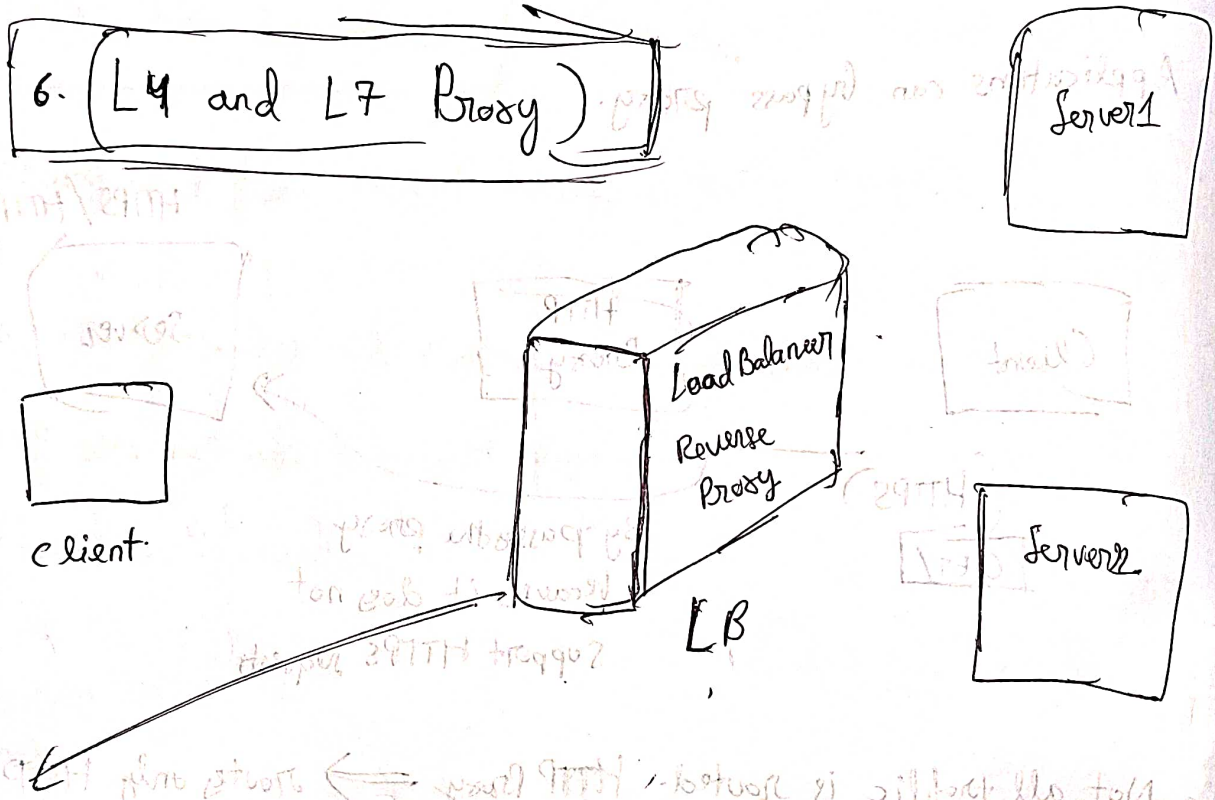
5. (Load Balancer vs Reverse Proxy)

Load Balancer is a type of Reverse Proxy.
It helps in managing and distributing traffic

6. (L4 and L7 Proxy)



Lets say based on some logic the load balancer decides to send the request to (Server 1).

