

Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique

Université de Carthage



Rapport du projet :

Implémentation d'un Système distribué de partage de fichiers

Encadrant : Mr KHALGUI

Réalisé par :

Fayadhi Aymen

Fourati Aymen

Année universitaire : 2022 - 2023

Plan

1. **Introduction**
2. **Première Partie** : *Présentation du projet*
 - 2.1. Gnutella : *Système de référence.*
 - 2.2. Architecture P2P adoptée.
3. **Deuxième Partie** : *Développement d'un système distribué de partage de fichier*
 - 3.1. Exécution de l'application.
4. **Conclusion** : *Les pistes d'améliorations identifiées*

Ce projet est une implémentation d'un système de partage de fichiers peer-to-peer (P2P) inspiré par l'exemple de Gnutella ou un peer agit à la fois comme client et serveur à travers sa fonction de recherche et de partage de fichiers.

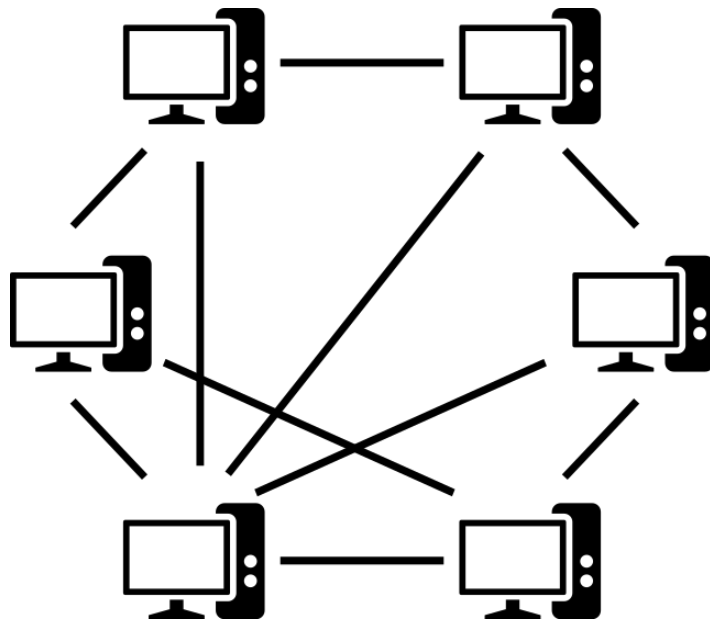
2.1 Gnutella : Système de référence

Diagram illustrating a query distribution in a peer-to-peer network. A central node (top) receives a 'Query' from a node on the right and sends a 'QueryHit' back. The central node then sends 'Query' messages to three other nodes (bottom-left, bottom-center, and bottom-right). The bottom-left node sends a 'QueryHit' back to the central node. A red curved arrow labeled 'File transfer: HTTP' connects the central node to the bottom-left node.

Nous avons jugé que l'architecture Gnutella était à plus approprié à notre use case de partage de fichier puisqu'elle est très simple a implémenter. En effet, le network design de ce système distribué est très souple. Avec un minimum de deux appareils dans un réseau local nous pouvons effectuer la recherche et le partage de fichiers.

2.2 Architecture P2P adopté :

Nous avons mis en œuvre la programmation java socket et les concepts de multithreading. Lorsque un peer rejoint le réseau, un thread serveur relatif à ce dernier est automatiquement instancié s'exécutant en arrière plan. Le fichier de configuration donné est lu pour trouver son homologue et lance un thread client pour chaque voisin. Lorsqu'un voisin est en ligne, une connexion est établie. Cette connexion est utilisée pour rechercher le fichier dans le réseau.



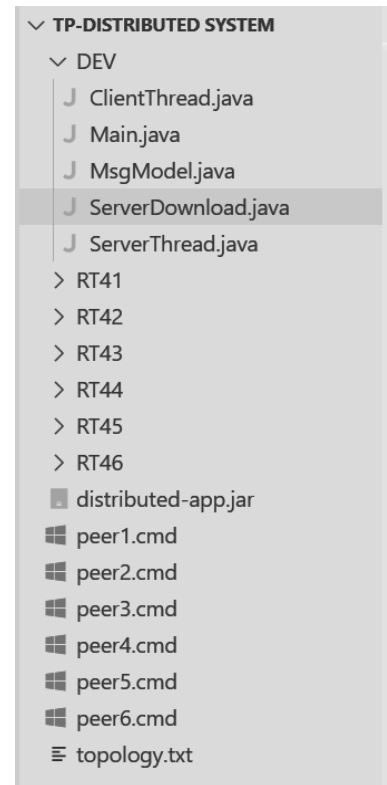
Deuxième Partie : Développement d'un système distribué de partage de fichier

2.1 Exécution de l'application :

Par contrainte de temps, nous n'avons pas pu implémenter notre système sur un réseau réel donc nous allons configurer une petite démonstration sur une seule machine avec une simulation d'un réseau à travers plusieurs terminaux.

