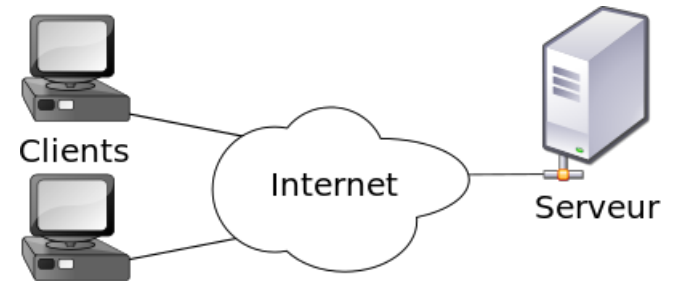


Transfert de données dans les architectures **pair-à-pair**

L'architecture client/serveur

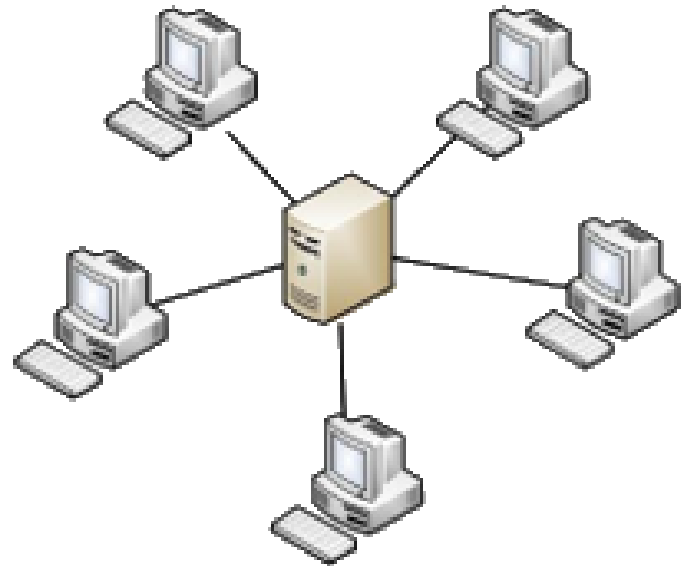
L'architecture client/serveur désigne un mode de communication entre plusieurs composants d'un réseau.

- Chaque entité est considérée comme un client ou un serveur.
- Chaque logiciel client peut envoyer des requêtes à un serveur. Un serveur peut être spécialisé en serveur d'applications, de fichiers, de terminaux, ou encore de messagerie électronique.



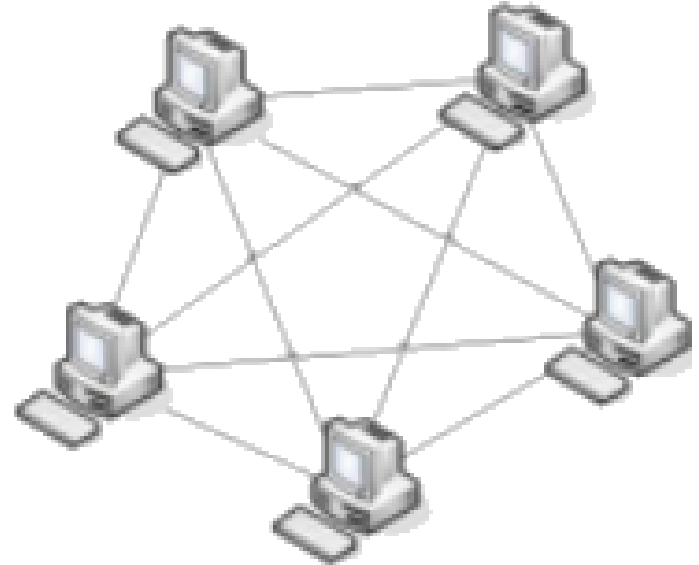
Architecture Centralisée

- **Client / Serveur**
- **Données stockées centralement**
- **Client se connecte au serveur**
 - Pour envoyer et recevoir des données
 - Pour communiquer avec autre clients
 - Pas de communication direct entre les clients
- **Serveur en panne**
 - Plus de données pour les clients
- **Nombre clients augmente**
 - Ressource nécessaires du serveur augmentent
- **Exemple: Serveur web**



