ECE438 Homework 2 Jiadong Hong

1 Choose all that Apply

1. Two distinct Web pages (for example, *www.intl.zju.edu.cn/students.html* and *www.intl.zju.edu.cn/research.html*) can be sent over the same persistent connection.

   (a) True

   (b) False

(a) In traditional HTTP/1.0, a separate connection was established for each resource request (e.g., an HTML page, an image, a stylesheet), which could result in a significant overhead due to the repeated connection setup and teardown. However, in HTTP/1.1, which is the most widely used version of HTTP as of my last knowledge update in January 2022, persistent connections are supported by default.

2. Is it possible for an organization's Web server and mail server to have exactly the same alias for the hostname(for example, foo.com?)

   (a) Yes

   (b) No

(a) Yes.

3. Knowing the alias for the mail server, What type of RR should a DNS client to query to get the cannonical name for the mail server?

   (a) A

   (b) NS

   (c) CNAME

   (d) MX

(c)

4. What protocol might be used if a user want to get email from user's mail server to his local PC?

   (a) SMTP

   (b) POP3

   (c) IMAP

   (d) HTTP

(b)

2 Short Answer Questions - 5 × 2 points

1. Briefly explain the advantages and disadvantages of the use of cookie

Advantages of Using Cookies:

1. **Session Management:** Cookies are commonly used to manage user sessions. They can store session identifiers that allow websites to recognize and remember users as they navigate between pages, which is essential for maintaining user login status and personalized experiences.
2. **Customization:** Cookies enable websites to personalize content and settings for individual users based on their preferences and behavior. This can enhance user experience and engagement.
3. **Tracking and Analytics:** Cookies are used for tracking user behavior on websites. They provide valuable data for web analytics, helping businesses understand how users interact with their sites and make data-driven decisions.
4. **Shopping Carts and E-commerce:** Cookies are essential for e-commerce websites to maintain shopping cart contents and remember user selections. This is crucial for a smooth shopping experience.
5. **Remembering User Preferences:** Cookies can store user preferences, such as language settings or display preferences, allowing websites to remember these choices across visits.

Disadvantages of Using Cookies:

1. **Privacy Concerns:** Cookies can raise privacy concerns, as they can track and store user data, potentially infringing on user privacy. Users may be uncomfortable with the amount of data collected.
2. **Security Risks:** Cookies can be vulnerable to various security threats, including cookie theft, session hijacking, and cross-site scripting (XSS) attacks if not implemented securely.
3. **Cross-Site Tracking:** Some users may be concerned about being tracked across different websites by third-party cookies. This can lead to privacy issues and unwanted advertisements.
4. **Limited Data Storage:** Cookies have a size limit (typically 4KB), which restricts the amount of data that can be stored. This can be a limitation for applications that require larger data storage.
5. **Browser Settings:** Users can disable or block cookies in their web browsers, which may disrupt some website functionality, such as login sessions or personalization.

2. Describe how Web caching can reduce the delay in receiving a requested object. Will Web caching reduce the delay for all objects requested by a user or for only some of the objects? Explain why.

How Web Caching Reduces Delay:

1. **Reduced Round-Trip Time:** When a user's request is served from a cache, it eliminates the need to fetch the object from the original web server. This reduces the round-trip time, as the object is already stored closer to the user, typically in the same geographical region or on the same network.
2. **Faster Retrieval:** Retrieving objects from a cache is faster because it doesn't involve the time-consuming process of establishing a connection with the original server, waiting for the server to process the request, and transferring data over the internet. Instead, the cache server can respond quickly with the cached copy.
3. **Less Congestion:** By serving content from a cache, it reduces the load on the original web server. This helps prevent congestion and overloads on the server, ensuring a more responsive and reliable user experience.