Nous présentons ci-dessous la conception de notre application. En effet, chaque peer aura une partie software qui contient le script de connexion au réseau ainsi que les modules de recherche et partage de fichiers et une partie fichier qui contient les dossiers de fichiers partagés. La partie DEV est le code source de l'application, elle se charge de la mise en connexion des peers par programmation java sockets.

Dans notre cas, nous avons prévu 6 dossiers de partages, partageons métaphoriquement la RT4 en six nœuds sur le même réseau.



```
1  peer1.port=4000
2  peer2.port=4001
3  peer3.port=4002
4  peer4.port=4003
5  peer5.port=4004
6  peer6.port=4005
7
8
9  peer1.next=2
10 peer2.next=3
11 peer3.next=4
12 peer4.next=5
13 peer5.next=6
14 peer6.next=1
15
16 peer1.serverport=50000
17 peer2.serverport=50001
18 peer3.serverport=50002
19 peer4.serverport=50003
20 peer5.serverport=50004
21 peer6.serverport=50005
```

Le fichier Topology.txt va dicter la topologie P2P adoptée ce qui aura des conséquences sur la performance.

Ci-contre, nous présentons le contenu de ce fichier : Chaque peer a deux ports: un port local et un port serveur. De plus, chaque nœud est lié par un autre nœud appelé nœud suivant nous donnant une architecture en anneau.

- l'objet message utiliser dans le processus de communication entre les peers

```
import java.io.Serializable;

public class MessageFormat implements Serializable {

    String fname;
    String msgId;
    int fromPeerId;
    int TTL_value;
}
```

- Instantiation du thread client chez un peer :

```
public void run()
{
    try{

        socket=new Socket("localhost",portofconnection);

        OutputStream os=socket.getOutputStream();
        ObjectOutputStream oos=new ObjectOutputStream(os);
        InputStream is=socket.getInputStream();
        ObjectInputStream ois=new ObjectInputStream(is);
        MF.fname=filetodownload;

        MF.msgId=msgid;
        MF.fromPeerId=frompeer_id;
        MF.TTL_value=TTL_value;
        oos.writeObject(MF);

        peersArray=(int[])ois.readObject();

    }
}
```

- **Instantiation du thread serveur :**

```

        try{
            socket=serverSocket.accept();
            System.out.println("Connected to client at
"+socket.getRemoteSocketAddress()+" with peer "+peer_id);
            new
Download(socket,FileDir,peer_id,msg).start();
        }

```

- **Transfert d'un fichier vers le fichier tampon pour être transféré :**

```

{
    File myFile = new
File(sharedDirectory+"//"+filename);
    long length = myFile.length();
    byte [] mybytearray = new byte[(int)length];
    //Sending file length of the file to be downloaded to the client
    oos.writeObject((int)myFile.length());
    oos.flush();
    FileInputStream fileInSt=new
FileInputStream(myFile);
    BufferedInputStream objBufInStream = new
BufferedInputStream(fileInSt);

    //transferring the contents of the file as stream of bytes

objBufInStream.read(mybytearray,0,(int)myFile.length());
    System.out.println("sending file of "
+mybytearray.length+ " bytes");
    oos.write(mybytearray,0,mybytearray.length);
    oos.flush();
}

```

- **Connexion d'un peer au réseau :**

```
java -jar distributed-app.jar topology.txt *PeerID* *Path_Shared_Directoy*
```

Exemple:

```
java -jar distributed-app.jar topology.txt 1 "C:\Users\Aymen.Fayadhi\Desktop\Distributed System\TP-Distributed System\RT41"
```