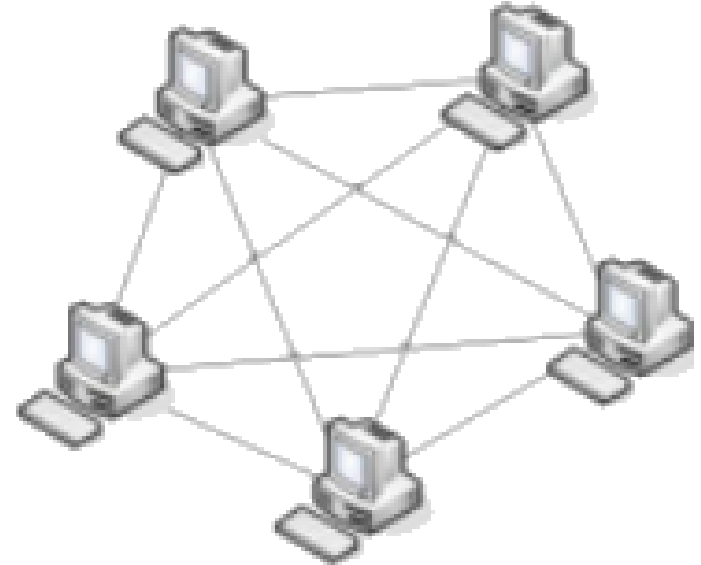
Architecture Décentralisée

- **Peer est client et/ou Serveur**
- **Pas de serveur central**
- **Communication direct entre peers**
- **Responsabilités partagées**
- **Tolérant aux fautes**



Définition (Peer-to-Peer):

- Le terme **pair à pair** (P2P) désigne un **modèle distribué** où les entités appelées **pairs** jouent le double rôle de **client et serveur** et interagissent afin d'offrir à une communauté un **service décentralisé**.
- Chaque nœud **paye** sa participation en donnant accès à une partie de ses ressources.



Définition (Peer-to-Peer):

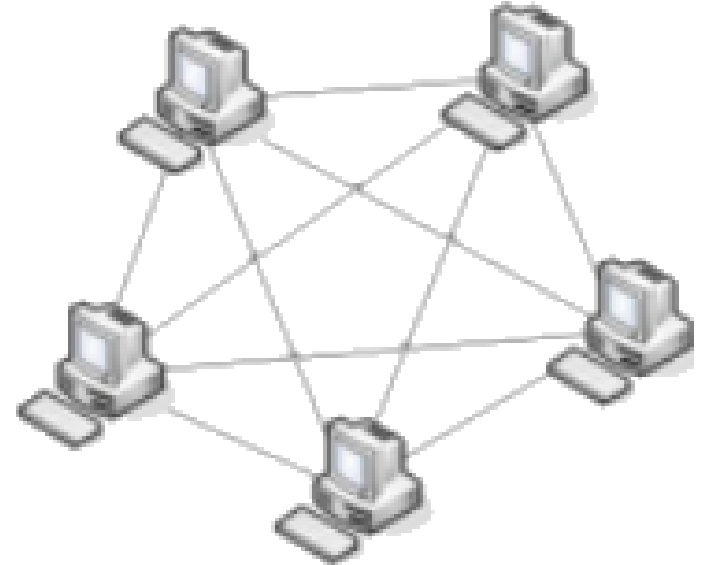
peer

noun [C] • **UK**  /piəʔ/ **US**  /pɪr/

peer *noun* [C] (EQUAL)

★ **C1** a person who is the same age or has the same social position or the same abilities as other people in a group:

- Un **processeur**, une **application**, une **machine**
- **Rôle**: peut être un **client** et/ou **serveur**
- **Client**: demande un **service**
- **Serveur**: **offre** un service, **exécute** un service



