Implementation Documentation

Assignment 2: Create and Distribute a Torrent File in a Peer-to-Peer Environment

---

1. Introduction

Peer-to-Peer (P2P) computing represents a distributed system architecture in which individual computers, referred to as peers, communicate directly with each other to share resources such as files, storage, and processing power. Unlike traditional client–server models, P2P systems eliminate centralized control, thereby improving scalability, fault tolerance, and efficiency.

BitTorrent is one of the most widely adopted P2P protocols for large-scale file distribution. It divides files into smaller fragments called pieces and allows peers to exchange these pieces simultaneously. This parallel data transfer significantly reduces download time while balancing network load among participants.

This implementation focuses on creating a torrent file and demonstrating decentralized file sharing using qBittorrent. The experiment includes torrent creation, seeding, peer discovery, downloading, and integrity verification, thereby providing hands-on exposure to real-world P2P communication.

---

2. Objectives

The main objectives of this implementation are:

- To understand the working principles of Peer-to-Peer computing
- To create and configure torrent files using BitTorrent technology
- To perform seeding and downloading between peers
- To observe decentralized peer communication
- To verify data integrity using cryptographic hash values
- To analyze the efficiency and reliability of BitTorrent-based file sharing

---

3. System Requirements

Hardware Requirements

- Laptop or Desktop Computer
- Minimum 4 GB RAM
- Stable Network Connection

Software Requirements

- Operating System: Windows / Linux
- Torrent Client: qBittorrent v5.1.4
- Public Tracker Services
- Non-copyrighted document file

---

4. Overview of BitTorrent Technology

BitTorrent operates by splitting large files into small pieces, typically ranging from a few kilobytes to several megabytes. Each piece is identified by a cryptographic hash value, ensuring integrity during transmission.

Key components include:

Seeder

A peer that possesses the complete file and uploads pieces to other peers.

Leecher

A peer that downloads the file but may also upload downloaded pieces simultaneously.

Tracker

A lightweight server that assists peers in discovering each other by maintaining peer lists.

Torrent File

A metadata file containing:

- File name and size
- Piece length
- Hash values
- Tracker URLs

Once peers connect, data transfer occurs directly without passing through the tracker.

---

5. System Architecture

The overall workflow of the system is:

File Selection → Torrent Creation → Seeding → Peer Discovery → Downloading → Integrity Verification

The architecture is decentralized, where every peer can function both as client and server. This eliminates single points of failure and enhances scalability.

---

6. Implementation Methodology

6.1 File Selection

A non-sensitive document file (`bit_torrent.docx`) is selected from the local system. This file serves as the shared resource.

6.2 Torrent File Creation

Using qBittorrent, a torrent file is created by selecting the source file and specifying tracker URLs. During this process:

- Piece size is set automatically
- Hybrid torrent format is chosen
- DHT (Distributed Hash Table) is enabled
- Public tracker URLs are added
- Torrent metadata is generated

The resulting `.torrent` file contains cryptographic hashes for every file piece, ensuring reliable reconstruction at receiving peers.

6.3 Seeding Process

The system that owns the original file acts as the seeder. After loading the torrent, qBittorrent begins announcing the torrent to the tracker and DHT network.

Once active, the seeder uploads file pieces upon request. The torrent status changes to *Seeding*, indicating readiness for peer connections.

6.4 Peer Discovery and Downloading

Another peer opens the same torrent file. The client contacts the tracker and DHT network to locate active seeders.

The downloading peer receives different file pieces in parallel and simultaneously uploads acquired pieces to others. This cooperative sharing improves throughput and reduces dependency on a single source.

6.5 Peer-to-Peer Communication

Peers communicate directly using TCP/UDP connections. Each peer dynamically exchanges available pieces based on rarity algorithms, ensuring balanced distribution across the swarm.

This decentralized exchange demonstrates the robustness and efficiency of BitTorrent.

