Universidad de Valparaíso CHILE

REDES DE COMPUTADORES: TAREA 1

Implementación de un cliente HTTP en C++

Entendimiento modelo 4+1

Nombres:

Nicolas Ignacio Toledo Toledo
Leonardo Esteban Arancibia Vicencio
Augusto Ignacio Pinochet Contreras

Profesor:

Gabriel Astudillo

15/11/2018
REDES DE COMPUTADORES I

Universidad de Valparaíso CHILE

INTRODUCCIÓN

Existen varios tipos de protocolos que se desarrollan a lo largo de la carga de una página web, algunos son vistos por el usuario y otros no, los primeros son claramente los objetos visualizados en la pantalla del computador, y los otros simplemente son complejos de entender. El propósito de este trabajo es investigar cuántos de esos protocolos (vistos y no vistos) son ejecutados al poder navegar por el internet, además, ampliar nuestros conocimientos en los lenguajes de programación y nuestras mentes como ingenieros, desarrollando un programa en el lenguaje C++, que permita poder guardar una página web en un archivo “txt”, todo esto realizándose desde consola.

Universidad de Valparaíso CHILE

OBJETIVOS

- Implementar cliente HTTP simple en C++14.
- Capturar el tráfico HTTP a través de wireshark hacía muchos servidores web
- Poder entender qué es HTTP y HTML, y cómo están relacionados
- Analizar cuales son las acciones y protocolos HTTP que se utilizan en la conexión
- Implementación del cliente HTTP basado en texto
- Guardar código HTML de una página web en un archivo de texto
- Poder entender de qué trata el modelo 4+1 e implementarlo en el proyecto

DESARROLLO

¿Que es HTTP?:

El Protocolo de transferencia de hipertexto, es el mecanismo primario para transportar objetos por la web y es una aplicación de protocolo (define cómo interactúan un cliente y un servidor.)

HTTP es un protocolo de [petición-respuesta], el cual significa que un cliente envía un mensaje de petición y el servidor retorna un mensaje de respuesta (el mensaje está basado en texto entendible para humanos.)

HTTP también es llamado un protocolo sin estado, a lo cual se refiere a que es un protocolo de comunicaciones que trata cada petición como una transacción independiente que no tiene relación con las solicitudes anteriores.

¿Qué es HTML?:

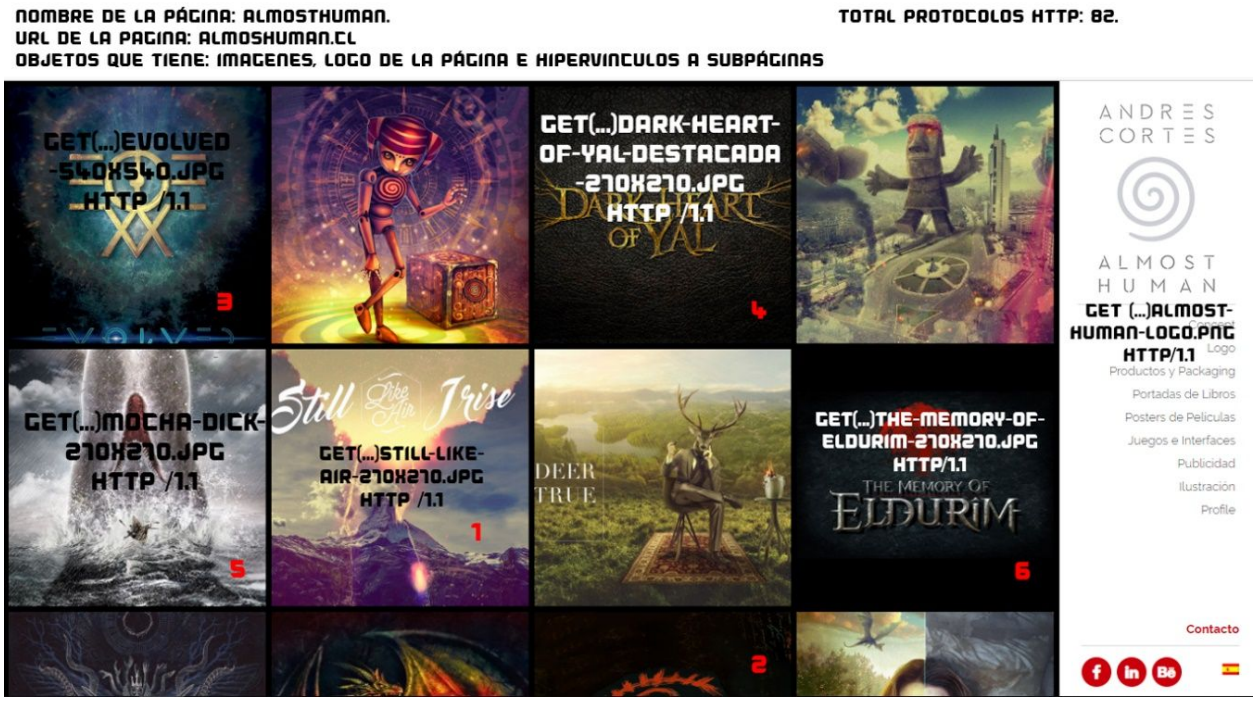
HyperText Markup Language(Lenguaje de Marcado para Hipertextos) es la representación estándar para documentos de hipertextos, HTML define el diseño y el formato de una página web y permite a los autores incrustar referencias de hipervínculo a otros recursos de la web.

