# CS-335a: Computer Networking

Department of Computer Science, University of Crete
Fall 2023

Professor: Maria Papadopouli
TA: Christina Brozi (brozi@csd.uoc.gr)

## Assignment 2 - Application Layer

*At the Top!*

## INDICATIVE SOLUTIONS

Objective: Get familiar with the HTTP protocol, Persistent vs. Non-Persistent HTTP connections and HTTP response and request messages.

### HTTP

**1. (20 points)** Answer the following questions on WWW.

i) Is WWW a client-server or peer-to-peer application ? Why ? (3 points)

WWW is a client-server application. Always-on servers receive requests from browsers (clients). Unlike peer-to-peer systems, in the client-server architecture of the Web, browsers do not directly communicate with each other.

ii) Briefly describe HTTP non-persistent and HTTP persistent. What are the benefits of HTTP persistent over HTTP non-persistent ? (5 points)

Non-persistent HTTP: each request (from client) and response (from server) pair is sent over a separate TCP connection.

Persistent HTTP: each request/response pair is sent over the same TCP connection.

Advantages of persistent HTTP: responses/requests can be pipelined in the same connection (reduces latency), reduces network congestion etc.

iii) We said that the classical HTTP protocol is stateless, that is once you finish your HTTP session all the client-server interactions are “forgotten” and state information cannot be used in following sessions. Briefly describe how cookies can be used to preserve state across sessions. (7 points)

An HTTP cookie is a small piece of data that a server sends to a user's web browser. The browser may store the cookie and send it back to the same server with later requests. An HTTP cookie usually indicates if two requests come from the same browser. For example, cookies are often used to keep users logged in to a website over many requests and responses. When a user sends a login request with a valid username and password, the server will generate a new session token and send it to the user as a cookie. In future requests, the browser will attach the session token cookie and send it to the server. The server maintains a mapping of session tokens to users, so when it receives a request with a session token cookie, it can look up the corresponding user and customize its response accordingly.

iv) How can web caches be used for improved user experience in remote and poorly connected areas? List one drawback of the use of Web caches. (5 points)

Web caching can bring the desired content “closer” to the user, possibly to the same LAN to which the user’s host is connected.

If caching is not set up correctly the browser might not be able to validate the cached content and the page may load outdated content.

**2.** **(5 points) A.** Consider the following string of ASCII characters that were captured by Wireshark when the browser sent an HTTP GET message (i.e., this is the actual content of an HTTP GET message). The characters `<cr><lf>` are carriage return and line-feed characters.

Answer the following questions, indicating where in the HTTP GET message below you find the answer.

```
GET /cs335/index.html HTTP/1.1<cr><lf>Host:
gaia.cs.umass.edu<cr><lf>User-Agent: Mozilla/5.0 ( Windows;U;
Windows NT 5.1; en-US; rv:1.7.2) Gec ko/20040804 Netscape/7.2
(ax) <cr><lf>Accept: ext/xml, application/xml,
application/xhtml+xml, text /html;q=0.9, text/plain;q=0.8,
image/png,*/*;q=0.5 Accept-Language: en-us, en;q=0.5
<cr><lf>AcceptEncoding: zip, deflate <cr><lf>Accept-Charset:
ISO -8859-1, utf-8;q=0.7,*;q=0.7<cr><lf>Keep-Alive: 300
<cr><lf>Connection:keep-alive<cr><lf><cr><lf>
```

i) What is the URL of the document requested by the browser ? (1 point)

`https://gaia.cs.umass.edu/cs335/index.html`

ii) What version of HTTP is the browser running ? (1 point)

`HTTP 1.1`

iii) Does the browser request a non-persistent or a persistent connection ? (1 point)

The browser requests a persistent connection as indicated by `Connection:keep-alive`

iv) What is the IP address of the host on which the browser is running ? (1 point)

No such information is provided in the HTTP GET message.

v) What type of browser initiates this message ? Why is the browser type needed in an HTTP request message ? (1 point)

Browser: `Mozilla 5.0`

Since some webpages might have different versions depending on the browser, the server needs to know this type of information in order to send the correct version of a web page to the client.

