

Create and Distribute a Torrent File in a Peer-to-Peer Environment

-- (2023115106)

Nagasurya N

1. Introduction

- Project Title: Creation and Distribution of a Torrent File in a P2P Environment.
- Abstract: A brief summary of what the project does (e.g., "This project demonstrates the decentralized distribution of files using the BitTorrent protocol...").
- Background: Briefly explain the shift from Client-Server to Peer-to-Peer (P2P) models.
- Objectives: What were you trying to achieve? (e.g., efficient file sharing, reducing server load, ensuring data integrity).

2. Distributed Systems Concepts

Explain the theory behind your project to show your academic understanding:

- P2P Architecture: Define unstructured vs. structured P2P networks.
- Decentralization: How the system functions without a central file server.
- Scalability: Why P2P systems handle high traffic better than traditional servers.
- Data Integrity: Mention how SHA-1/SHA-256 hashing is used to verify file pieces.

3. Project Objectives

The objectives of this project are listed below:

- To study the working of Peer-to-Peer architectures
- To create torrent metadata files for data sharing
- To implement decentralized file transfer mechanisms
- To improve download performance through parallel sharing
- To understand practical distributed system concepts

4. Background and Motivation

Modern distributed applications require efficient mechanisms to transfer large volumes of data across networks. Centralized file distribution systems often suffer from limitations such as high server load, low scalability, and single points of failure. These challenges have led to the adoption of decentralized communication models.

This project focuses on implementing a torrent-based file sharing system that operates in a peer-to-peer distributed environment. By allowing users to exchange file fragments directly, the system ensures improved performance, robustness, and efficient resource utilization.

5. Challenges & Solutions

- Peer Churn: What happens when a seeder goes offline?
- Network Issues: Handling NAT traversal or firewall blocks.
- Synchronization: Ensuring pieces are requested in a way that doesn't cause bottlenecks.

6. Project Workflow

Provide a step-by-step walkthrough:

1. Creation: Generating the .torrent file from the source data.
2. Seeding: The initial peer making the file available.
3. Discovery: How a new peer finds the initial seeder via the tracker.
4. Distribution: The exchange of pieces between multiple peers.

7. Operational Workflow

The workflow begins when a user generates a torrent metadata file for the selected content. This metadata is shared through the tracker, enabling other peers to identify available data sources.

Peers request and exchange file fragments based on availability. Once a peer receives all fragments, it reconstructs the original file and continues sharing it with others, contributing to the network's sustainability.

8. System Architecture

The system architecture follows a decentralized Peer-to-Peer model with minimal reliance on centralized components. The primary elements of the system include torrent creator, tracker, seeders, and leechers.

The tracker acts as a coordination service, while actual file data transfer occurs directly between peers. Each peer functions both as a client and a server, allowing efficient sharing of resources. This architecture enhances scalability and ensures high availability even when some nodes fail.