Propriétés (Peer-to-Peer):

- Pas de coordination centralisée
- Pas de base de données centralisées
- Aucun nœud n'a une **vision globale** du système
- Tous les services et données sont **accessibles** depuis n'importe quel nœud
- Les nœuds sont **autonomes**
- Les nœuds et connexions sont **non fiables**



Classes d'applications

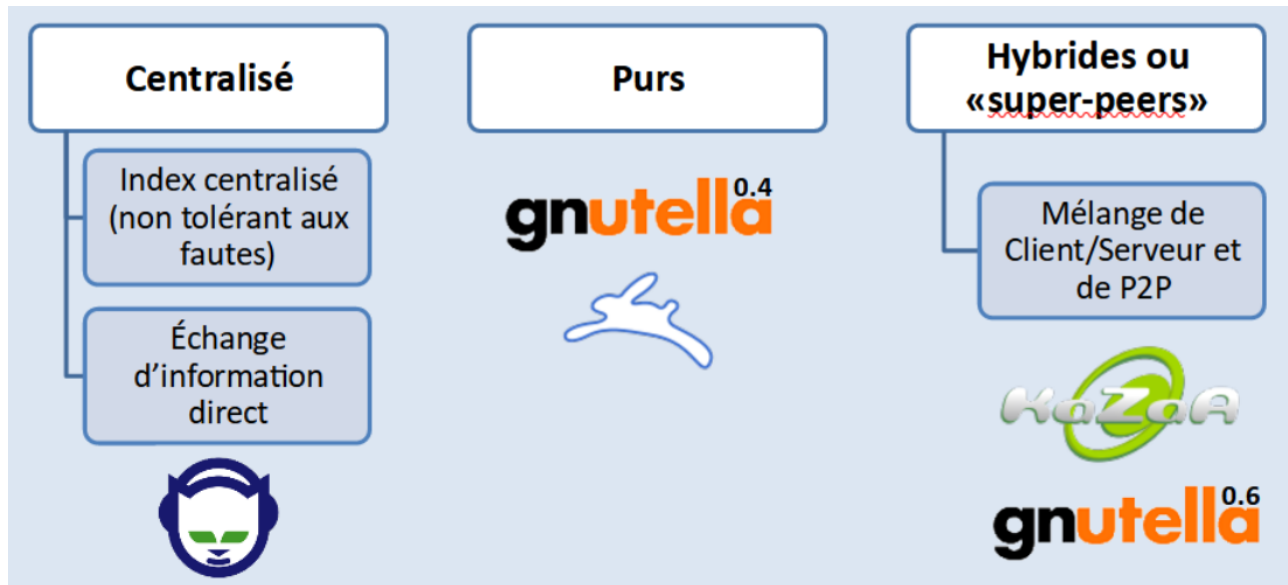
- **Partage de fichiers: Napster, Gnutella, KaZaa....**
- **La messagerie instantanée avec ICQ, AIM.**
- **GridComputing: Seti@home, Folding@home, Enigma@home.**
- **Streaming: Peercast, Skype.**
- **Jeux en réseaux.**

Fonctionnalités d'un système (Peer-to-Peer) :

- **Découverte de ressources**
- **Gestion des mises-à-jour**
- **Passage à l'échelle**
- **Tolérance aux fautes**
- **Sécurité**

Classes de systèmes (Peer-to-Peer) :

-





- **Napster** a été lancé en juin 1999 par Shawn Fanning comme service de partage de fichiers musicaux au format MP3.

La facilité de trouver et de télécharger des fichiers musicaux a rapidement rendu Napster très populaire.

Les réseaux à haute vitesse dans les dortoirs de collèges sont devenus surchargés, Napster monopolisant jusqu'à 60 % du trafic de ces réseaux.

À son apogée, le service Napster avait environ 80 millions d'utilisateurs enregistrés.

