Name Ni, Tianshu Date: 2023.10.12

(last name, first name)

NYU ID: N16371283

Course Section: 001

**Assignment 3**

**Total in points** (100 points total):

**Professor’s Comments:**

**Affirmation of my Independent Effort:** Tianshu Ni

**(Sign here)**

* 1. Problem 1:
     1. What transport layer protocol SMTP uses and why?

It uses TCP as its transport layer protocol. The reason is that TCP provides a reliable, connection-oriented communication which ensures that the email messages are delivered in the correct order without errors. SMTP requires this reliability to ensure that email messages are transmitted accurately and in their entirety.

* + 1. Research and describe the protocols used in WhatsApp and iMessage?

- WhatsApp: The app primarily uses the XMPP for its messaging. For voice and video calls, WhatsApp uses the RTP. The communication is end-to-end encrypted using the Signal Protocol.

- iMessage: Apple's iMessage uses the Apple Push Notification Service to send messages. It also uses end-to-end encryption to protect the content of the messages, ensuring that only the sender and the recipient can read them.

* + 1. Briefly describe two different server placement approaches in CDN.

1) Proactive Server Placement: In this approach, content servers are placed at locations anticipating potential user demand, even before there's any actual demand. This can be based on predictions about where demand will likely grow.

2) Reactive Server Placement: Here, servers are placed or content is replicated in response to observed user demand. As the demand in a particular region grows, the CDN will place a server there or replicate more content to an existing server.

* + 1. Explain the meaning of HTTP response codes:

1) 201: Created. This means that the request has been fulfilled and has resulted in one or more new resources being created

2) 403: Forbidden. The server understands the request, but it refuses to authorize it.

3) 500: Internal Server Error. This is a generic error message, given when an unexpected condition was encountered, and no more specific message is suitable.

* + 1. Describe the difference between download-and-delete mode vs. download-and- keep mode in POP3.

- Download-and-delete mode: Messages are downloaded to the client and then deleted from the email server. This means once the email is downloaded, it only exists on the client device, and not on the email server.

- Download-and-keep mode: Messages are downloaded to the client but remain on the email server as well. This allows the user to access the same emails from different devices since the emails still exist on the central server.

* + 1. Name two benefits of P2P over client-server architecture.

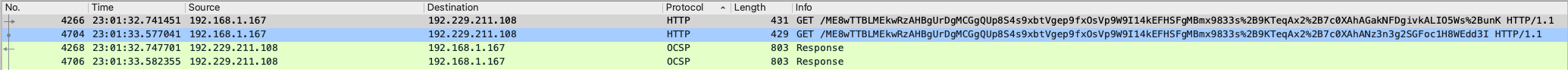
1) Scalability: P2P networks can scale easily since each new participant brings additional resources to the network. This contrasts with client-server models where increasing clients can overload a central server.

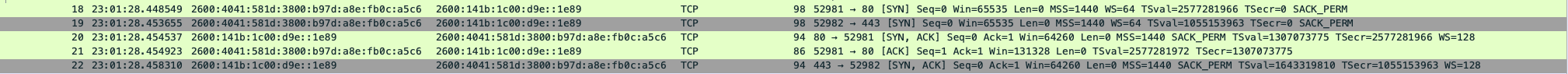
2) Resilience: Since there's no central server, P2P networks are generally more resistant to faults and shutdowns. If one node goes offline, the network can still function.

* + 1. Visit [http://ww.ibm.com](http://ww.ibm.com/) in your favorite browser. Which protocols are used in application layer and transport layer? Provide proof via screenshot(s) using WireShark.

Application layer: OCPS, HTTP

Transport layer: TCP





* 1. Problem 2:

Are the following statements true or false?

1. In order to communicate with a process in a host, one only needs the IP address of the host. False
2. SSL is a transport layer protocol. False
3. Both UDP and TCP guarantee the delivery of the packet from source to destination. False
4. If an application uses UDP, that means that application does not need reliable data transfer. True
5. Problem 3:

If you download a HTML file which references to 8 images, how many RTTs are needed?

* 1. Your browser uses non-persistent HTTP?

- 1 RTT for the initial TCP connection establishment.

- 1 RTT to request and receive the HTML file.

For each of the 8 images, we will need:

- 1 RTT for the TCP connection establishment.

- 1 RTT to request and receive the image.

