# Application Layer

Annalise Tarhan

November 7, 2020

## 1 True or False

### 1.1 A user requests a webpage that consists of some text and three images. For this page, the client will send one request message and receive four response messages.

False. The client will send separate requests for each image.

### 1.2 Two distinct web pages can be sent over the same persistent connection.

True, as long as the web pages reside on the same server.

### 1.3 With nonpersistent connections between browser and origin server, it is possible for a single TCP segment to carry two distinct HTTP request messages.

False. Depending on browser settings, clients can send requests and receive responses in parallel instead of serially, but only one per segment.

### 1.4 The *Date* header in the HTTP response message indicates when the object in the response was last modified.

False. It indicates when the response was created and sent by the server. There is a separate "last modified" field.

### 1.5 HTTP response messages never have an empty message body.

False. The "HEAD" method returns an empty body.

## 2 What protocols do SMS, iMessage, and WhatsApp use? How do they differ?

SMS, short message service, is a text messaging service that is standard on mobile phones. Its protocol was originally defined as part of GSM, the Global System for Mobile Communications, and is realized by the Mobile Application Part of the SS7 set of telephony signaling protocols. MAP provides an application layer for nodes in GSM networks. SMS is now available on mobile devices using different protocols, including CDMA.

iMessage is Apple's proprietary messaging system. It is based on Apple Push Notification service, which sends notifications to Apple devices.

WhatsApp uses a version of the open standard Extensible Messaging and Presence Protocol, which was developed by the open-source community.

SMS and WhatsApp users are identified by their phone numbers, while iMessage users are identified by their Apple accounts. SMS messages are sent by wireless carriers to recipients who may or may not be customers of the same company. iMessage and WhatsApp, on the other hand, only deliver messages to other iMessage and WhatsApp users through their proprietary applications. SMS is tied to a phone's data plan and can only send text messages, while iMessage and WhatsApp can handle any form of text, image, or video, as well as voice or video calls, and messages and calls can be sent over WiFi.

## 3 Consider an HTTP client that wants to retrieve a Web document at a given URL, without knowing the IP address of the HTTP server. What transport and application-layer protocols besides HTTP are needed in this scenario?

To access the document, the client first needs to find the IP address of the server using DNS, which is an application-layer protocol that runs on UDP. To retrieve the document, HTTP will use TCP.

# 4 Consider the string of ASCII characters that were captured by Wireshark.

## 4.1 What is the URL of the document requested by the browser?

gaia.cs.umass.edu/cs453/index.html
This is a concatenation of the Host field and the path following the GET method.

## 4.2 What version of HTTP is the browser running?

HTTP/1.1
This follows the path and precedes the Host field.

## 4.3 Does the browser request a non-persistent or a persistent connection?

Persistent. The Connection field says keep-alive, not close.

## 4.4 What is the IP address of the host on which the browser is running?

No way to know.

## 4.5 What type of browser initiates this message? Why is the browser type needed in an HTTP request message?

Mozilla/5.0. Including the browser type allows server to send the most appropriate version of the object when it has different versions available.

# 5 Consider the text below, which shows the response sent from the server in response to the previous HTTP GET request.

## 5.1 Was the server able to find the document? What time was the document reply provided?

Yes it was, since the status is 200 OK. It was provided at 12:39:45 GMT.

## 5.2 When was the document last modified?

It was last modified on Saturday December 10, 2005 at 18:27:46 GMT, as indicated by the Last-Modified header line.

### 5.3 How many bytes are there in the document being returned?

3874, according to the Content-Length header line.

### 5.4 What are the first 5 bytes of the document being returned? Did the server agree to a persistent connection?

<!doc
Yes, it did, as indicated by "Connection: Keep-Alive."

## 6 Obtain the HTTP/1.1 specification.

### 6.1 Explain the mechanism used for signaling between the client and server to indicate that a persistent connection is being closed. Can the client, the server, or both signal the close of a connection?

By default, connections remain open, but both client and server can signal the close of a connection using the Connection header field. Once one party sends the close token, that request becomes the last one for the connection.