Napster



Napster v2.0 BETA 7

File Actions Help

Home Chat Library Search Hot List Transfer Discover Help

Filename	File Size	User	Status	Speed	Progress	Rate	Time Left
Comedy - Maclean & Maclean - The Gross Manual [14 Sure Fire Jo...	299,200 of 2,220,560	ldhull	Downloading...	Unknown	13%	5.06 k/s	00:06:19
(Comedy) Maclean & Maclean - Johnny Marijuana Seed.mp3	270,048 of 7,381,800	mrbm	Downloading...	Cable	3%	6.50 k/s	00:18:13
Comedy - Maclean & Maclean - The Gross Manual [14 Sure Fire Jo...	539,936 of 2,220,560	big_bang_	Downloading...	Cable	24%	7.33 k/s	00:03:49
MacLean and Maclean - Bowser and Blue.mp3	5,606,005	adie_m	Getting Info...				
Andrews Sisters - Little Toot.mp3	375,392 of 8,528,129	F4phantom...	Downloading...	DSL	4%	4.14 k/s	00:32:50
78's CHILDREN'S STORY RECORD - LITTLE TOOT.mp3	89,512 of 5,840,561	dude_way	Downloading...	14.4	1%	1.10 k/s	01:26:59
Disney Parks Toontown-Little Toot.mp3	64,608 of 2,354,597	hitteerblitter	Downloading...	Cable	2%	1.28 k/s	00:29:47
Maclean & Maclean - Little Toot.MP3	12,288 of 602,112	KASDJFLJ...	Downloading...	Cable	3%	0.24 k/s	00:41:39
Disney Parks Toontown-Little Toot.mp3	185,648 of 2,354,597	Creano1	Downloading...	56K	9%	5.82 k/s	00:06:12
Maclean & Maclean - Little Toot.mp3	210,944 of 882,688	stearch	Downloading...	56K	24%	4.44 k/s	00:02:31
Big Tiny Little & Mickey Finn - Toot Toot Tooti.mp3	40,128 of 2,888,844	abear69	Downloading...	Cable	1%	1.89 k/s	00:25:08
Comedy - Maclean & Maclean - Little Toot.mp3	6,816 of 882,688	catlins2136	Downloading...	DSL	0%	0.95 k/s	00:26:45
Brenda Lee - Toot Toot Tootsie Goodbye (Little Miss Dynamite, CD...	44,368 of 2,250,710	angel3339	Downloading...	Unknown	1%	1.10 k/s	00:33:26
MacLean & MacLean - Little Toot (ed1).mp3	115,600 of 1,180,444	123412341...	Downloading...	14.4	9%	4.68 k/s	00:03:47
Maclean & Maclean - Little Toot.mp3	882,688	raybauduc	Getting Info...				

Filename	File Size	User	Status	Speed	Progress	Rate	Time Left
Crosby, Stills, Nash And Young - Ohio.Mp3	2,725,888 of 3,709,496	cybemoose	Uploading...	56K	73%	1.55 k/s	00:10:33
lolipop lust kill - jesus chrysler.mp3	3,339,493	4stringMF	File Complete				
Simpsons - The Life Of Homer Simpson.mp3	1,406,976 of 2,756,736	astude67	Uploading...	56K	51%	1.45 k/s	00:15:30
Fear Factory - Cars.mp3	3,921,141	tubbedbid	File Complete				
Comedy - Richard Pryor - New Niggers.mp3	3,796,451	tubbedbid	File Complete				
Unknown - Mopar (Chrysler) 440 6-pak V8 race-buit - real men don't...	869,049	TYLERSEX...	File Complete				
Comedy - Jerky Boys - Fanning My Balls.mp3	675,840 of 1,784,264	tubbedbid	Uploading...	14.4	38%	7.33 k/s	00:02:31
Ben Stiller & Andy Dick - Can We Get It Together Again.mp3	1,230,600	fluidpockets...	Waiting...				

