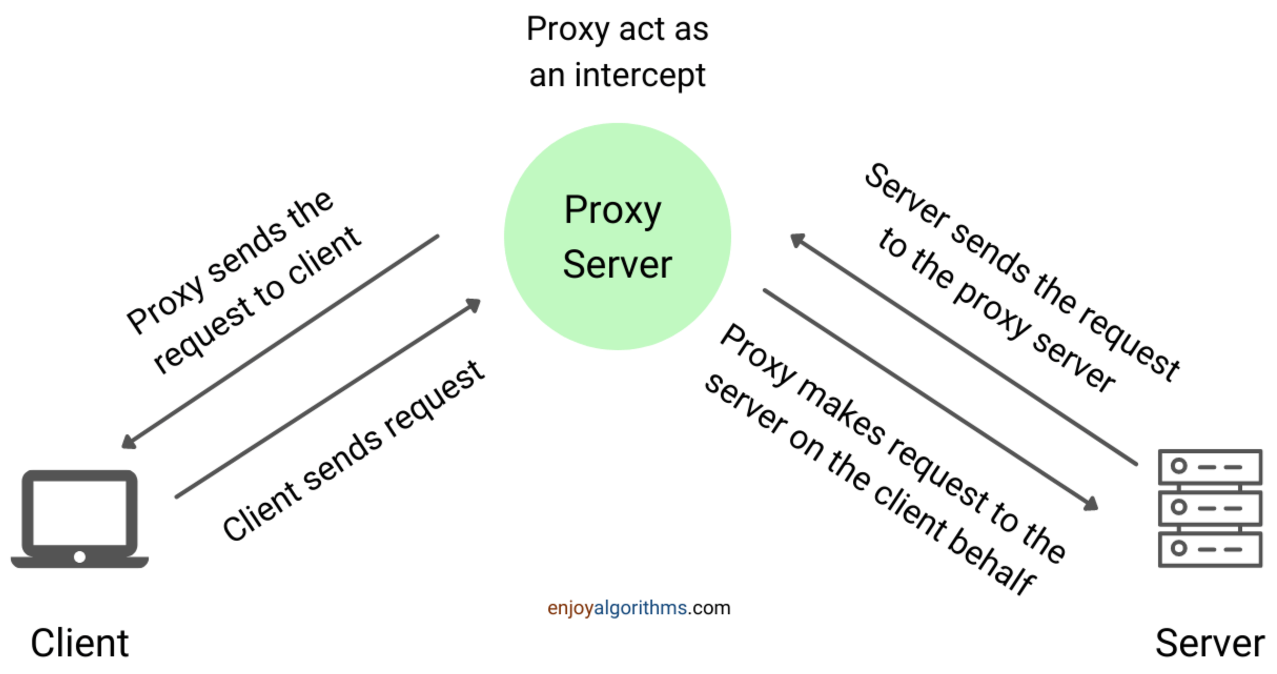
Proxies in System Design (Forward and Reverse Proxy)

In this blog, we will learn about Proxy, an essential concept for a system design interview. Many of us have used tools that help us maintain our privacy or anonymity, or remain safe from third-party users eavesdropping on our connection. Do you know how it works and how it is useful? Let's find out!

What is Forward Proxy?

A forward proxy often referred to as a proxy or proxy server, is a server that sits in front of a group of client machines. When a client issues a request to communicate with a server, instead of going directly to the server, the request goes to the forward proxy, which then forwards the request to the server.

In other words, when clients send requests to websites on the Internet, the forward proxy intercepts these requests and communicates with the servers on behalf of the clients, acting as an intermediary. This allows us to perform certain actions before or after the request gets through to the original destination.


Proxy act as an intercept
Proxy Server
Proxy sends the request to client
Client sends request
Server sends the request to the proxy server
Proxy makes request to the server on the client behalf
enjoyalgorithms.com
Client
Server

Why do we use a forward proxy?

There are several reasons to use a forward proxy when designing a system:

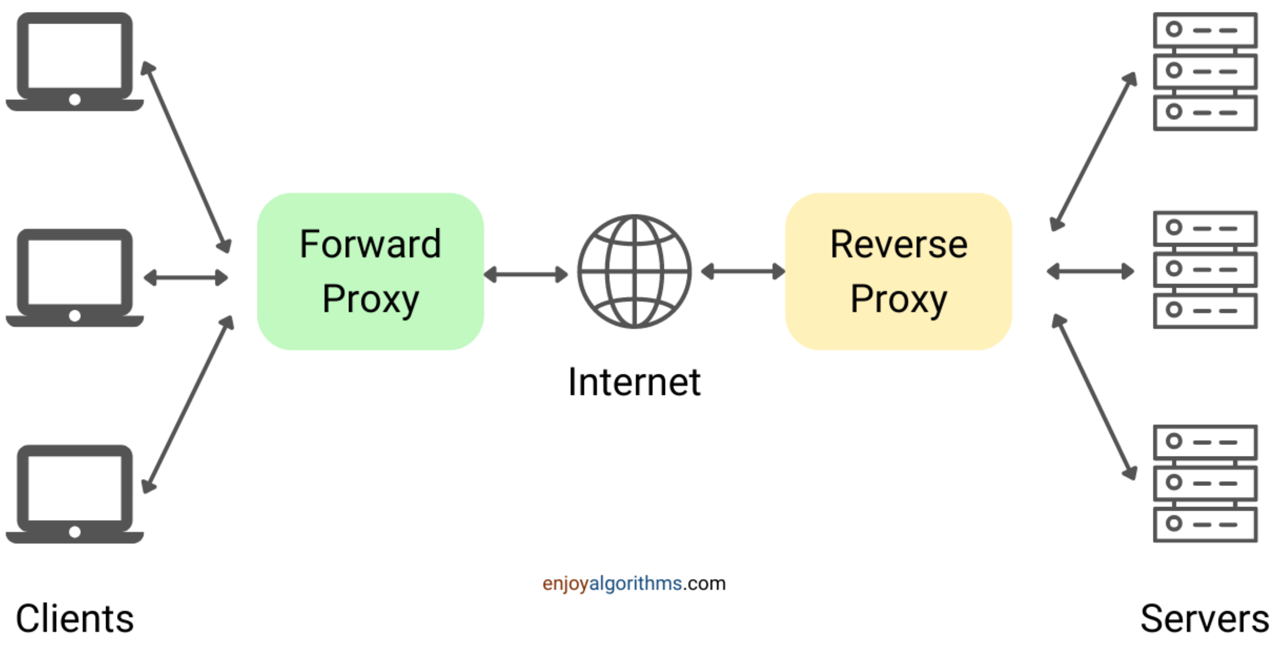
- Forward proxy can **filter network traffic** based on predefined rules. For example, an organization can use a proxy server to block access to specific websites or restrict access to certain types of content. Proxy servers can also enforce security policies and protect against malware and other malicious activities.
- Forward proxy can **provide anonymity** for clients by hiding their IP addresses. This can be useful for accessing content that is restricted in certain regions or for protecting clients' privacy and security.
- Forward proxy can **optimize network traffic** by compressing data, removing unnecessary headers, and reducing the size of transferred data. This reduces bandwidth usage and improves network performance.

