Create and Distribute a Torrent File in a Peer-to-Peer Environment ---- (2023115063) Pradeep K

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.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

3. 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.

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. 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.

6.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.

7.Torrent Creation Process

Torrent creation involves generating a metadata file that describes the original content. This metadata includes file name, file size, piece length, hash values of file segments, and tracker information.

The original file is divided into fixed-size pieces, and each piece is hashed using cryptographic algorithms. The generated torrent file does not contain actual data but serves as a reference for peers to locate and download the content from the network.

8.Tools and Technologies

List everything you used to build and test the project:

- **Programming Language:** (e.g., Python, Java, Node.js).
- **Libraries/Frameworks:** (e.g., `libtorrent` for Python, `WebTorrent` for JS, or custom socket programming).
- **Development Environment:** (e.g., VS Code, IntelliJ, Linux/Windows).
- **Network Simulation:** (e.g., Localhost testing, Docker containers, or multiple physical PCs).

9.Peer-to-Peer vs. Client-Server Paradigms

This project highlights a fundamental shift in network architecture. In a traditional client-server model, the server's bandwidth is a bottleneck; as more users join, the performance per user drops. In contrast, the P2P model

demonstrates **reverse-scalability**: as more peers join the swarm, the total available bandwidth of the system increases because every leecher is also a potential uploader. This makes P2P environments exceptionally resilient against "Flash Crowds" (sudden spikes in popularity) and significantly reduces the infrastructure costs for the original content distributor.

10.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.

11. Cloud Deployment

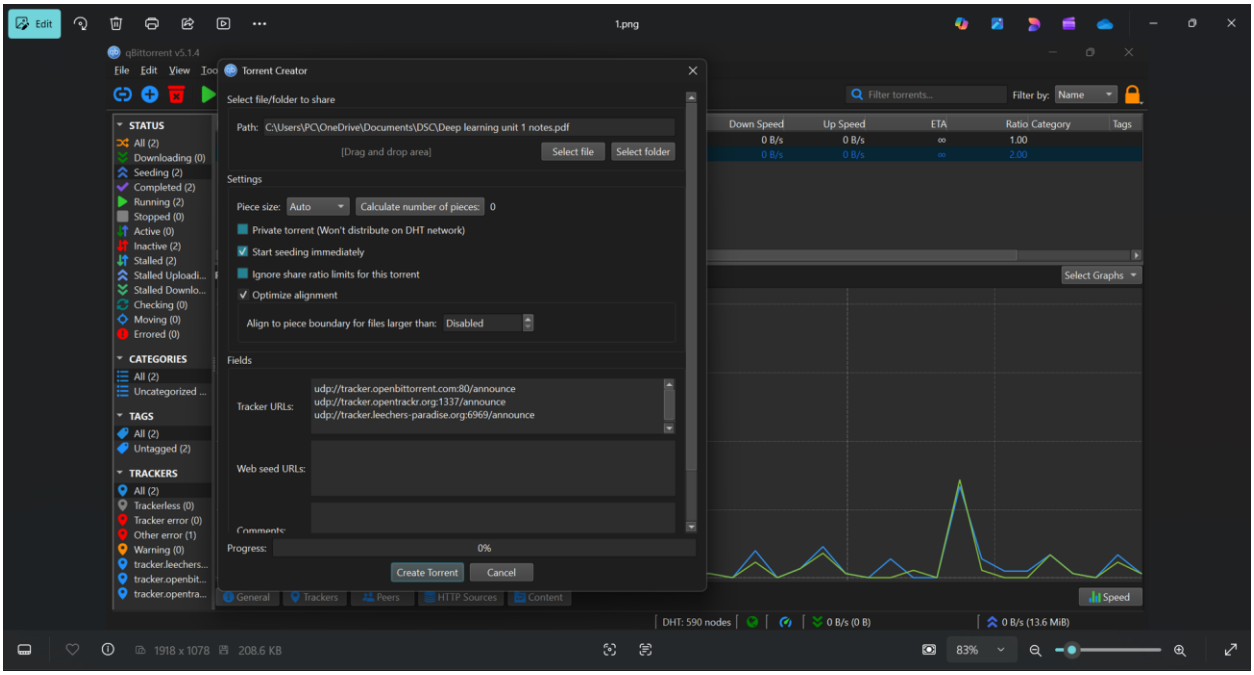
To improve accessibility and performance, the tracker component can be hosted on a cloud platform. Cloud deployment allows peers from different locations to connect seamlessly.