**(5 points) B.** The text below shows the reply sent from the server in response to the HTTP GET message in the question above. Answer the following questions, indicating where in the message below you find the answer.

```
HTTP/1.1 200 OK<cr><lf>Date: Tue, 07 Mar 2008
12:39:45GMT<cr><lf>Server: Apache/2.0.52 (Fedora)<cr><lf>
Last-Modified: Sat, 10 Dec 2005 18:27:46 GMT<cr><lf>ETag:
”526c3-f22-a88a4c80”<cr><lf>AcceptRanges: bytes<cr><lf>
Content-Length: 3874<cr><lf>Keep-Alive: timeout=max=100
<cr><lf>Connection: Keep-Alive<cr><lf>Content-Type:
text/html; charset= ISO-8859-1<cr><lf><cr><lf><!doctype html
public ”-//w3c//dtd html 4.0 transitional//en”><lf><html><lf>
<head><lf> <meta http-equiv=”Content-Type”
content=”text/html; charset=iso-8859-1”><lf><meta
name=”GENERATOR” content=”Mozilla/4.79 [en] (Windows NT
5.0; U) Netscape]”><lf> <title>CMPSCI 453 / 591 /
NTU-ST550ASpring 2005 homepage</title><lf></head><lf>
<...>
```

i) Was the server able to successfully find the document or not ? What time was the document reply provided ? (1 point)

Status code (`200`) and phrase (`OK`) indicate that the server successfully found the requested document. The reply was provided on `Tue, 07 Mar 2008 12:39:45 GMT`

ii) When was the document last modified ? (1 point)

The document (index.html) was last modified on `Sat, 10 Dec 2005 18:27:46 GMT.`

iii) How many bytes are there in the document being returned ? What are the first 5 bytes of the document being returned ? Explain your answer. (2 points)

There are `3874` bytes in the document being returned.
An HTML document starts with: `<!doctype html (...)`
and since each character is one byte, the first five bytes will be `<!doc`

iv) Did the server agree to a persistent connection ? (1 point)

The server agrees to a persistent connection as indicated by `Connection: Keep-Alive`

**3.** **(15 points)** A browser wants to access the webpage xyz.com. Suppose the webpage is an HTML document that contains 5 JPEG images.

i) Create a sequence diagram of the communication process (request/response messages) between the client and the server.

ii) For the sake of simplicity, assume each Round-Trip Time (RTT) is 100 ms and the time to transmit each object is 10 ms. What is the total response time (i.e. the duration from the initiation of the TCP connection by the browser until it receives all requested objects) ?

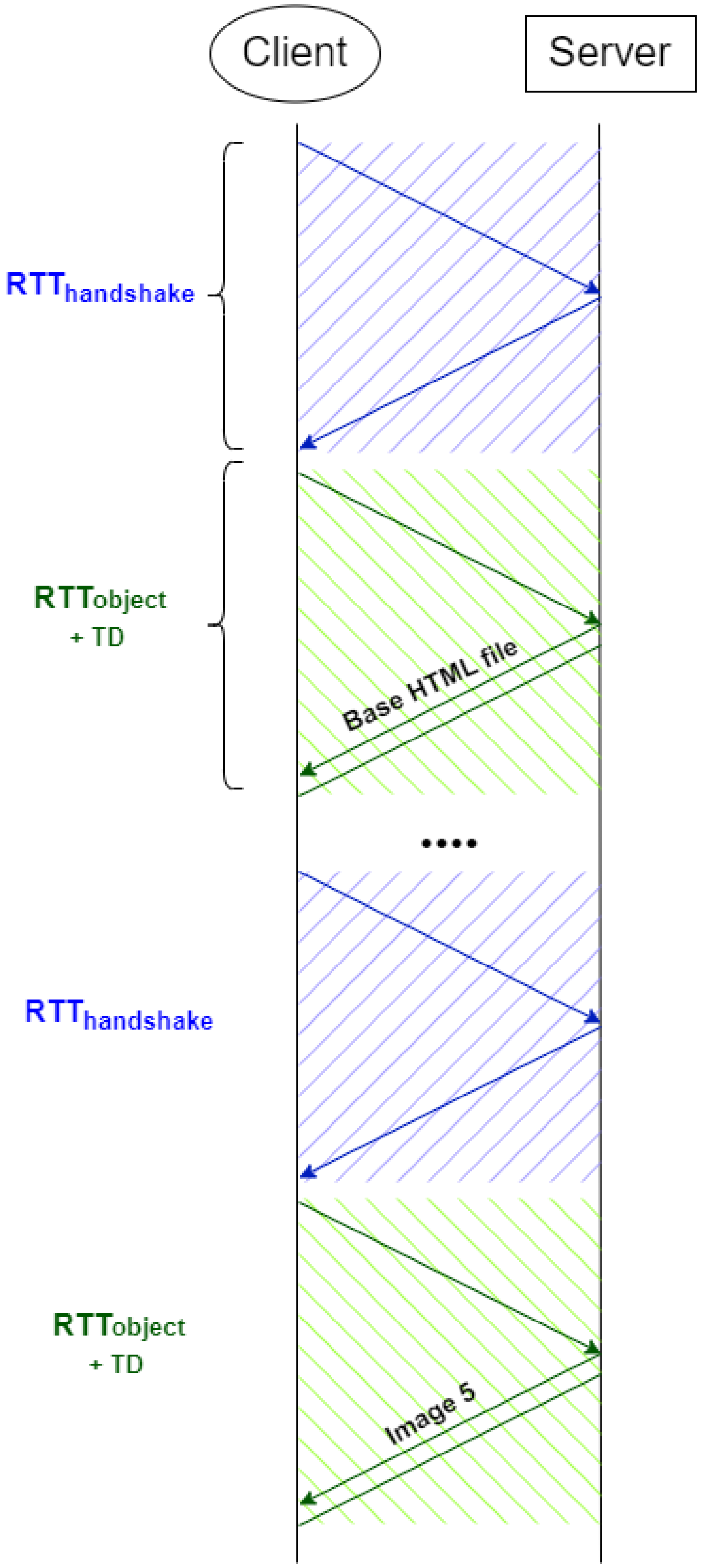
Answer the above questions for each of the following cases:

The server has to send N = 5 JPEG images + 1 base HTML file = 6 objects.

A. Non-Persistent HTTP (without parallel connections) (3 + 2 points)

In this case the server must send the 6 objects over 6 different TCP connections. Each transmission consists of a RTT for the handshake ($RTT_{handshake}$), a RTT for the transmission of the object ($RTT_{object}$) and the transmission delay (TD).

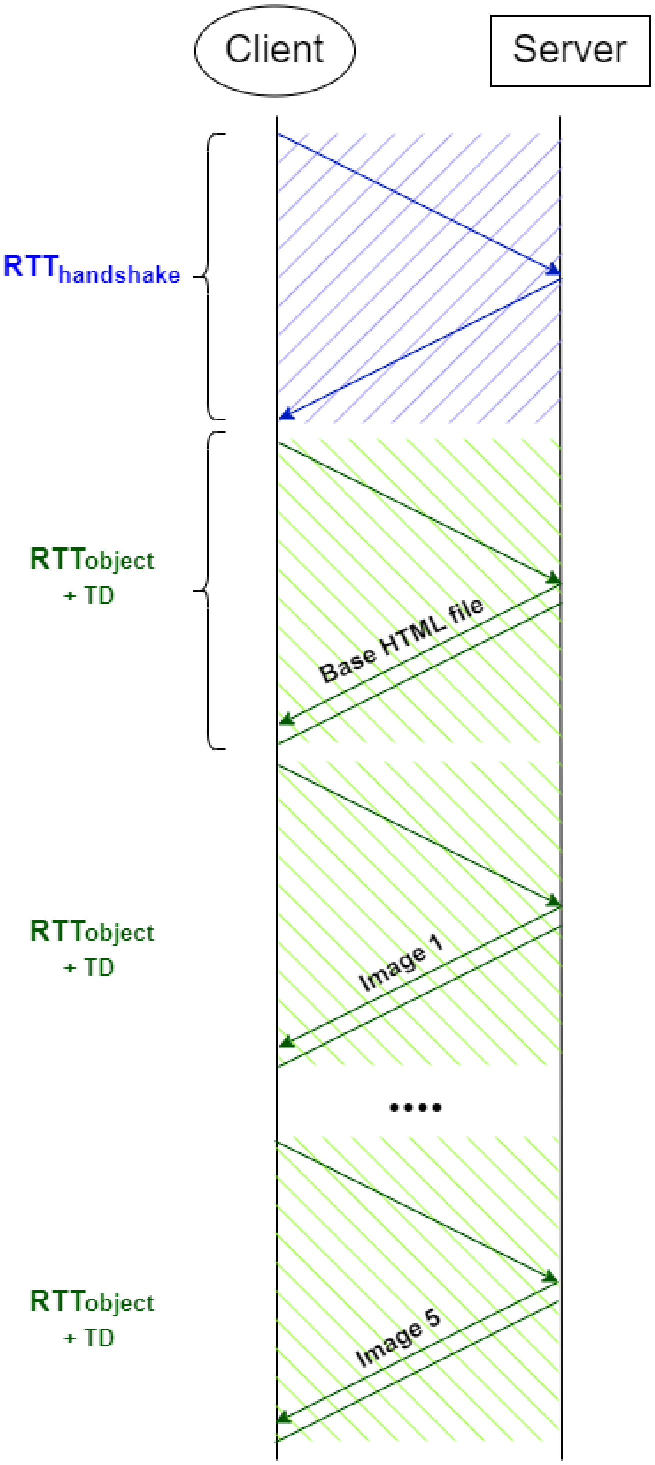
$$\text{ResponseTime} = (RTT_{handshake} + RTT_{object} + TD)*\ N = (100 + 100 + 10)*6 = 1260\ \text{ms.}$$