(1st option) TCP connection from (Client to LB), then LB again creates a new TCP connection from (LB to Server 1)

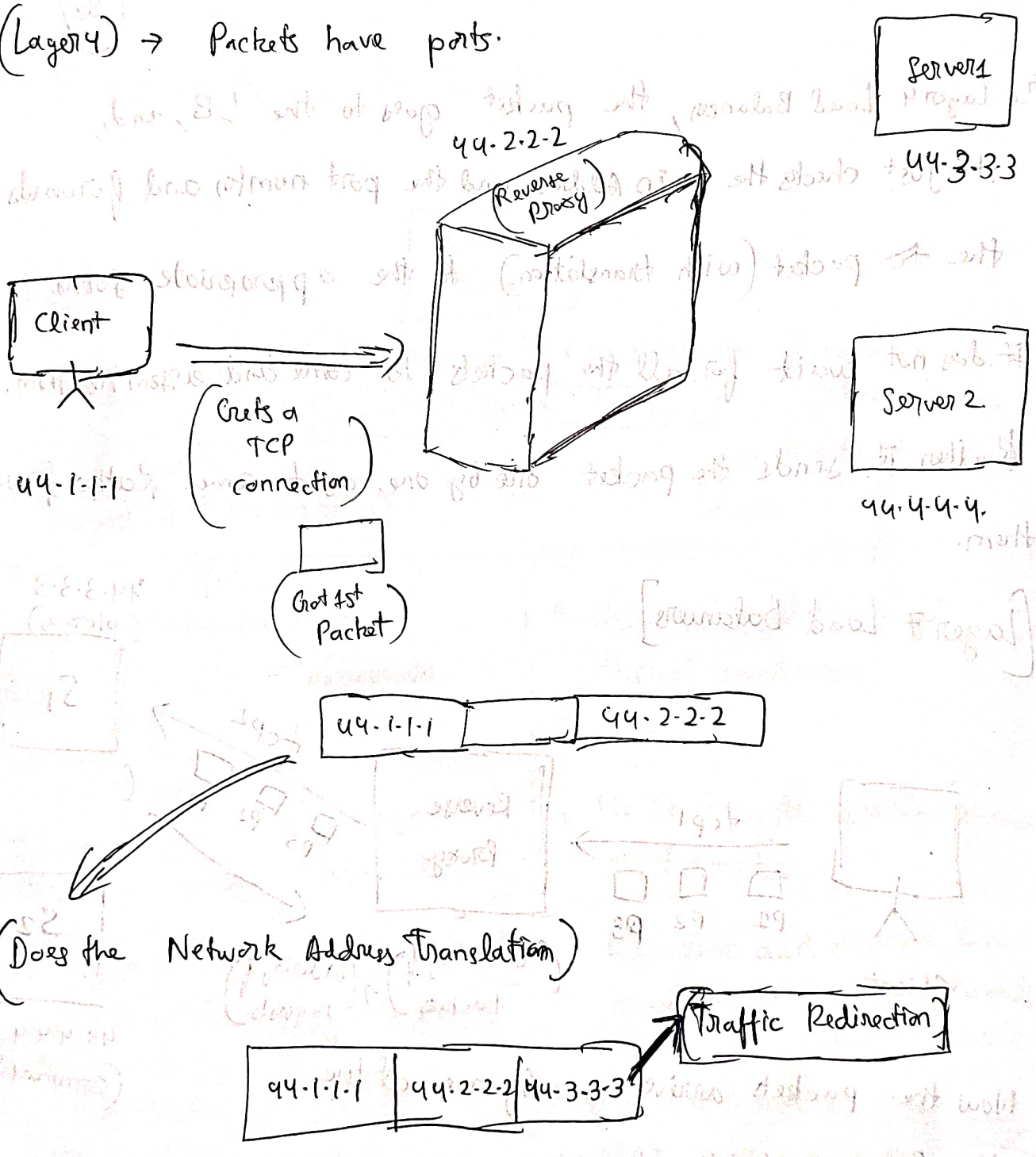
(2nd option) the Load Balancer maintains a pool of TCP connections with the servers so that it does not have to create connections again and again.