¿Cuál es la relación de HTML con HTTP?:

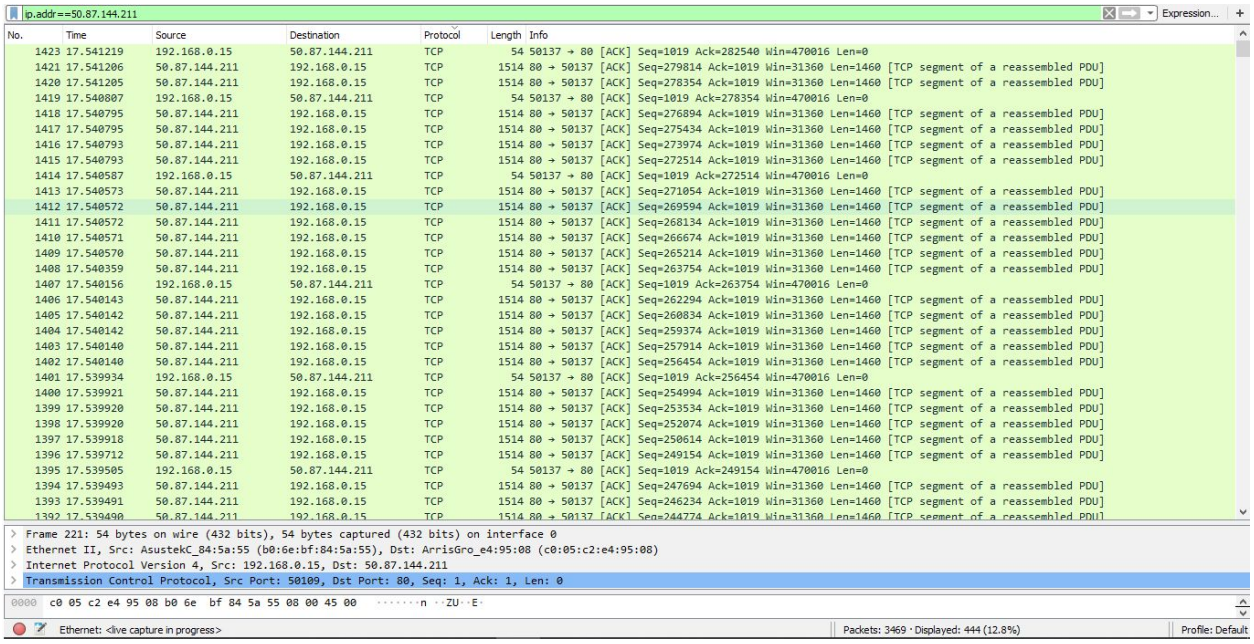
La relación existente entre http y html es que el protocolo de transferencia de hipertexto(http) sirve para transferir páginas de formato de hipertexto(html) desde un servidor web a un navegador web, para intercambiar páginas web entre servidor y buscador.(cliente-servidor, petición-respuesta).