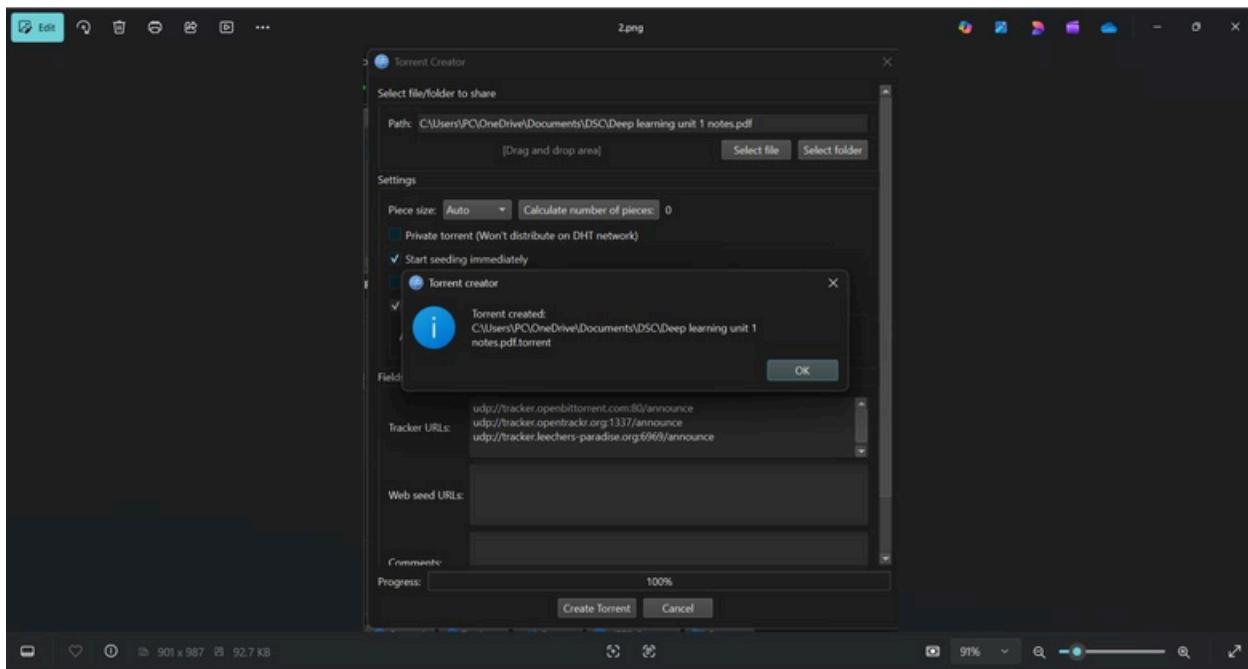
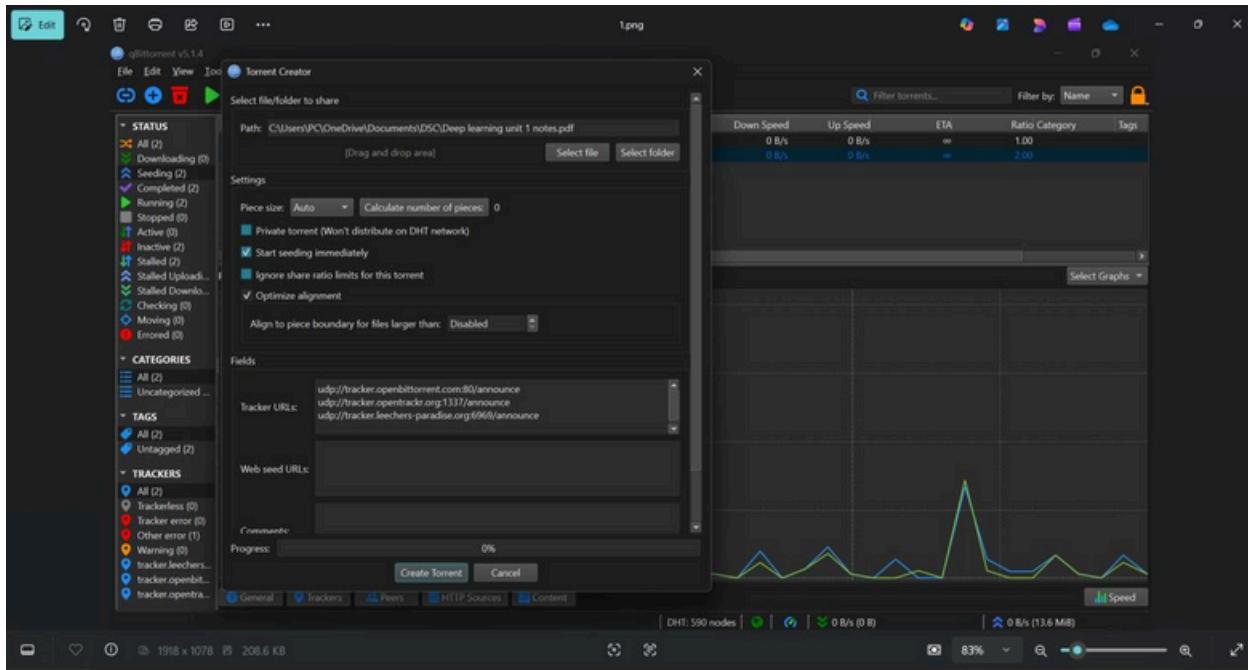
9. Testing and Results

