

UDP File Downloading System

- Design Report

Members:

Bryan Ang Wei Ze (2301397)

Tham Kang Ting (2301255)

Low Yue Jun (2301302)

Contents

1. Design Introduction	2
1.1 System Architecture.....	2
1.2 Reliable Communication Protocol	3
1.2.1 Protocol Selection and Implementation.....	3
1.2.2 Message Formats	4
1.3 Transaction Sequence	5
1.4 Multi-threading Model	6
1.4.1 Server-side Threading	6
1.4.2 Client-side Threading.....	7
1.5 Optimizations	7
2. Verification Methodology	8
2.1 Testing Scenarios.....	8
2.2 Test Results	9
2.3 Performance Analysis	10
3. Individual Contribution	12

1. Design Introduction

This document outlines the implementation of a reliable file transfer system over UDP. The system allows multiple clients to connect to a server, view available files, and download them with guaranteed delivery, even in the presence of network disruptions.

1.1 System Architecture

The system is built around a client-server architecture:

1. Server Component:

- TCP listener for accepting client connections and processing commands
- UDP socket for sending file data
- Multi-threaded TaskQueue system for handling concurrent clients
- Session management system for tracking active downloads
- Monitoring system for detecting and recovering from stalled transfers

2. Client Component:

- TCP connection for sending commands and receiving responses
- UDP socket for receiving file data
- Multi-threaded design to separate UI, TCP, and UDP processing
- File reassembly and tracking system
- Progress monitoring and reporting

Communication uses:

- **TCP:** For control messages (commands and responses)
- **UDP:** For file data transfer with a reliability layer

1.2 Reliable Communication Protocol

1.2.1 Protocol Selection and Implementation

We implemented a **Go-Back-N** protocol for reliable data transfer over UDP, which provides:

- Efficient throughput with sliding window mechanism
- Good balance between complexity and performance
- Automatic recovery from packet loss

Key features of our Go-Back-N implementation:

1. Sliding Window:

- Configurable window size (default: 5 packets)
- Sequential packet transmission within the window
- Window advancement based on received acknowledgments
- Efficient window boundary management

2. Sequence Numbering:

- Each packet is assigned a sequence number based on file offset
- Sequence numbers are used for ordering and acknowledgment
- Enables detection of missing or out-of-order packets

3. Acknowledgment System:

- Client acknowledges each received packet by sequence number
- Cumulative acknowledgment approach
- Fast acknowledgment processing to maintain throughput

4. Timeout and Retransmission:

- Configurable timeout (default: 500ms)
- Smart retransmission of unacknowledged packets
- Retry limit to prevent infinite retransmission loops
- Exponential backoff for congestion control

5. Session Management:

- Unique session IDs for each download
- Comprehensive session state tracking
- Session monitoring for detecting stalled downloads
- Graceful session cleanup for completed or abandoned transfers

1.2.2 Message Formats

TCP Messages:

Message Type	Command ID	Description	Additional Fields
REQ_LISTFILES	0x4	Request file list	None
RSP_LISTFILES	0x5	Response with file list	Number of files (2 bytes) List Length (4 bytes) File entries (variable)
REQ_DOWNLOAD	0x2	Request download	Client IP (4 bytes) Client Port (2 bytes) Filename Length (4 bytes) Filename (variable)
RSP_DOWNLOAD	0x3	Download information	Server IP (4 bytes) Server Port (2 bytes) Session ID (4 bytes) File Length (4 bytes)
REQ_QUIT	0x1	Terminate connection	None
DOWNLOAD_ERROR	0x30	Download error	None

UDP Messages:

1. Data Packet:

- Flags (1 byte): LSB 0 indicates data packet
- Session ID (4 bytes): Identifies the download session
- File Offset (4 bytes): Position in the file
- Data Length (4 bytes): Length of the data in this packet
- File Data (variable): Actual file data

2. ACK Packet:

- a. Flags (1 byte): LSB 1 indicates ACK packet
- b. Session ID (4 bytes): Identifies the download session
- c. ACK Number (4 bytes): Sequence number being acknowledged

1.3 Transaction Sequence

The typical file download transaction follows this sequence:

1. Connection Establishment:

- Client connects to server via TCP
- Server accepts connection and creates a client handler thread

