Osaze Ogieriakhi

CPTS360 PA4

11784953

Proxy Implementation Summary

The proxy program handles HTTP/1.0 GET requests by acting as an intermediary between clients and servers. It listens on a specified port for incoming client connections, processes their requests, retrieves resources from target servers, and forwards responses back to the clients.

Key Steps

1. **Setup and Listening**: The proxy validates input arguments, sets up a listening socket, and waits for client connections.
2. **Request Handling**:
    a. Reads the client’s request using a buffered interface.
    b. Validates that the request method is GET.
    c. Parses the URI to extract the hostname, port, and resource path.
3. **Server Communication**:
    a. Connects to the target server.
    b. Constructs and forwards an HTTP/1.0 GET request, including necessary headers (e.g., Host, User-Agent).
4. **Response Forwarding**: Reads the server's response line-by-line and relays it to the client.

Parsing and Validation

The parse_request function extracts host and path details from the URI, with support for optional port numbers. Default settings (e.g., port 80) are used if not specified. The proxy logs errors for unsupported methods or connection issues.

Testing

The proxy was tested with Telnet to ensure proper formatting of requests and accurate forwarding of server responses. This implementation demonstrates key networking concepts using sockets while maintaining simplicity and adherence to HTTP/1.0 standards.

Accessing Problem

Some URLs may not be accessible using the proxy in Firefox due to several limitations. The proxy only supports GET requests, so URLs requiring other HTTP methods like POST or PUT will fail. Additionally, it handles only HTTP/1.0, while Firefox often uses newer versions like HTTP/2 or HTTP/3, leading to incompatibility. Another significant limitation is the lack of HTTPS support, as the proxy cannot handle SSL/TLS tunneling required for secure connections. Furthermore, the proxy sends minimal headers, which may not include required fields like Accept-Encoding or Cache-Control that modern servers expect. These limitations can also affect Firefox's ability to load dynamic resources such as scripts or multimedia files. Addressing these issues would require upgrading the proxy to support HTTPS, HTTP/1.1, and additional headers to align with current web standards.