6.6 File Integrity Verification

Each downloaded piece is verified using SHA-based hash values stored in the torrent file. If corruption is detected, the affected piece is automatically re-requested.

After all pieces pass verification, the torrent reaches 100% completion, confirming that the received file matches the original exactly.

---

7. Cloud or Local Deployment

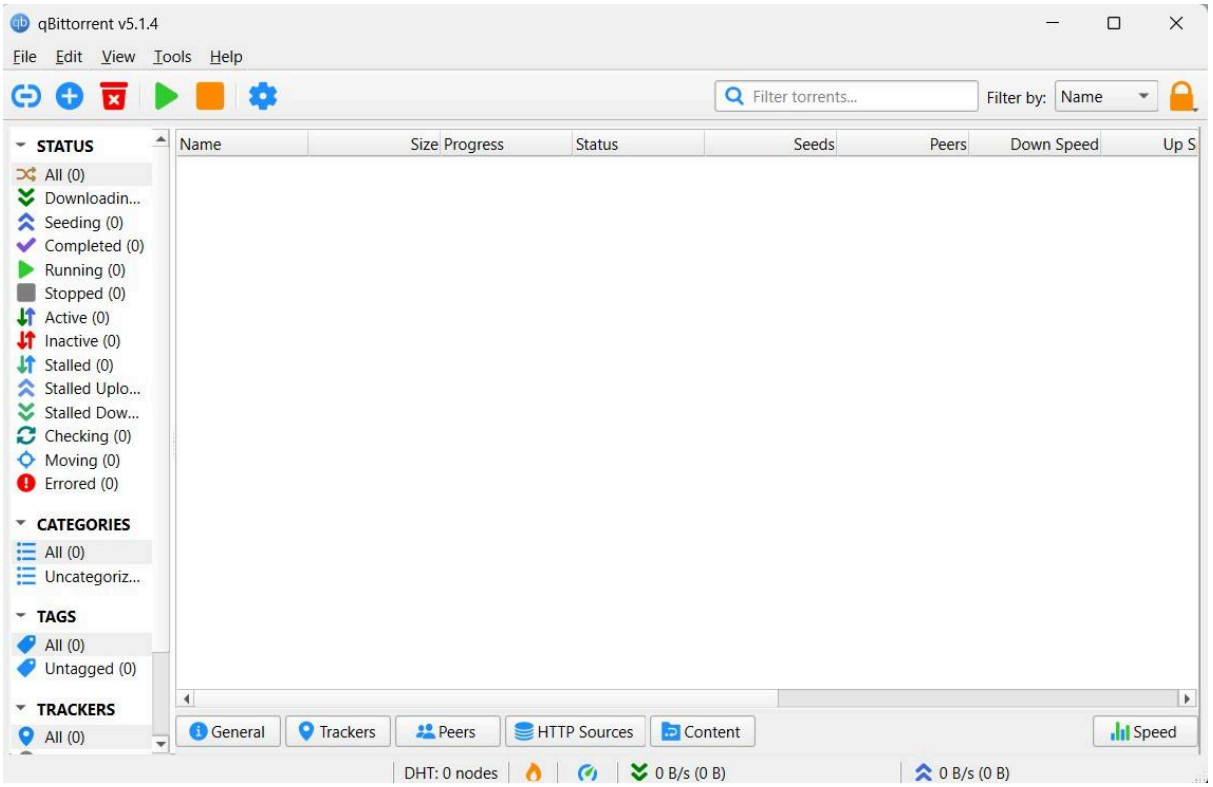
Torrent clients can operate on either local machines or cloud-hosted systems. Cloud deployment enables public IP accessibility, simulating real-world distributed environments. This experiment supports both modes, demonstrating flexible deployment.