2. File Discovery:

- Client sends REQ_LISTFILES command
- Server processes the request, scans the download directory
- Server sends RSP_LISTFILES with available files
- Client displays the file list to the user

3. Download Initiation:

- Client sends REQ_DOWNLOAD with IP, port, and filename
- Server verifies file existence and prepares for transfer
- Server assigns a session ID and creates a download session
- Server sends RSP_DOWNLOAD with server IP, port, session ID, and file size
- Server spawns a file transfer thread for this session

4. File Transfer (Go-Back-N Protocol):

- Server sends packets within the current window
- Client receives packets and sends ACKs for each received packet
- Server advances window based on received ACKs
- If packets are lost, server retransmits unacknowledged packets after timeout
- Transfer continues until all file data is sent and acknowledged
- Client assembles the file from received packets
- Client reports progress during download

5. Transfer Completion:

- Server confirms all packets have been acknowledged
- Server closes the download session
- Client finalizes the downloaded file
- Client displays download statistics (time, speed)

6. Error Handling:

- If file not found, server sends DOWNLOAD_ERROR
- If network disruption occurs, retransmission handles recovery
- If maximum retries are exceeded, server logs error and closes session
- Client detects and reports stalled downloads

1.4 Multi-threading Model

1.4.1 Server-side Threading

1. Main Thread:

- Handles initial setup and TCP listener
- Accepts incoming client connections
- Dispatches client connections to the TaskQueue

2. TaskQueue Worker Threads:

- Process TCP commands from clients
- Handle file listing and download requests
- Create download sessions

3. File Transfer Threads:

- One per active download session
- Read file data and create packets
- Handle window management and retransmissions
- Track acknowledgments and manage session state

4. UDP Receiver Thread:

- Receives and processes ACK packets
- Updates session state based on acknowledgments

5. Session Monitor Thread:

- Periodically checks for stalled or inactive sessions
- Cleans up timed-out sessions
- Logs session statistics

1.4.2 Client-side Threading

1. Main Thread:

- Handles user interface and command input
- Processes user commands

2. TCP Receiver Thread:

- Receives and processes TCP messages from server
- Handles file listings and download responses
- Creates download sessions

3. UDP Receiver Thread:

- Receives and processes UDP data packets
- Sends acknowledgments
- Assembles file data
- Tracks download progress

4. Session Monitor Thread:

- Monitors active downloads for progress
- Detects stalled downloads
- Reports download statistics

1.5 Optimizations

1. Socket Buffer Optimization:

- Increased socket buffer sizes (1MB) for better performance
- Reduces the chance of packet loss due to buffer overflow

2. File I/O Efficiency:

- Pre-allocation of file space on the client side
- Efficient seeking and writing at specific file offsets
- Buffered I/O for better performance

3. Protocol Tuning:

- Configurable window size for network condition adaptation
- Adjustable timeout values for different network environments
- Retry limiting to prevent resource exhaustion

4. Memory Management:

- Reuse of packet buffers to reduce memory allocation overhead
- Proper buffer sizing based on maximum packet size
- Efficient vector and map usage for data storage

2. Verification Methodology

2.1 Testing Scenarios

The implementation was verified through comprehensive testing:

1. Functionality Testing:

- a. Basic file listing and downloading without network disruption
- b. Multiple simultaneous client connections and downloads
- c. Files of various sizes (small to 200MB)
- d. Different window sizes and protocol configurations

2. Reliability Testing:

- a. Network disruption using Ethernet Break Circuit
- b. Multiple brief disconnections during a transfer
- c. Extended disconnection during transfer
- d. High packet loss simulation
- e. Client-side disconnection and reconnection

3. Performance Testing:

- a. Large file transfers (100MB+)
- b. Multiple concurrent downloads
- c. Various window sizes to find optimal settings
- d. Different network conditions and latencies

4. Error Handling Testing:

- a. File not found scenarios
- b. Invalid commands
- c. Malformed packets
- d. Server shutdown during transfer
- e. Client crash and recovery

2.2 Test Results

Our implementation successfully handled all test scenarios:

1. Basic Functionality:

- File listing worked correctly with directories of various sizes
- Downloaded files matched source files (verified by checksum)
- Multiple clients could connect and download simultaneously

2. Reliability:

- Files transferred correctly despite network disruptions
- Recovery from packet loss was automatic
- Protocol correctly handled out-of-order packets

