1. (10 points) This question focuses on the concept of Autonomous System (AS) and Autonomous System Number (ASN). Find a tool (an online one is fine, it does not have to be a command line utility) that allows you to retrieve AS information and answer the following questions:
    1. What is the ASN of the University of Vermont (UVM)?
        i. AS1351
    2. List the IPv4 and IPv6 prefixes that are included in the UVM's AS.
        i. Main v4 Subnets
            1. 64.17.117.192/27
            2. 67.217.113.0/26
            3. 132.198.0.0/16
            4. 192.149.109.0/24
            5. 207.136.213.0/24
            6. 207.136.219.192/27
            7. 207.136.252.128/26
        ii. v6 subnets
            1. 2620:104:e000::/40
    3. Which ASs is the UVM AS connected to?
        i. Receiving traffic from
            1. AS174
            2. AS36351
            3. AS6939
            4. AS13536
            5. AS22652
            6. AS28186
            7. AS10578
            8. AS11537
            9. AS47787
            10. AS11164
    4. Does UVM serve as a transit AS for some other AS?
        i. Forwarding traffic to
            1. AS4134
            2. AS12389
            3. AS20940
            4. AS4755
            5. AS13335
            6. AS9009
            7. AS40676
            8. AS55410

9. AS199524
10. AS8966

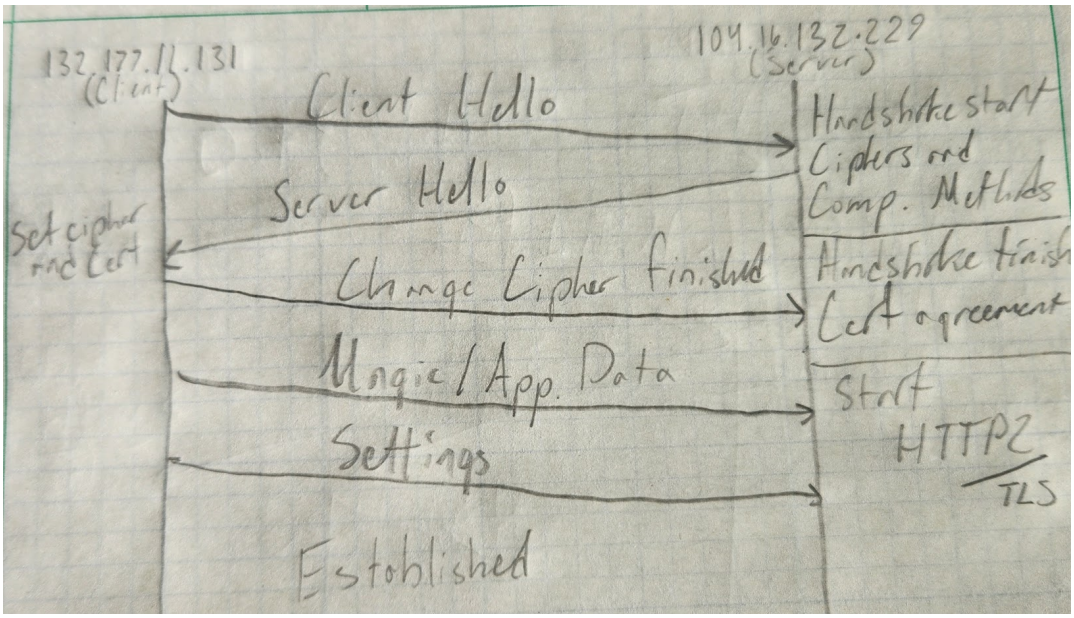
2. (10 points) Run traceroute from a computer on the NH network (e.g., agate) to www.uvm.edu. List the ASs and their ASNs encountered along the way.

```
-bash-5.1$ traceroute -A www.uvm.edu
traceroute to www.uvm.edu (132.198.101.197), 30 hops max, 60 byte packets
 1  _gateway (132.177.4.4) [AS46887/AS11745/AS3549]  0.408 ms  0.368 ms  0.345 ms
        Lighttower Fiber Networks, UNH, Level 3 Parent
 2  csxrtr-1009.cs.unh.edu (132.177.9.33) [AS46887/AS11745/AS3549]  0.793 ms  0.740 ms
0.718 ms
        Lighttower Fiber Networks, UNH, Level 3 Parent
 3  cs01-unh.cs.unh.edu (132.177.9.2) [AS46887/AS11745/AS3549]  0.445 ms  0.494 ms  0.516
ms
        Lighttower Fiber Networks, UNH, Level 3 Parent
 4  fatcat.unh.edu (132.177.100.4) [AS46887/AS11745/AS3549]  1.498 ms  1.503 ms  1.483 ms
        Lighttower Fiber Networks, UNH, Level 3 Parent
 5  unh-re-nox300gw1.nox.org (18.2.0.85) [AS10578/AS3]  4.513 ms  4.525 ms  4.534 ms
        Harvard University, MIT
 6  nox300gw1-uvm-re.nox.org (18.2.0.150) [AS10578/AS3]  11.496 ms  11.790 ms  11.754 ms
        Harvard University, MIT
 7  te0-0-2-3.gw3-gw1.uvm.edu (132.198.255.129) [AS1351]  11.787 ms  11.812 ms  11.664 ms
        UVM Internal
 8  vl55.aa.uvm.edu (132.198.200.19) [AS1351]  11.690 ms  11.658 ms  11.880 ms
        UVM Internal
 9  pc1-2-mann-N77K6.aa.uvm.edu (132.198.200.46) [AS1351]  11.840 ms  12.040 ms  12.345
ms
        UVM Internal
10  www.uvm.edu (132.198.101.197) [AS1351]  11.547 ms  11.999 ms  11.848 ms
        Destination
-bash-5.1$ whoami
rjs1070
-bash-5.1$ hostname
agate.cs.unh.edu
-bash-5.1$
```

