Protocol Document

Trevor Mee

01/29/2025

Project 1

COP4635 - Systems and Networks 2

Table of Contents

Table of Contents..... 2

1 Overview..... 3

2 Communication..... 3

2.1 Client Request 3

2.2 Server Processing..... 3

2.3 Server Response..... 3

2.4 Connection Handling..... 4

3 OSI Breakdown..... 4

3.1 Application Layer..... 4

3.2 Transport Layer..... 4

3.3 Network Layer..... 4

1 Overview

This document describes the protocol and message exchange between a browser client and a HTTP server implemented in `httpServer.cpp`. The server follows the HTTP/1.1 standard and uses TCP sockets to connect with a client via a web browser or using command line tools. The server processes GET requests while listening on port 60001.

2 Communication

Communication between the server and client follows these steps.

2.1 Client Request

The client initiates a TCP connection to the server's IP address on port 60001. The client sends an HTTP GET method to retrieve a resource.

2.2 Server Processing

The server reads the incoming request and parses the HTTP method, file, path, and version. For GET requests, the server attempts to open the requested file from the project directory. If the file exists, the server reads its content and responds with a successful response. If the file is missing or the method is not supported, the server responds with an error message.

2.3 Server Response

After processing, the server responds with either a successful response or an error message. Successful responses look similar to “HTTP/1.1 200 OK” while error messages look similar to “404 Not Found”.

2.4 Connection Handling

Once the server has been created, binded, and begins listening on port 60001, the server remains open while waiting for incoming client connections. After server processing, the connection is closed with the client. Additionally, the server has the capability of a graceful shutdown (Ctrl + C) via the `SIGINT` signal.

3 OSI Breakdown

The following sections describe how each layer from the OSI Reference Model is used in this project.

3.1 Application Layer

The application layer supports the HTTP/1.1 protocol and processes GET requests.

3.2 Transport Layer

The transport layer encapsulates the TCP segments to prepare the segments for transportation. The transport layer utilizes the socket(), listen(), and accept() system calls to create a TCP socket, listen for client connections, and establish a connection to a client.

3.3 Network Layer

The network layer routes the client request to the server's IP address and port number. The server can bind to any IP address through INADDR_ANY and uses the IPv4 protocol defined by AF_INET address family.