Concurrent Downloads: 25 Concurrent Uploads: 1

Clear Finished Cancel

Online (fgjkdtdvbfvcv): Sharing 2703 files. Currently 526,589 files (2,246 gigabytes) available in 8,411 libraries.

Napster



Client

Se connecter à un serveur Napster

Télécharger la liste des fichiers musicaux

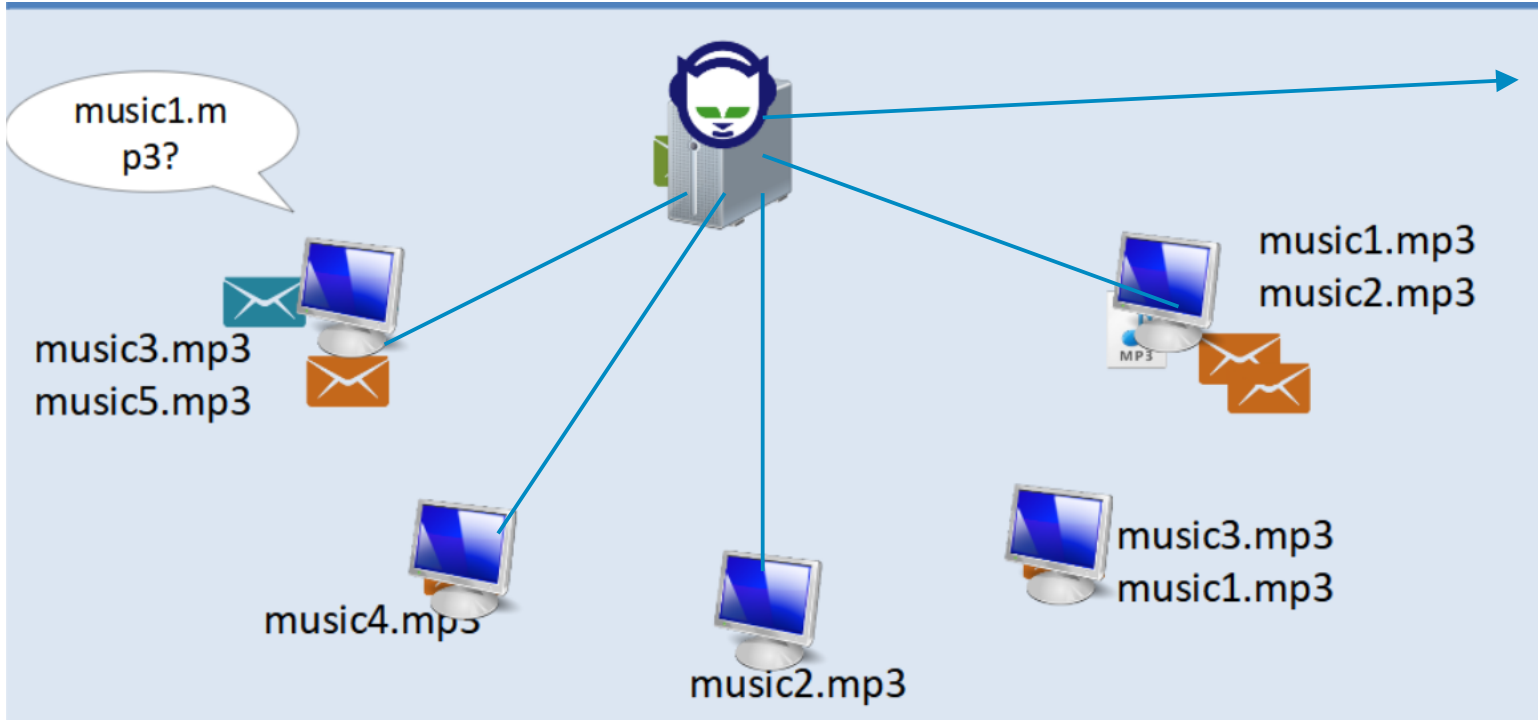
Le serveur conserve la liste <filename,ip_address, portnum>