Web Caching and Objects Requested by a User:

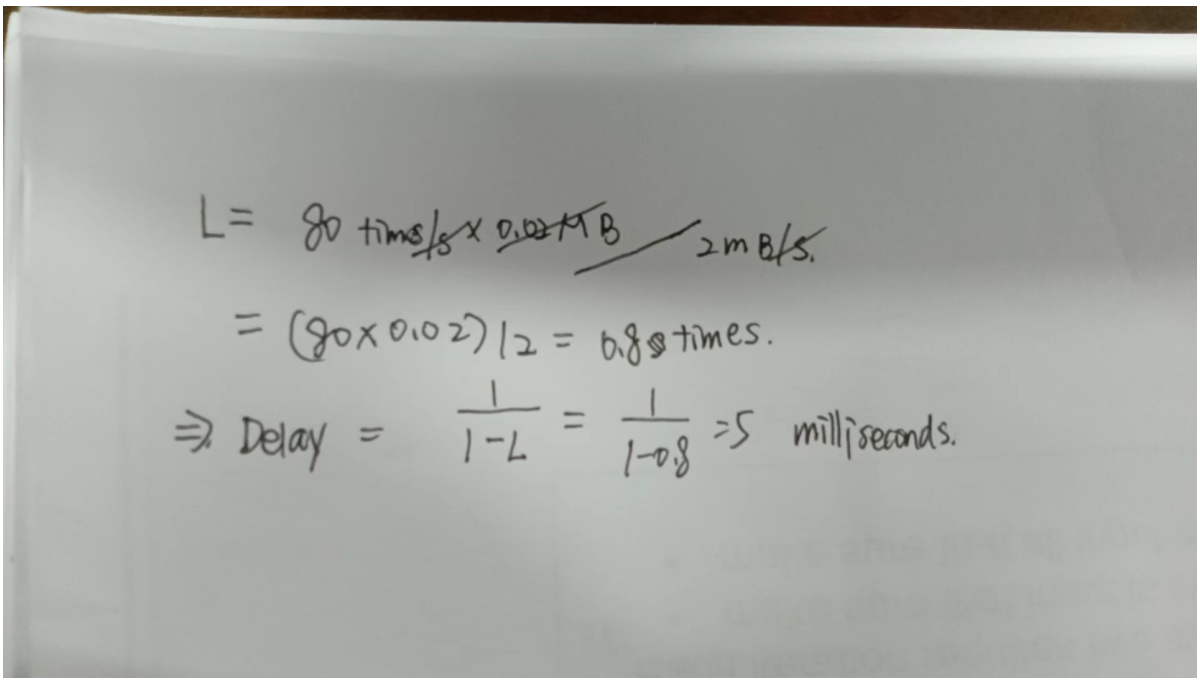
Web caching reduces the delay for some of the objects requested by a user, but not necessarily for all objects. The extent to which web caching reduces the delay depends on several factors, including:

1. **Cache Hit vs. Cache Miss:** When a user requests an object, the caching server checks if it has a cached copy (cache hit) or if it needs to fetch the object from the original server (cache miss). If the object is in the cache (cache hit), the delay is reduced. If it's not in the cache (cache miss), the delay is not reduced, and the object must be fetched from the original server.
2. **Cache Policies:** Cache servers are configured with policies that determine what objects to cache, how long to keep them, and when to invalidate them. These policies vary, and not all objects may be cached or kept for an extended period. Frequently accessed or popular objects are more likely to be cached.
3. **Cache Size:** Cache servers have limited storage capacity. Less frequently accessed or less popular objects may not be cached due to space limitations. As a result, only a subset of objects is cached.

