

TCP OVER UDP

Contents

About the project:.....	3
Project structure:	3
User Manual:.....	4

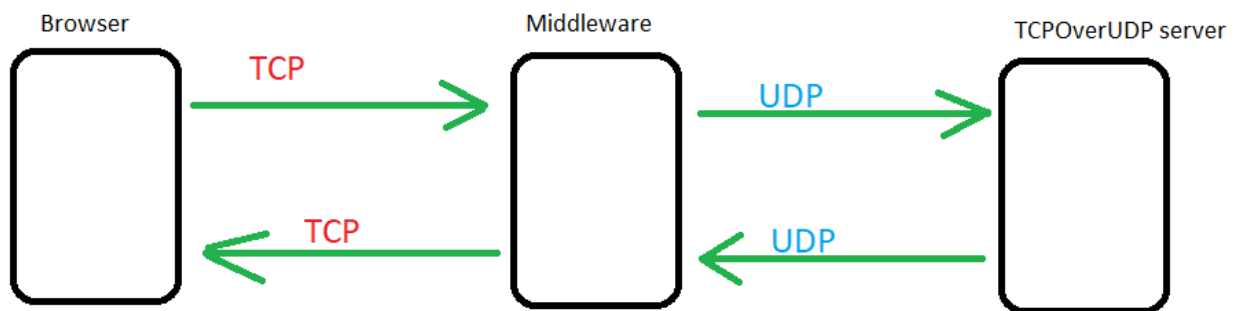
About the project:

Our project is a sophisticated to-do list webpage that allows users to add tasks or to mark tasks as completed. Our project communicates seamlessly with a powerful Python server built with the socket library. The objective of this project is to create an innovative system that simulates the functioning of Transmission Control Protocol (TCP) packets using a User Datagram Protocol (UDP) connection. This is achieved by extending the user space of the application, while still ensuring reliability and supporting the Hypertext Transfer Protocol (HTTP) on top of UDP.

Project structure:

The project is a sophisticated system that aims to create a reliable TCP-like protocol over UDP for the HTTP requests and responses. The architecture is designed as a three-layer structure consisting of the client, middleware, and TCPOverUDPserver. The client, which is a web browser, sends raw HTTP requests to the middleware using the TCP protocol. The middleware acts as a proxy server and forwards these raw requests to the TCPOverUDPserver. Since web browsers cannot communicate with UDP packets, the middleware is needed to accept the requests using TCP and forward them to the TCPOverUDPserver using UDP. The TCPOverUDPserver then handles the requests with implementations for GET, POST, DELETE, and OPTIONS methods, and returns the response to the middleware. The middleware then forwards the response to

the client using TCP. All communications between the middleware and the TCPOverUDPserver are done using UDP with `ack_numbers`, `sequence_numbers`, `flags`, `headers_info`, `response_data` embedded in those UDP packets which ensures reliable data transfer. The system is designed to provide a reliable and efficient alternative to the TCP protocol over the less reliable UDP protocol. The project structure is carefully designed to ensure the efficiency, reliability, and scalability of the system while maintaining compatibility with the HTTP protocol. The structure is clearly represented in the following figure:



User Manual:

You can run the project in three steps:

- 1) Run ***server.py*** that is in folder **code** → **serverSide**.
 - 2) Run ***middleware.py*** that is in folder **code** → **serverSide**.
 - 3) Open the ***index.html*** that is in folder **code** → **clientSide**.
- Note: open ***index.html*** on vsCode using live server extension.