---

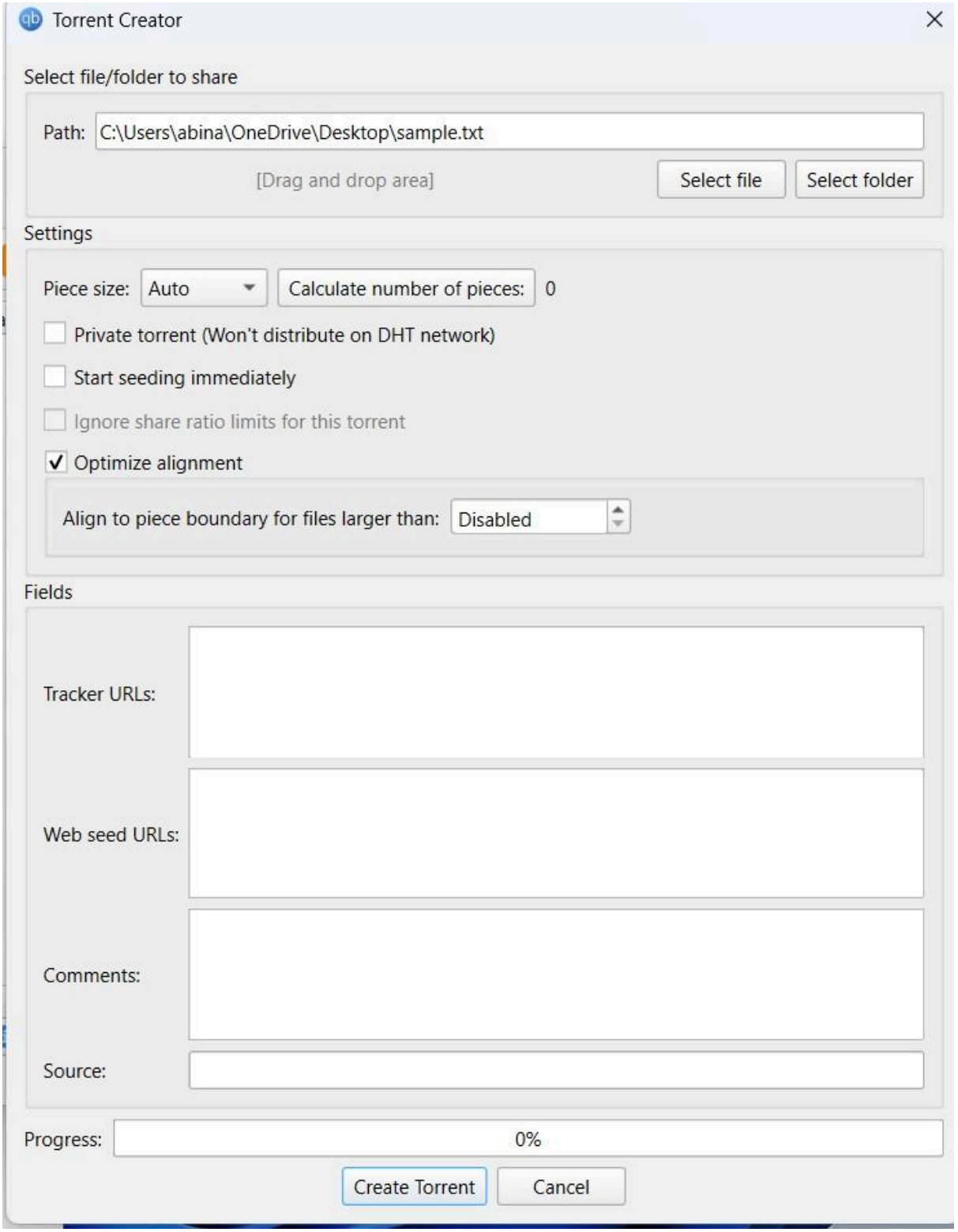
8. Results and Observations

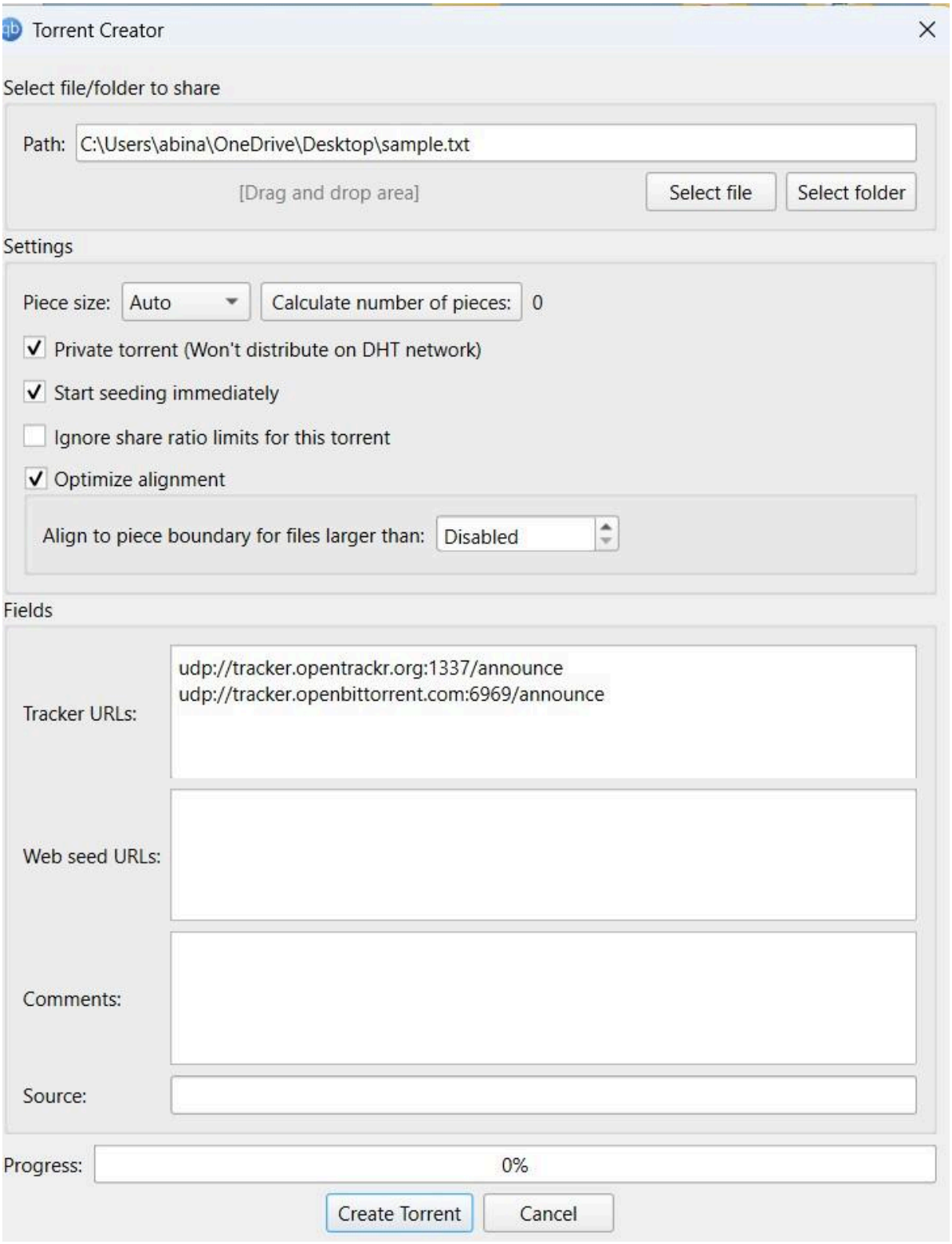
- Torrent file was created successfully
- Seeder became active and announced to tracker
- Leecher discovered peers
- File downloaded completely
- Hash verification confirmed integrity
- Torrent achieved seeding status after completion

