Centralized Peer-to-Peer Courseware Sharing System for UCS

Thin Su Su Win, Kalyar Myo San
University of Computer Studies, Yangon
thinsusu.win@gmail.com,kalyar.myosan@gmail.com

Abstract

Today, thanks to the development of a number of advanced P2P file sharing applications, the reach and scope of peer networks has increased dramatically. Moreover, Centralized application architecture also become important in P2P file sharing system since file sharing can be used based around the central server system which directs traffic between individual registered users. On the other hand, established courseware system in University community encourages the development of detailed treatments of the subject matter and also reduces the cost of lesson development.

In this paper, Centralized Peer-to-Peer system is implemented to share university courseware between students and teachers. Furthermore, Multithreading technique is used in the development of this system to control the uploading/downloading files by multi-users and, the system also restricted the file sharing among users for the security purpose.

Keywords: P2P, Centralized P2P, Courseware, Multithreading

1. Introduction

Efficient and effective resource sharing systems are invaluable in modern economic systems such as the Internet. P2P is a Networking Software where clients communicate with each other rather than go through a server. The most common way of linking computers so they can share data or processing power is via client-server architecture. Servers store data and applications that PCs can access. A different way to achieve data and application sharing is by allowing PCs to connect directly with each other and act as both clients and servers. This is the basis of Peer-to-Peer computing. A number of years ago applications that allow these direct connections were developed for use in office environments with a small number of PCs. Programs such as AppleTalk Network and Windows for Workgroups are early examples of this technology known as Peer-to-Peer (P2P). Recent developments have made it possible to expand P2P technology to computers connected to the Internet as well as those connected through a network. [3]

This paper is organized as follow: Section 2 describes P2P System, Centralized P2P System, Courseware Systems, and Multithreading. The overview of the system design is presented in Section 3. Finally, System implementation and conclusion of the system are described in Section 4 and 5.

2. Background Theory

2.1. Peer-to-peer Systems

File sharing P2P applications focus on storing and retrieving information from various peers in the network. The model that popularized this class of application is the content exchange model. A P2P application should be able to locate other peers in the network. Once an application is able to locate other peers, it should be able to communicate with them using messages. P2P systems consider that all nodes or peers are equal for sharing information.

Peer-to-Peer network can be classified into two types: structured peer-to-peer network and unstructured peer-to-peer network. [1, 3]

In structured peer-to-peer networks, connections in the overlay are fixed. They typically use distributed hash table-based (DHT) indexing.

Unstructured peer-to-peer networks do not provide any algorithm for organization or optimization of network connections. In particular, three models of unstructured architecture are defined. In pure peer-to-peer systems the entire network consists solely of equipotent peers. There is only one routing layer, as there are no preferred nodes with any special infrastructure function.

Hybrid peer-to-peer systems allow such infrastructure nodes to exist often called super nodes. In centralized peer-to-peer systems, a central server is used for indexing functions and to bootstrap the entire system. [4]

2.2. Advantages of Centralized P2P System

Centralized - Resources and data security are controlled through the server.
Scalability - Any or all elements can be replaced individually as needs increase.
Flexibility - New technology can be easily integrated into system.
Interoperability - All components (client/network/server) work together.
Accessibility - Server can be accessed remotely and across multiple platforms. [4]

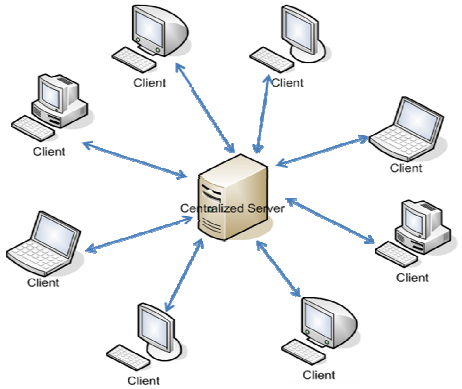

Figure 1: Centralized P2P System

2.3. Courseware Systems

Courseware is educational material intended as kits for teachers or trainers or as tutorials for students, usually packaged for use with a computer. Courseware can encompass any knowledge area, but information technology subjects are most common. Courseware is frequently used for delivering education about the personal computer and its most popular business applications, such as word processing and spreadsheet programs. Courseware is also widely used in information technology industry certification programs, such as the Microsoft Certified Systems Engineer (MCSE) and the Computing Technology Industry Association's A+ examination.