Esquemas de páginas

Página 1



Sniffeo página 1



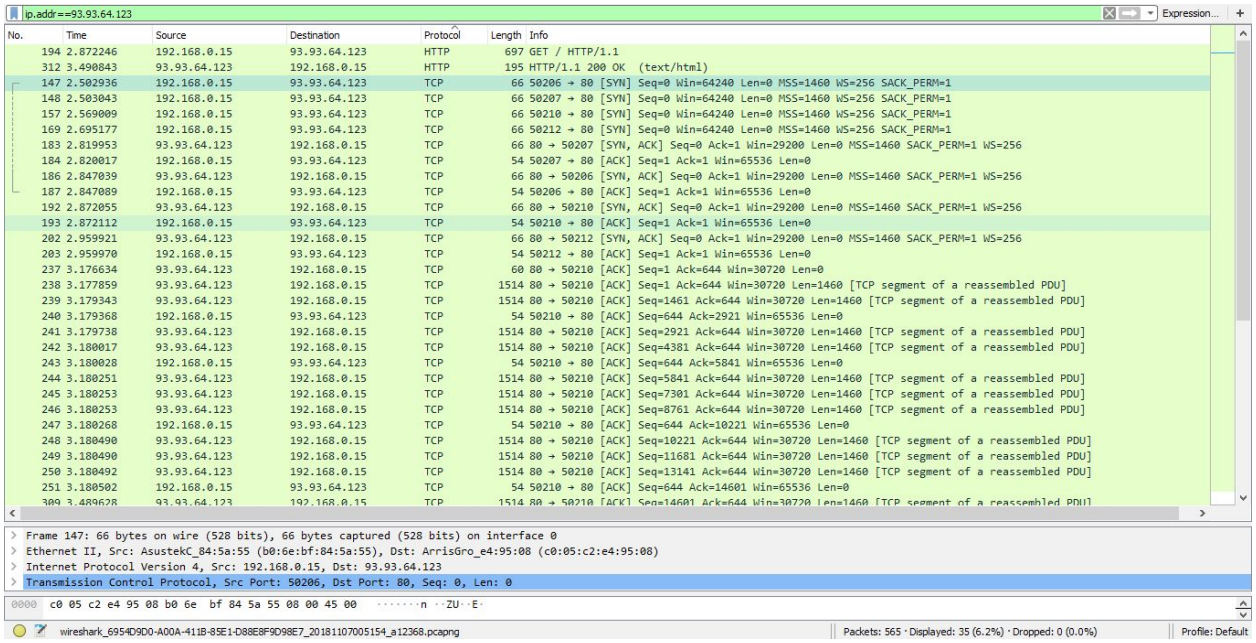
Página 2

NOMBRE DE LA PÁGINA: LECTURALIA.COM
URL DE LA PAGINA: HTTP://WWW.LECTURALIA.COM
OBJETOS QUE TIENE LA PÁGIOA: HIPERVINCULOS HACIA SUBPÁGINAS
(SRC DEL HIPERVÍNCULO SIN PETICION HTTP)

TOTAL PROTOCOLOS HTTP: 2

GET / HTTP/1.1
HTTP/1.1 200 OK
(TEXT/HTML)

Sniffeo página 2



Código final

Presentaremos capturas de pantalla del código final para posteriormente explicar cuáles fueron los cambios realizados al código fuente

```
/*
 *   C++ sockets on Unix and Windows
 *   Copyright (C) 2002
 *
 *   This program is free software; you can redistribute it and/or modify
 *   it under the terms of the GNU General Public License as published by
 *   the Free Software Foundation; either version 2 of the License, or
 *   (at your option) any later version.
 *
 *   This program is distributed in the hope that it will be useful,
 *   but WITHOUT ANY WARRANTY; without even the implied warranty of
 *   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 *   GNU General Public License for more details.
 *
 *   You should have received a copy of the GNU General Public License
 *   along with this program; if not, write to the Free Software
 *   Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
 */

#include "YASL.h"      // For Socket and SocketException
#include "checkArgs.h"
#include <iostream>    // For cerr and cout
#include <cstdlib>     // For atoi()
#include <fstream>
using namespace std;

const uint32_t RCVBUFSIZE = 1024;    // Size of receive buffer

int main(int argc, char *argv[]) {

    checkArgs* argumentos = new checkArgs(argc, argv);
    ofstream archivo;
    archivo.open("fracaso.txt", ios::out);

    std::string servAddress;
```

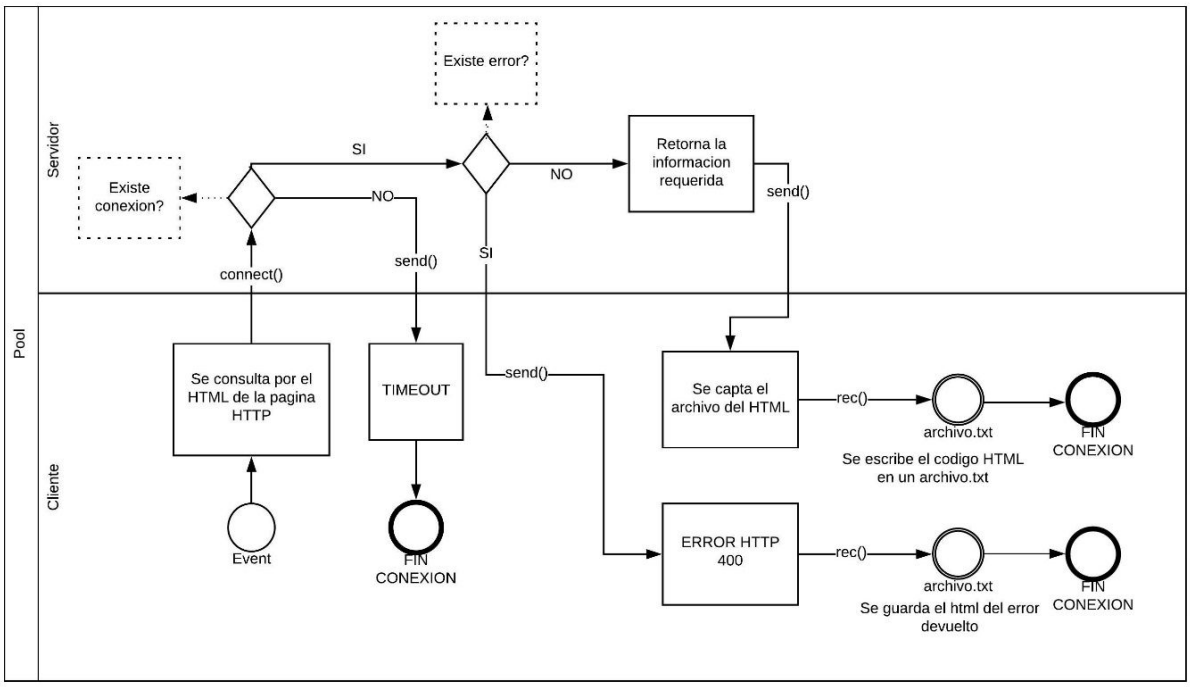
Básicamente lo que se hizo en la línea 24 y 25, fue poder implementar lo necesario para poder realizar el punto de poder guardar la página web en una archivo. Además se modificó la cantidad de bits del RCVBUFSIZE a 1024.