3. Performance:

- Achieved good throughput with optimized window sizes
- Multiple concurrent downloads showed fair resource sharing
- Large files transferred successfully with consistent speed

4. Error Handling:

- Appropriate error messages displayed for all error conditions
- No resource leaks observed during stress testing
- Graceful cleanup of aborted transfers

2.3 Performance Analysis

```
[20:28:24] Received ACK: Session 3, Seq 6399
[20:28:24] Session 3 progress: 99%
[20:28:24] Sent packet: Session 3, Seq 6400, Offset 104774400, Size 16371
[20:28:24] Sent packet: Session 3, Seq 6401, Offset 104790771, Size 16371
[20:28:24] Sent packet: Session 3, Seq 6402, Offset 104807142, Size 16371
[20:28:24] Sent packet: Session 3, Seq 6403, Offset 104823513, Size 16371
[20:28:24] Sent packet: Session 3, Seq 6404, Offset 104839884, Size 16371
[20:28:24] Sent packet: Session 3, Seq 6405, Offset 104856255, Size 1345
[20:28:24] Received ACK: Session 3, Seq 6400
[20:28:24] Received ACK: Session 3, Seq 6401
[20:28:24] Received ACK: Session 3, Seq 6402
[20:28:24] Received ACK: Session 3, Seq 6403
[20:28:24] Received ACK: Session 3, Seq 6404
[20:28:24] Received ACK: Session 3, Seq 6405
[20:28:24] Timeout for session 3, retry 1/5
[20:28:24] Timeout for session 3, retry 2/5
[20:28:25] Timeout for session 3, retry 3/5
[20:28:25] Timeout for session 3, retry 4/5
[20:28:25] Timeout for session 3, retry 5/5
[20:28:25] Maximum retries limit reached for session 3, waiting 5 more seconds for any final ACKs
[20:28:27] No new ACKs during final wait, ending transfer
[20:28:27] Last chunk (Seq 6405) is ACKed - transfer is complete
[20:28:27] File transfer completed successfully: Session 1
[20:28:28] No new ACKs during final wait, ending transfer
[20:28:28] Last chunk (Seq 6405) is ACKed - transfer is complete
[20:28:28] File transfer completed successfully: Session 2
[20:28:30] No new ACKs during final wait, ending transfer
[20:28:30] Last chunk (Seq 6405) is ACKed - transfer is complete
[20:28:30] File transfer completed successfully: Session 3

[20:28:06] Starting download from 192.168.50.193:8002
[20:28:06] Session ID: 1
[20:28:06] File size: 100.00 MB
[20:28:06] Output file: D:\Dest1\download_1
[20:28:06] Session 1: 0% complete, 532.91 KB/s
[20:28:07] Session 1: 5% complete, 7.56 MB/s
[20:28:08] Session 1: 10% complete, 7.81 MB/s
[20:28:08] Session 1: 15% complete, 7.84 MB/s
[20:28:09] Session 1: 20% complete, 7.86 MB/s
[20:28:09] Session 1: 25% complete, 7.75 MB/s
[20:28:10] Session 1: 30% complete, 7.68 MB/s
[20:28:11] Session 1: 35% complete, 7.50 MB/s
[20:28:12] Session 1: 40% complete, 7.32 MB/s
[20:28:12] Session 1: 45% complete, 7.26 MB/s
[20:28:13] Session 1: 50% complete, 7.26 MB/s
[20:28:14] Session 1: 55% complete, 7.20 MB/s
[20:28:15] Session 1: 60% complete, 7.18 MB/s
[20:28:15] Session 1: 65% complete, 7.15 MB/s
[20:28:16] Session 1: 70% complete, 7.14 MB/s
[20:28:17] Session 1: 75% complete, 7.16 MB/s
[20:28:17] Session 1: 80% complete, 7.19 MB/s
[20:28:18] Session 1: 85% complete, 7.20 MB/s
[20:28:19] Session 1: 90% complete, 7.22 MB/s
[20:28:19] Session 1: 95% complete, 7.20 MB/s
[20:28:21] Session 1: 100% complete, 6.90 MB/s