Client recherche

- 1-envoyer des mots-clés au serveur
- 2-Le serveur effectue une recherche par mots-clés dans sa liste.
- 3-Le serveur envoie une liste des hôtes.
- 4-Le client ping chaque host dans la liste.
- 5-Le client télécharge le fichier depuis le meilleur (ping) hôte

Fonctionnement



Filename	info
Music1.mp3	@192.168.2.3

Déboires judiciaires

- La facilité de télécharger des chansons individuelles avec Napster et les services ultérieurs est souvent cité comme la cause de la fin de l'ère des albums en musique populaire.
- Le logiciel est retiré en 2001, après 2 ans de procédure judiciaire aux États-Unis pour infraction à la législation sur le droit d'auteur.
- aujourd'hui, Napster devient service musical payant.

Bilan «purs»

- **Complètement décentralisé**
- **Très tolérant aux fautes**
- **S'adapte bien à la dynamique du réseau**
- **Simple, robuste et passe à l'échelle (pour le moment)**
- **Gros consommateur de bande passante**
- **Pas de garantie de succès, ni d'estimation de la durée des requêtes.**
- **Pas de sécurité.**

Bilan «super-peers»

- **Client/Serveur + P2P**
- **Éviter les problèmes dus à l'hétérogénéité de la bande passante des nœuds.**
- **Tous les nœuds ne sont plus égaux:**
 - Nœuds avec une bonne bande passante sont organisés en P2P: les super-peers.
 - Nœuds avec une faible bande passante sont rattachés en mode client/serveur à super-peers (Cluster)
 - Super-peers disposent d'un index des ressources de leur cluster.
- **Utilisé dans KaZaa.**

gnutella

gnutella

- C'était le premier réseau décentralisé peer-to-peer de son genre. a été imaginé en 2000 par Tom Pepper et Justin Frankel.



Fonctionnement Version 0.4

- chaque poste agit comme un serveur et un client et est nommé « **servent** » (contraction de **serveur** et **client**).
- Pour lancer une recherche, un servent interroge tous ses voisins en leur envoyant un message de recherche (Query). Ses voisins font de même avec leur propre voisins.
- Un champ **TTL** est associé au message de recherche pour comptabiliser le nombre de retransmissions restantes. Quand celle-ci est nulle, le message n'est plus renvoyé. (**inondation**).



Fonctionnement Version 0.4

- Les serveurs ayant des fichiers qui répondent à la requête renvoient leur réponse (QueryHit)(nom du fichier + leur adresse IP) au voisin qui leur a retransmis la requête. La réponse remonte ainsi de proche en proche jusqu'au serveur qui a initié la requête.
- Le serveur initiateur de la requête va ensuite choisir les fichiers à télécharger en envoyant directement une requête de téléchargement au serveur qui possède le fichier.



Ultrapeers

Fonctionnement Version 0.6

FastTrack System

- Les peers sont divisés en deux catégories :
- 1-les ultrapeers (ou **superpeer**) qui sont des serveurs **stables** ayant une bonne **connectivité**
- 2-les clients (**leaf**) sont rattachées à 3 ultrapeers et les ultrapeers admettent entre 30 à 45 clients et 30 ultrapeers.



Le protocole BitTorrent

Généralités

- Protocole de type "Peer to Peer".
- Inventé par Bram Cohen en 2001.
- **Objectif**
Faciliter la distribution de gros fichiers en diminuant la charge sur les serveurs.
- **Succès croissant**
Nombreuses organisations proposent des téléchargements via ce protocole (RedHat, Eclipse ...)

Glossaire

Torrent

Torrent : Extension du fichier contenant les informations sur les données à télécharger.

Ex: (Filename.**torrent**)



Glossaire

Seeder

Seeder : Client ne faisant que d'uploader (possède tout le fichier).