La otra gran modificación del código, fue poder implementar la codificación necesaria de archivos en C++, para así poder guardar todo el “echoBuffer” en archivo declarado en segmentos de códigos anteriores.

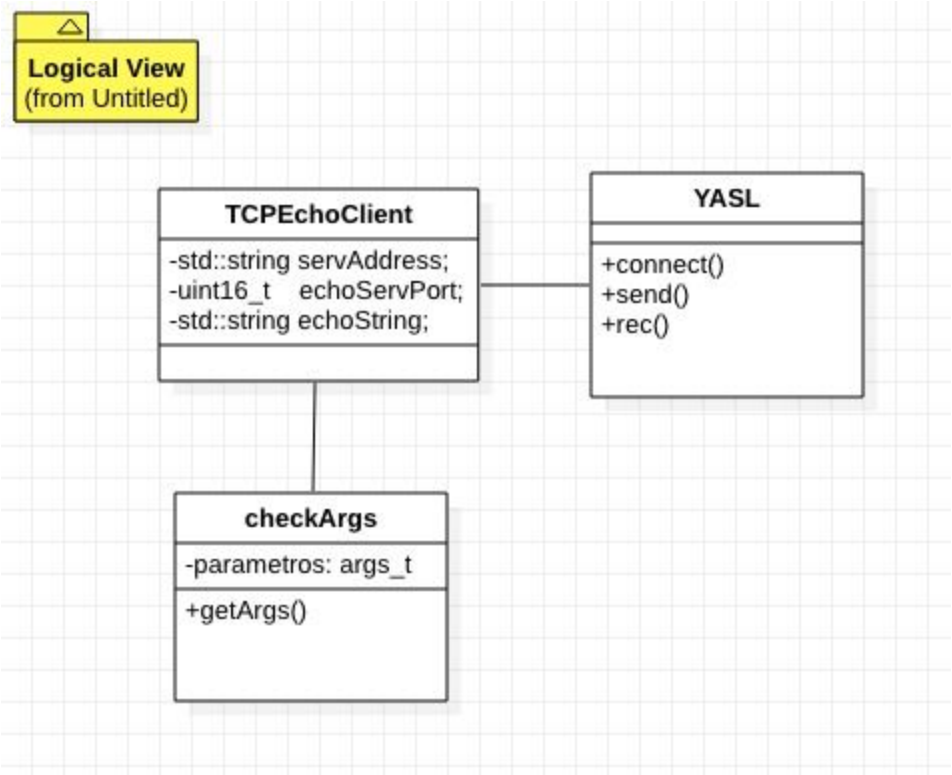
Arquitectura de software

Representamos las arquitecturas del software a través del modelo 4+1 en donde tomaremos 2 “Vistas” de este modelo las cuales serán: la Vista Lógica y la Vista Proceso.

Vista Proceso



Vista lógica



CONCLUSIÓN

El poder trabajar codificando a través de comunicaciones y no linear o secuencialmente, cambia mucho el método de trabajo, el pensamiento que uno tiene al poder codificar o poder programar de manera no online, es muy distinta a cuando el programador depende de la red para ver si su código funciona, la implementación de un cliente HTTP da a entender como es de complejo un navegador web y todo lo que tiene que hacer para que le usuario pueda ver lo que está cargando en un servidor que está a quizás cuantos kilómetros de su pantalla. En definitiva, programar dependiendo de una red, es un mundo distinto.

LINKOGRAFÍA

http://lecturalia.com

http://almosthuman.cl

https://github.com

http://staruml.io