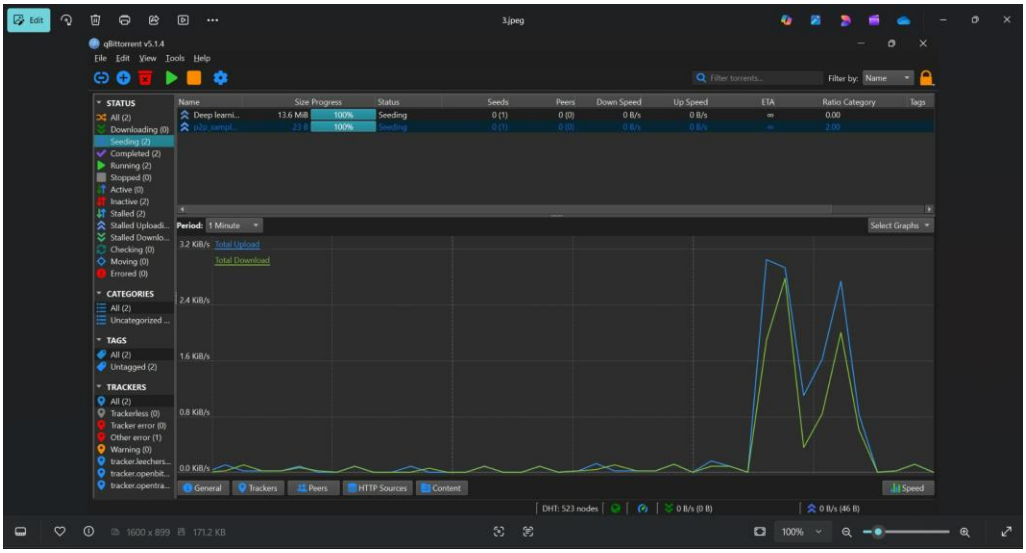
The use of cloud infrastructure provides better availability, scalability, and fault tolerance. It also helps simulate real-world distributed environments for testing and analysis.

12.outputs

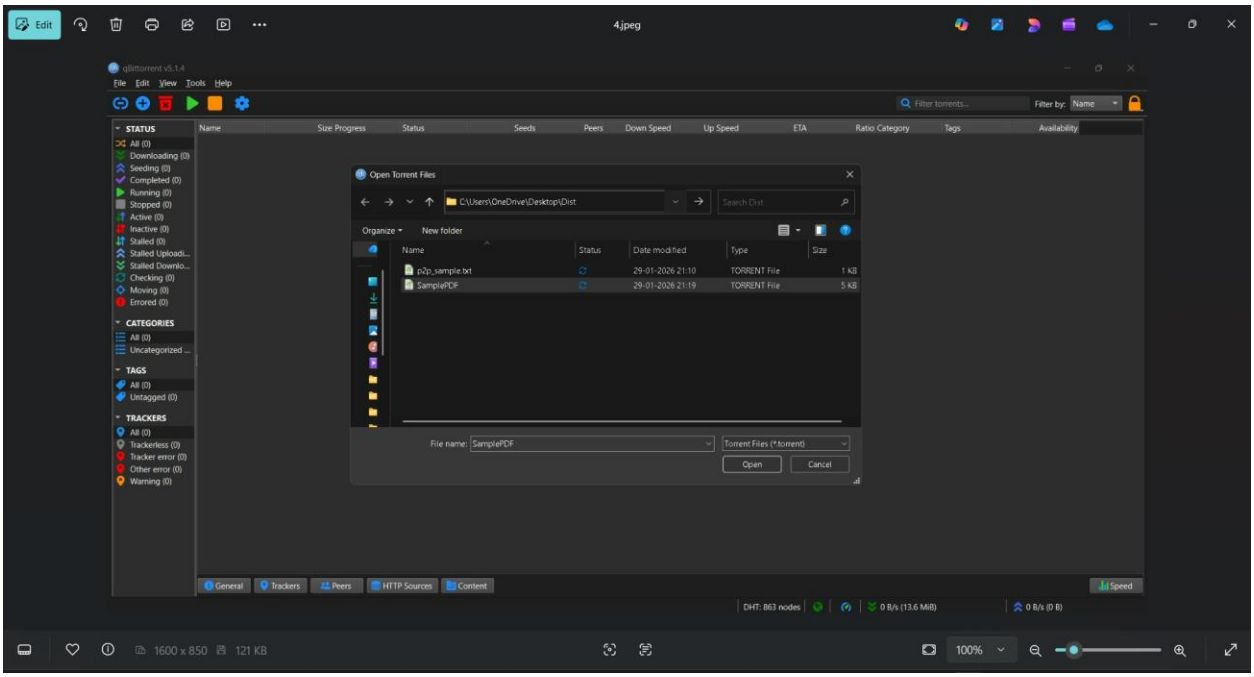
The implemented system effectively demonstrates torrent-based file sharing in a distributed network. Observations indicate reduced reliance on centralized servers, improved transfer speeds, and efficient utilization of network resources.