### 6.2 What encryption services are provided by HTTP?

None

### 6.3 Can a client open three or more simultaneous connections with a given server?

The RFC states that a client should not maintain more than 2 connections with any server or proxy.

### 6.4 Either a server or a client may close a transport connection between them if either one detects the connection has been idle for some time. Is it possible that one side starts closing a connection while the other side is transmitting data via this connection?

Yes, it is possible, so both sides must be able to recover if it happens.

## 7 Suppose you click on a link to a webpage, but the IP address of the URL is not already cached. Suppose $n$ DNS servers are visited in order to obtain the IP address, taking $RTT_i$ time each, $i \in 1..n$. Further, let $RTT_0$ indicate the round trip time between your host and the server containing the object. Assuming zero transmission time of the object, how much time elapses from when the client clicks on the link until the client receives the object?

$2 * RTT_0 + \sum_{i=1}^{n} RTT_i$
It takes two round trips to obtain the object, since the TCP connection requires a handshake first.

## 8 Suppose the HTML file in the previous question references eight very small objects on the same server. Neglecting transmission times, how long does it take in each of the following cases?

### 8.1 Non-persistent HTTP with no parallel TCP connections?

$2 * RTT_0 + 2 * 8RTT_0 = 18RTT_0$

### 8.2 Non-persistent HTTP with the browser configured for 5 parallel connections?

$2 * RTT_0 + 2 * 2 * RTT_0 = 6RTT+_0$

### 8.3 Persistent HTTP?

$2 * RTT_0 + 8 * RTT_0 = 10RTT_0$
Or, with pipelining, $2 * RTT_0 + 1 * RTT_0 = 3RTT_0$

# 9 Consider an institutional network connected to the internet. Suppose that the average object size is 850,000 bits, the average request rate is 16 requests per second, and the time it takes from when the router on the internet side forwards an HTTP request until it receives the response is three seconds on average. The average delay between the institution router and the internet router is $\mathit{\Delta}/(1-\mathit{\Delta}\beta)$ where $\mathit{\Delta}$ is the average time required to send an object over the access link and $\beta$ is the arrival rate of objects to the access link. The access link is 15 Mbps.

## 9.1 Find the total average response time.

average response time = avg access delay + avg internet delay<br>
avg access delay = $\mathit{\Delta}/(1-\mathit{\Delta}\beta)$<br>
$\mathit{\Delta}$ = avg time to access link<br>
$\mathit{\Delta} = 850,000\ bits * \frac{Mb}{10^6\ bits} * \frac{1\ sec}{15\ Mb} = .0567$ seconds<br>
$\beta$ = arrival rate of objects to access link<br>
$\beta$ = 16 objects per second<br>
$\mathit{\Delta}/(1-\mathit{\Delta}\beta)$<br>
$.0567/(1-.0567*16) = .61$ seconds

avg internet delay = 3 seconds<br>
average response time = .61+3 = 3.61 seconds

## 9.2 Now suppose a cache is installed and the miss rate is 0.4. What is the total response time?

average response time = .6(0) + .4(avg access delay + avg internet delay)<br>
avg access delay = $\mathit{\Delta}/(1-\mathit{\Delta}\beta)$<br>
$\mathit{\Delta}$ = avg time to access link = .0567 seconds<br>
$\beta$ = arrival rate of objects to access link = total rate of objects * miss rate<br>
$\beta = \frac{16\ objs}{sec} * .4 = 6.4$ objects per second<br>
$\mathit{\Delta}/(1-\mathit{\Delta}\beta) = .0567/(1-.0567*6.4) = .089$ seconds

avg internet delay = 3 seconds<br>
average response time = .6(0) + .4(.089+3) = 1.24 seconds

## 10 Consider a short 10-meter link with a transmission rate of 150 bits per second. Suppose that data packets are 100,000 bits long and control packets are 200 bits long. Assume $N$ parallel connections each get $1/N$ of the link bandwidth. Using the HTTP protocol with downloaded objects of 100 Kbits where the initial object contains 10 object references, would parallel downloads via parallel instances of non-persistent HTTP make sense in this case? Would persistent HTTP? Would there be significant gains over the non-persistent case?

