

Implementación de un Servidor Proxy HTTP en C++ usando Sockets de POSIX.

Angel David Talero Peñuela
Departamento de Ingeniería de Sistemas,
Facultad de Ingeniería
Pontificia Universidad Javeriana
Bogotá, Colombia
angel_talero@javeriana.edu.co

Samir Alejandro Sánchez
Departamento de Ingeniería de Sistemas,
Facultad de Ingeniería
Pontificia Universidad Javeriana
Bogotá, Colombia
sanchezsamir@javeriana.edu.co

Gerardo Hidalgo Carroll
Departamento de Ingeniería de Sistemas,
Facultad de Ingeniería
Pontificia Universidad Javeriana
Bogotá, Colombia
gerardo.hidalgoc@javeriana.edu.co

Abstract - In the last decades the importance of the web pages has become the main anchor of the businesses of the XXI century, becoming the largest showcase in the world, where any need is consulted and solved in the network, for the above mentioned it is important to understand the functioning of cyberspace and the great functionalities that allow us. Group 5 of the subject "communication and networks" explains how is the functioning of the web pages by means of a software that implements the connection of server-client requests, presenting a proxy server itself that guides the connection with other servers that are in the network accepting the respective requests for the presentation of the web pages that handle the "HTTP" protocol.

Keywords - HTTP, Server, Web, connection, Clients, Protocols, Proxies

Resumen - En las últimas 25/10/2021 décadas la importancia de las páginas web se ha convertido en el ancla principal de los negocios del siglo XXI, volviéndose la vitrina más grande del mundo, donde cualquier necesidad es consultada y solucionada en la red, por lo anterior comentado es importante comprender el funcionamiento del ciberespacio y las grandes funcionalidades que nos permiten. El grupo 5 de la materia "comunicación y redes" explica como es el funcionamiento de las páginas web mediante un software que implementa la conexión de solicitudes de servidor-cliente, presentando un servidor proxy propio que guía la conexión con otros servidores que se encuentran en la red aceptando las respectivas solicitudes para la presentación de las páginas web que manejan el protocolo "HTTP".

Palabras clave - HTTP, Servidor, Web, conexión, Clientes, Protocolos

I. INTRODUCCIÓN

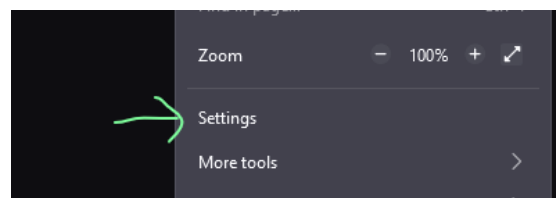
El ciberespacio es una localización intangible, pero libre para el desarrollo de herramientas de todo tipo, donde se busca la implementación de contenido y servicios, que están disponibles las 24 horas del día los 7 días de la semana, no tiene ningún tipo de mandato o gobierno que interfiera en este espacio, en la antigüedad las páginas web eran utilizadas para temas totalmente corporativos o educativos, por el motivo de que el internet no era para todo público. Pero con la creación de los computadores de escritorio, portátiles y celulares, se logró implementar la red como un servicio para toda la sociedad, lo que ha generado que lleguemos a la era actual (Era de la información), donde el conocimiento de años de investigación la encontremos a un clic de distancia. Pero todo esto es posible, mediante procesos complejos para el correcto funcionamiento de la internet, por eso mismo el grupo 5 de "comunicación y redes" explicara uno de los procesos necesarios para el ingreso de sitios web, entraremos en la comprensión de conceptos básicos para el entendimiento del

programa presentado en este documento, que tiene la intención de implementar el funcionamiento del servidor proxy con su respectiva conexión al cliente y a los servidores que tengan contratados los dueños de las páginas web a consultar, estas tendrá como requisito el protocolo "HTTP", lo cual son páginas web que no contienen certificados "SSL" que convierte la información que se esté transfiriendo entre el cliente y el servidor en datos cifrados, lo que permite mayor seguridad, pero en nuestro caso deben ser páginas que no posean este paso de encriptación y certificado.

