**深 圳 大 学 实 验 报 告**

**课程名称：­ 计算机网络（Computer Networks）**

**实验名称： Application Layer Assignment**

**学院： 电子与信息工程学院**

**专业： 电子信息工程**

**指导教师： 毕宿志**

**报告人：陈应权 班级： 06 学号： 2022280297**

**实验时间： 2023年10月20日-2023年11月4日**

**实验报告提交时间： 2023年11月4日**

**教务部制**

|  |
| --- |
| 1. **Purpose of experiment** 2. Deepen the understanding of TCP and UDP connections. 3. Learn to program in python to implement the process of building a basic website and simulating TCP and UDP connections. 4. Understand and apply FTP protocol to achieve file transfer. 5. **Experimental principle** 6. The principle of the experiment 1 is based on the client-server model of HTTP protocol.    1. Create a connection socket    2. Receive HTTP requests    3. Parse the request    4. Retrieve the file    5. Construct an HTTP response    6. Send the response    7. Error handling 7. The principle of this experiment is based on the client-server model using the UDP protocol.    1. Create the client and server    2. Send ping messages    3. Receive pong messages    4. Calculate RTT    5. Handle lost messages    6. Print the results   (3)  The principle of this experiment is based on simple TCP-based FTP protocol using Python and Socket.  For the server:   1. Listen for connections and file requests from the client. 2. Check whether the requested file exists. 3. Send the file or an error message to the client.   For the client:   1. Connect to the server and send the file name. 2. Receive the file or an error message from the server. 3. **Content**   （1）Web server lab   1. Create a connection socket: The web server uses the socket library to create a TCP socket, which is bound to a specific IP address and port to listen for client connection requests. 2. Receive HTTP requests: When a client (browser) initiates a connection request, the server accepts the connection and receives the HTTP request message through the socket. The HTTP request message consists of a request line, request headers, and a request body. 3. Parse the request: The server parses the received HTTP request message and extracts the filename from the request line to determine the specific file requested by the client. 4. Retrieve the file: The server reads the corresponding file content from the server's file system based on the requested filename. 5. Construct an HTTP response: The server constructs an HTTP response message, including response headers and a response body. The response headers contain status line, date, server information, content length, content type, and other relevant information. 6. Send the response: The server sends the complete HTTP response message back to the client (browser) through the socket. Upon receiving the response, the browser processes it based on the response status code and content. 7. Error handling: If the requested file does not exist on the server, the server returns a 404 Not Found error message, indicating that the requested file was not found.   (2)UDP Pinger   1. Create the client and server: First, we need to create a UDP client and a UDP server. The client will send ping messages, and the server will receive the ping messages and send corresponding pong messages as responses. 2. Send ping messages: The client uses a UDP socket to send ping messages to the server. Each ping message has a unique identifier and sequence number to track the messages and calculate the RTT. 3. Receive pong messages: The server receives the ping messages and sends pong messages as responses. The pong messages will contain the same identifier and sequence number as the ping messages. 4. Calculate RTT: The client starts a timer after sending a ping message and stops the timer upon receiving the corresponding pong message. By calculating the time difference between sending and receiving, the RTT can be determined. 5. Handle lost messages: Since UDP is an unreliable protocol, the sent messages may get lost. The client waits for a certain period of time (e.g., 1 second) to receive pong messages. If no pong message is received within the waiting time, the client assumes that the message was lost and prints a corresponding message. 6. Print the results: The client prints the sequence number, send time, receive time, and calculated RTT for each ping message. For lost messages, the client prints an appropriate message.   (3) FTP protocol  For the server:   1. Listen for connections and file requests from the client. 2. Check whether the requested file exists. 3. Send the file or an error message to the client.   For the client:   1. Connect to the server and send the file name. 2. Receive the file or an error message from the server. |
| 1. **Conclusion and discussion**   In these three experiments, we explored the principles and implementation of different network protocols using Python.   1. Python Web Server: We learned how to create a simple web server using Python and handle client requests. The server listened for incoming connections, received HTTP requests, parsed the requests, retrieved the requested files, and sent back HTTP responses. This experiment demonstrated the basic functionality of a web server and how it interacts with clients. 2. Ping Program: We delved into the principles of a ping program based on the UDP protocol. We implemented the program using Python and the Socket library. The program allowed us to send ping messages to a target host, receive pong messages in response, calculate the Round-Trip Time (RTT), and handle lost messages. This experiment provided insights into network troubleshooting and latency measurement. 3. TCP-based FTP Protocol: We explored the principles of a simple FTP protocol based on the TCP protocol. Using Python and Socket, we implemented a basic FTP server and client. The server listened for connections, checked file existence, and sent files or error messages to clients. The client connected to the server, sent file requests, and received files or error messages. This experiment demonstrated file transfer between a server and a client using TCP. |
| 指导教师批阅意见：  成绩评定：  指导教师签字：  年 月 日  备注： |

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。