Courseware can include:

- Material for instructor-led classes
- Material for self-directed computer-based training (CBT)
- Web sites that offer interactive tutorials
- Material that is coordinated with distance learning, such as live classes conducted over the Internet
- Videos for use individually or as part of classes

Courseware systems are the software designed to be used in educational programs. [5]

2.4. Multithreading

Multithreading is introduced to exploit thread-level parallelisms within a processor. In computer science, threads of execution results from a fork of a computer program are split into two or more concurrently running tasks. The implementation of threads and processes differs from one operating system to another, but in most cases, a thread is contained inside a process. Multiple threads can exist within the same process and share resources such as memory, while different processes do not share these resources. On a single processor, multithreading generally occurs by time-division multiplexing (as in multitasking): the processor switches between different threads. The context switching generally happens frequently enough that the user perceives the threads or tasks as running at the same time. On a multiprocessor or multi-core system, the threads or tasks will generally run at the same time, with each processor or core running a particular thread or task. [2]

3. Proposed System

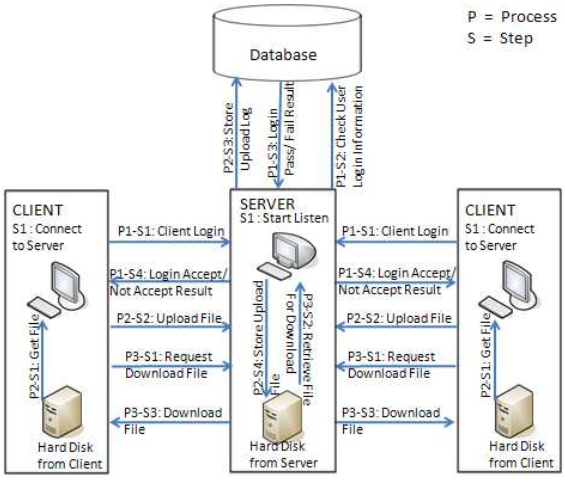

Figure 2: Overview of the System

The proposed is developed in two different applications: server-side application and client-side application. Server-side application is to use on the Server of the University. Server-side application is only available for administrator who has full authority to access the server-side application. Full authorize access such as: upload/ download files, manage files, and manage users.

In client-side application, users can upload or download files. There are 11 user types available for client-side application. Those are: 1st year CT, 1st year CS,2nd year CT, 2nd year CS, 3rd year CT, 3rd year CS, CT HONS, CS HONS, CT Master, CS Master and Teacher user types. 1st year CT users can only download files from CT folder and its sub

folders of 1st year CT folder. 1st year CS users can only download files from CS folder and its sub folders of 1st year CS folder. Similar accessible level to 2nd year CT, 2nd year CS, 3rd year CT, 3rd year CS, CT HONS, CS HONS, CT Master, CS Master with 1st year CT/CS users. For Teacher user type users, they are accessible to all folders and allow both uploading and downloading files. For the rest of the user types (students from CT/CS), they are allow to upload files at Knowledge Folder.

4. System Implementation

In this paper, the system used multithreading technique to control the uploading/ downloading files by multi-users. Without using a multi-threading technique, file transferring can only be done one at a time. It means, when a client is uploading files or downloading files from server, another client has to wait until the processes of first client is finished. When using multithreading technique, socket programming is using in the development of the technique. Therefore, server is allowed to talk with multi clients.