B. Persistent HTTP without pipelining (3 + 2 points)

In this case the server sends the objects sequentially over the same TCP connection.

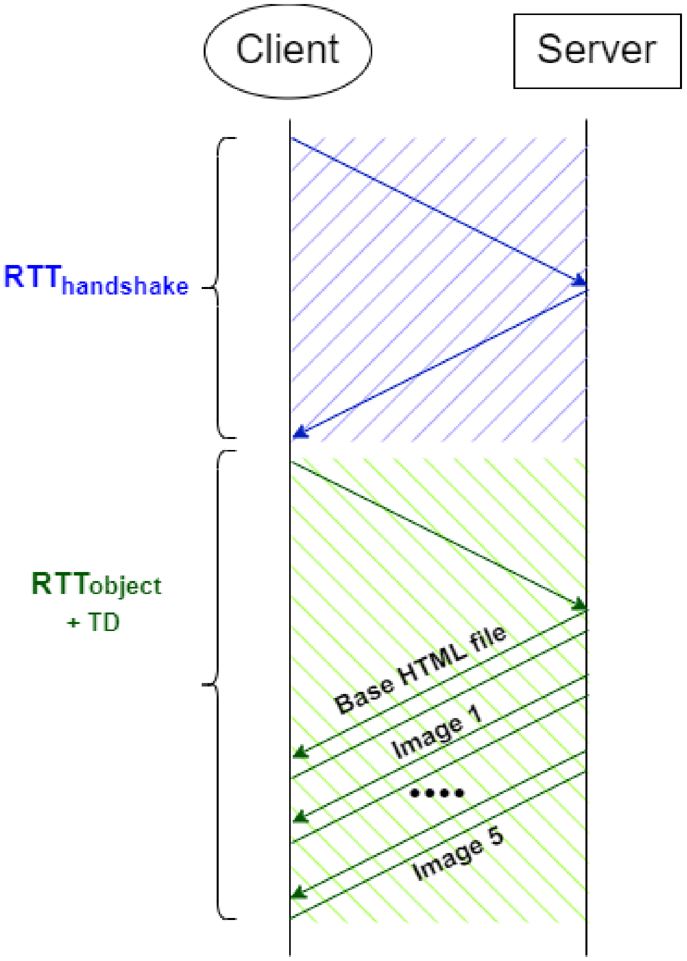
ResponseTime = $\text{RTT}_{\text{handshake}}$ + ($\text{RTT}_{\text{object}}$ + TD) * N = 100 + (100 +10) * 6 = 760 ms.



C. Persistent HTTP with pipelining (3 + 2 points)

In this case the server sends the objects in parallel (without waiting for an answer from the client).

ResponseTime = $RTT_{handshake}$ + $RTT_{object}$ + TD = 100 + 100 + 10 = 210 ms.



Objective: Understand the operational mechanisms of the Domain Name System (DNS) and the range of services it offers.