In this case, the bottleneck is the extremely slow transmission rate. The extra handshakes over the non-persistent connection and any gains from sending the objects in parallel are insignificant by comparison.

There are two types of packets being sent in this scenario: small control packets and large data packets. Since the connections aren't persistent, each request involves two control packets for a handshake, one control packet for the request, and a final data packet to send the object. The time it takes to send each is the number of bits divided by the available bandwidth.

Time for a control packet, all bandwidth available: $200/150 = 1.33$ seconds
Time for a control packet, 10% of bandwidth: $200/(150/10) = 13.3$ seconds
Time for an object packet, all bandwidth available: $100000/150 = 666.7$ seconds
Time for an object packet, 10% of bandwidth: $100000/(150/10) = 6667$ seconds

Time to receive initial object: $3 * 1.33 + 666.7 = 671$ seconds
Time to receive next ten objects: $3 * 13.3 + 6667 = 6707$ seconds

Total time for parallel downloads on non-persistent connection: $671 + 6707 = 7378$ seconds

For a persistent connection, only the first handshake is required, but the request control packets still are. This means the total time is $3 * 1.33 + 666.7 + 13.3 + 6667 = 7351$ seconds.

# 11 Continuing with the previous scenario, suppose the link is shared by Bob with four other users. Bob uses parallel instances of non-persistent HTTP, and the other four users use non-persistent HTTP without parallel downloads.

## 11.1 Do Bob's parallel connections help him get webpages more quickly?

Yes. Because the bandwidth is distributed evenly to each connection, Bob's parallel connections give him a larger share.

## 11.2 If all five users open five parallel instances of non-persistent HTTP, then would Bob's parallel connections still be beneficial?

Yes, but only to defend an equal share of the available bandwidth, not to get any more than any of the other four users.

# 12 Write a simple TCP program for a server that accepts lines of input from a client and prints the lines onto the server's standard output.

```
from socket import *
serverPort = 12001
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((‘’, serverPort))
serverSocket.listen(1)
print(‘The server is ready to receive’)
while True:
        connectionSocket, addr = serverSocket.accept()
        sentence = connectionSocket.recv(1024).decode()
        print(sentence)
        connectionSocket.close()
```

## 13 What is the difference between “MAIL FROM:” in SMTP and “From:” in the mail message itself?

MAIL FROM is used as a command, identifying the beginning of a new message. The mail address it identifies is the same as that in the From in the mail message.

## 14 How does SMTP mark the end of a message body? How about HTTP? Can HTTP use the same method as SMTP to mark the end of a message body?

SMTP marks the end of a message with a line containing a single period. HTTP messages typically include a ‘Content-length’ header, so the receiver knows exactly how many bytes to expect. The only exception is that when it is time to close the connection, hosts can omit the ‘Content-length’ header and close the connection instead. No, HTTP can’t use the same method as SMTP, because the HTTP message could be in any format, as opposed to SMTP, which must be in 7-bit ASCII.

## 15 Read RFC 5321 for SMTP. What does MTA stand for? Identify the malicious host that generated the spam e-mail below.

MTA stands for mail transfer agent. When they are used, the originator of the message sends the message to an MTA, which sends it on to the recipients with its own address as the return path. This avoids the inevitable error messages generated by a mass mailing from arriving in the inbox of the originator. In this message, the generator of the spam e-mail is asusus-4b96 ([58.88.21.177]).

## 16 Read the POP3 RFC, RFC 1939. What is the purpose of the UIDL POP3 command?

UIDL, which stands for “unique-id listing,” is a command to retrieve metadata. It takes an optional message-number as an argument and returns either a line containing information about the indicated message or a line for each message in the mailbox if no message-number was given. The information line consists of the message-number of the message and the unique-id of the message.

