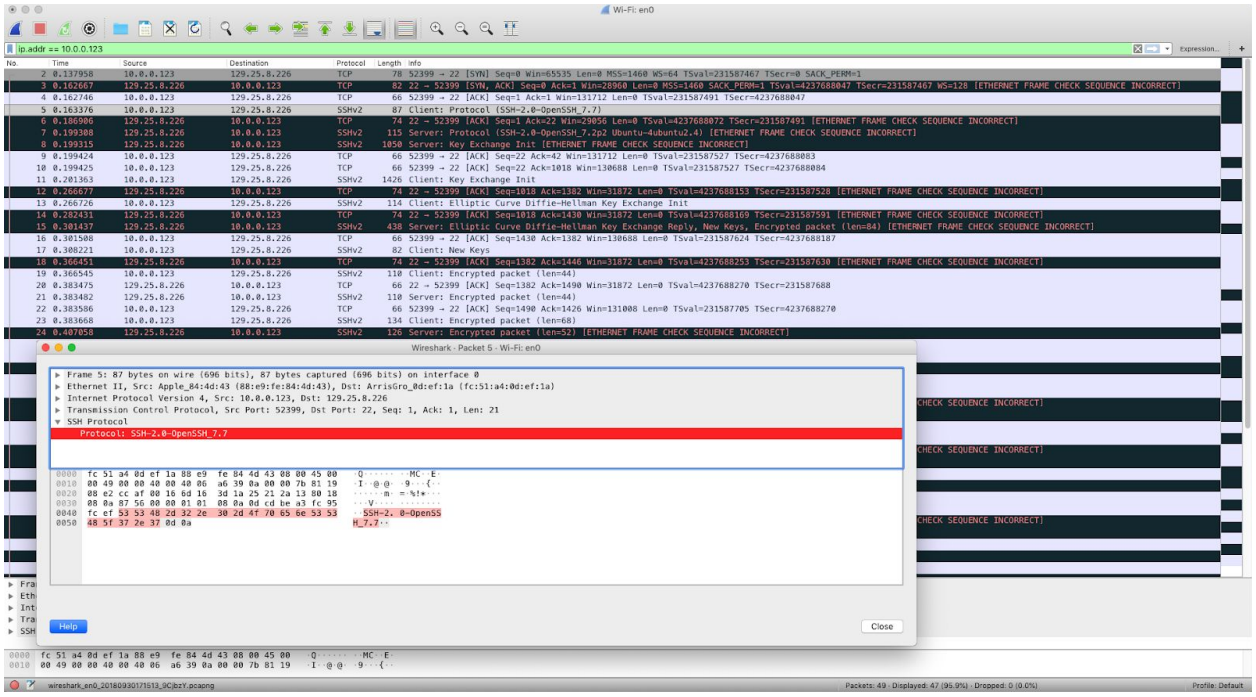
This next screenshot I tried making a HTTP request and seeing the packets come through. I was originally looking for a TCP protocol but after noticing the TLS requests, this made sense because all the the sites I was accessing (YouTube, Facebook) were all HTTPS so the data was being encrypted.



This screenshot is an showing the packets from me sending an email. The thing that interested me was the server actually sending packets back to my laptop which makes me think that there is some processing going on with the address i sent it to, whether its valid or invalid, or if the mail was received to the server.

The computer is addressed at each layer by establishing the network layer of where the destination and source of the packet and then the transport layer in our case TCP to maintain a conversation. There is also multiple protocols being used at each layer TCP and TSL in our case.

Now capture packets of traffic that uses encryption (SSL, SSH, etc.) and analyze the packets. What is encrypted and what isn't? What can WireShark decode and what can't it? Do you think WireShark should be able to decode the rest – and what would it need? Do you think that this could happen?



The SSH packets themselves are encrypted the TCP ack requests are not. I do not think Wireshark should be able to decode SSH and SSL traffic, that would be a serious issue if my information in a key exchange or password are not encrypted and secure.

4. [15 points] What is an RFC

RFC (Request for comments) is a document that describes a specification for internet standards. They contain protocol, programs, research that are reviewed by a

committee and published which includes technologies or previous protocols that it may obsolete. If we were to implement a network protocol we would need to define our protocol model with all reply codes, functions, interactions and what is being accomplished by using this protocol. This includes security and a standard base protocol.

5. [10 points] When you're writing an application – you have a few choices of how to talk with the other copies of your application – TCP and UDP are some of the examples. Are there other ways that you can think in which application talk to each other? What do you think that you must specify to talk to the other application? What is required? What can be optional (options)?

You could write your own application and protocol to communicate to each other udp and tcp are just some of the popular transport layer protocols. It does not have to TCP and UDP you can use SPX or RDP as long as it meets the requirements of the transport layer. To talk to the other application there needs to be a destination to send the data, it is up to the protocol whether or not how secure or reliable that transfer of data is, but that transfer of data is required. Something that is option in this case would be knowing which packets are lost or the security of those packets being sent.

6. [no points for this one] – What do you want to get out of this course? What topics are you most interested in?

I hope to get a better understanding of computer networks from both a higher level and lower level understanding with some hands on practice. Most interested in security, and different types of networks (P2P, P2M).