1. HTTP and FTP require a reliable transportation protocol, such as TCP, where data won't be lost, or at least the possibility percentage of losing data is minimal, unlike UDP where data can be lost in transportation without any regard to that loss. It is possible to have a reliable application protocol on top of UDP, such as the DNS application protocol, where the DNS gets resolved on the local machine, and the probability of encountering data loss is minimal on the local machine, therefore the UDP protocol is reliable as a transportation layer for the DNS application protocol.

2. Time sensitive applications suffer the most from transmissions delays. Applications such as trading apps for stock exchange where a millisecond would make a huge difference for the brokers to gain or lose money can suffer the most, and this is one of the reasons why the servers for the trading apps are physically close to the trading centers. Besides that, VoIP apps, screensharing apps, online games, can also suffer from the most from transmissions delays.

3. POP3 is a very simple protocol. There are only 3 states in this protocol which are, authorization, transaction, and update. There are a few commands which are list, retr, and dele. And only two responses ok, and err. Also, there are only two modes for emails, either download-and-delete or download-and-keep where deletion and keeping are on the server-side. However, this simplicity causes a problem where no state information can be shared across different POP3 sessions. There are no remote folders, you download the emails and then create folders locally and assign emails to those local folders, and even if you have a download-and-keep mode, the emails won't be assigned to any folders remotely.

   On the other side, IMAP assigns emails to recipient's inbox folder remotely, it allows users to create new folders and move emails around to those new folders, not just locally, but remotely. It keeps the state information across different IMAP sessions with no struggle. It also allows emails to be searched, remotely, across different folders. Besides that, IMAP has commands that permit the user agents to obtain components of messages, this is very helpful for people with slow internet connections. However, and with all of that in mind, IMAP is a very complicated protocol with a lot of features, which defy the POP3 simplicity.

   IMAP, in terms of features, is superior to POP3, as can be seen, however, it is also quite complicated in comparison to POP3's simplicity.

4. Unlike the non-persistent connections where at least two RTTs are required to for each object reference to download, the persistent connection establishes the three-way handshake one time and then does takes one RTT to download each other object that needs to be downloaded using the same TCP connection. However, on the downside, having multiple persistent connections open can be resources-consuming on the machine, which is a disadvantage if those persistent connections aren’t closed.

5. One of the major DNS vulnerabilities is DDoS bandwidth-flooding attack where people send illegal, malfunctional, repetitive requests to the root server to exhaust the resources of the root server and bring down the website. A method of prevention for this vulnerability would be packet filtering to identify bad and illegal requests. Besides that, this DNS vulnerability can be used on the top-level domain servers, by doing DDoS attack to send tons of requests to the top-level-domain servers, leading to problems with resolving DNS. A method to prevent this vulnerability would be to cache local DNS servers and use the cached resources to respond to the queries.

   Another major DNS vulnerability is the DNS poisoning attack, where the person doing the attack sends bogus messages to a DNS server to trick the server into accepting those bogus messages into its cache. A method to prevent this vulnerability would be to throttle the servers or intercept and filter packets.

6. As a compromise between network reliability and performance, the number thirteen was chosen for the number of root server addresses in the world. Thirteen is based on a constrain of nameserver IPv4. At the same time number of actual servers is much larger (500+), and this was allowed by DNS clustering allowing each root server to have multiple computers linked to each other, which in turn lead to an increase in the reliability of DNS without damaging the performance of the server.

7. By using the dig command and the type option for facebook.com, the following addresses and names were found for each type:
   a. type A: 157.240.18.35
   b. type AAAA: 2a03:2880:f127:283:face:b00c:0:25de
   c. type MX: msgin.vvv.facebook.com

8. There was a total of eight requests before an error was reported by ping saying that “name or service not known”. Those eight requests were divided into four parts, where each part is of 2 requests. Between each part 5 seconds elapsed before the next part started sending its two requests. In total, the requests took 15 seconds before ping failed.

9. I tried resolving facebook.com using Cloudflare’s 1.1.1.1, Google’s 8.8.8.8, and OpenDNS’ 208.67.222.222. The results, in time, to resolve facebook.com were as following:

    a. Cloudflare: 0.013668469 seconds
    b. Google: 0.022037473 seconds
    c. OpenDNS: 0.018931773 seconds

    Those results are following the same trend as in Cloudflare’s announcement. We can see that Cloudflare resolves Facebook in 13 milliseconds, which is pretty close to the 10 milliseconds in the announcements. At the same time, OpenDNS resolves the website in approximately 19 milliseconds, which is also pretty close to the 20 milliseconds in the announcement. However, Google is not as farfetched as in the announcement, where the graph shows Google at approximately 32 milliseconds, while my results show that Google’s DNS resolved facebook.com for me in 22 milliseconds. Although the results for Google are quite different, it is still following the same trend from Cloudflare’s announcement where Google had the highest number of milliseconds.

10. By doing the following command:

    ```
    $ nslookup -type=NS luther.edu 208.67.222.222
    ```

    I was able to find the following name servers:

    ```
    nameserver = dns-2.iastate.edu.
    nameserver = martin.luther.edu.
    nameserver = dns.uni.edu
    ```

    And I was able to extract IP addresses for each nameserver by the following commands

    ```
    $ host dns-2.iastate.edu  →  dns-2.iastate.edu has address 129.186.67.145
    $ host martin.luther.edu  →  martin.luther.edu has address 192.203.196.20
    $ host dns.uni.edu        →  dns.uni.edu has address 134.161.1.32
    ```