Hence, 2 RTTs for the HTML file + 16 RTTs for the 8 images = 18 RTTs in total.

* 1. Your browser uses persistent HTTP?

- 1 RTT for the initial TCP connection establishment.

- 1 RTT to request and receive the HTML file.

- For each of the 8 images, because the connection is persistent, you only need 1 RTT to request and receive the image.

Hence, RTTs for the HTML file + 8 RTTs for the images = 10 RTTs in total.

1. Problem 4:
   1. Consider the following HTTPS request: authority: [www.google.com](http://www.google.com/)

method: GET path: / scheme: https accept:

text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\* accept-encoding: gzip, deflate, br

accept-language: en-US,en;q=0.9,tr-TR;q=0.8,tr;q=0.7

user-agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_15\_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.130 Safari/537.36

* + 1. What method is used? GET
    2. What is the path that is requested? /
    3. What type of response is accepted?

text/html, application/xhtml+xml, application/xml (with a quality factor of 0.9), image/webp, image/apng, any other type (/).

* + 1. Why multiple user agents are listed?

The "user-agent" string helps the server identify the browser and the platform from which the request is coming. The multiple values listed (like "Mozilla/5.0", "AppleWebKit/537.36", "Chrome/79.0.3945.130", and "Safari/537.36") are there for historical and compatibility reasons. Web browsers include multiple tokens in the User-Agent string to ensure compatibility with websites that expect certain strings. It doesn't mean that multiple browsers are being used simultaneously.

* 1. Consider the following HTTPs response: content-length: 66708

content-type: text/html; charset=UTF-8 date: Thu, 27 Feb 2022 02:24:24 GMT

server: gws

set-cookie: 1P\_JAR=2020-02-27-02; expires=Sat, 28-Mar-2020 02:24:24 GMT; path=/; domain=.google.com; Secure; SameSite=none

status: 200

1. Was the request successful? Yes, given status code 200.
2. What is the returned content type? text/html; charset=UTF-8
3. Explain what is set-cookie header is used for?

The "set-cookie" header is used by the server to instruct the client (usually a web browser) to store a specific piece of data (a cookie) and send it back with every subsequent request to the same domain. In this case, the server is setting a cookie named "1P\_JAR" with a value of "2020-02-27-02", and it provides some additional attributes like its expiration date, path, domain, and some flags (Secure and SameSite).

1. Is the document last modified at Thu, 27 Feb 2022 02:23:23 GMT? (Yes or No)

No. The provided information only indicates when the response was generated or sent (date: Thu, 27 Feb 2022 02:24:24 GMT), but it doesn't give the last modification time of the document itself.

1. Problem 5:

Consider Figure 2.12 from the course textbook (shown below). Assume that the average object size is 300 Kbits, and the average requests rates from institutional browsers to origin servers are 80 requests per second. Moreover, suppose that the time it takes for the router on the access link side to receive the object from its request is 2 seconds. Also ignore the delay of the outgoing traffic from the institutional network to origin servers.

A diagram of a network

Description automatically generated

Answer the following questions:

* 1. What is the traffic intensity at the access LAN?

Traffic intensity = (Average request rate) × (Average object size) = 80 requests/second × 300 Kbits/request = 24,000 Kbps

* 1. What is the traffic intensity at the access link (from Internet router to institutional router)?

Since there's no caching in the given scenario, the traffic intensity at the access link will be the same as that at the access LAN, which is 24 Mbps.

* 1. List two solutions to mitigate the delay in the access link?

1. Increase the Bandwidth: Upgrading the access link to have a higher bandwidth can reduce the delay caused by congestion.

2. Implement Caching: By placing a cache server (or Content Delivery Network - CDN) closer to the institutional network, frequently requested content can be served from the cache, reducing the load on the access link.

* 1. If cache is deployed with a 0.40 hit rate within the institutional network, what would be the total delay (if less than 0.80 traffic intensity assume 0.01 sec delay)?

First, calculate the effective request rate considering the hit rate:

Effective request rate = (1 - Cache hit rate) × Average request rate = (1 - 0.40) × 80 requests/second = 0.60 × 80 = 48 requests/second

Then, calculate the traffic intensity with the effective request rate:

Traffic intensity = 48 requests/second × 300 Kbits/request = 14,400 Kbps or 14.4 Mbps

If the traffic intensity is less than 0.80 (or 80% of the 15 Mbps access link, which is 12 Mbps), we assume a delay of 0.01 seconds. Since 14.4 Mbps is greater than 12 Mbps, we consider the given delay of 2 seconds.