What is Reverse Proxy?

A reverse proxy is a proxy server that sits between one or more servers. Reverse proxies are the exact opposite of forward proxies in terms of their interaction pattern: Forward proxy acts on behalf of clients, a reverse proxy acts on behalf of servers. So, unlike a forward proxy, which is used to protect clients, a reverse proxy is used to protect servers.

- When clients send requests to the origin server of a website, the requests are intercepted by the reverse proxy server. The reverse proxy server will then send requests to and receive responses from the origin server. In other words, the client thinks it is directly interacting with the server, and the request actually goes to the reverse proxy without the client knowing about it (or does not know that some other server actually processed its request).
- To simplify the difference between a forward and reverse proxy: A forward proxy ensures that no origin server ever communicates directly with that specific client. On the other hand, a reverse proxy ensures that no client ever communicates directly with that origin server.

Let's take an example: Suppose we type "https://www.enjoyalgorithm.com" in our browser. Our browser makes a DNS query to get the IP address of enjoyalgorithm.com. If enjoyalgorithm.com uses a reverse proxy and configures it correctly, the DNS query will return the reverse proxy's IP address.



What are the use cases of reverse proxy?

A reverse proxy is handy when we design a complex system, and it can be used for many purposes, such as:

- **Security:** When we use a reverse proxy, a website’s origin server IP address is abstracted from the attackers. So to exploit any vulnerabilities, malicious clients can not access them directly. Many reverse proxy servers include features that help protect backend servers against distributed denial-of-service (DDoS) attacks, such as rejecting traffic from specific IP addresses of the client (blacklisting) or limiting the number of requests accepted from each client.
- **Load Balancing:** A website with millions of users visiting every day may find it hard to handle such huge traffic with a single server. Instead, we can use more than one server and use reverse proxy as a load balancing solution to distribute the traffic among servers and prevent any server from getting overloaded.
- **Caching:** A proxy server can cache frequently requested resources such as web pages, images, and videos. This allows the cached content to be served directly from the proxy server, without needing to make requests to the original server. This reduces the load on the web server, speeds up the delivery of content, saves bandwidth, and reduces network latency.
- **SSL encryption:** If you're looking to reduce the computational load on your origin server when it comes to SSL (or TLS) encryption, a reverse proxy can be a great solution. We can configure reverse proxy to decrypt all incoming requests and encrypt all outgoing responses i.e. your origin server won't have to deal with the encryption/decryption process for each individual client request. This can be a major relief for your server's resources, freeing them up to handle other tasks more effectively.

Disadvantages of reverse proxy

Reverse proxies are not always useful, and it has their own drawbacks:

- Adding a reverse proxy to a system can increase its complexity, which can make it more difficult to manage and troubleshoot issues.
- If the reverse proxy fails, the entire system can be affected, as all traffic goes through it. It is important to ensure redundancy and failover mechanisms are in place to mitigate this risk.
- While a reverse proxy can improve performance by caching and load balancing, it can also introduce additional latency and overhead due to the additional processing required for each request.
- When SSL encryption is used, the reverse proxy must handle SSL certificates, which can be challenging to manage if there are many backend servers with different SSL configurations.

Reverse Proxy vs. Load Balancer

Reverse proxy servers and load balancers are components of a client-server system design. They both act as intermediaries between clients and servers, performing efficiency-enhancing functions.

A **load balancer** distributes incoming client requests among a group of servers, returning the response to the appropriate client from the selected server. When a site requires multiple servers, load balancers are commonly deployed because a single server cannot efficiently handle the volume of requests. So the task of the load balancer is to distribute requests in a way that makes the best use of each server's capacity and prevents overload on any server.

A **reverse proxy** accepts a request from a client or set of clients, forwards it to a server, and returns the client's response from the server. Deploying a reverse proxy often makes sense, even with just one web server or application server. While deploying a load balancer only makes sense when multiple servers are available.

Thanks to Navtosh for his contribution in creating the first version of this content. If you have any queries/doubts/feedback, please write us at contact@enjoyalgorithms.com. Enjoy learning, Enjoy system design, Enjoy algorithms!