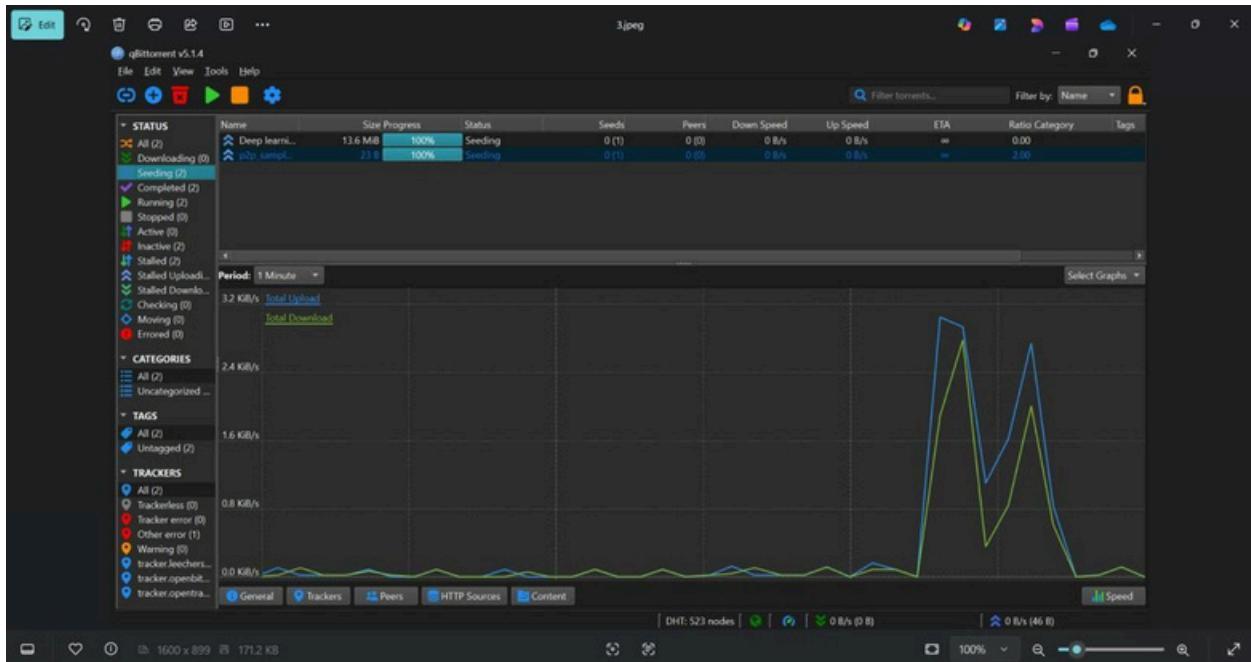
- Test Cases: Show what happened when 2 peers joined, then 5, then 10.
- Performance Metrics: Download speeds vs. number of seeders.
- Integrity Check: Screenshots or logs showing that the file was verified successfully after download.