[Layer 4 Load Balancer] :

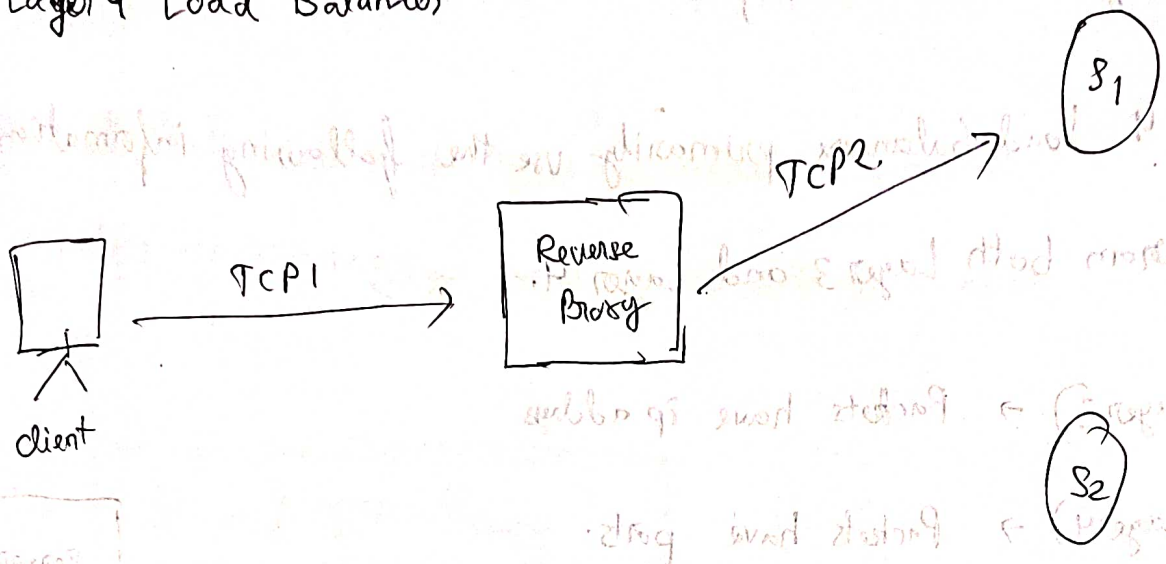
L4 load balancers primarily use the following information from both Layer 3 and Layer 4.

(Layer 3) → Packets have ip address

(Layer 4) → Packets have ports.