1. Problem 6:

Consider distributing a file of size 25 Gbits to N peers. The server has an upload rate of us=50Mbps and each peer has a download rate of di=4.2 Mbps and upload rate u. For N= 10, 100, 1000 and u= 200 Kbps, 600 Kbs, 2 Mbps, calculate the minimum distribution time for each of the combinations of N and u for both client-server distribution and P2P distribution. Hint: use the formula presented in class and in the slides).

For N = 10

1. u = 0.2 Mbps

- C-S:

t\_{cs} = 10 x (25,000 / 50) = 5,000 seconds

- P2P:

t\_1 = 25,000 / 50 = 500 seconds

u\_{agg} = 50 + 9(0.2) = 51.8 Mbps

t\_{p2p\_rest} = 25,000 / 51.8 = 482.63 seconds

t\_{p2p} = 500 + 482.63 = 982.63 seconds

2. u = 0.6 Mbps

- C-S:

t\_{cs} = 10 x (25,000 / 50) = 5,000 seconds

- P2P:

t\_1 = 25,000 / 50 = 500 seconds

u\_{agg} = 50 + 9(0.6) = 55.4 Mbps

t\_{p2p\_rest} = 25,000 / 55.4 = 451.26 seconds

t\_{p2p} = 500 + 451.26 = 951.26 seconds

3. u = 2 Mbps

- C-S:

t\_{cs} = 10 x (25,000 / 50) = 5,000 seconds

- P2P:

t\_1 = 25,000 / 50 = 500 seconds

u\_{agg} = 50 + 9(2) = 68 Mbps

t\_{p2p\_rest} = 25,000 / 68 = 367.65 seconds

t\_{p2p} = 500 + 367.65 = 867.65 seconds

For N = 100

1. u = 0.2 Mbps

- C-S:

t\_{cs} = 100 x (25,000 / 50) = 50,000 seconds

- P2P:

t\_1 = 25,000 / 50 = 500 seconds

u\_{agg} = 50 + 99(0.2) = 69.8 Mbps

t\_{p2p\_rest} = 25,000 / 69.8 = 358.17 seconds

t\_{p2p} = 500 + 358.17 = 858.17 seconds

2. u = 0.6 Mbps

- C-S:

t\_{cs} = 100 x (25,000 / 50) = 50,000 seconds

- P2P:

t\_1 = 25,000 / 50 = 500 seconds

u\_{agg} = 50 + 99(0.6) = 109.4 Mbps

t\_{p2p\_rest} = 25,000 / 109.4 = 228.43 seconds

t\_{p2p} = 500 + 228.43 = 728.43 seconds

3. u = 2 Mbps

- C-S:

t\_{cs} = 100 x (25,000 / 50) = 50,000 seconds

- P2P:

t\_1 = 25,000 / 50 = 500 seconds

u\_{agg} = 50 + 99(2) = 248 Mbps

t\_{p2p\_rest} = 25,000 / 248 = 100.81 seconds

t\_{p2p} = 500 + 100.81 = 600.81 seconds

For N = 1000

1. u = 0.2 Mbps

- C-S:

t\_{cs} = 1000 x (25,000 / 50) = 500,000 seconds

- P2P:

t\_1 = 25,000 / 50 = 500 seconds

u\_{agg} = 50 + 999(0.2) = 249.8 Mbps

t\_{p2p\_rest} = 25,000 / 249.8 = 100.08 seconds

t\_{p2p} = 500 + 100.08 = 600.08 seconds

2. u = 0.6 Mbps

- C-S:

t\_{cs} = 1000 x (25,000 / 50) = 500,000 seconds

- P2P:

t\_1 = 25,000 / 50 = 500 seconds

u\_{agg} = 50 + 999(0.6) = 649.4 Mbps

t\_{p2p\_rest} = 25,000 / 649.4 = 38.48 seconds

t\_{p2p} = 500 + 38.48 = 538.48 seconds

3. u = 2 Mbps

- C-S:

t\_{cs} = 1000 x (25,000 / 50) = 500,000 seconds

- P2P:

t\_1 = 25,000 / 50 = 500 seconds

u\_{agg} = 50 + 999(2) = 2,048 Mbps

t\_{p2p\_rest} = 25,000 / 2,048 = 12.21 seconds

t\_{p2p} = 500 + 12.21 = 512.21 seconds

1. Problem 7:

Develop algorithms for generating each of the codes of NRZI, Bipolar-AMI, Pseudoternary, Manchester, Differential Manchester, B8ZS, and HDB3 from NRZ-L.

