**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： 廖祖颐 班级： 06 学号： 2022110131**

**实验时间： 2023.11.1**

**实验报告提交时间： 2023.11.5**

**教务部制**

|  |
| --- |
| 1. **Purpose of experiment**   **Assignment 1: Web Server Lab：**   * Gain hands-on experience in creating a rudimentary HTTP server using Python. * Grasp the foundational concepts of how servers process * Grasp how servers process respond to web requests.   **Assignment 2: UDP Pinger：**  The primary objective of this experiment is to design and implement a simple UDP-based client-server communication system that can send and receive ping messages.  **Assignment 3: FTP protocol**  The objective of this experiment is to develop a basic understanding of the File Transfer Protocol (FTP) through the creation of a simple TCP-based file transfer system. This system consists of a server to serve files upon request and a client to request and receive files, demonstrating the core concepts of FTP.   1. **Experimental principle**   **Assignment 1: Web Server Lab：**  **HTTP Protocol:** An application-layer protocol used for data transfer, based on TCP connections.  **Request/Response Model:** The client sends an HTTP request to the server, and the server processes the request, returning an appropriate response.  **TCP Socket Programming:** Using the socket API to create TCP connections, listen on ports, accept connections, and send/receive data.  **Assignment 2: UDP Pinger：**  This section explains the foundational theory of the experiment which revolves around UDP's nature as a connectionless and unreliable protocol. The experiment will utilize UDP to understand its behavior, particularly focusing on the calculation of Round Trip Time (RTT) and simulating packet loss to emulate real-world scenarios.  **Assignment 3: FTP protocol**  The foundation of this experiment is the client-server model, where the server acts as the resource provider and the client as the service requester. The **socket** module in Python is utilized to establish and handle TCP/IP networking, allowing the server to manage file requests and the client to initiate file transfers.   1. **Content**   **Assignment 1: Web Server Lab：**  **Initiation**: Using Python's socket module, a server was set up to listen on port 1234. This port was chosen to avoid conflict with commonly used ports.  **Awaiting and Reading Requests:** The server is programmed to wait for incoming connections. Upon establishing a connection, it reads the client's request, specifically identifying the desired file.  **Responding to Requests:**  If the requested file exists, it's read from the server's file system and sent back to the client.  If the file doesn’t exist, the server responds with a standard HTTP 404 error, notifying the client of the unavailable content.  **Connection Management:** After responding, the server terminates that particular connection. The current server configuration allows it to process just one request before halting its operations.  **Assignment 2: UDP Pinger：**   * UDP Client Implementation: Details the client's role in sending ping messages, handling server responses, calculating RTTs, and managing timeouts. * UDP Server Implementation: Describes the server's behavior in receiving ping messages and randomly deciding whether to respond, thus simulating packet loss. * Measurement and Analysis: Discusses how RTT data will be collected and analyzed, including the calculation of the mean RTT and packet loss ratio.   **Assignment 3: FTP protocol：**   * **Server Script (server.py):**   + Initializes a socket and listens on port 12345.   + Accepts client connections and processes file requests.   + Checks file existence and transfers it with "START" and "END" signals if found.   + Sends "ERROR" signal if the file is absent. * **Client Script (client.py):**   + Establishes a connection with the server's socket.   + Sends file request and handles server's response.   + Receives file data between "START" and "END" signals.   + Manages "ERROR" signal by notifying the user of unavailability. |
| 1. **Conclusion and discussion**   **Assignment 1: Web Server Lab：**  This experiment offered a basic, yet insightful, look into the operations of an HTTP server. It provided clarity on how servers communicate with clients, handle file requests, and manage connections.  However, the server in its current form has limitations:   * It can process only one request at a time. * The error responses can be enriched for a more informative user experience.   Future enhancements could include introducing multi-threading for concurrent request handling and expanding the range of HTTP responses for a broader set of scenarios.      **Assignment 2: UDP Pinger：**  The conclusion summarizes the findings that UDP is fast but comes with reliability issues. The experiment highlights UDP's characteristics and suggests that while UDP is suitable for many applications, it requires additional mechanisms for scenarios needing reliable delivery.      **Assignment 3: FTP protocol：**  The experiment has successfully implemented a fundamental FTP system, showcasing the capability of transferring files using TCP/IP protocols with basic command signals. |
| 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。