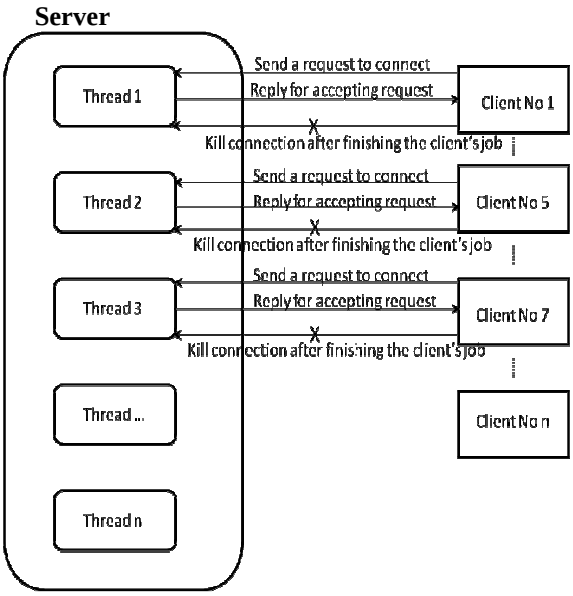

Figure 3: Multithreading and how it works

As soon as server is open and the server side program is running, threads are opened and ready to work with clients. When a client sends a request for connecting to the server, server checks the available threads and allow client to connect to an available thread. As shown in Figure 3, Client No 1 requests to the server first, therefore, the server allows Client No 1 to connect at Thread 1 and allows doing its jobs. Client No 7 is sending request to server after Client No 1, as a result, the server allows Client No 7 to connect at Thread 2 and allows doing its jobs and same apply to Client No 5.

When clients are finished to do their jobs and disconnect with the server, the server will kill the current connection with the client and restore as a new thread to wait for new request from another clients. For example, when Client No 7 disconnect with the server, Thread 2 kill the connection with Client No 7 and restore as a new thread for another clients. If Client No 6 is request to the server for connection while Client No 1 and No 5 are connected at Thread 1 and 3, and Thread 2 is open, the server will allow Client No 6 to connect at Thread 2 instead of allowing connecting at Thread 4.

4.1. File Uploading (client)

```
begin
    send upload file name to server
    while (wait for server command)
        accept server command
        if (server command=accept) then
            send upload file
        else
            show deny message
        end
    end
end
```

File Uploading (server)

```
begin
    while (wait for client request)
        accept upload file name from client
        if (accepted file name not in deny list)
         then
            accept upload file
            write to disk
        end
    end
end
```

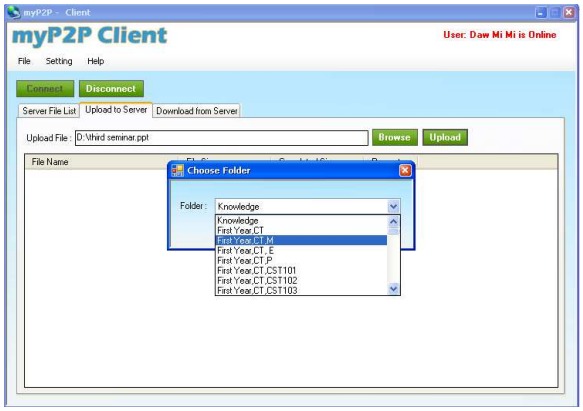

Figure 4: File Upload

4.2. File Downloading (client)

```
begin
        check download permission
        if (permission = true)
        begin
                send request file name to server
                while (wait for server command)
                        accept server command
                        if (server command = found) then
                            download requested file
                            write file to disk
                        else
                            show not found message
                        end
                end
        end
end
```

File Downloading (server)

```
begin
        while (wait for client request)
        begin
                accept request file name from client
                search requested file name in disk
                if (founded requested file name) then
                begin
                        send requested file to client
                end
        end
end
```

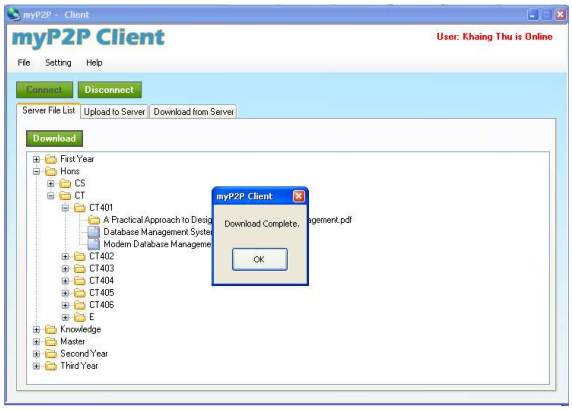

Figure 5: File Download

4.3. Deny EXE File Upload

EXE, INF, INI, BAT file types are prohibited to upload on the system to prevent the software virus as describe in Figure 6; unless, the EXE files are archived with ZIP or RAR to protect automatically run on the system. When users upload data from the client-side application, the uploaded files will be stored on central server.

```
begin
        accept upload file name from client
        check accepted file extension
        if (file extension = exe or ini or inf ) then
                send deny message to client
        else
                send accept message to client

        end
end
```