# 17 Consider accessing e-mail with POP3.

## 17.1 Suppose you've configured your POP mail client to operate in download-and-delete mode. Complete the following transaction.

```
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: blah blah ...
S: ... blah
S: .
C: dele 1
C: retr 2
S: ...
C: dele 2
C: quit
S: +OK POP3 server signing off
```

## 17.2 Suppose you've configured your POP mail client instead to operate in download-and-keep mode. Complete the following transaction.

```
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: blah blah ...
S: ... blah
S: .
C: retr 2
S: blah blah ...
S: ... blah
S: .
C: quit
S: +OK POP3 server signing off
```

### 17.3 Suppose you've configured your POP mail client to operate in download-and-keep mode and that five minutes after the session in part (2), you again access POP to retrieve new e-mail. Suppose that in the five-minute interval no new messages have been sent to you. Provide a transcript of this session.

Since no new messages have arrived, this session is identical to the previous one. It retrieves each message regardless of whether it has already been downloaded.

```
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: blah blah ...
S: ... blah
S: .
C: retr 2
S: blah blah ...
S: ... blah
S: .
C: quit
S: +OK POP3 server signing off
```

## 18

### 18.1 What is a *whois* database?

A database containing contact and registration information for domain names.

### 18.2 Use various whois databases on the Internet to obtain the names of two DNS servers. Indicate which whois databases you used.

ab-31-204-180-44.mxc.ru
31.204.180.44
whois.domaintools.com: Ltd "Maxima"
whois.net: Ltd "Maxima"

ns-track5-a.siteage.net
81.7.4.5
lookup.icann.org: Whois Privacy Corp.
whois.domaintools.com: Whois Privacy Corp.

## 18.3 Use nslookup on your local host to send DNS queries to three DNS servers: your local DNS server and the two DNS servers you found in part (2). Try querying for Type A, NS, and MX reports. Summarize your findings.

Basic queries
The first DNS server, Ltd “Maxima,” returned the same information as the local DNS server for my query, but “Whois Privacy Corp.” couldn’t be reached.

Type=“A”
The results were exactly the same. Ltd “Maxima” and the local DNS server returned the same information as they did without a type argument.

Type=“NS”
The local server and Ltd “Maxima” returned the name server in the “Non-authoritative answer” section, but Ltd “Maxima” also returned a list of servers in the “Authoritative answers can be found from” section.

Type=“MX”
The results were similar to type NS. The local server returned the non-authoritative answer with an address to a mail exchanger, and the Ltd “Maxima” server additionally returned a list of servers where authoritative answers can be found.

## 18.4 Use nslookup to find a Web server that has multiple IP addresses. Does the Web server of your institution have multiple IP addresses?

berkeley.edu has a different IP address for each school: cs.berkeley.edu, ieor.berkeley.edu, and engineering.berkeley.edu all have IP addresses different from berkeley.edu.

## 18.5 Use the ARIN whois database to determine the IP address range used by your university.

AIRN doesn’t seem to return a result.

## 18.6 Describe how an attacker can use whois databases and the nslookup tool to perform reconnaissance on an institution before launching an attack.

Whois databases and nslookup can be used to create a list of related entities and their IP addresses, as well as all IP addresses used by an institution. Multiple points of entry can make the attack more effective or more likely to be effective.

### 18.7 Discuss why whois databases should be publicly available.

Whois databases should be publicly available for the same reason addresses and phone numbers should be publicly available. They can be used maliciously, but they are also important for accountability.

## 19 Read the man page for *dig*.

### 19.1 Starting with a root DNS server, initiate a sequence of queries for the IP address for your department's Web server by using *dig*. Show the list of the names of DNS servers in the delegation chain in answering your query.

g.root-servers.net
i.edu-servers.net
adns1.berkeley.edu

### 19.2 Repeat part (a) for several popular websites.

nyt.com:
b.root-servers.net
a.gtld-servers.net
dns2.p06.nsone.net

reddit.com:
k.root-servers.net
d.gtld-servers.net
ns-1887.awsdns-43.co.uk

netflix.com:
m.root-servers.net
h.gtld-servers.net
ns-1984.awsdns-56.co.uk

## 20 Suppose you can access the caches in the local DNS servers of your department. Can you propose a way to roughly determine the Web servers that are most popular among the users in your department? Explain.

If you can see which servers are most recently accessed, that will roughly represent their popularity. Less frequently accessed servers will still appear in the cache if accessed recently enough, but they won't stay near the top for very long.

