Continuous File Synchronization System

October 31, 2024

1 Introduction

1.1 Motivation

The increasing prevalence of multi-device usage in both personal and professional environments has created a critical need for seamless file synchronization. Users frequently work across multiple devices (laptops, smartphones, tablets) and require their files to be consistently updated and accessible across all platforms.

We choose to develop the File Synchronization System as our graduation project because it address very interesting challenges in distributed systems and Real-time synchronization reliability. This project gives us a chance to learn more about distributed computing, networks and back-end engineering while tracking a real-life problem. Additionally, we are particularly interested in consensus protocols to ensure data consistency and resolve conflicts effectively, enhancing our understanding of these critical concepts in distributed systems.

1.2 Problem Statement

In a world where digital collaboration, remote work, and data accessibility are becoming increasingly essential, users often need real-time access to the most up-to-date versions of files across multiple devices. Traditional file-sharing services can be limited by connectivity issues, centralized storage constraints, or restrictions in updating files consistently across distributed devices. Users struggle with maintaining consistent file versions across multiple devices, leading to:

- Version conflicts and lost work
- Manual file transfer overhead
- Storage redundancy
- Security vulnerabilities in file sharing
- Inconsistent file access across different platforms

Hence, developing an efficient, decentralized file synchronization system to ensure continuous file availability and consistency across multiple devices in real time is crucial.

1.3 Goals

he primary goal of this project is:

- Develop a robust file synchronization system that maintains file consistency across multiple devices
- Implement real-time synchronization with minimal latency

- Create an efficient conflict resolution mechanism
- Ensure secure file transfer and storage

From the Technical Objectives:

- Design a scalable architecture supporting multiple concurrent users
- Implement efficient delta-sync algorithms to minimize bandwidth usage
- Develop cross-platform compatibility (Windows, macOS, Linux)
- Create an intuitive user interface for managing synchronization

1.4 Related Work

1.4.1 Currently Available Solutions

Several file synchronization solutions currently exist in the market, each offering distinct features and trade-offs in terms of performance, ease of use, and security. Some of the most popular file sync systems include:

- **Syncthing**: Syncthing is an open-source, decentralized file synchronization application designed to replace proprietary sync and cloud services. It enables users to synchronize files directly between devices on a local network or over the internet without storing data on third-party servers. Syncthing emphasizes data privacy, ensuring that data is encrypted during transfer and only accessible by the devices involved in the sync process. However, due to its decentralized nature, managing complex sync setups may require manual configuration.
- **Dropbox**: Dropbox is a widely-used cloud storage service that includes file synchronization as a core feature. Dropbox offers seamless sync across multiple devices, enabling users to access files on desktop and mobile platforms. Its user-friendly interface and collaboration features make it ideal for everyday users and small teams. However, Dropbox relies on centralized storage, which can raise privacy concerns, and its file syncing speed can vary based on network and storage limitations.
- **Google Drive**: Google Drive combines cloud storage and file synchronization, allowing users to store and access files on any device with internet connectivity. The service includes robust integration with other Google applications, such as Google Docs and Sheets, making it popular for collaborative work. However, similar to Dropbox, Google Drive uses a centralized storage model, and users must trust Google with their data privacy.
- **OneDrive**: Microsoft OneDrive is a cloud-based solution that integrates tightly with Windows and the Microsoft Office suite. It offers both cloud storage and file synchronization features, making it a strong option for

users in the Microsoft ecosystem. OneDrive provides tools for sharing and collaboration, particularly for enterprise users. Like Google Drive and Dropbox, OneDrive is centralized and may not meet the privacy needs of all users.

- **rsync**: rsync is a command-line utility for Unix-based systems that offers efficient, incremental file synchronization. While not specifically a file sync service, rsync provides high-speed sync capabilities over networks and is popular for backups and transferring large amounts of data. However, its usage requires familiarity with command-line operations, making it less accessible to everyday users.

- **Nextcloud**: Nextcloud is an open-source, self-hosted cloud storage and file synchronization solution. It allows users to set up their own server, keeping full control over data. Nextcloud provides additional collaborative tools and security options tailored for businesses and privacy-focused users. However, it requires setup and maintenance on the part of the user, which may not be suitable for non-technical users.

1.4.2 Limitations of Current Solutions

While each of these solutions has its own strengths, they share certain limitations. Centralized systems, like Dropbox, Google Drive, and OneDrive, come with privacy and data ownership concerns. Decentralized and self-hosted solutions, like Syncthing and Nextcloud, address privacy but can be challenging to configure for less technical users. These tools may also face performance and reliability issues when syncing large datasets over fluctuating network connections.