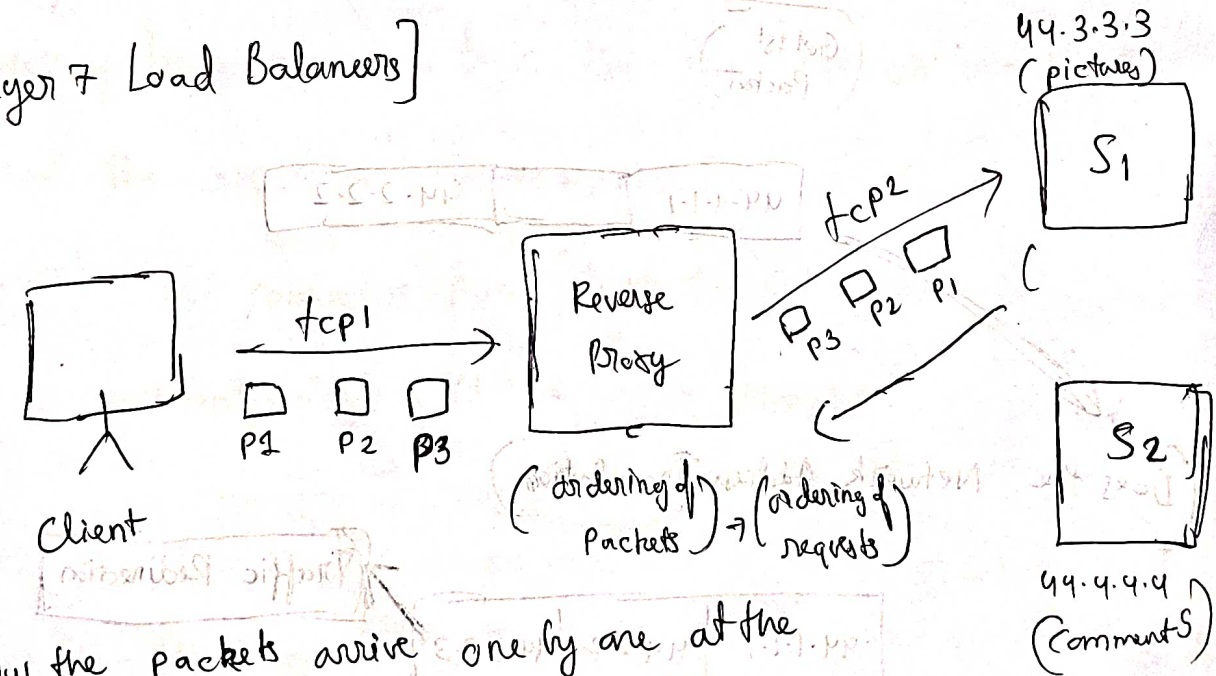


In Layer 4 Load Balancer



In Layer 4 Load Balancer, the packet goes to the LB, and, it just checks the ip Address and the port number and forwards the packet (with translation) to the appropriate server. It does not wait for all the packets to come and assemble them. Rather it sends the packet one by one as it comes. Rather forwards them.

[Layer 7 Load Balancers]

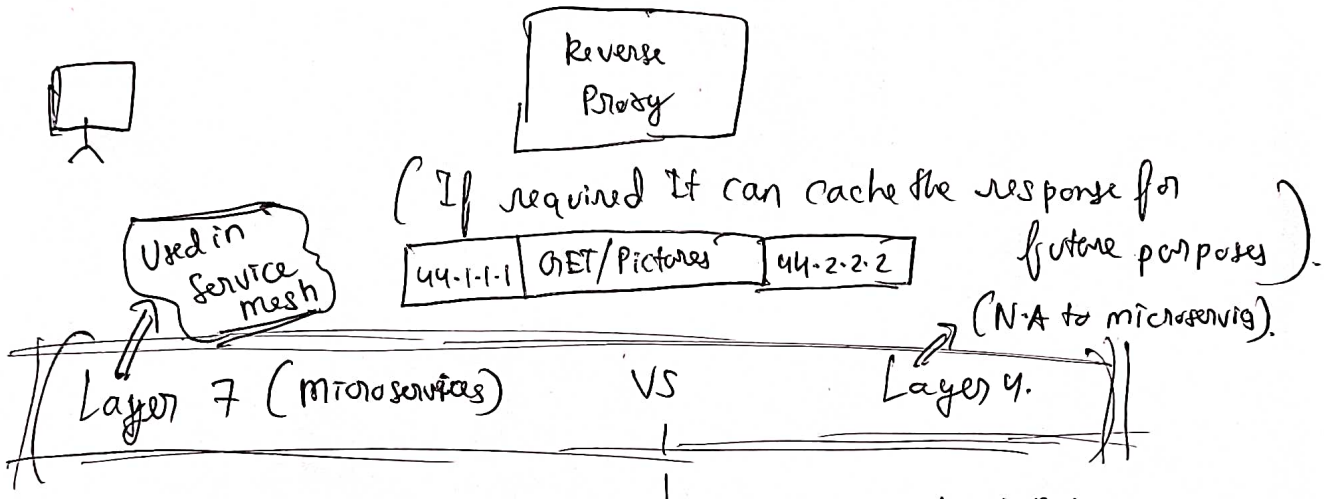


Now the packets arrive one by one at the Load Balancer. And the Load Balancer will this time wait

It waits for all the packets to arrive at the Reverse Proxy. The Load balancer will assemble them back to the Layer 7 mode and see the full content.

And As per the requested data can drive you to the required server.

Load Balancer will have the request



* Smart Load Balancing - Can see data and route to appropriate microservices.
* Data can be cached, As both request and response are decrypted on LB
* More time taking as it looks at data and assembles them (Less Secure)
* Two TCP connections, TLS is terminated at proxy and new TCP [Proxy → Server]

* No smart Load Balancing - typical Round Robin.
* No Caching As simply packets are forwarded.
* Efficient and does not look at content and simply forwards. (More Secure).
* One TCP connections, only NAT is used to redirect to server.