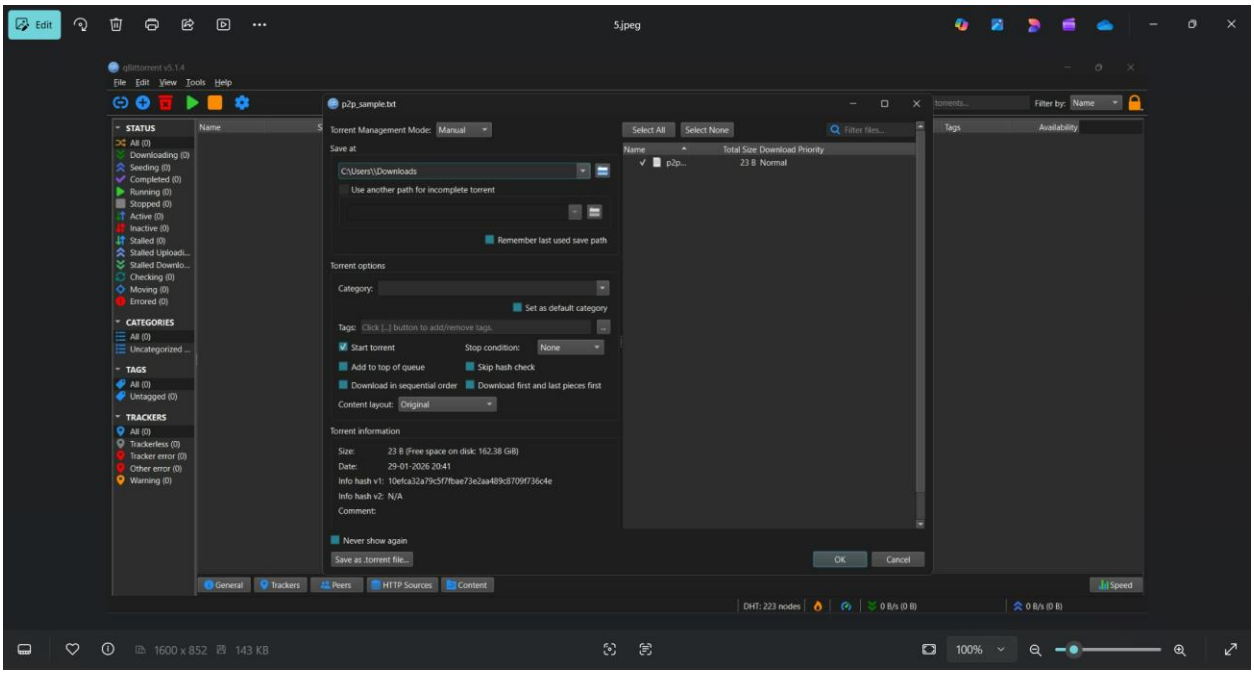
SENDER OUTPUT

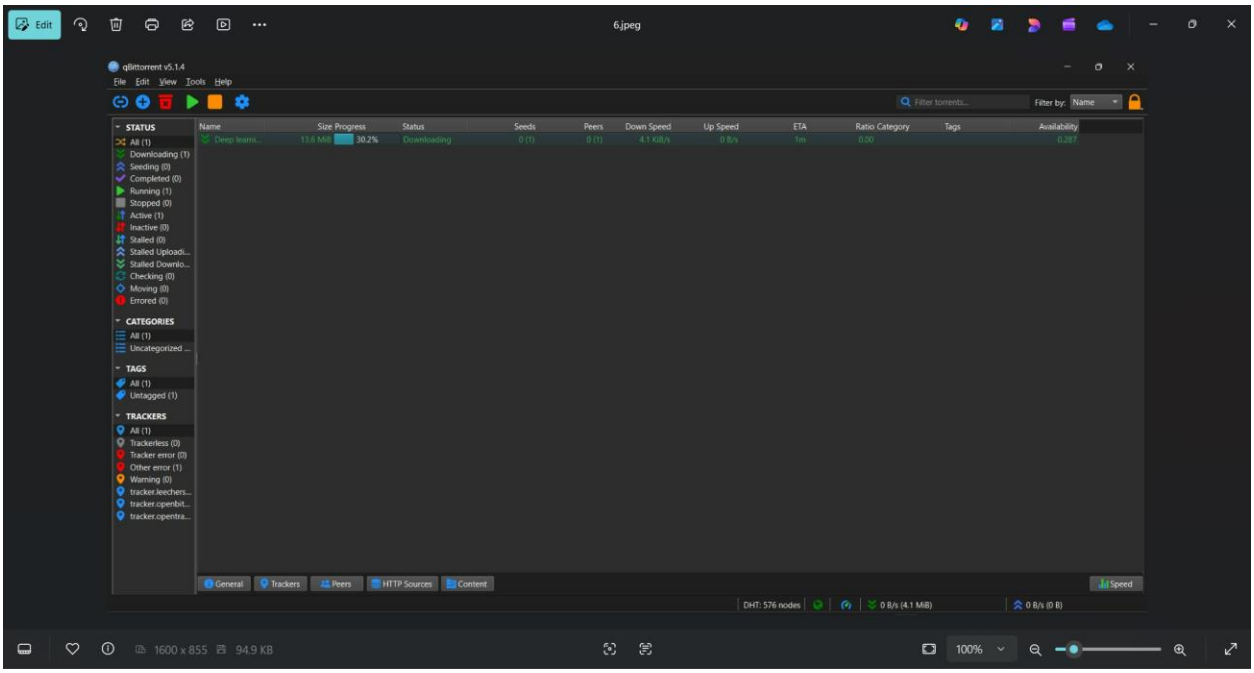




RECEIVER OUTPUT:


4.jpeg
Edit
qBittorrent v5.1.4
File Edit View Tools Help
Open Torrent Files
C:\Users\OneDrive\Desktop\Dist
Organize
New folder
p2p_sample.txt
SamplePDF
File name: SamplePDF
Torrent Files (*.torrent)
Open
Cancel
1600 x 850
121 KB
100%


5.jpeg
Edit
qBittorrent v5.1.4
File Edit View Tools Help
p2p_sample.txt
Torrent Management Mode: Manual
Save at
C:\Users\Downloads
Use another path for incomplete torrent
Remember last used save path
Torrent options
Category:
Set as default category
Tags:
Start torrent
Stop condition: None
Add to top of queue
Skip hash check
Download in sequential order
Download first and last pieces first
Content layout: Original
Torrent information
Size: 23 B (Free space on disk: 162.38 GiB)
Date: 29-01-2026 20:41
Info hash v1: 10efca32a79c5f7fbae73e2aa489c8709f736c4e
Info hash v2: N/A
Comment:
Never show again
Save as .torrent file...
OK
Cancel
1600 x 852
143 KB
100%



13.conclusion

The implemented system successfully demonstrates the creation and distribution of torrent files in a peer-to-peer environment. The results show efficient file transfer, reduced server load, and effective use of network bandwidth.

In conclusion, the project highlights the advantages of torrent-based P2P systems in distributed computing. The implementation reinforces core distributed system concepts such as decentralization, scalability, and reliability, making it a practical solution for large-scale data distribution.