* + 1. NRZI

prev\_state = 'LOW'

for bit in input\_stream:

if bit == '1':

if prev\_state == 'LOW':

prev\_state = 'HIGH'

else:

prev\_state = 'LOW'

output\_stream.append(prev\_state)

* + 1. Bipolar-AMI

prev\_polarity = 'POSITIVE'

for bit in input\_stream:

if bit == '1':

if prev\_polarity == 'POSITIVE':

output\_stream.append('+')

prev\_polarity = 'NEGATIVE'

else:

output\_stream.append('-')

prev\_polarity = 'POSITIVE'

else:

output\_stream.append('0')

* + 1. Pseudoternary

prev\_polarity = 'POSITIVE'

for bit in input\_stream:

if bit == '0':

if prev\_polarity == 'POSITIVE':

output\_stream.append('+')

prev\_polarity = 'NEGATIVE'

else:

output\_stream.append('-')

prev\_polarity = 'POSITIVE'

else:

output\_stream.append('0')

* + 1. Manchester

for bit in input\_stream:

if bit == '0':

output\_stream.append('HIGH-LOW')

else:

output\_stream.append('LOW-HIGH')

* + 1. Differential Manchester

prev\_state = 'LOW'

for bit in input\_stream:

if bit == '0':

if prev\_state == 'LOW':

output\_stream.append('LOW-HIGH-LOW')

prev\_state = 'LOW'

else:

output\_stream.append('HIGH-LOW-HIGH')

prev\_state = 'HIGH'

else:

if prev\_state == 'LOW':

output\_stream.append('HIGH-LOW')

prev\_state = 'HIGH'

else:

output\_stream.append('LOW-HIGH')

prev\_state = 'LOW'

* + 1. B8ZS

prev\_polarity = 'NONE'

i = 0

while i < len(input\_stream):

if i <= len(input\_stream) - 8 and input\_stream[i:i+8] == '00000000':

if prev\_polarity == 'POSITIVE':

output\_stream.append('000-0+0-')

else:

output\_stream.append('000+0-0+')

i += 8

else:

if input\_stream[i] == '1':

if prev\_polarity == 'POSITIVE':

output\_stream.append('-')

prev\_polarity = 'NEGATIVE'

else:

output\_stream.append('+')

prev\_polarity = 'POSITIVE'

else:

output\_stream.append('0')

i += 1

* + 1. HDB3

prev\_polarity = 'NONE'

ones\_since\_last\_violation = 0

i = 0

while i < len(input\_stream):

if i <= len(input\_stream) - 4 and input\_stream[i:i+4] == '0000':

if ones\_since\_last\_violation % 2 == 0: // even number of ones

if prev\_polarity == 'POSITIVE':

output\_stream.append('B00V') // 'B' is a bipolar violation

else:

output\_stream.append('V00B')

else: // odd number of ones

if prev\_polarity == 'POSITIVE':

output\_stream.append('000V')

else:

output\_stream.append('000B')

ones\_since\_last\_violation = 0

i += 4

else:

if input\_stream[i] == '1':

ones\_since\_last\_violation += 1

if prev\_polarity == 'POSITIVE':

output\_stream.append('-')

prev\_polarity = 'NEGATIVE'

else:

output\_stream.append('+')

prev\_polarity = 'POSITIVE'

else:

output\_stream.append('0')

i += 1

1. Problem 8:

The following describes a signal encoding technique. Binary data is represented as input am for m = 1, 2, 3, … A new set of binary numbers is then produced as follows:

b0 = 1

bm = (am + bm-1) mod 2

These numbers are then encoded as: cm = bm – bm-1

On reception, the original data is recovered via am = cm mod 2

* 1. Confirm that the received values of am are equal to the transmitted values of am.

Given:

1. a\_m is the original binary data.

