

CS335a: Computer Networks

Professor: Maria Papadopouli

TA: Giorgos Mellios (csdp1395@csd.uoc.gr)

Deadline: 05/11/2025

SUBMISSION GUIDELINES

- Your report should be in PDF format.
- Please submit your assignment via the e-learn platform.
- The maximum grade you can get is 120 with 20 out of the 120 points being BONUS.
- To prevent plagiarism, in each assignment series a random sample of students will be selected for further oral examination.

Assignment 2: Application Layer

Exercise 1 (25pts + 5pts)

- i) (3p) Explain the main differences between client-server and peer-to-peer architectures. Provide one modern real-world example of each.
- ii) (3p) What are the main benefits and drawbacks of running applications over UDP instead of TCP? Mention at least two examples of real applications that use UDP and explain why.
- iii) (3p) Describe the term application-layer protocol. What are the key elements it defines, and how do they enable interoperability among systems?
- iv) Imagine you are designing a real-time multiplayer educational platform (for example, an interactive quiz, collaborative lab, or simulation game) where multiple users interact simultaneously over the Internet.
1. (4p) Specify the type of transport service your application would require in terms of:
 - Reliability

- Timing / delay tolerance
- Throughput
- Security

Justify each requirement based on the expected user experience.

2. (2p) Choose an appropriate transport protocol (e.g., TCP, UDP, or a hybrid approach such as UDP with application-level reliability).

Explain your choice in terms of performance, scalability, and interactivity.

3. (2p) Propose one application-layer feature that would improve user experience under varying network conditions (e.g., packet loss, latency spikes, or limited bandwidth).

Describe how this feature adapts to changing network performance.

4. (+5 bonus points) Suggest a novel or non-trivial feature that distinguishes your system from existing real-time platforms (such as Kahoot, Quizizz, or multiplayer online games).

Your innovation may relate to:

- Network protocol design,
- Synchronization and fairness mechanisms,
- Adaptive content delivery,
- Real-time feedback, or
- Cross-platform optimization.

v) (8p) Compare **DNS**, **SMTP**, and **HTTP** in terms of:

1. Type of communication (push/pull)
2. Statefulness
3. Typical transport protocol used
4. Default port numbers

Exercise 2 (20pts)

i) (3p) Explain the difference between **non-persistent** and **persistent** HTTP connections. How many TCP connections are typically established when a page with 10 embedded objects is loaded under each approach?

ii) (4p) Describe how **pipelining** in HTTP/1.1 reduces response time compared to standard persistent HTTP. What limitations caused browsers to eventually disable pipelining by default?

iii) (5p) Consider an institutional network using a web proxy cache.

1. Explain how **proxy caching** reduces overall response time and bandwidth consumption.
2. What HTTP headers are used by clients and servers to control caching behavior?
3. How can a cache validate that its stored copy of an object is still fresh?

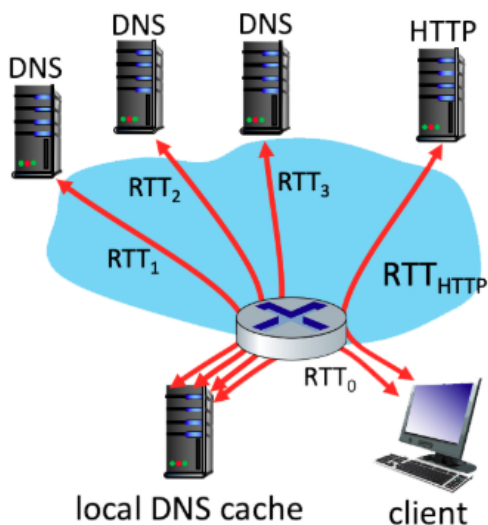
iv) (4p) Explain how **cookies** allow HTTP servers to maintain state across multiple interactions of the client

Include in your answer the four components that are involved in cookie-based state management.

v) (4p) A user reports that some pages load outdated versions of content even though they were updated on the server.

1. Identify and explain one HTTP mechanism that could help prevent this issue.
2. Include an example of an HTTP header used for that purpose.