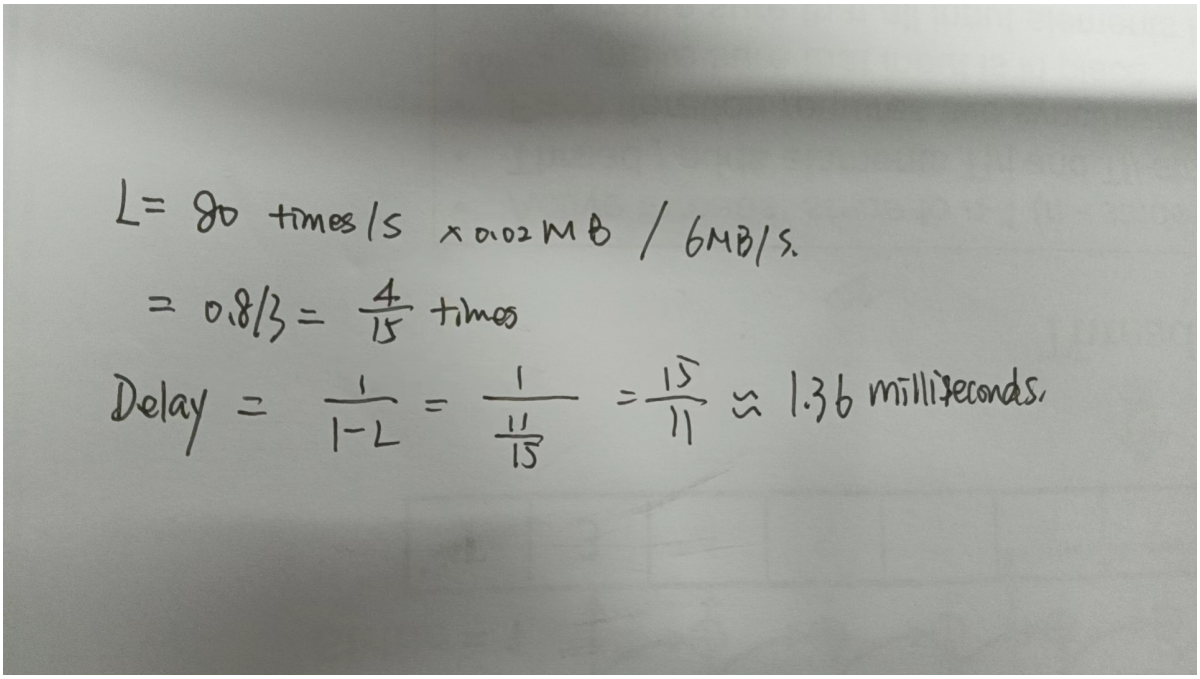
3 Web Caching - 7 x 3 points

Assume a group of students in an institution want to access a private server A outside of the institution. The bottleneck link from the institution to this server supports a bitrate of 2MB/S. Assume the average request rate from the institution is 80 times/s and each request is 0.02MB. Assuming there is no other traffic within or outside of the institution, answer the following questions. Assume that queueing delay dominates so you can neglect the much smaller propagation delays, transmit times, and processing delays