## 21 Suppose your department has a local DNS server for all computers in the department. Can an ordinary user determine if an external website was likely accessed from a computer in your department a couple of seconds ago?

The best approach would be to use the dig command. If the result is cached because it has been accessed recently, the query time will be very small.

## 22 Consider distributing an $F = 15$ Gbit file to $N$ peers. The server has an upload rate of $u_s = 30$ Mbps and each peer has a download rate of $d_i = 2$ Mbps and an upload rate of $u$. For $N = 10,\ 100,$ and $1,000$ and $u = 300$ Kbps, 700 Kbps, and 2 Mbps, prepare a chart giving the minimum distribution time for each combination for both client-server distribution and P2P distribution.

For a client-server architecture, the time to distribute the file is the maximum of the time it takes for the server to upload $N$ copies of the file and the time it takes for the slowest client to download one copy: $D_{cs} = max\{N * F/u_s, F/d_{min}\}$ The upload speed of each client is irrelevant.

| N | Time |
|---|---|
| 10 | 128 min |
| 100 | 853 min |
| 1,000 | 142 hours |

For a P2P architecture, the time to distribute the file is the maximum of the time it takes for the server to distribute on copy of the file, the time it takes for the slowest peer to download the file, and the time it takes for the entire system to upload $N$ copies of the file: $D_{P2P} = max\{F/u_s, F/d_{min}, NF/(u_s + \sum_{i=1}^{N} u_i)\}$. Where the time is 128 minutes, the bottleneck is the slowest download. For the rest, the bottleneck is the the time to upload $N$ copies of the file.

| N\u | 300 Kbps | 700 Kbps | 2 Mbps |
|---|---|---|---|
| 10 | 128 min | 128 min | 128 min |
| 100 | 432 min | 260 min | 128 min |
| 1,000 | 793 min | 358 min | 128 min |

# 23 Consider distributing a file of $F$ bits to $N$ peers using a client-server architecture. Assume a fluid model where the server can simultaneously transmit to multiple peers, transmitting to each peer at different rates, as long as the combined rate does not exceed $u_s$.

## 23.1 Suppose that $u_s/N \leq d_{min}$. Specify a distribution scheme that has a distribution time of $NF/u_s$.

Split the transmission time allotted to each peer equally, with a rate of $u_s/N$.

## 23.2 Suppose that $u_s/N \geq d_{min}$. Specify a distribution scheme that has a distribution time of $F/d_{min}$.

Split the transmission time allotted to each peer equally, with a rate of $d_{min}$.

### 23.3 Conclude that the minimum distribution time is in general given by $max\{NF/u_s, F/d_{min}\}$.

If $u_s/N \leq d_{min}$, the minimum distribution time is $NF/u_s$ and $NF/u_s \geq F/d_{min}$. Conversely, if $u_s/N \geq d_{min}$, the minimum distribution time is $d_{min}$ where $NF/u_s \leq F/d_{min}$. Since at least one must be true, then in all cases, the distribution time $max\{NF/u_s, F/d_{min}\}$.

## 24 Consider distributing a file of $F$ bits to $N$ peers using a P2P architecture. Assume a fluid model. Assume that $d_{min}$ is very large, so that peer download bandwidth is never a bottleneck.

### 24.1 Suppose that $u_s \leq (u_s + u_1 + ... + u_N)/N$. Specify a distribution scheme that has a distribution time of $F/u_s$.

The server should split the file into $N$ pieces with sizes proportional to each peer's upload speed. Once those pieces have been distributed into the system, most of the work will be done by the faster peers.

### 24.2 Suppose that $u_s \geq (u_s + u_1 + ... + u_N)/N$. Specify a distribution scheme that has a distribution time of $NF/(u_s + u_1 + ... + u_N)$.

Since the server's upload speed is faster than the average peer's upload speed, it will continue doing significant work, represented by the parts of the file it keeps to itself. Again, it splits the file into $N$ pieces proportional to the peers' upload speeds, but this time it reserves part of the file to distribute itself after those $N$ pieces have been distributed into the system.