Exercise 3 (20pts)



Suppose within your Web browser you click on a link to obtain a Web page. The IP address for the associated URL is not cached in your local host, so a DNS lookup is necessary to obtain the IP address. Suppose that four DNS servers are visited before your host receives the IP address from DNS. The first DNS server visited is the local DNS cache, with an RTT delay of $RTT_0 = 2$ msec. The second, third and fourth DNS servers contacted have RTTs of 17, 10, and 27 msec, respectively. Initially, let's suppose that the Web page associated with the link contains exactly one object, consisting of a small amount of HTML text. Suppose the RTT between the local host and the Web server containing the object is $RTT_{HTTP} = 87$ msec.

- i) (4p) Assuming zero transmission time for the HTML object, how much time (in msec) elapses from when the client clicks on the link until the client receives the object?
- ii) (4p) Now suppose the HTML object references 8 **very small objects** on the same server. Neglecting transmission times, how much time (in msec) elapses from when the client clicks on the link until the base object and all 8 additional objects are received from the web server at the client, assuming **non-persistent HTTP** and **no parallel TCP** connections?
- iii) (4p) Suppose the HTML object references 8 very small objects on the same server, but assume that the client is configured to support a maximum of 5 parallel TCP connections, with **non-persistent HTTP**.
- iv) (4p) Suppose the HTML object references 8 very small objects on the same server, but assume that the client is configured to support a maximum of 5 parallel TCP connections, with **persistent HTTP**.

v) (4p) What are the differences between these methods:

1. Non-Persistent HTTP (without parallel connections)
2. Persistent HTTP without pipelining
3. Persistent HTTP with pipelining

Exercise 4 (20pts)

i) (5p) Suppose a client at **student.csd.uoc.gr** needs to resolve **www.harvard.edu**. List in order, all DNS servers contacted during an iterative resolution process and describe what each returns.

ii) (5p) Now assume the same query is done using recursive resolution. What changes?

iii) (5p) Explain DNS caching and how stale entries can affect performance or correctness.

iv) (5p) Explain how DNS-based content delivery enables users to be directed to geographically closer servers. Reference how CDNs like Akamai use DNS for performance optimization.

Exercise 5 (15pts)

i) (3p) Outline the role of each of the following components in email delivery:

1. User Agent (UA)
2. Mail Server
3. Mail Access Protocol

ii) (4p) Describe the steps involved when Alice (alice@csd.uoc.gr) sends an email to Bob (bob@mit.edu) using SMTP.

iii) (4p) Compare SMTP and HTTP in terms of:

1. Direction of data transfer (push vs pull)
2. Connection persistence
3. Message format

iv) (4p) IMAP allows messages to remain on the server. List two advantages and one drawback of this design.

Exercise 6 (BONUS 15pts)

In this exercise, you will use the command-line tool **dig (Domain Information Groper)** to explore how the **Domain Name System (DNS)** operates in practice. **dig**, provides detailed information about DNS queries, responses, and authoritative name servers.

You can perform all commands on the department's Linux machines or any Unix-based system.

i) Run the following command:

dig www.google.com

1. (1p) What is the IP address returned for www.google.com?
2. (1p) Which DNS server answered your query? Was it authoritative or non-authoritative? Explain how you can tell.
3. (1p) Run the same command again immediately. Did the query time change? Why?
4. (1p) Briefly explain how DNS caching reduces latency in name resolution.

ii) Retrieve the MX (Mail Exchange) records for the University of Crete:

dig uoc.gr MX

1. (1p) List all returned mail servers and their priorities.
2. (1p) Which mail server has the highest preference (lowest number)?

Query for the NS (Name Server) records of the top-level domain .gr:

dig gr NS

3. (1p) How many TLD servers are listed?

4. (1p) What is the purpose of having multiple TLD servers?

Query for the A record of www.uoc.gr:

dig www.uoc.gr A

5. (1p) Identify the IP address returned and the Time To Live (TTL). What does TTL indicate?

iii)

Use the trace option to follow the full resolution path:

dig +trace www.google.com

Answer the following:

1. (1p) List the sequence of DNS servers contacted (root, TLD, authoritative).
2. (1p) Which server provides the final IP address for www.google.com?
3. (1p) How is iterative resolution visible in the output?
4. (1p) Explain why dig might only show a single response line on some networks, and how to fix this.

iv)

Run:

dig @8.8.8.8 www.csd.uoc.gr

1. (1p) What does @8.8.8.8 specify? Compare the response time with your local resolver.
2. (1p) Why might the results differ between your local resolver and Google's DNS in terms of authority and latency?

For this exercise, provide screenshots of each command's output. Write short explanations (1–3 sentences per question). Make sure your answers clearly indicate which section and question they correspond to.