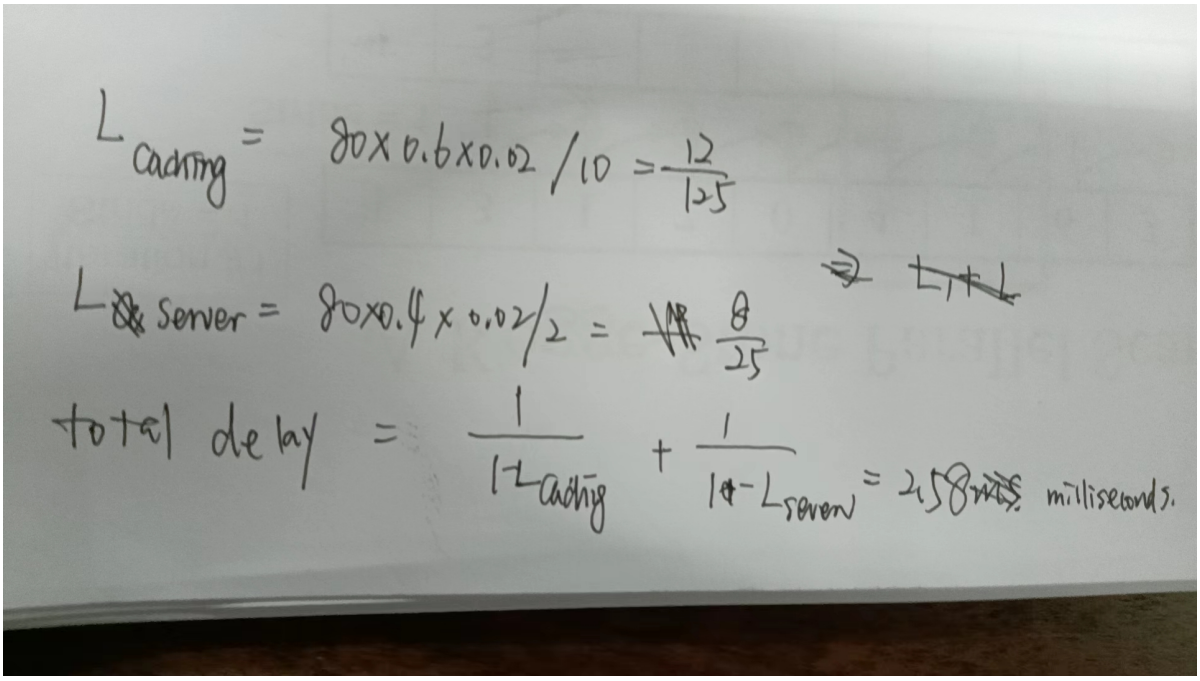
1. What is the average access time for a user in the institution to access this server? Assume the queuing delay is 1/(1-L) milliseconds, where L is the fraction of link usage. (Your answer should be in milliseconds).



2. To improve network performance, we now increase the bitrate of this bottleneck link to 6MB/s. Calculate the average access time again. Your unit should be milliseconds and must be computed up to 2 decimal places.



3. Another way to improve network performance is to add a cache server within the institution without increasing the bandwidth of bottleneck link. The bitrate to the cache server is 10MB/s. Assume there is a 60% cache hit rate. The queuing delay for both cache server and server A follows the formula in Q1. Calculate the average access time in this case. (Assume the network knows cache server so no additional delays are needed to find that cache server; also, your unit should be milliseconds, computed to 2 decimal places).