10. Outputs

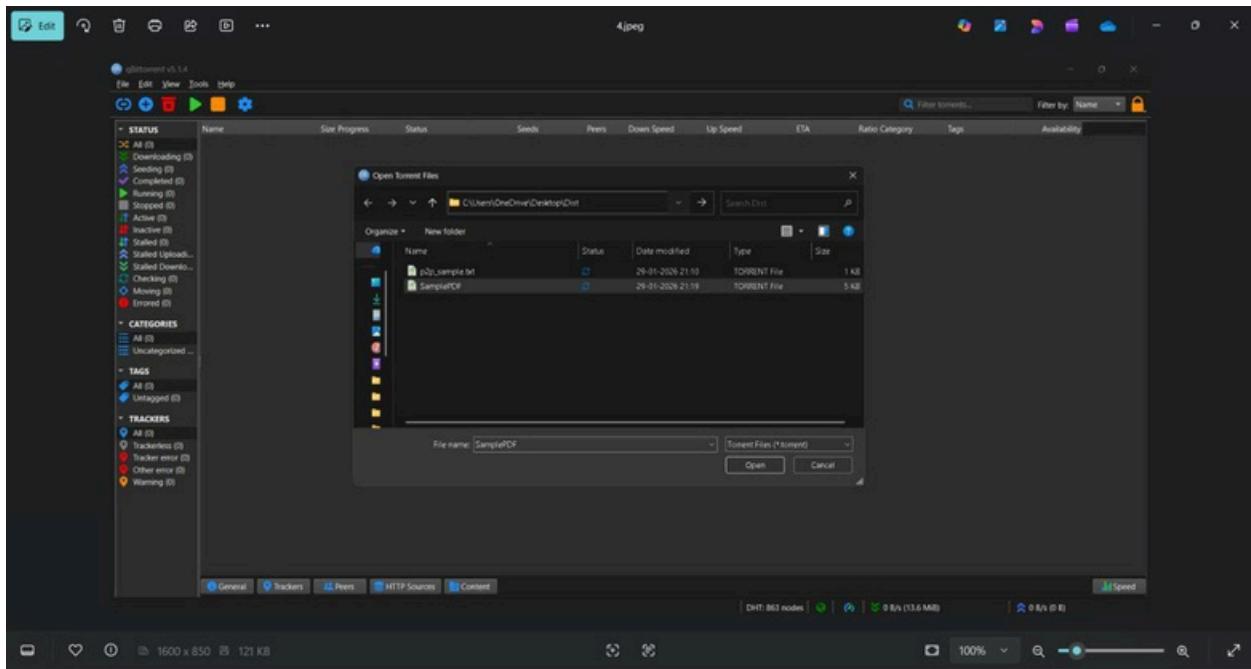
The implemented system effectively demonstrates torrent-based file sharing in a distributed network. Observations indicate reduced reliance on centralized servers, improved transfer speeds, and efficient utilization of ne

