Design Document

Developed By:

- Hamza Haroon
- Sidra Khan
- Hassan Nawaz

AI Voice Assistant Protocol (VIP)

Computer Networks

Design and Implementation of a Real-World Application Layer Protocol

---

1. Project Overview

The **AI Voice Assistant Protocol (VIP)** project is a real-world implementation of a custom **Application Layer Protocol** that enables structured communication between a web-based client and a backend server for voice-based interaction.

The system emphasizes **networking principles**, particularly:

- Custom protocol design
- TCP/IP communication
- Structured message formatting
- Session management
- Error handling mechanisms

The application is implemented using **Python and Streamlit**, where Streamlit serves as the client-side protocol interface.

---

2. Problem Statement

Traditional voice assistants rely heavily on complex AI models, but many academic networking courses require an understanding of **how application-layer protocols function**.

This project addresses the problem of:

Designing and implementing a real-world application-layer communication protocol that manages voice data, commands, and responses using structured message exchange over TCP.

---

3. System Objectives

The system is designed to fulfill the following objectives:

- To design and implement a **custom protocol (VIP)**.
- To ensure **reliable communication** using TCP.
- To manage **client-server sessions** efficiently.

- To handle errors using structured protocol messages.
- To demonstrate real-world protocol behavior in a working application.

---

4. Scope of the Project

Included:

- Custom protocol design
- Client-side UI implementation
- Server-side protocol simulation
- Message parsing and formatting
- Session management

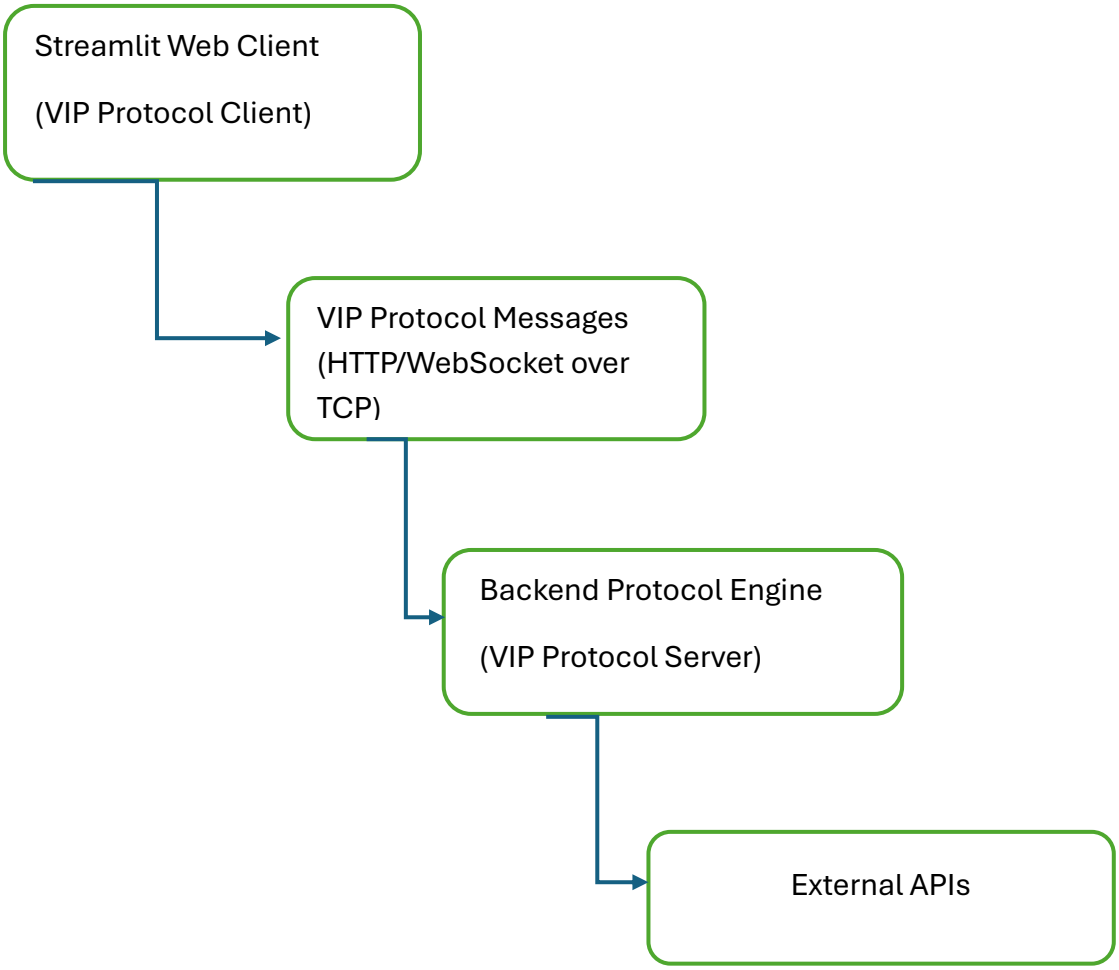
Excluded:

- Complex machine learning models
- External cloud AI services
- Advanced speech model training

---

5. System Architecture

High-Level Architecture



Layered Architecture (TCP/IP Model)

Layer	Role in This Project
Application	VIP Protocol implementation
Transport	TCP
Network	IP
Data Link	Ethernet / Wi-Fi
Physical	Physical hardware medium

---

6. Protocol Design (VIP)

Protocol Name

“” Voice Interaction Protocol (VIP) “”

6.2 Protocol Type

- Application Layer Protocol
- Request–Response based
- Text-based structured messages

Message Structure

Each VIP message contains four logical parts:

1. Start Line
2. Header Fields
3. Blank Line
4. Optional Message Body

Generic Format:

```
<Message-Type> <Session-ID> <Body-Length>
Header-Name: Header-Value
Header-Name: Header-Value
```

Header Field Definitions

Header Name	Description
Message-Type	Type of message
Session-ID	Unique session identifier
Body-Length	Size of message body
Audio-Format	WAV

Encoding	BASE64
Response-Type	TEXT / SPEECH
Content-Length	Body size
Error-Code	Numeric error identifier
Error-Message	Description of error

Message Types

Message Type	**Description**
INIT_SESSION	Establishes new session
VOICE_DATA	Transfers audio data
COMMAND_TEXT	Internal command parsing
BOT_RESPONSE	Server response to client
ERROR	Error reporting
END_SESSION	Terminates active session

---

7. Component Design

Client Component (Streamlit UI)

Responsibilities:

- Acts as protocol client
- Initiates sessions
- Sends voice data
- Displays server responses
- Manages user interaction

Implemented using:

- streamlit
- Pillow
- base64

Server Component (Protocol Engine)

Responsibilities:

- Parses VIP protocol messages
- Manages sessions
- Processes requests
- Generates responses
- Handles errors

Implemented using:

- Python socket programming
- Threaded request handling

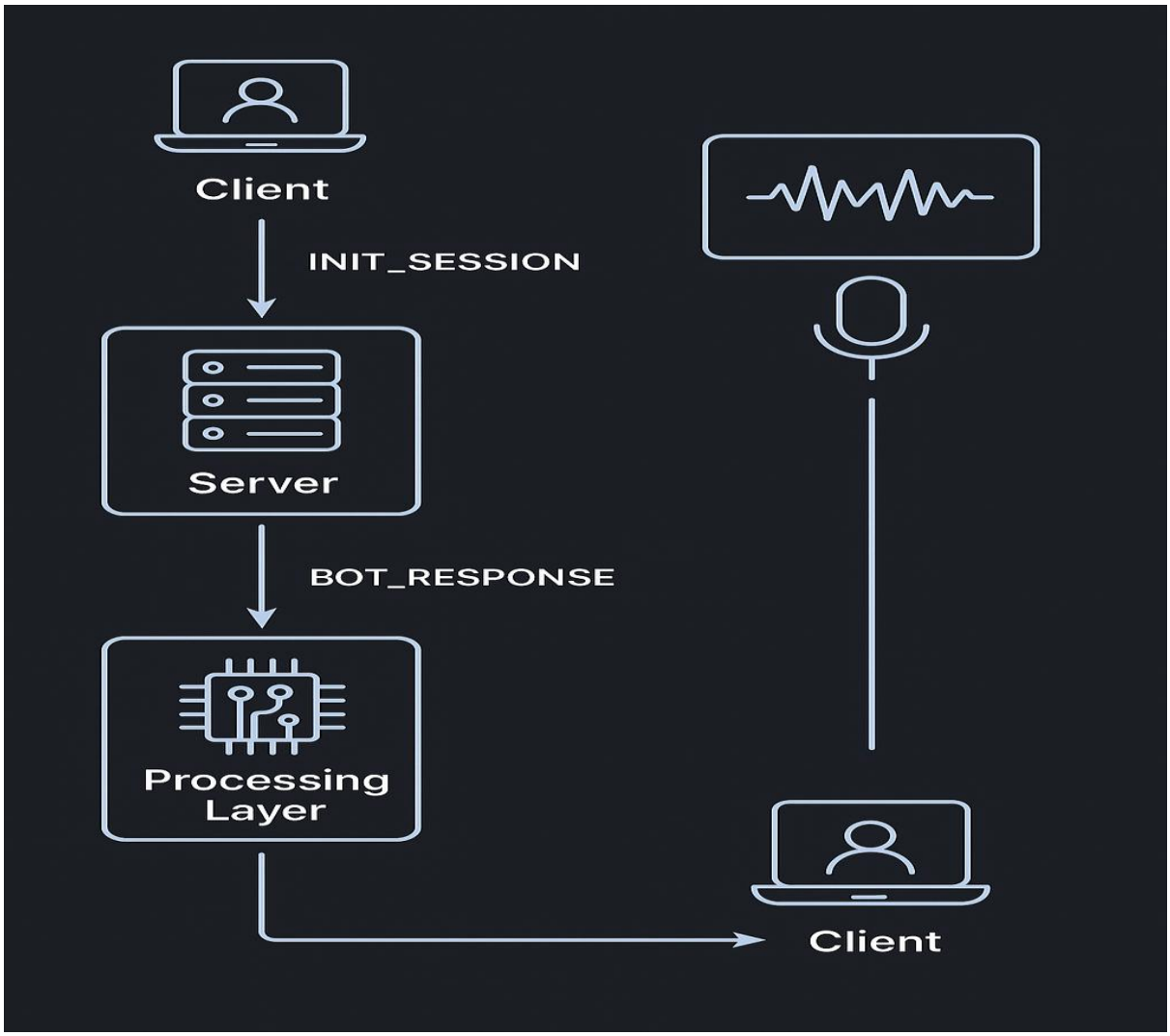
Session Manager

Responsibilities:

- Generates unique Session IDs
- Maps sessions to connected clients
- Cleans inactive sessions

---

8. Data Flow Design



---

9. Error Handling Design

Error Scenario	Protocol Response
Invalid Audio	ERROR 400
Session Not Found	ERROR 404
Internal Server Error	ERROR 500

---

10. Non-Functional Design Goals

Feature	Design Approach
Performance	Lightweight text-based protocol
Reliability	TCP-based delivery
Security	HTTPS / TLS-ready
Scalability	Multi-client threaded server
Availability	Persistent listening server

---

11. Technology Stack

Component	Technology
Programming Language	Python 3.10
UI Framework	Streamlit
Networking	TCP Sockets
Encoding	Base64
Data Format	Text-based protocol structure

---

12. Deployment Design

Local Deployment:

```
pip install streamlit speechrecognition pyttsx3 pillow

streamlit run app.py
```

---

13. Testing Strategy

Test Type	Description
Unit Testing	Header parser validation
Integration Testing	Client-Server flow

Stress Testing	Multiple session handling
Fault Testing	Invalid packets simulation

---

14. Limitations

- Streamlit abstracts low-level socket control.
- Voice processing simulated for networking focus.
- Prototype version not industrial-grade.

---

15. Future Improvements

- Raw TCP streaming
- UDP-based real-time audio
- Packet-level encryption
- Wireshark integration

---

16. Conclusion

The **AI Voice Assistant Protocol (VIP)** successfully demonstrates the principles of **real-world Application Layer Protocol Design**. It implements structured communication, session handling, error management, and reliable TCP-based data transfer, fulfilling the complete academic requirements of the Computer Networks assignment.