4 Traceroute - 4 × 3 points

In the next 2 figures, you will see a series of results from running traceroute (with the -q 1 option to send one probe per hop). For each of the results, please answer the following questions:

```
traceroute to www.google.com (216.58.192.132), 64 hops max, 52 byte packets
 1  0148-cslgeneral-net.gw.uiuc.edu (192.17.100.1)  0.958 ms
 2  t-core1-2.gw.uiuc.edu (172.20.101.29)  0.660 ms
 3  t-exiti1.gw.uiuc.edu (130.126.0.162)  0.321 ms
 4  t-fw1.gw.uiuc.edu (130.126.0.134)  0.716 ms
 5  t-exite1.gw.uiuc.edu (130.126.0.141)  1.067 ms
 6  t-dmzo.gw.uiuc.edu (130.126.0.202)  1.087 ms
 7  ur1rtr-uiuc.ex.ui-iccn.org (72.36.127.1)  1.100 ms
 8  t-ur2rtr.ix.ui-iccn.org (72.36.126.66)  1.413 ms
 9  r-equinix-isp-ae0-2244.wiscnet.net (216.56.50.49)  4.007 ms
10  74.125.49.37 (74.125.49.37)  4.113 ms
11  209.85.254.157 (209.85.254.157)  4.390 ms
12  216.239.42.149 (216.239.42.149)  4.459 ms
13  216.239.42.153 (216.239.42.153)  4.375 ms
14  ord36s01-in-f132.1e100.net (216.58.192.132)  4.414 ms
```

```
traceroute to www.auckland.ac.nz (130.216.159.127), 64 hops max, 52 byte packets
 1  0148-cslgeneral-net.gw.uiuc.edu (192.17.100.1)  0.967 ms
 2  t-core1-1.gw.uiuc.edu (172.20.101.25)  0.536 ms
 3  t-exit1.gw.uiuc.edu (130.126.0.242)  0.407 ms
 4  t-fw1.gw.uiuc.edu (130.126.0.134)  0.666 ms
 5  t-exite1.gw.uiuc.edu (130.126.0.141)  0.937 ms
 6  t-dmzo.gw.uiuc.edu (130.126.0.202)  12.626 ms
 7  ur1rtr-uiuc.ex.ui-iccn.org (72.36.127.1)  1.051 ms
 8  t-ur2rtr.ix.ui-iccn.org (72.36.126.66)  1.576 ms
 9  internet2-710rtr.ex.ui-iccn.org (72.36.127.158)  4.107 ms
10  et-7-1-0.4070.rtsw.kans.net.internet2.edu (198.71.45.15)  21.305 ms
11  et-4-1-0.4070.rtsw.salt.net.internet2.edu (198.71.45.19)  41.337 ms
12  et-4-1-0.4070.rtsw.salt.net.internet2.edu (198.71.45.19)  41.280 ms
13  aarnet-1-is-jmb-776.lsanca.pacificwave.net (207.231.241.149)  81.268 ms
14  et-1-2-1.pe1.a.koa.aarnet.net.au (113.197.15.86)  205.814 ms
15  et-1-2-1.pe1.a.koa.aarnet.net.au (113.197.15.86)  205.753 ms
16  et-1-0-0-202.and12-nsh.reannz.co.nz (182.255.119.201)  205.921 ms
17  br-cpf1-north.net.auckland.ac.nz (130.216.95.106)  206.111 ms
18  cxj-alfa-430.net.auckland.ac.nz (130.216.95.122)  208.200 ms
19  cxj-alfa-430.net.auckland.ac.nz (130.216.95.122)  207.881 ms
20  *
21  www.auckland.ac.nz (130.216.159.127)  206.567 ms
```

1. Which hop(s) (if any) is transoceanic

2. Based on the RTT to the last hop, what's the furthest away the corresponding server could possibly be located? (Note: use speed of packet propagation: (2×10^8) m/s.)

3. Sometimes the RTT of a subsequent hop is lower than the RTT of a previous one. Give one reason for this.

Traceroute to www.google.com

1. Transoceanic hop: None
   This traceroute stays within the US, no transoceanic hops.
2. RTT of last hop is 4.414ms, thus max distance of server is:
   Speed * Time = 2*10^8 * 0.004414 = 88280 meters = 88.28 km
3. Reason for lower RTT in subsequent hops:
   Routing change, more optimized path taken hence shorter propagation distance.

Traceroute to www.auckland.ac.nz

1. Transoceanic hop: From hop 10 onward it's transpacific.
2. RTT of last hop is 206.567ms, thus max distance of server is: Speed * Time = 2*10^8 * 0.206567 = 41313400 meters = 41331 km
3. Same as above, lower RTT in subsequent hops could be due to routing change and path optimization.

5 HTTP

5 HTTP - 7 × 3 points

Suppose a webpage has nothing but 10 large images each of size 10 MB. A client wants to access the webpage and load the images in his browser. The RTT between the client and the server is 40 ms and the transmission rate at the server is 500 MB/s. How long will it take to load the webpage in each of the following cases? (Note: the size of the object for indexing is negligible.) For all answers, please answer in milliseconds, and include the detail

1. Using Non-Persistent HTTP?
2. Using Persistent HTTP?
3. Using Pipelined Persistent HTTP?

1. **Using Non-Persistent HTTP:**
   - 10 requests x 2 (request and response) x 40 milliseconds (RTT) = 800 milliseconds
   - Plus image transfer time: 10 images x 10 MB / 500 MB/s = 200 milliseconds