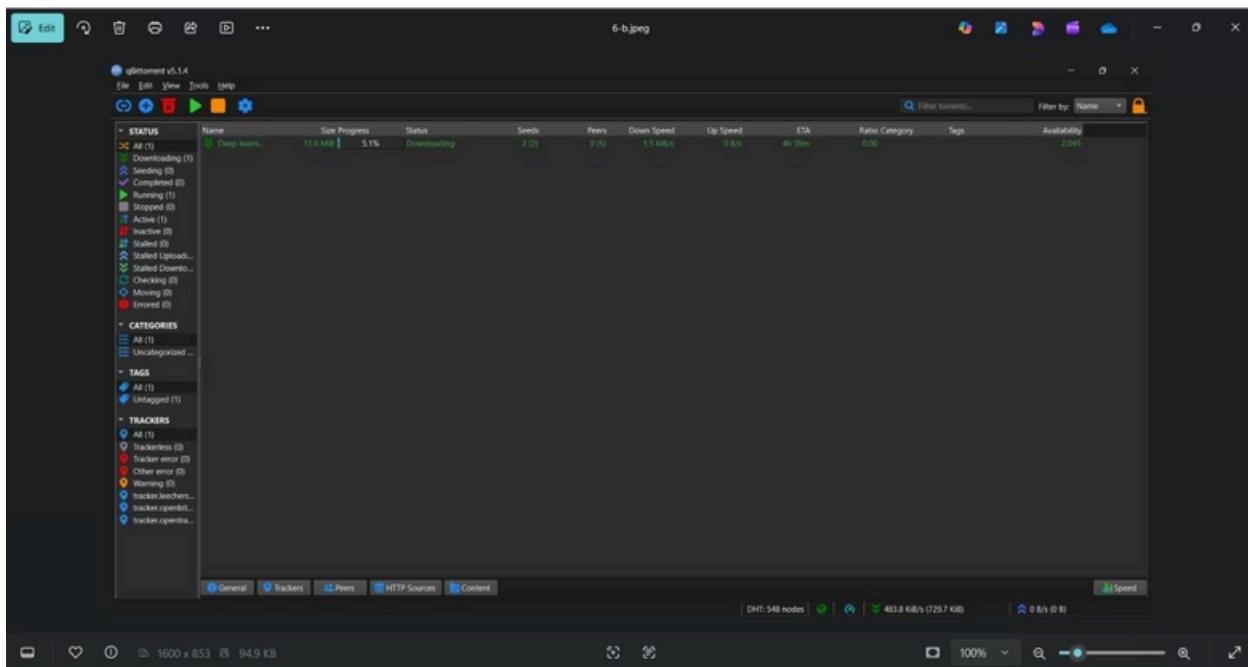
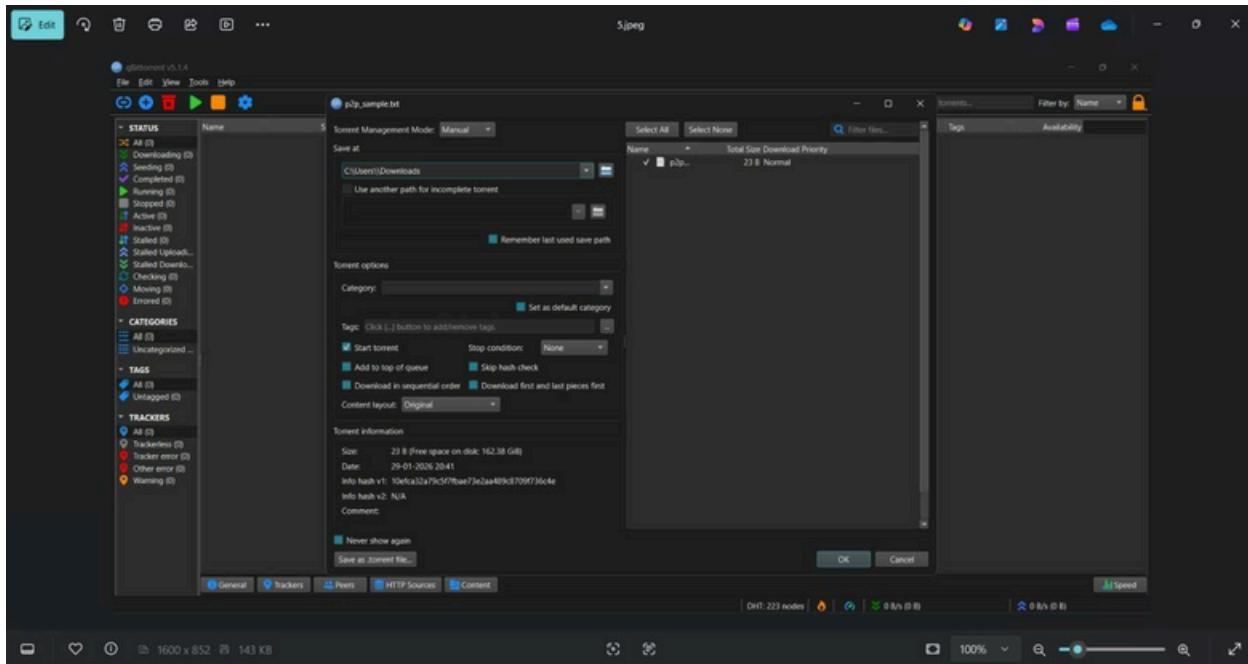
SENDER:





RECEIVER:





11. Conclusion

In conclusion, torrent-based P2P systems provide a robust and scalable solution for large-scale file sharing. The project effectively applies distributed system principles such as decentralization, concurrency, and reliability, making it a practical implementation for modern data distribution needs.