## DNS

**4. (20 points)** Answer the following questions on DNS.

i) Describe 4 services/functions provided by DNS. (8 points)

1) Translates hostnames to IP addresses
2) Host aliasing
3) Mail server aliasing
4) Performs load distribution among replicated servers

ii) Could a centralized DNS server work? Indicate three possible disadvantages of such architecture in the context of DNS ? (2 points)

Using a single DNS server to handle all requests could not work for the following reasons:

1) The centralized DNS would not be able to handle large amounts of requests
2) Single point of failure: if the centralized DNS stops working, so will the internet
3) Delays in requests coming from regions that are “far” from the DNS

iii) Suppose a web browser wants to know the IP address of www.csd.uoc.gr. Describe the name resolution process assuming iterative queries are used. Which query type is considered best practice: iterative or recursive ? Explain your answer. (10 points)

1) The host sends a DNS query to the local DNS server for the IP address of `www.csd.uoc.gr.`
2) The local DNS server forwards the message to the root DNS server which replies with the IP address of the TLD servers responsible for `.gr` domains.
3) The local DNS server sends the query to a TLD DNS server, which in turn replies with the IP address of the authoritative DNS server, or the IP address of an intermediate DNS server who knows the authoritative DNS server.
4) If the TLD DNS server uses an intermediate DNS server, the intermediate server replies to the local DNS server with the IP address of the authoritative DNS server.
5) The local DNS server sends the message to the authoritative DNS server, which replies with the IP address of the hostname `www.csd.uoc.gr.`

Iterative queries are more efficient because they distribute the load among multiple servers.

5. **(10 points)** The `dig` (domain information groper) command is a flexible tool for interrogating DNS name servers. It performs DNS lookups and displays the answers that are returned from the queried name server(s).
Use https://networking.ringofsaturn.com/Tools/dig.php to run the following command:

```
dig chat.openai.com
```

i) What does this command do ? (2 points)

The domain information groper (dig) command is used to perform DNS queries. In this case it performs a DNS lookup and retrieves DNS records for `chat.openai.com`

ii) What does every line (not starting with ";") of the command's output mean ? (6 points)

```
;; global options: +cmd
```

The options applied to the domain query, by default cmd.

```
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 38676
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0,
ADDITIONAL: 1
```

This section contains details about the answer. The header includes the opcode, which is the action performed by the dig command, which in this case is a query, the status NOERROR which means that the request was successful and a random id number for the request/response pair. The flags stand for query, recursion desired (which means that the DNS lookup was done recursively) and recursion available. QUERY: 1 is the number of queries in this session (in this case 1). ANSWER: 3 is the number of answers in the response (in this case 3). AUTHORITY: 0 is the number of answers provided by Authoritative DNS servers (in this case 0). ADDITIONAL: 1 is additional information.

```
;; OPT PSEUDOSECTION:
```

This section contains information about the Extension system for DNS, flags and the UDP packet size.

```
;; QUESTION SECTION:
```

This section displays the query data which contain the domain name queried, the type of the query and the record.

```
;; ANSWER SECTION:
chat.openai.com. 300   IN    CNAME chat.openai.com.cdn.cloudflare.net.
```

The answer section contains the name of the server that was queried, the TTL (time after which the record should be removed from the cache), the class of the query (in this case internet), the type of the query (in this case CNAME, which maps a hostname to another hostname).

```
chat.openai.com.cdn.cloudflare.net. 300 IN A 104.18.37.228
chat.openai.com.cdn.cloudflare.net. 300 IN A 172.64.150.28
```

These are type A queries which map the hostname to two IP addresses

```
;; Query time: 38 msec
```

The total time needed for the response.

```
;; SERVER: 47.190.90.80#53(47.190.90.80) (UDP)
```

The IP address and the port number of the DNS server.

```
;; WHEN: Thu Nov 02 06:12:03 CST 2023
```

Date and time when the command was run.

```
;; MSG SIZE  rcvd: 152
```

The size of the DNS server reply.

iii) Why in the records ‘chat.openai.com.’ ends with a dot ? (2 points)

The dot at the end of the domain name indicates an absolute address, which means that the DNS resolver will not view it as relative and will not append its default domain to the domain provided.

Objective: Delve into the different protocols utilized in email.

## EMAIL - SMTP

**6.** **(10 points)** Suppose Alice, with a Web-based e-mail account (like Hotmail or Gmail), sends a message to Bob, who accesses his mail from his mail server using POP3. Discuss how the message gets from Alice to Bob. Make sure to list all application layer protocols that are used to move the message between the two hosts.