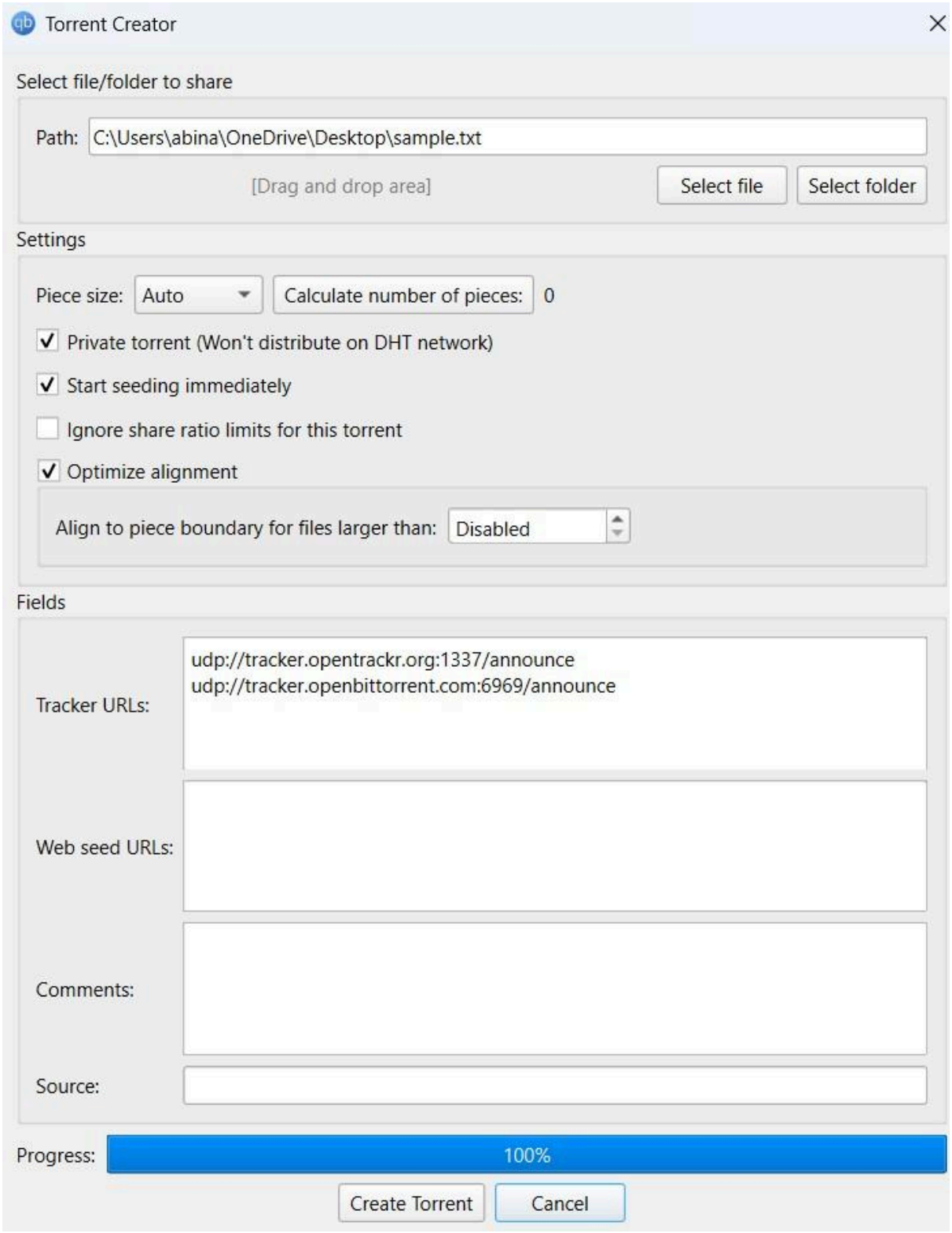
The experiment validated decentralized file transfer.

