

# Peer-to-Peer File Sharing System: A Robust and Scalable Implementation

Yugal Khanal  
2302704

February 13, 2025

## Abstract

The proliferation of distributed systems has led to increased interest in peer-to-peer (P2P) architectures for file sharing. This dissertation presents the design, implementation, and evaluation of a robust P2P file sharing system that addresses key challenges in scalability, fault tolerance, and security. The system implements a hybrid architecture combining centralized tracking with distributed file storage, featuring chunked file transfer, piece verification, and concurrent downloading capabilities.

The implementation includes sophisticated features such as tracker-based peer discovery, UPnP port mapping for NAT traversal, and a comprehensive piece management system for handling large file transfers. Through extensive testing and evaluation, the system demonstrates reliable performance under various network conditions while maintaining data integrity and transfer efficiency.

This work contributes to the field by implementing novel approaches to common P2P challenges, including peer availability management and fault-tolerant file transfers, while providing insights into the practical considerations of building distributed systems.

**Keywords:** Peer-to-Peer Networks, Distributed Systems, File Sharing, Network Programming, Fault Tolerance

# Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Background and Motivation . . . . .	5
1.2	Project Objectives . . . . .	5
1.3	Problem Statement . . . . .	5
1.4	Project Scope . . . . .	5
<b>2</b>	<b>Literature Review</b>	<b>6</b>
2.1	History of P2P Systems . . . . .	6
2.2	BitTorrent Protocol Analysis . . . . .	6
2.3	Modern P2P Applications . . . . .	6
2.4	Security Challenges in P2P Networks . . . . .	6
2.5	Distributed Hash Tables and Peer Discovery . . . . .	6
<b>3</b>	<b>System Architecture</b>	<b>7</b>
3.1	High-Level Design . . . . .	7
3.2	Component Overview . . . . .	7
3.3	Network Protocol Design . . . . .	7
3.4	Data Flow Architecture . . . . .	7
3.5	Storage System Design . . . . .	7
<b>4</b>	<b>Implementation Details</b>	<b>8</b>
4.1	Tracker Implementation . . . . .	8
4.2	Peer Discovery and Management . . . . .	8
4.3	File Chunking and Transfer Protocol . . . . .	8
4.4	Concurrent Download Management . . . . .	8
4.5	Error Handling and Recovery . . . . .	8
4.6	Security Implementation . . . . .	8
<b>5</b>	<b>Technical Challenges and Solutions</b>	<b>9</b>
5.1	Network NAT Traversal . . . . .	9
5.2	File Integrity Verification . . . . .	9

5.3	Peer Availability Management . . . . .	9
5.4	Performance Optimization . . . . .	9
5.5	Fault Tolerance Implementation . . . . .	9
<b>6</b>	<b>Testing and Evaluation</b>	<b>10</b>
6.1	Performance Metrics . . . . .	10
6.2	Scalability Testing . . . . .	10
6.3	Network Resilience . . . . .	10
6.4	Security Testing . . . . .	10
6.5	User Experience Testing . . . . .	10
<b>7</b>	<b>Conclusion and Future Work</b>	<b>11</b>
7.1	Project Achievements . . . . .	11
7.2	Limitations . . . . .	11
7.3	Future Improvements . . . . .	11
7.4	Final Reflections . . . . .	11
<b>A</b>	<b>Code Listings</b>	<b>12</b>
A.1	Core Components . . . . .	12
<b>B</b>	<b>Testing Data</b>	<b>13</b>
<b>C</b>	<b>User Manual</b>	<b>14</b>

# List of Figures

# List of Tables

# Chapter 1

## Introduction

### 1.1 Background and Motivation

[Discussion of the evolution of P2P systems and their role in modern networking]

### 1.2 Project Objectives

[Clear enumeration of project goals and success criteria]

### 1.3 Problem Statement

[Detailed description of the challenges in P2P file sharing]

### 1.4 Project Scope

[Outline of what the project encompasses and its boundaries]

# Chapter 2

## Literature Review

### **2.1 History of P2P Systems**

[Evolution of P2P architectures and protocols]

### **2.2 BitTorrent Protocol Analysis**

[Detailed examination of BitTorrent's approach]

### **2.3 Modern P2P Applications**

[Survey of current P2P implementations]

### **2.4 Security Challenges in P2P Networks**

[Analysis of security considerations]

### **2.5 Distributed Hash Tables and Peer Discovery**

[Review of peer discovery mechanisms]



# Chapter 3

## System Architecture

### 3.1 High-Level Design

[System overview with architectural diagrams]

### 3.2 Component Overview

[Detailed description of system components]

### 3.3 Network Protocol Design

[Protocol specifications and communication patterns]

### 3.4 Data Flow Architecture

[Data flow diagrams and explanations]

### 3.5 Storage System Design

[File storage and management architecture]

# Chapter 4

## Implementation Details

### 4.1 Tracker Implementation

[Details of the tracking system]

### 4.2 Peer Discovery and Management

[Peer handling mechanisms]

### 4.3 File Chunking and Transfer Protocol

[File transfer implementation details]

### 4.4 Concurrent Download Management

[Concurrency handling approaches]

### 4.5 Error Handling and Recovery

[Error management strategies]

### 4.6 Security Implementation

[Security measures and protocols]

# Chapter 5

## Technical Challenges and Solutions

### 5.1 Network NAT Traversal

[NAT handling implementation]

### 5.2 File Integrity Verification

[Data verification mechanisms]

### 5.3 Peer Availability Management

[Peer management strategies]

### 5.4 Performance Optimization

[Performance improvements]

### 5.5 Fault Tolerance Implementation

[Fault handling approaches]

# Chapter 6

## Testing and Evaluation

### 6.1 Performance Metrics

[Performance testing results]

### 6.2 Scalability Testing

[Scalability analysis]

### 6.3 Network Resilience

[Network testing results]

### 6.4 Security Testing

[Security evaluation]

### 6.5 User Experience Testing

[Usability assessment]

# **Chapter 7**

## **Conclusion and Future Work**

### **7.1 Project Achievements**

[Summary of accomplishments]

### **7.2 Limitations**

[Project limitations]

### **7.3 Future Improvements**

[Potential enhancements]

### **7.4 Final Reflections**

[Concluding thoughts]

# Appendix A

## Code Listings

### A.1 Core Components

```
1 type Tracker struct {  
2     fileIndex      map[string]*FileInfo  
3     peerIndex      map[string]map[string]bool  
4     peerLastSeen   map[string]time.Time  
5     mu             sync.RWMutex  
6 }
```

Listing A.1: Tracker Implementation

# Appendix B

## Testing Data

[Detailed test results and analysis]

# Appendix C

## User Manual

[System usage instructions]