[20:28:21] Download completed: download_1
[20:28:21] Size: 100.00 MB
[20:28:21] Time: 14s
[20:28:21] Speed: 6.90 MB/s
```

```
[20:28:09] Starting download from 192.168.50.193:8002
[20:28:09] Session ID: 3
[20:28:09] File size: 100.00 MB
[20:28:09] Output file: D:\Dest3\download_3
[20:28:10] Session 3: 0% complete, 207.63 KB/s
[20:28:11] Session 3: 5% complete, 6.63 MB/s
[20:28:12] Session 3: 10% complete, 6.51 MB/s
[20:28:12] Session 3: 20% complete, 6.64 MB/s
[20:28:13] Session 3: 25% complete, 6.71 MB/s
[20:28:14] Session 3: 30% complete, 6.71 MB/s
[20:28:14] Session 3: 35% complete, 6.74 MB/s
[20:28:15] Session 3: 40% complete, 6.77 MB/s
[20:28:16] Session 3: 45% complete, 6.85 MB/s
[20:28:17] Session 3: 50% complete, 6.85 MB/s
[20:28:17] Session 3: 55% complete, 6.90 MB/s
[20:28:18] Session 3: 60% complete, 6.94 MB/s
[20:28:19] Session 3: 65% complete, 6.99 MB/s
[20:28:19] Session 3: 70% complete, 7.04 MB/s
[20:28:20] Session 3: 75% complete, 6.86 MB/s
[20:28:21] Session 3: 80% complete, 6.70 MB/s
[20:28:22] Session 3: 85% complete, 6.65 MB/s
[20:28:23] Session 3: 90% complete, 6.62 MB/s
[20:28:24] Session 3: 95% complete, 6.66 MB/s
[20:28:24] Session 3: 100% complete, 6.71 MB/s
[20:28:24] Download completed: download_3
[20:28:24] Size: 100.00 MB
[20:28:24] Time: 14s
[20:28:24] Speed: 6.71 MB/s
```

```
[20:28:07] Starting download from 192.168.50.193:8002
[20:28:07] Session ID: 2
[20:28:07] File size: 100.00 MB
[20:28:07] Output file: D:\Dest2\download_2
[20:28:07] Session 2: 0% complete, 347.55 KB/s
[20:28:08] Session 2: 5% complete, 7.36 MB/s
[20:28:08] Session 2: 10% complete, 7.72 MB/s
[20:28:09] Session 2: 15% complete, 7.84 MB/s
[20:28:10] Session 2: 20% complete, 7.63 MB/s
[20:28:10] Session 2: 25% complete, 7.64 MB/s
[20:28:11] Session 2: 30% complete, 7.11 MB/s
[20:28:12] Session 2: 35% complete, 7.10 MB/s
[20:28:13] Session 2: 40% complete, 7.09 MB/s
[20:28:14] Session 2: 45% complete, 7.05 MB/s
[20:28:14] Session 2: 50% complete, 7.03 MB/s
[20:28:15] Session 2: 55% complete, 7.04 MB/s
[20:28:16] Session 2: 60% complete, 6.99 MB/s
[20:28:16] Session 2: 65% complete, 6.98 MB/s
[20:28:17] Session 2: 70% complete, 7.01 MB/s
[20:28:18] Session 2: 75% complete, 7.05 MB/s
[20:28:18] Session 2: 80% complete, 7.08 MB/s
[20:28:19] Session 2: 85% complete, 7.11 MB/s
[20:28:20] Session 2: 90% complete, 7.04 MB/s
[20:28:21] Session 2: 95% complete, 6.82 MB/s
[20:28:22] Session 2: 100% complete, 6.83 MB/s
[20:28:22] Download completed: download_2
[20:28:22] Size: 100.00 MB
[20:28:22] Time: 14s
[20:28:22] Speed: 6.83 MB/s
```

Performance observations:

- Server can handle multiple download requests simultaneously
- Transfer speed is similar for all clients

3. Individual Contribution

Name	SIT ID	DigiPen ID	Contribution
Bryan Ang Wei Ze	2301397	bryanweize.ang	Responsible for the UDP implementation, Go-Back-N protocol design, file transfer threads, and session management
Tham Kang Ting	2301255	kangting.t	Focused on system architecture, TCP protocol, TaskQueue integration, client-side implementation, and documentation
Low Yue Jun	2301302	yuejun.low	Handled server-side implementation, file management, reliability testing, and error handling