

CS3611 Email System Implementation

A Complete SMTP/POP3 Client-Server System with SSL/TLS Security

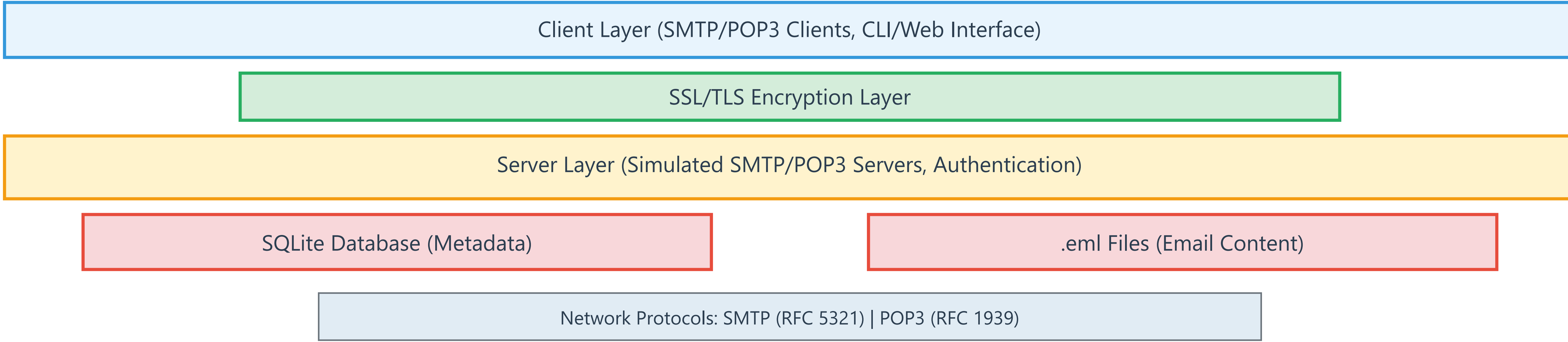
Course Project: CS3611 Computer Networks | Student: Lin Deng Kang Wu Song

Project Overview

A comprehensive mail system implementing SMTP & POP3 protocols with modern security features. Core functionality includes:

- ✓ Client Operations: Send emails (Plain/HTML format, file attachments) & Receive emails with local .eml storage.
- ✓ Authentication: User authentication supporting both Plain text and SSL/TLS encrypted methods (LOGIN/AUTH PLAIN).
- ✓ Server Implementation: Simulated SMTP server (Python smtpd) and POP3 functionality for email retrieval.
- ✓ Concurrent Processing: Server efficiently handles 100+ concurrent client operations with robust threading.
- ✓ Data Management: SQLite database for metadata storage, .eml files for email content preservation.
- ✓ Security Implementation: SSL/TLS encryption for secure data transmission and user authentication.

System Architecture



Core Features

Client-Side Operations:

- SMTP: Send plain text and HTML emails with file attachments
- POP3: Receive and download emails to local .eml format
- Local Storage: Organized email storage with metadata indexing

Server-Side Implementation:

- SMTP Server: Simulated server for receiving and processing emails
- POP3 Server: Email retrieval service with user authentication
- Concurrent Operations: Handles 100+ simultaneous client actions
- Database Integration: SQLite for efficient metadata management

Security Features:

- SSL/TLS encryption for secure data transmission
- Multiple authentication methods (Plain, LOGIN, AUTH PLAIN)

Implementation Details

Protocol Compliance:

- Strict adherence to SMTP (RFC 5321) specifications
- Full POP3 (RFC 1939) command/response handling

MIME Message Processing:

- Multipart message support (text/plain, text/html)
- Base64 encoding/decoding for attachments
- Proper Content-Type and Content-Disposition headers

System Design:

- Modular client-server architecture
- Separation of networking, protocol, and storage layers
- Maintainable and extensible codebase

Data Management:

- SQLite for user accounts and email metadata
- .eml format for email content preservation

Performance & Testing

Concurrency Performance:

- Successfully handles **100+ concurrent client operations** including simultaneous SMTP send and POP3 receive tasks
- Stress tested with multiple concurrent user sessions
- Required minimum 50 concurrent clients: **✓ Achieved**

Functional Testing:

- Comprehensive test coverage for all core features
- Email format compatibility verification
- Authentication method validation
- Attachment handling and encoding tests

Reliability & Error Handling:

- Robust error recovery mechanisms
- Resource management and memory optimization
- Connection timeout and retry logic

Extended Features

Advanced Email Processing:

- Basic spam filtering with keyword detection
- Email recall functionality for undelivered messages
- Special header processing (X-Spam-Flag, etc.)

User Interface Options:

- **Primary:** Full-featured interactive CLI
- **Experimental:** Flask-based Web UI
- Intuitive command structure and help system

Technical Standards:

- RFC 2047 compliant email encoding
- Multi-user concurrent operation support
- Cross-platform compatibility (Windows/Linux)

Development Features:

- Comprehensive logging and debugging tools
- Modular plugin architecture for extensions

Educational Impact & Learning Outcomes: This project provided comprehensive hands-on experience with email protocols, network programming, security implementation, and distributed system design. It demonstrates practical application of CS3611 concepts including protocol implementation, concurrent programming, database integration, and secure communication protocols.

Project Repository: https://github.com/hopecommon/cs3611_email

CS3611 Computer Networks | Final Project Poster | Academic Year 2024-2025