1. Application-layer addresses are composed of the host IP address and a port number. Explain why port numbers are needed.

The IP address identifies the computer, but the port number identifies an application running on the computer. Since there are many programs running on the computer at any time, a port number allows incoming data to be delivered to the correct program. The port number is used by the transport protocol to determine the correct receiving application. Standard well-known port numbers of common applications (such as FTP (ports 20 and 21), HTTP (port 80), SSH (port 22), DNS (port 53), IMAP (port 143), etc) are established by the Internet Assigned Numbers Authority (IANA). Each IP address has 65,356 possible port numbers associated with it, so one IP address could potentially handle messages for up to 65,356 applications.

2. The Soccer World Cup 2014 is going to take place in Brazil and you were hired to help get the country's network ready for the event. Suppose that the country is interconnected through a 100 Mbps network. However, the link that connects Brazil to the Internet is only 10 Mbps.

Assume that the delay to retrieve an object from the Internet is on average 3 seconds, while the delay to get an object of similar size residing in the country's network is 10ms. You then decide to install a Web cache in the country's network whose hit rate is 45%.

i. What's the average delay to access an object? Compare it to the initial "cache-less" configuration.

The cache is useful due to the large difference between bandwidth within the country's network, and access to the Internet outside Brazil. The reduction from 100Mbps inside to 10Mbps outside represents a bottleneck that can cause large amounts of delay if there is a high amount of traffic going outside the country. Caching can help reduce delay by keeping some of the Internet traffic inside the country.

\[
\text{delay when hitting the cache} = 10\text{ms} \\
\text{delay when retrieving the object from the Internet (missing the cache)} = 3\text{s} \times 1000 = 3000\text{ms} \\
\text{hit rate} = .45
\]

Average delay with cache:
\[
.55(3000\text{ms}) + .45(10\text{ms}) = 1650\text{ms} + 4.5\text{ms} \\
= 1654.5 \text{milliseconds average delay per access.}
\]

Without cache:
\[
= 3000 \text{milliseconds average delay per access.}
\]

ii. Is there a way to further decrease the average delay? Explain.

Yes. We could try increasing the hit rate of the cache by increasing the cache size. We could also increase the hit rate by installing additional Web caches and distributing requests among them.

3. HTTP uses TCP as its underlying transport protocol.

i. Why do you think the designers of HTTP picked TCP as its transport protocol?

TCP was designed to be robust in the worst possible communication conditions. Its design as a reliable transport protocol enables the use of statistical multiplexing—regardless of how many packets are dropped in the network, TCP ensures that the message will get through, eventually. Having a separate
TCP protocol protects the application from concerning itself with the details of receiving packets and putting them in order. HTTP (detailed in RFC 2616) is concerned with exchanging data, and the transport layer ensures that the data is exchanged completely and correctly. Both protocols can then be less complex than one huge protocol would be.

According to the RFC2616, HTTP usually uses TCP as the transport protocol because TCP offers reliable transfer. A reliable data transfer guarantees that packets will be received in order and will be uncorrupted.

ii. What is the consequence of using TCP when considering delay?
The reliability which TCP guarantees can cause increased delay due to congestion control and flow control efforts. TCP also increases the amount of traffic, since it requires additional messages to establish reliable communication, duplicate messages, and ACK messages. Without these controls, however, even more delay would result from a flow or congestion event.

iii. What are the trade-offs if HTTP used UDP instead?
As mentioned before, HTTP had assumed that the transport protocol it uses would be reliable. UDP is a connectionless protocol and would ideally transfer data faster. However, since UDP is unreliable and only a “best-effort“ protocol, the designers of HTTP would have to design mechanisms to handle reliable data transfer if UDP were used.

4. A Web page has 7 embedded objects. Assume that the time to transmit the page and its embedded objects is negligible when compared to the propagation delay and service time within the network, which combined are approximately 150ms.

What is the total time perceived by the user between clicking on a link and having the entire object rendered if the user’s browser employs:

i. Non-persistent HTTP.
We assume that each embedded object can be contained in one packet. Each time we send a packet across the network (one way), we incur a delay of 150ms. Note: Data can piggyback requests for objects on the ACKs from client to the server.

Let X be the time for establishing 7 TCP connections = 7 * 3 * 150ms
Let Y be time for Web server to send 7 embedded objects = 7 * 150ms
Total time = X + Y = 4200ms.

ii. Persistent HTTP.
Let X be the time for establishing 1 TCP connection = 3 * 150ms
Let Y be the time to for Web server to send 7 objects and receive 7 ACKs = 14 * 150ms
Total time = X + Y = 2550ms.

5. Describe how caching is used in DNS to improve performance.
When a client needs to obtain the IP address of a name, it sends a DNS query. The DNS server and the client can cache the resolved name, so that when a similar query is received, the DNS lookup can be handled faster.

6. DNS may also use a feature called "negative caching".
i. What is negative caching (we haven’t covered it in class, so you will need to look it up).

In general, a negative cache is when a cache stores a failure of an attempted operation. If the operation is attempted again, the cache will return a failure message, instead of having the system attempt the operation over again. According to RFC 1034 (the DNS standard), DNS servers can optionally cache unsuccessful attempts to resolve an address, so that any future servers that attempt to resolve the address will receive the error from the DNS server, thus saving resources that would otherwise have been wasted attempting to resolve the non-existent address.

Negative caching occurs when DNS servers cache the results of unsuccessful name resolution attempts.

ii. How does it improve performance?

If the DNS server already knows that a name cannot be resolved, it won’t waste time and resources trying to resolve the name again.

7. For each of the Web requests below, list the sequence of steps to resolve the Web server’s name. Assume names are not initially cached (but could be cached as they are accessed) and name resolution is done recursively.

i. Web client at ucsc.edu is requesting an object from www.ucsc.edu.
   The web client at UCSC contacts the local DNS server for UCSC.
   The local DNS server returns the IP address of www.ucsc.edu to the web client.


   1. The web client at UCSC queries the local DNS server for www.comcast.com
   2. The local DNS server queries the root domain server for www.comcast.com
   3. The root DNS server queries a top level domain DNS server for www.comcast.com
   4. The top level domain DNS server for .com queries authoritative DNS server of Comcast for www.comcast.com
   5. The authoritative DNS server returns the IP address of www.comcast.com to the top level domain DNS server.
   6. The top level domain server returns IP address to the root DNS server.
   7. The root DNS server returns the IP address to the local DNS server of UCSC.
   8. The local DNS server returns the IP address to the web client.

iii. Another Web client at ieee.org is requesting the same object as above from www.comcast.org.
   The same process would occur, but the search would end with the top level domain DNS server for .org, rather than with the root server.