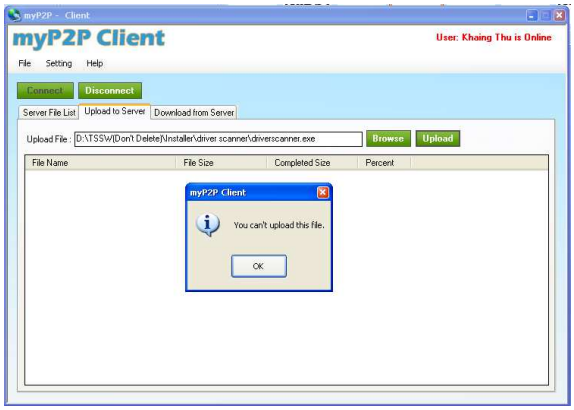

Figure 6: Deny EXE file upload

4.4. System Alert

The system will alert to users if new files are uploaded. For example, in Figure 7, if teachers upload new files under 1st year CT folder, the system will alert to the first year CT students that new files are uploaded.

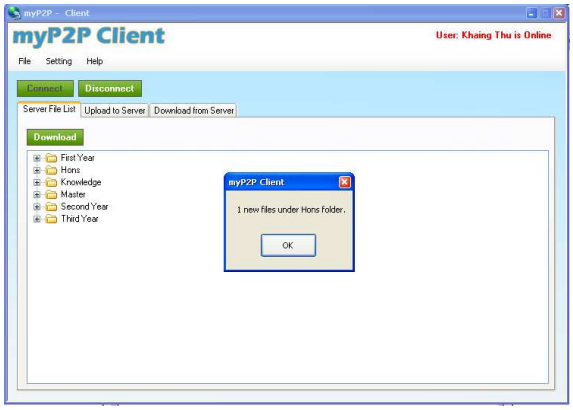

Figure 7: Alert for New Files upload

The system will also alert to the users file Hard Disk size is exceed to limited size as describe in Figure 8.

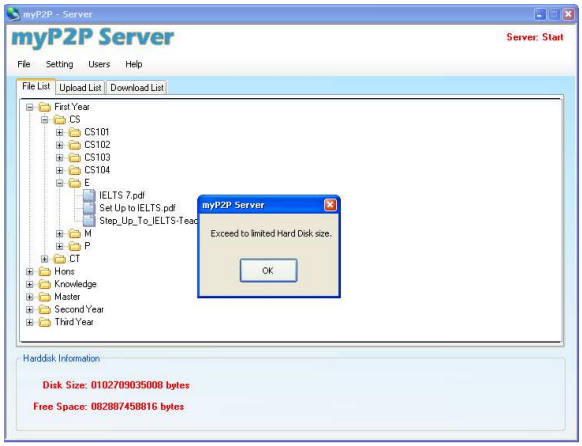

Figure 8: Alert for Limited Hard Disk size

5. Conclusion

Peer-to-Peer (P2P) file sharing software has brought a hot discussion on P2P file sharing among all businesses. The system is implemented based on centralized P2P system and it can support enhanced security of file sharing in centralized p2p system. The system is useful for both teachers and students whom can be run on Local Area Network (LAN) with low costing rate. The proposed system is being focused on usefulness and cost effectiveness for university learning system.

6. References

[1] John F. Buford, Heather Yu, Eng Keong Lua, *P2P Networking and Applications.*
ISBN 30-12374-214-5

[2] *Manning Publications. .NET Multithreading, 2003.*
ISBN 9781930110540 (1930110545)

[3] Ramesh Subramanian and Brian Goodman (eds), *Peer-to-Peer Computing: Evolution of a Disruptive Technology*, ISBN 1-59140-429-0

[4] http://www.wikipedia.org
http://en.wikipedia.org/wiki/Peer-to-peer
Retrieved on: 2010-04-01

[5] http://www.webopedia.com
http://www.webopedia.com/TERM/C/courseware.html
Retrieved on: 2010-02-07