



# Pauta Auxiliar 2

## Proxies y Conexiones Persistentes

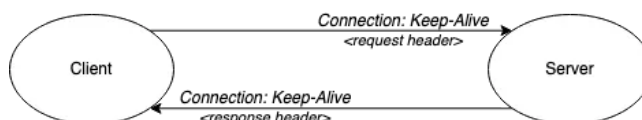
29 de marzo, 2023

### Conceptos relevantes

- Proxy: Servidor intermediario entre un cliente y un servidor. El proxy toma los request de un cliente, puede que evalúe la petición, y luego la redirige hacia el servidor de donde obtiene una respuesta, y finalmente envía esta respuesta al cliente.
- Keep-Alive: Valor para header *Connection* el cual indica que la conexión entre cliente y servidor debe ser persistente. Es decir, no debe cerrarse el canal de comunicación una vez llegada la respuesta del servidor al cliente.

### P1. Proxies y Keep-Alive

En HTTP 1.0, luego de terminado el intercambio entre dos entidades, el canal se cerraba, por lo que cada conexión debía ser restablecida para poder hacer una nueva solicitud. En orientado a conexión, esto resulta muy costoso. Para solucionar este problema, HTTP 1.0 comenzó a reutilizar canales de conexión, indicado por el header *Connection: Keep-Alive*.



Si el servidor no soporta conexiones Keep-Alive, simplemente no manda el header *Connection* en la respuesta y cierra la conexión. Sin el header, el cliente sabe que ese canal no va a ser reutilizado en caso de querer hablar de nuevo, sino que debe establecer uno nuevo.

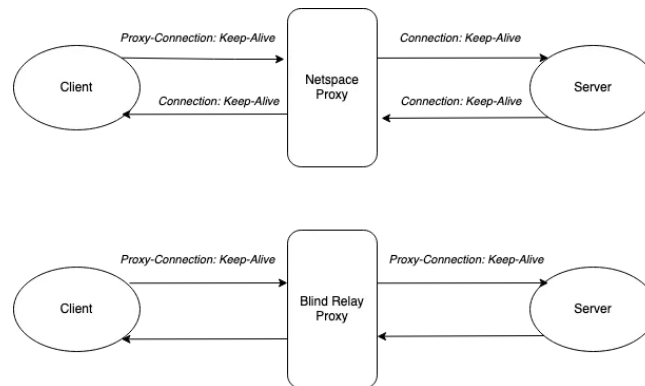
Hasta aquí todo bien. Pero, ¿Qué ocurre si se agrega un proxy que simplemente redirige los mensajes que le llegan? ¿Existirá algún problema?

**R:** Este tipo de proxies que simplemente redirigen los mensajes de un cliente a un servidor, no leen el contenido de lo que reciben, simplemente lo envían; incluido el header *Connection*. Luego, el servidor ve que le llega una request con el header *Connection* con valor *Keep-Alive*, a lo que responde también con el mismo header y valor, y mantiene la conexión entre sí mismo y el proxy abierta.

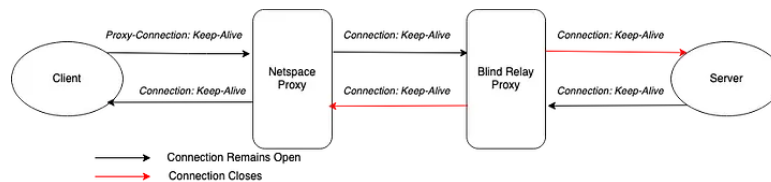
Luego, el proxy reenvía esa respuesta del servidor al cliente, este recibiendo la confirmación de mantener la conexión abierta, por lo que también lo hace, pero con el proxy. En este momento, tanto cliente como servidor están manteniendo un canal de comunicación abierto con el proxy.

El proxy no soporta *Keep-Alive*, simplemente reenvía paquetes HTTP, por lo que ahora cierra ambas conexiones al ver que la request ya fue resuelta. Acto seguido, descarta los mensajes por parte del cliente, ya que no hay canal establecido, a pesar de que el cliente cree que sí. Y el servidor queda con el socket ocupado, esperando nuevas requests por parte del proxy, las cuales nunca llegan.

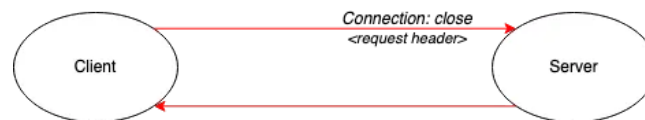
Ante lo anterior, se creó una solución. Esta consiste en un nuevo header, *Proxy-Connection*. Luego, si el proxy es capaz de mantener la conexión abierta, elimina ese header y añade un nuevo header *Connection* al paquete antes de reenviarlo al servidor. Si el proxy no sabe manejarlo, simplemente reenviará este header, el cual va a ser ignorado por el servidor y no responderá con el header *Connection: Keep-Alive*, cerrando así la conexión y evitando el problema. El siguiente diagrama lo ilustra.



Aun así, este modelo se cae en caso de existir un proxy que solo reenvía mensajes, además del proxy que si maneja conexiones persistentes.



Finalmente, en HTTP/1.1 se diseñó para siempre trabajar con conexiones persistentes, a menos que se especifique lo contrario. También se utiliza el header *Connection*, pero solo con el valor *close*, el cual indica cuando se debe cerrar la conexión.



Si quieren leer más: <https://www.oreilly.com/library/view/http-the-definitive/1565925092/ch04s05.html>

## P2. Por qué bytes

Los headers en HTTP se encuentran codificados en texto plano, sin embargo, muchos otros protocolos de comunicación prefieren usar directamente bytes ¿Por qué será esto? ¿Cuál es su utilidad práctica?

**R:** Durante la auxiliar se mostró un pequeño código en el que se ejemplificó cómo utilizar bytes para guardar la información de los headers. Esto resulta en una mejor utilización de los recursos, reduciendo el tamaño de lo que se quiere enviar. Se subirá el código a material docente.