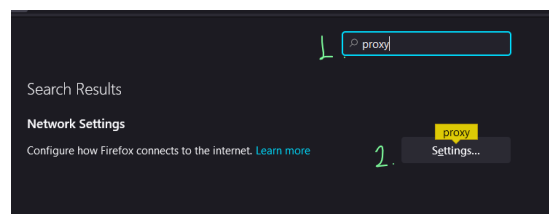
II. CÓMO SE CONECTA EL PROGRAMA AL NAVEGADOR

Para realizar la conexión del programa con el navegador se tiene que hacer uso de la opción de 'Proxy', al invocar el ejecutable del programa, se debe enviar por parámetro el número del puerto TCP que será utilizado para escuchar las peticiones del navegador (ej. *ProxyServer 8024*), la configuración de Proxy en el navegador depende del navegador web que se esté usando, para este ejemplo utilizaremos *Mozilla Firefox*:

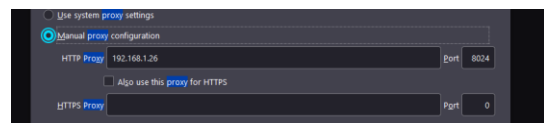
1. Se abre la página de configuraciones del navegador



2. Se busca la opción de Proxy



3. Se selecciona 'Configuración Manual', se coloca la IP del computador que está corriendo el programa, y el puerto que se mandó por parámetro:



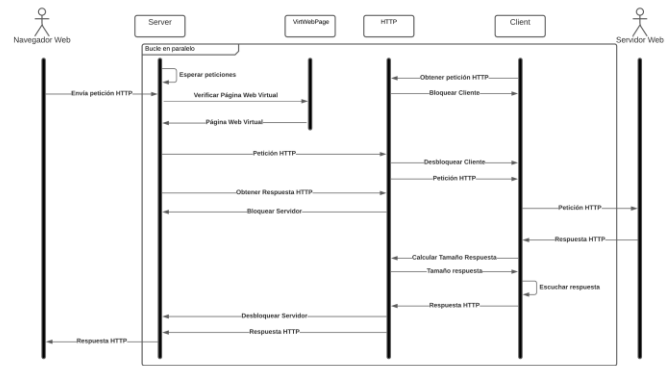
III. VISTA GENERAL

El funcionamiento del programa se da a través de *hilos*, una característica que le permite a nuestro programa ejecutar ambas clases: Cliente y Servidor, de manera paralela y así poder recibir y enviar solicitudes desde y hacia varios equipos al mismo tiempo.

Esto se da a través de cuatro clases, al ejecutarse el programa y manejarse los argumentos se crea la clase *VirtWebPage* quien leerá el archivo de páginas web virtuales y será el encargado de manejarlas, luego se crea la clase HTTP quien será la clase intermedia encargada de intercambiar peticiones y solicitudes HTTP entre la clase Client y Server así como bloquear y desbloquear los hilos asociados a estas clases para el correcto funcionamiento de la concurrencia, finalmente la clase Server se encarga de recibir y enviar solicitudes desde y hacia el navegador web y la clase Client envía y recibe solicitudes desde y hacia el servidor web de destino, estas dos ultimas clases poseen un método llamado *listener* (Heredado de una interfaz 'Connection'), que recibe como parámetro una referencia a la clase HTTP que coordina las clases.

para iniciar el funcionamiento de las clases Server y Client se debe llamar a este método *listener* a través de un hilo, a partir de este momento ambas clases se encuentran en un bucle infinito en el que están constantemente esperando a una petición HTTP, para detener la ejecución de estas dos clases y sus hilos asociados y detener el programa en general, se puede generar una interrupción presionando las teclas Ctrl+C (*Señal SIGINT de POSIX*)

El diagrama de secuencias que modela el funcionamiento de todas estas clases en conjunto es el siguiente:



IV. CLASE SERVER

El módulo encargado de recibir las peticiones del navegador es la clase "Server", quien escucha las peticiones HTTP por el Puerto que fue especificado al llamar al programa, una vez una solicitud llega a la clase 'Server', este se encarga de verificar si se trata de una página web virtual y a su vez resuelve el nombre de host haciendo uso del sistema DNS, una vez se ha identificado la IPv4 de destino, el Servidor envía la petición a una clase intermedia llamada 'HTTP' quien será la encargada de hacerle saber a la clase 'Cliente' sobre la petición, el hilo de la clase Servidor se bloqueará hasta que reciba la respuesta de la clase 'Cliente' que será enviada a través de la clase intermedia 'HTTP', una vez reciba la solicitud, el hilo Servidor se desbloqueará y enviará la respuesta de vuelta al navegador web.

V. CLASE CLIENT

El módulo encargado de recibir las peticiones del server de destino es la clase Client, esta clase se encarga de recibir las peticiones, reenviárselas al servidor, leer la información y determinar cuántos bits de data llegan por el puerto, finalmente reenviar esta información a la clase Server para que pueda llegar al navegador

VI. REFERENCIAS

- [1] Free Software Foundation (2018) 'Bash manual page', Linux (Version 4.99) [Computer program]. Copy of manual text available at <http://man7.org/linux/man-pages/man1/bash.1.html>