### 24.3 Conclude that the minimum distribution time is in general given by $max\{F/u_s, NF/(u_s + u_1 + ... + u_N)\}$.

Since either $u_s \geq (u_s+u_1+...+u_N)/N$ or $u_s \leq (u_s+u_1+...+u_N)/N$ must be true, the minimum distribution time is given by either $F/u_s$ or $NF/(u_s+u_1+...+u_N)$, whichever is larger.

# 25 Consider an overlay network with $N$ active peers, with each pair of peers having an active TCP connection. Additionally, suppose that the TCP connections pass through a total of $M$ routers. How many nodes and edges are there in the corresponding overlay network?

Each user is a node and each user-user connection is an edge. In an overlay network with $N$ peers, there are $N$ nodes and $N(N-1)/2$ edges.

# 26 Bob joins a BitTorrent torrent, but doesn't want to upload any data.

## 26.1 Bob claims that he can receive a complete copy of the file that is shared by the swarm. Is that possible?

Yes, but it will be slow. Since peers randomly choose peers to optimistically unchoke, he will eventually receive enough data to form a complete copy of the file. He will never be in anyone's top four, though, so optimistic unchoking is the only way he will receive data.

## 26.2 Bob further claims that he can further make his free-riding more efficient by using a collection of multiple computers in the computer lab in his department. How?

All of them will receive data from optimistic unchoking, but if they are operating separately, they won't receive the data efficiently, since each computer will be trying to assemble its own complete set and there will be overlap. He could try to share that data in a private torrent network, or he could just play nice with everyone else.

## 27 Consider a DASH system for which there are $N$ video versions with different rates and qualities and $N$ audio versions with different rates and qualities. Suppose we want to allow the player to choose at any time any of the $N$ video versions and any of the $N$ audio versions.

### 27.1 If we create files so that the audio is mixed in with the video, so the server sends only one media stream at a given time, how many files will the server need to store?

$N$

### 27.2 If the server instead sends the audio and video streams separately and has the client synchronize the streams, how many files will the server need to store?

$2N$

## 28 Install and compile the TCPClient and UDP Client on one host and TCPServer and UDPServer on another host.

### 28.1 Suppose you run TCPClient before you run TCPServer. What happens? Why?

The server is not available to complete the handshake, so the connection won't be made.

### 28.2 Suppose you run UDPClient before you run UDPServer. What happens? Why?

As long as the UDPServer is running by the time data is sent, there won't be a problem.

### 28.3 What happens if you use different port numbers for the client and server sides?

The client and server won't connect.

## 29 Suppose that in UDPClient.py, after we create the socket, we add the line: $clientSocket.bind(('', 5432))$ Will it be necessary to change UDPServer.py? What are the port numbers for the sockets in UDPClient and UDPServer? What were they before making this change?

No, it will not be necessary to change the server. The server is informed of the client's address during serverSocket.recvfrom(), and it accepts any address the client gives it. UDPClient and UDPServer both set serverPort to 12000 and changing the client's port number doesn't affect the server's port number. Before the change, the operating system sets the client's port number and the client relayed that information to the server. Now, it will relay the new port number, 5432, instead.

## 30 Can you configure your browser to open multiple simultaneous connections to a website? What are the advantages and disadvantages of having a large number of simultaneous TCP connections?

Chrome allows up to six simultaneous TCP connections to a single domain. The advantage is that files can be downloaded faster. The disadvantage is that it can become inefficient and hog resources.

## 31 We have seen that Internet TCP sockets treat the data being sent as a byte stream but UDP sockets recognize message boundaries. What are one advantage and one disadvantage of a byte-oriented API versus having the API explicitly recognize and preserve application-defined message boundaries?

The obvious disadvantage of sending discrete packets is the extra overhead of packaging them with enough data for the boundaries to be recognizable. The advantage is that for applications where discrete packets are a natural fit, such as messages, the boundaries already exist and the application itself doesn't have to work out a system for replicating them.

## 32 What is the Apache Web server? How much does it cost? What functionality does it currently have?

Apache is free, open-source web server software that serves almost 30% of the world's busiest websites. It implements many features, such as authentication, as compiled modules.