- Initialization de 6 peers dans la simulation de réseau :

```

C:\WINDOWS\system32\cmd. x + v
initializing nodes...
Peer 1 has joined the network.
Linking media on : C:\Users\Aymen.Fayadhi\Desktop\Distributed System\TP-Distributed System\RT41
Adopted topology : topology.txt

-----

File downloader (1)

-----

>>

C:\WINDOWS\system32\cmd. x + v
initializing nodes...
Peer 2 has joined the network.
Linking media on : C:\Users\Aymen.Fayadhi\Desktop\Distributed System\TP-Distributed System\RT42
Adopted topology : topology.txt

-----

File downloader (1)

-----

>>

C:\WINDOWS\system32\cmd. x + v
initializing nodes...
Peer 3 has joined the network.
Linking media on : C:\Users\Aymen.Fayadhi\Desktop\Distributed System\TP-Distributed System\RT43
Adopted topology : topology.txt

-----

File downloader (1)

-----

>>

C:\WINDOWS\system32\cmd. x + v
initializing nodes...
Peer 4 has joined the network.
Linking media on : C:\Users\Aymen.Fayadhi\Desktop\Distributed System\TP-Distributed System\RT44
Adopted topology : topology.txt

-----

File downloader (1)

-----

>>

C:\WINDOWS\system32\cmd. x + v
initializing nodes...
Peer 5 has joined the network.
Linking media on : C:\Users\Aymen.Fayadhi\Desktop\Distributed System\TP-Distributed System\RT45
Adopted topology : topology.txt

-----

File downloader (1)

-----

>>

C:\WINDOWS\system32\cmd. x + v
initializing nodes...
Peer 6 has joined the network.
Linking media on : C:\Users\Aymen.Fayadhi\Desktop\Distributed System\TP-Distributed System\RT46
Adopted topology : topology.txt

-----

File downloader (1)

-----

>>

```

- Recherche d'un fichier à partir du peer 1 dans le peer 6 et réponse du peer 6:

```

C:\WINDOWS\system32\cmd. x + v
1
Search :
ML2017.txt
Connected to client at /127.0.0.1:49691 with peer 1
server thread for peer1
got request from 6
Found: ML2017.txt
Local Search Completed
sending to 2
Peers containing the file are:
6
Enter the peer from where to download the file:

C:\WINDOWS\system32\cmd. x + v
>>
Connected to client at /127.0.0.1:49688 with peer 4
server thread for peer4
got request from 3
Found: ML2017.txt
Local Search Completed
sending to 5

C:\WINDOWS\system32\cmd. x + v
Connected to client at /127.0.0.1:49689 with peer 5
server thread for peer5
got request from 4
Found: ML2017.txt
Local Search Completed
sending to 6

C:\WINDOWS\system32\cmd. x + v
File downloader (1)

-----

>>
Connected to client at /127.0.0.1:49690 with peer 6
server thread for peer6
got request from 5
Found: ML2017.txt
Local Search Completed
sending to 1

C:\WINDOWS\system32\cmd. x + v
Local Search Completed
sending to 3
Connected to client at /127.0.0.1:49692 with peer 2
server thread for peer2
got request from 1
duplicate
Found: ML2017.txt

C:\WINDOWS\system32\cmd. x + v
File downloader (1)

-----

>
connected to client at /127.0.0.1:49687 with peer 3
server thread for peer3
got request from 2
Found: ML2017.txt
Local Search Completed
sending to 4

```


- **Téléchargement du fichier :**

```
C:\WINDOWS\system32\cmd. x + v
got request from 6
Found: ML2017.txt
Local Search Completed
sending to 2
Peers containing the file are:
6
Enter the peer from where to download the file:
6
bytes transferred: 36
ML2017.txt Successfully downloaded ! C:\Users\Aymen.Fayadhi\Desktop\Distribut
ed System\TP-Distributed System\RT41
File: ML2017.txt downloaded from Peer 6 to Peer 1

C:\WINDOWS\system32\cmd. x + v
>>
Connected to client at /127.0.0.1:49688 with peer 4
server thread for peer4
got request from 3
Found: ML2017.txt
Local Search Completed
sending to 5

C:\WINDOWS\system32\cmd. x + v
Connected to client at /127.0.0.1:49692 with peer 2
server thread for peer2
got request from 1
duplicate
Found: ML2017.txt

C:\WINDOWS\system32\cmd. x + v
File downloader (1)

>
Connected to client at /127.0.0.1:49687 with peer 3
server thread for peer3
got request from 2
Found: ML2017.txt
Local Search Completed
sending to 4

C:\WINDOWS\system32\cmd. x + v
sending file of 36 bytes
sending file of 36 bytes
sending file of 36 bytes
sending file of 36 bytes
sending file of 36 bytes
sending file of 36 bytes
sending file of 36 bytes
sending file of 36 bytes
sending file of 36 bytes
sending file of 36 bytes
sending file of 36 bytes
sending file of 36 bytes
```

Conclusion : *Les pistes d'améliorations identifiées :*

Bien que notre application ne soit loin d'être achevée, nous avons bien réussi à simuler le fonctionnement d'un système distribué de partage de fichier sur un réseau local a six peer par une topologie en anneau. Cela dit, nous considérons essentiel de pouvoir passer à une implémentation sur un réseau au lieu de l'utilisation du localhost sur une seule machine. De plus, il est impératif d'ajouter une interface graphique utilisateur pour simplifier le processus de communication.