   Total time: 800 milliseconds + 200 milliseconds = 1000 milliseconds
2. **Using Persistent HTTP:**

   Calculation:
   - 1 connection establishment and termination, requiring 2 x 40 milliseconds = 80 milliseconds
   - 10 requests x 40 milliseconds (RTT) = 400 milliseconds
   - Plus image transfer time: 10 images x 10 MB / 500 MB/s = 200 milliseconds

   Total time: 80 milliseconds + 400 milliseconds + 200 milliseconds = 680 milliseconds
3. **Using Pipelined Persistent HTTP:**

   Calculation:

- 1 connection establishment and termination, requiring 2 x 40 milliseconds = 80 milliseconds
- 10 requests x 40 milliseconds (RTT) = 400 milliseconds
- Image transfer time: 10 images x 10 MB / 500 MB/s = 200 milliseconds

Total time: 80 milliseconds + 400 milliseconds + 200 milliseconds = 680 milliseconds

6 Client-Server

Imagine spreading an F-bit file among N peers using a client-server structure. Let the server have a maximum upload capacity μs, and each client c has a download capacity dc. Assume the server can serve multiple clients simultaneously and fluidly set the rate for each client rc.

1. Suppose μs/N ≤ dmin, where dmin = minc dc is the minimum download rate. How would you set the rates rc for each client so that the file is fully distributed to all clients in a minimum time? (i.e. you are minimizing the time that the slowest client receives the file.) What would the distribution time be?

2. Suppose now that μs/N > dmin. How would you set the rates rc now to fully distribute the file to the clients in a minimum time? And what would this time be?

Case 1: μs/N ≤ dmin

In this scenario, the server's maximum upload capacity is limited by the minimum download capacity of the clients. To minimize the time it takes to distribute the file to all clients, we evenly distribute the server's upload capacity among the clients:

1. Set the rate rc for each client to dmin. This ensures that each client receives data at their minimum download rate, so no client has to wait for an extended period.
2. The total size of the file is F bits, and the server's maximum upload capacity is μs. Therefore, the distribution time is F / μs. Since μs/N ≤ dmin, this time represents the time required for the slowest client to receive the file.

Distribution time: F / μs

Case 2: μs/N > dmin

In this case, the server's maximum upload capacity allows a higher total upload rate without being limited by the minimum download rate of the clients. Therefore, we can distribute the file more quickly:

1. Calculate the total available upload rate, which is μs.
2. Distribute the server's upload rate evenly among the N clients, with each client's rate rc = μs / N.
3. The total size of the file is F bits, and each client receives data at a rate of rc. Therefore, the distribution time is F / (μs / N) = (F * N) / μs.

In this scenario, the distribution time is shorter because the server's upload capacity allows a faster distribution rate.

Distribution time: (F * N) / μs

7 DNS

7 DNS - 7 + 3 points

The task requires using the **dig** command to provide answers. To ensure accurate results, it is recommended to perform these steps from a computer located on a campus network. The user can refer to the dig documentation to understand how to utilize it.

1. Starting from one of the root servers a–m.root-servers.net, perform an **iterative** lookup for the host www.eecs.mit.edu to get the ip address. For instance, you can initiate the search by using the following command:
dig @h.root-servers.net www.eecs.mit.edu
(You may first use **dig www.eecs.mit.edu** to get the canonical name and use the canonical name for the following query.)
show the process of your query and point out:
(1) the domain name of the name server being visited
(2) the IP address of the name server that is currently being used
(3) For how long can you store the results in cache.