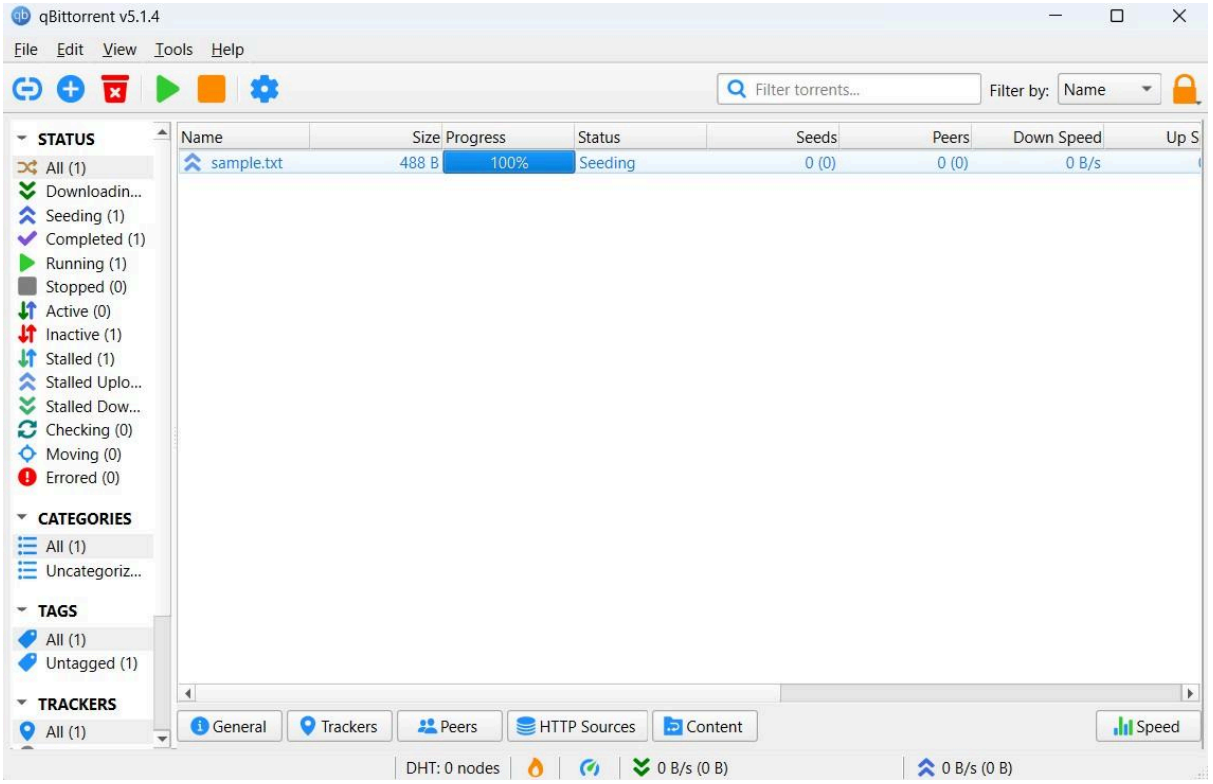


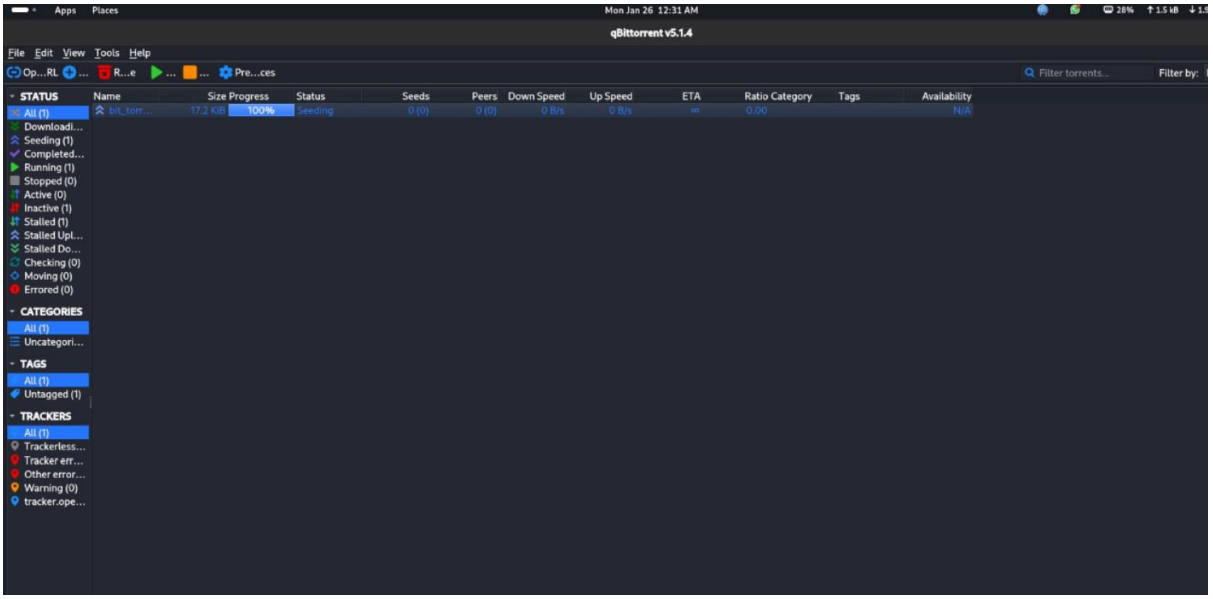

```
Assignment 2 – Peer-to-Peer File Sharing using BitTorrent

Student Name: Abinash
Tool Used: qBittorrent
Cloud Platform: AWS EC2

This file is shared using BitTorrent protocol in a peer-to-peer environment.
The torrent was created on a local machine and downloaded on a cloud instance.
Public trackers were used for peer discovery.

The purpose of this file is to demonstrate decentralized file sharing and integrity verification.

Course: Distributed Systems and Computing
```