2. The calculation for b\_m is:

b\_0 = 1

b\_m = (a\_m + b\_{m-1}) mod 2

3. The encoded signal c\_m is:

c\_m = b\_m - b\_{m-1}

4. At the receiving end, the original data a\_m is recovered using:

a\_m = c\_m mod 2

Now we confirm that the received values of a\_m are equal to the

transmitted values of a\_m:

For this, let's express c\_m in terms of a\_m and b\_{m-1} and then derive

a\_m:

From the given:

c\_m = b\_m - b\_{m-1}

b\_m = (a\_m + b\_{m-1}) mod 2

Substituting the value of b\_m:

c\_m = (a\_m + b\_{m-1}) mod 2 - b\_{m-1}

On reception:

a\_m = c\_m mod 2

a\_m = (a\_m + b\_{m-1}) mod 2 - b\_{m-1} mod 2

Without loss of generality, let's consider the possible values of a\_m and

b\_{m-1}:

1. When ( a\_m = 0 ) and ( b\_{m-1} = 0 ):

a\_m = 0 mod 2 - 0 mod 2 = 0

2. When ( a\_m = 0 ) and ( b\_{m-1} = 1 ):

a\_m = 1 mod 2 - 1 mod 2 = 0

3. When ( a\_m = 1 ) and ( b\_{m-1} = 0 ):

a\_m = 1 mod 2 - 0 mod 2 = 1

4. When ( a\_m = 1 ) and ( b\_{m-1} = 1 ):

a\_m = 2 mod 2 - 1 mod 2 = 1

In all cases, the derived a\_m matches the original a\_m. Thus, the received values of a\_m are confirmed to be equal to the transmitted values of a\_m.

* 1. What kind of encoding is this?

Differential Encoding

1. Problem 9:

One positive side effect of bipolar encoding is that a bipolar violation (i.e., two consecutive + pulses or two consecutive – pulses separated by any number of zeros) indicates to the receiver that an error occurred in transmission. However, upon receipt of such a violation, the receiver does not know which bit is in error (i.e., the receiver only knows that an error has occurred).

For the received bipolar sequence + - 0 + - 0 - +, which has one bipolar violation, construct two scenarios that will produce this same received bit pattern. Each of the two scenarios should involve a different transmitted bit stream with one transmitted bit being converted via an error.

Scenario 1: An error causes a '0' to be received as a '+'.

- Original bit stream: "+ - 0 + - 0 - 0"

- Encoded as: + (for '1') - (for next '1') 0 (for '0') + (for next '1') - (for next '1') 0 (for '0') - (for next '1') 0 (for '0')

- Due to an error, the last '0' is received as '+', giving the received pattern "+ - 0 + - 0 - +".

Scenario 2: An error causes a '-' to be received as a '+'.

- Original bit stream: "+ - 0 + - 0 - -"

- Encoded as: + (for '1') - (for next '1') 0 (for '0') + (for next '1') - (for next '1') 0 (for '0') - (for next '1') - (for next '1')

- Due to an error, the last '-' (representing '1') is received as '+', giving the received pattern "+ - 0 + - 0 - +".

1. Problem 10:
   1. Considering the bit pattern 10101, encode this data using ASK, BFSK, and BPSK.

ASK: [Amplitude, 0, Amplitude, 0, Amplitude]

BFSK: [f1, f2, f1, f2, f1]

BPSK: [0°, 180°, 0°, 180°, 0°]

* 1. Assuming a required bit error rate of 10-6, what SNR ratio is required to achieved a bandwidth efficiency of 1.0 for ASK, FSK, PSK, and QPSK?

ASK:

BER = Q(\sqrt{2\*SNR})

BFSK:

BER = Q(\sqrt{SNR})

BPSK:

BER = Q(\sqrt{2\*SNR})

QPSK:

BER = Q(\sqrt{2\*SNR})

Given the BER value of 10^{-6}:

ASK: 12

BFSK: 24

BPSK: 12

QPSK: 12

* 1. Consider an audio signal with spectral components in the range 300 to 3000 Hz. Assume that a sampling rate of 7000 samples per second is used to generate a PCM signal.
     1. For SNR = 40 dB, what is the number of uniform quantization levels needed?

Using the given SNR of 40 dB:

40 = 6.02N + 1.76

N =6.35

Number of levels = 2^N = 2^7 = 128 levels.

* + 1. What is the required data rate?

Data rate = 7000 \* 7 = 49,000 bits per second