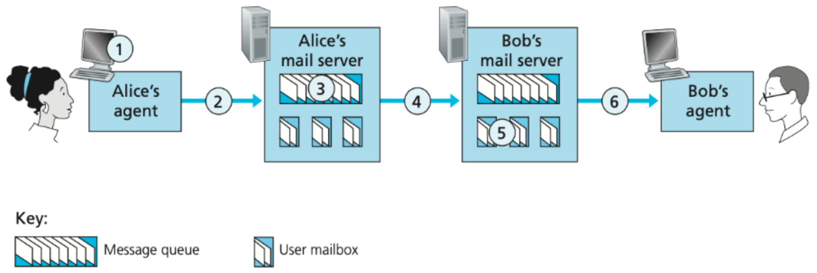


**Figure 2.15** ♦ Alice sends a message to Bob

1) The message is first sent from Alice's host to her mail server over HTTP.
2) Alice's email server uses the Domain Name System (DNS) to look up the Mail Exchanger (MX) records for Bob's email server. The MX record specifies the hostname of the server that is responsible for handling email messages for a given domain.
3) Alice's email server uses SMTP to connect to Bob's email server. It sends the email message to Bob's email server, which receives the message using the same protocol.
4) Bob's email server stores the message in a mailbox associated with Bob's email account. Bob can retrieve the message from his mailbox using the Post Office Protocol version 3 (POP3).

**7.** **(15 points)** Send an email to majordomo@csd.uoc.gr with subject "All subscribed lists" and as message the text "which". Locate the source* of both the email you sent and the response that you got from majordomo.

* Open the email (from webmail.csd.uoc.gr) > click on the ▯ symbol on the top right corner > click on "Show Source" from the drop down menu

i) What exactly do we learn from the **Received** header fields of the response ? What is the difference between **Deliver-To** and **Delivered-To** headers ? (7 points)

```
Received: from csd.uoc.gr (localhost [127.0.0.1])
```

Where the email was received from and the actual IP address of the sender (used as a verification method). In this case the localhost IP address is used because the webmail application is probably running on the same machine as the mail server.

```
by ermis.csd.uoc.gr (Postfix)
```

The host that added the header

```
with SMTP
```

The protocol that was used for the transfer (Simple Mail Transfer Protocol)

```
id BEA6B3C028E
```

A unique email id for this host

```
for <brozi@csd.uoc.gr>;
```

The recipient of the email

```
Thu,  2 Nov 2023 15:01:34 +0200 (EET)
```

The date and time the message was sent

`Deliver-To:` displays the address of the recipient who received the delivery. Deliver-To header is added during the event of the delivery

`Delivered-To:` annotates an email delivery event. The header field contains information about the individual address used to effect that transition.The Delivered-To: header field is added at the time of delivery, when responsibility for a message transitions from the Message Handling Service (MHS) to an agent of the specified individual recipient address.

ii) Why are there no **Received** header fields in the sent email ? (3 points)

The sent email doesn’t have any Received header fields since it just transfers the email from the sender’s user agent to the sender’s mail server and doesn’t get a response from the server or the recipient for that transfer.

iii) How is the end of the header fields encoded ? (2 points)

With an empty line.

iv) Estimate how long it took majordomo to compose and send its response. (3 points)

In the sent message the date and time field is:

```
Date: Thu,  2 Nov 2023 13:01:32 +0000
```

The actual time the mail was sent is +2:00 hours, therefore `15:01:32` .

In the response message the date and time field is (in the Received header):

```
Thu, 2 Nov 2023 15:01:34 +0200 (EET)
```

It took majordomo approximately 2 seconds to compose and send the response message.

## **Brainstorming!** (15 points– the 10 out of 15 points is bonus)

You envision an innovative adaptive video streaming system: it dynamically adjusts the streaming operation based on the ***user and network*** conditions, aiming to optimize the overall user experience. Note: the adaptation does not refer ONLY to the bitrate!

Give an overview of the architecture of your system (show in a diagram the various components) as well as the protocol at the application-layer (5 points).

Think about metrics to assess the ***perceived quality of experience*** (5 points). Argue about the advantages of your architecture and protocol (3 points).

Describe the experiments you need to run to support your claims about the attractive performance features of your system (2 points).

Feel free to target a **specialized domain** (e.g., video streaming that targets specific settings or market segment-user population or type of content or type of environment or devices);

You can assume that you can collaborate with the appropriate content provider and the availability of content is not an issue.

**Extra points for innovative applications-paradigms!** However you need to support your claim that your application is ***original*** (e.g., differs from existing systems/applications).