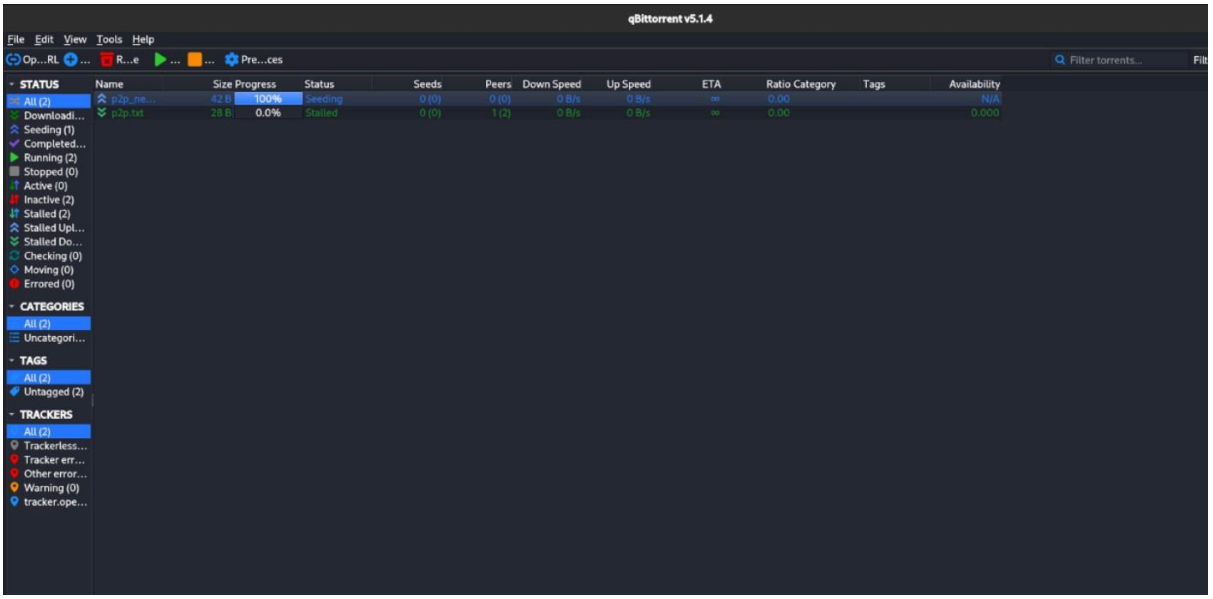

Torrent Creator
Select file/folder to share
Path:
C:\Users\abina\OneDrive\Desktop\sample.txt
[Drag and drop area]
Select file
Select folder
Settings
Piece size:
Auto
Calculate number of pieces:
0
Private torrent (Won't distribute on DHT network)
Start seeding immediately
Ignore share ratio limits for this torrent
Optimize alignment
Align to piece boundary for files larger than:
Disabled
Fields
Tracker URLs:
Web seed URLs:
Comments:
Source:
Progress:
0%
Create Torrent
Cancel


Torrent Creator
Select file/folder to share
Path:
C:\Users\abina\OneDrive\Desktop\sample.txt
[Drag and drop area]
Select file
Select folder
Settings
Piece size:
Auto
Calculate number of pieces:
0
Private torrent (Won't distribute on DHT network)
Start seeding immediately
Ignore share ratio limits for this torrent
Optimize alignment
Align to piece boundary for files larger than:
Disabled
Fields
Tracker URLs:
udp://tracker.opentrackr.org:1337/announce
udp://tracker.openbittorrent.com:6969/announce
Web seed URLs:
Comments:
Source:
Progress:
0%
Create Torrent
Cancel


Torrent Creator
Select file/folder to share
Path: C:\Users\abina\OneDrive\Desktop\sample.txt
[Drag and drop area]
Select file
Select folder
Settings
Piece size: Auto
Calculate number of pieces: 0
Private torrent (Won't distribute on DHT network)
Start seeding immediately
Ignore share ratio limits for this torrent
Optimize alignment
Align to piece boundary for files larger than: Disabled
Fields
Tracker URLs:
udp://tracker.opentrackr.org:1337/announce
udp://tracker.openbittorrent.com:6969/announce
Web seed URLs:
Comments:
Source:
Progress:
100%
Create Torrent
Cancel


qBittorrent v5.1.4
File Edit View Tools Help
Filter torrents...
Filter by: Name
STATUS
All (1)
Downloadin...
Seeding (1)
Completed (1)
Running (1)
Stopped (0)
Active (0)
Inactive (1)
Stalled (1)
Stalled Uplo...
Stalled Dow...
Checking (0)
Moving (0)
Errored (0)
CATEGORIES
All (1)
Uncategoriz...
TAGS
All (1)
Untagged (1)
TRACKERS
All (1)
Name Size Progress Status Seeds Peers Down Speed Up S
sample.txt 488 B 100% Seeding 0 (0) 0 (0) 0 B/s
General Trackers Peers HTTP Sources Content Speed
DHT: 0 nodes
0 B/s (0 B)
0 B/s (0 B)


Apps Places
Mon Jan 26 12:31 AM
qBittorrent v5.1.4
File Edit View Tools Help
STATUS
CATEGORIES
TAGS
TRACKERS
Name Size Progress Status Seeds Peers Down Speed Up Speed ETA Ratio Category Tags Availability



---

9. Advantages of BitTorrent P2P Model

- Eliminates centralized bottlenecks
- Improves download speeds via parallel transfers
- Automatically verifies data integrity
- Scales efficiently with peer count
- Provides fault tolerance

---

10. Applications

- Software distribution
- Academic content sharing
- Large dataset delivery
- Media distribution
- Distributed cloud storage

---

11. Conclusion

This implementation successfully demonstrates torrent creation and decentralized file distribution using BitTorrent technology. By performing seeding, downloading, peer communication, and integrity verification, the experiment confirms the efficiency and

reliability of P2P networks. The results highlight BitTorrent’s scalability and robustness, making it suitable for modern distributed applications.
