Exercise 1: Request Routing in a CDN.

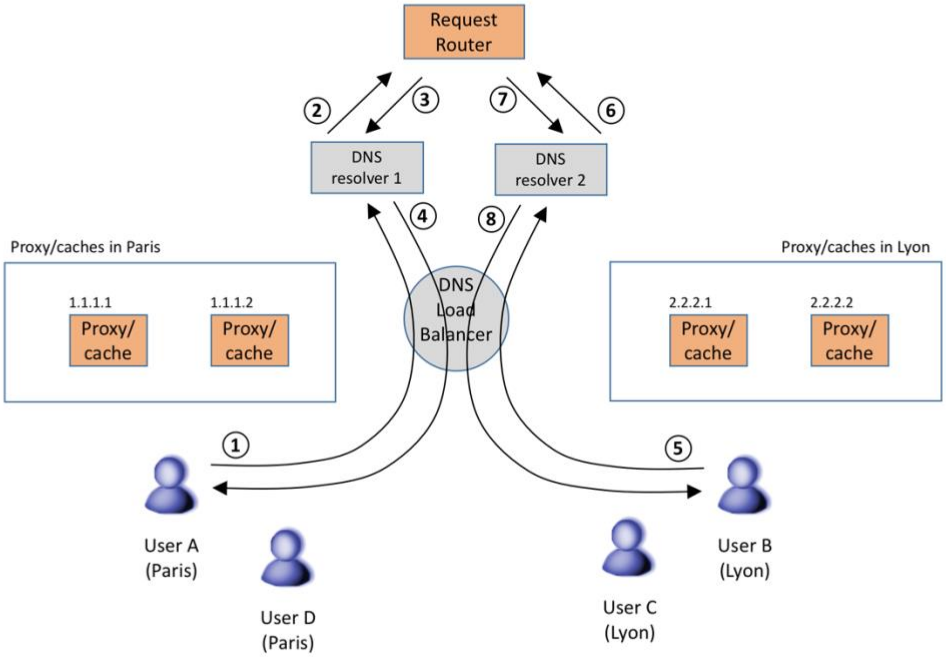
The Request Router is the component of a CDN (Content Delivery Network) that directs users to the best proxy/cache servers that distribute requested content.

A common request routing technique is based on DNS. With this technique, a Request Router receives DNS queries from a user through the DNS resolver of the user's operator. These DNS queries relate to a domain name whose zone is delegated to the Request Router and which is supposed to be associated with the CDN.

Consider a simplified scenario where a network operates with the following assumptions:

1. The operator has two centralized DNS resolvers that are used with a load balancing mechanism.
2. The two DNS resolvers do not communicate with each other.
3. The load balancer directs DNS queries to either DNS resolver with a simple round robin mechanism, which means that one query is sent to the first resolver, the next query is sent to the second resolver, the next query to the first resolver, and so on.
4. Both resolvers will have any DNS request delegated to the Request Router in case of no entry in the cache.
5. Resolvers do not know the load status and availability of the caching servers, unlike the Request Router which is aware of them in real time.
6. The Request Router responds to requests with the address of the least loaded cache server (and closest to the source of the DNS request, if it has the ability to locate it).
7. At the start of the exchanges that follow, the two resolvers have no cached entries and the proxy/cache servers serve no requests.
8. When a resolver receives a DNS response from the Request Router, it caches it for a TTL (Time To Live) time that matches the time specified in the DNS response.
9. Negligible DNS message propagation and processing time.

The following diagram shows DNS exchanges that take place in this operator's network. For simplicity, in the interest of this exercise, we take a case where only 4 users are active, 2 in Paris and 2 in Lyon.



At time T, the following exchanges start

User A in Paris issues a DNS query for cdn.example.com. This request is relayed by the load balancer to the DNS 1 resolver.

Source:	Destination:	Message:	
UserA	DNS Resolver	DNS Request	cdn.example.com

The DNS 1 resolver does not have a cached DNS entry for cdn.example.com, so it relays the request to the CDN's Request Router.

Source:	Destination:	Message:	
DNS Resolver 1	Request Router	DNS Request	cdn.example.com

The Request Router receives the DNS request. It does not know whether the user is in Paris or Lyon, because the source address of the query is one of the operator's resolver, which is centralized. Assuming that all the proxies/caches in Paris and Lyon are completely free, it chooses a cluster randomly, for example the one in Paris, and issues a response specifying the addresses of the proxy/cache servers in Paris:

Source:	Destination:	Message:	
Request Router	DNS Resolver 1	DNS Reply	cdn.example.com 300 IN A 1.1.1.1
			cdn.example.com 300 IN A 1.1.1.2

The number 300 represents the TTL in seconds. DNS 1 resolver caches the response from the previous point, which will therefore be valid for 300 seconds, and sends it back to user A.

Source:	Destination:	Message:	
DNS Resolver 1	User A	DNS Reply	cdn.example.com 300 IN A 1.1.1.1
			cdn.example.com 300 IN A 1.1.1.2

The subsequent steps required to establish an HTTP session between User A and a proxy/cache in Paris are not shown here, as they are not the subject of this exercise.

At time T + 100 seconds User B in Lyon issues a DNS query type A for cdn.example.com

At time T + 150 seconds, the proxy/cache server with the address 1.1.1.1 goes down

At time T + 200 seconds, user C in Lyon sends a DNS query for cdn.example.com

At time T + 350 seconds, user D in Paris sends a DNS query for cdn.example.com

Complete the following table with all interactions:

Time T	**Source:**	**Destination:**	**Message:**	

Did user C receive the address of the nearest and most available proxy/cache servers?

Yes ☐ NO ☐

Did user C receive the address of the nearest and most available proxy/cache servers?

Yes ☐ NO ☐

If you answered no to at least one of the questions above, you have identified cases where the DNS responses are not optimal with respect to the objective of directing users to the closest and available proxy/cache servers. Among the solutions mentioned below, could you indicate whether they might solve this problem?

Solution	**Description**	**Does it solve the identified problem? (Yes / No)**
Decrease DNS TTL	Reduce the value of the DNS TTL (Time To Live) in the responses provided by the CDN’s Request Router, so that responses remain in the DNS resolvers’ cache for a shorter time.	
Increase the number of proxy/cache servers	Add 3 more proxy/cache servers in Paris and Lyon.	
Implement EDNS0 Client-Subnet	By using the EDNS0 Client-Subnet extension, resolvers send the Request Router part of the user’s IP address originating the DNS query, which allows the Request Router to better localize them. In addition, resolvers keep separate cache entries for responses intended for one subnet versus another, with configurable subnet precision (e.g., /24). Here we assume users in Paris and Lyon have IP addresses in different /24 subnets.	
Cool the DNS resolvers	Install powerful air conditioning to lower the physical CPU temperature of the DNS resolvers.	
Increase the number of Request Routers	Add another Request Router, possibly in a different network location.	
Use the BGP protocol	Use the AS path prepending metric of BGP.	