2. Can you explain why the DNS protocol tends to utilize UDP rather than TCP?

(1)

```
; <<>> DiG 9.18.12-0ubuntu0.22.04.3-Ubuntu <<>> @h.root-servers.net
www.eecs.mit.edu
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 30289
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 13, ADDITIONAL: 27
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: dab1e4c207fe06bcdfe58d2865450a6147c3b9124937628b (good)
;; QUESTION SECTION:
;www.eecs.mit.edu.              IN      A

;; AUTHORITY SECTION:
edu.                    172800  IN      NS      c.edu-servers.net.
edu.                    172800  IN      NS      g.edu-servers.net.
edu.                    172800  IN      NS      a.edu-servers.net.
edu.                    172800  IN      NS      d.edu-servers.net.
edu.                    172800  IN      NS      k.edu-servers.net.
edu.                    172800  IN      NS      b.edu-servers.net.
edu.                    172800  IN      NS      i.edu-servers.net.
edu.                    172800  IN      NS      h.edu-servers.net.
edu.                    172800  IN      NS      j.edu-servers.net.
edu.                    172800  IN      NS      e.edu-servers.net.
edu.                    172800  IN      NS      f.edu-servers.net.
```

```
edu.                    172800  IN      NS      m.edu-servers.net.
edu.                    172800  IN      NS      l.edu-servers.net.

;; ADDITIONAL SECTION:
a.edu-servers.net.      172800  IN      A       192.5.6.30
b.edu-servers.net.      172800  IN      A       192.33.14.30
c.edu-servers.net.      172800  IN      A       192.26.92.30
d.edu-servers.net.      172800  IN      A       192.31.80.30
e.edu-servers.net.      172800  IN      A       192.12.94.30
f.edu-servers.net.      172800  IN      A       192.35.51.30
g.edu-servers.net.      172800  IN      A       192.42.93.30
h.edu-servers.net.      172800  IN      A       192.54.112.30
i.edu-servers.net.      172800  IN      A       192.43.172.30
j.edu-servers.net.      172800  IN      A       192.48.79.30
k.edu-servers.net.      172800  IN      A       192.52.178.30
l.edu-servers.net.      172800  IN      A       192.41.162.30
m.edu-servers.net.      172800  IN      A       192.55.83.30
a.edu-servers.net.      172800  IN      AAAA    2001:503:a83e::2:30
b.edu-servers.net.      172800  IN      AAAA    2001:503:231d::2:30
c.edu-servers.net.      172800  IN      AAAA    2001:503:83eb::30
d.edu-servers.net.      172800  IN      AAAA    2001:500:856e::30
e.edu-servers.net.      172800  IN      AAAA    2001:502:1ca1::30
f.edu-servers.net.      172800  IN      AAAA    2001:503:d414::30
g.edu-servers.net.      172800  IN      AAAA    2001:503:eea3::30
h.edu-servers.net.      172800  IN      AAAA    2001:502:8cc::30
i.edu-servers.net.      172800  IN      AAAA    2001:503:39c1::30
j.edu-servers.net.      172800  IN      AAAA    2001:502:7094::30
k.edu-servers.net.      172800  IN      AAAA    2001:503:d2d::30
l.edu-servers.net.      172800  IN      AAAA    2001:500:d937::30
m.edu-servers.net.      172800  IN      AAAA    2001:501:b1f9::30

;; Query time: 39 msec
;; SERVER: 198.97.190.53#53(h.root-servers.net) (UDP)
;; WHEN: Fri Nov 03 22:57:54 CST 2023
;; MSG SIZE  rcvd: 868
```

- (Server Domain Name): h.root-servers.net
- (Server IP Address): 198.97.190.53
- there is no answer section so the TTL is unknown.

(2)

The DNS protocol primarily uses UDP for its query and response mechanism because, compared to TCP, UDP is faster and requires less overhead. UDP is a connectionless protocol, making it more suitable for simple, lightweight, and quick communication. However, if the response size exceeds the maximum UDP payload size or if reliability is crucial, DNS can fall back to using TCP. UDP is the default choice due to its efficiency and low-latency characteristics.