3. (20 points) Consider the trace https://www.cloudshark.org/captures/3c138210b8fa that captures an HTTPS session and answer the questions below. The trace captures encrypted traffic, so normally you would not be able to see the application layer payload. The HTTPS request captured in the trace was made by command line utility curl with the environment variable SSLKEYLOGFILE set to make curl output the session key to a

file. As a side note, many browsers and networking libraries can be made to output session keys for network traffic debugging purposes. If you are curious, see this article for more details. The session key was then uploaded to CloudShark together with the trace to enable payload decryption. If you are curious how the same trace looks without access to the key, check https://www.cloudshark.org/captures/82f9df1966e2.

1. What is the version of TLS used in the HTTPS transaction captured in the trace?
    i. TLSv1.3(Frame 8)
2. How many packets does it take to establish the TLS session (i.e., past 3-way handshake all the way to the packet that carries the HTTP HEAD request)?
    i. 5
3. Draw and annotate a ladder diagram showing the packets used to establish the TLS session.



4. Which packets contain the server certificate(s)?
    i. Frame 10, both EDCSA and SHA256 Certificates
5. Who issued the certificate(s)?
    i. Baltimore CyberTrust, and Cloudflare
6. After the secure connection is established, the client starts communicating using HTTP/2. How does the client know that the server is capable of handling HTTP/2?
    i. In the inital server request, the TLS record for the Handshake protocol ‘Extended Extensions’ has a field for App Layer protocol

negotiation, which is set to 2, thus confirming that the server supports HTTP2

7. How many HTTP/2 streams are used during the captured session? What is their purpose?
    i. 2, the first/0th stream was used to establish initial HTTP2 connection information from the client to server, and the second/1st was to inform the client of HTTP status code Moved Permanently from server to client.
8. Does the server support HTTP/3? How is that indicated?
    i. Possibly, although the way that the connection would be initially established would have been different. ALPN doesn't include HTTP 3 as a proper identifier https://www.iana.org/assignments/tls-extensiontype-values/tls-extensiontype-values.xhtml#alpn-protocol-ids
9. In which packet does the client notify the server that the TLS connection is being closed?
    i. Frame 21, by the description of the alert, 'close notify'
10. The TCP connection close captured in the trace does not follow either of the two standard methods. Describe the exchange of packets that close the connection. Try to guess what packet triggers what response.
    i. Frame 22 includes the final FIN-ACK sent by the client, which was sent alongside the TLS close notify packet. These prompted the server to agree on complete communication, and it sends out another FIN-ACK. This FIN-ACK was then acknowledged by the client, and the server also acknowledged the close.