Glossaire

Leecher

Leecher : Client désirant télécharger des parties (ne possède pas l'intégralité des données)

Glossaire

Tracker

Tracker : Chef d'orchestre du téléchargement. Contient les informations concernant les clients

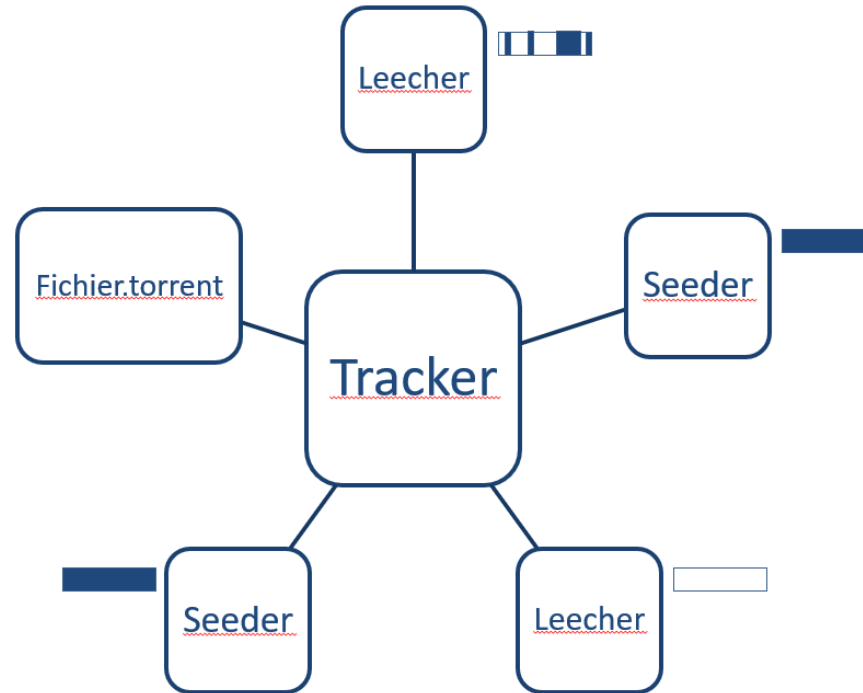
Fonctionnement

- Pas de système de recherche;
- Récupération d'un fichier d'informations Torrent;
- Téléchargement d'un ensemble de fichiers;
- Chaque client contient une ou toutes les parties des données à télécharger.

Fonctionnement

Architecture

-



Encodage du fichier Torrent

BEncoding

est un **encodage** utilisé par BitTorrent pour stocker et transmettre des données structurées.

Il supporte 4 différentes types de valeurs:

(String, Integer, List, Dictionarie)

Encodage du fichier Torrent

BEncoding

String

Description: Normal Strings [series of continuous characters]

Format: <length>:<data>

Exemple: 7:network

Encodage du fichier Torrent

BEncoding

Integer

Description: Normal integers

Format: i<integer>e

Exemple: i2017e ; i5e

Encodage du fichier Torrent

BEncoding

List

Description: They are lists of types [strings, integers, lists, dictionaries].

Format: l<contents>e // Contents are bencoded.

Exemple: l 8:advanced7:network e

Encodage du fichier Torrent

BEncoding

Dictionaries

Description: They are a mapping of keys to values

Format d<keys><values>e

Contents are bencoded with no separators.

Exemple: d3:onei1e3:twoi2e5:threei3e4:four4ee

Structure d'un fichier Torrent

BEncoding

.torrent

Contient les informations concernant les fichiers et le Tracker:

announce : URL du Tracker

info : informations sur les fichiers

name : nom du fichier torrent

files : description des fichiers

length : taille d'un fichier

path : chemin d'un fichier

piece length : taille d'une partie

pieces : chaîne de taille multiple de 20 (SHA1 de chaque parties)

Structure d'un fichier Torrent

BEncoding

.torrent

Key	Description
Info	A dictionary that describes the files
-length	Length of file in bytes (integer)
-md5sum(optional)	A 32 character hexadecimal string corresponding to the MD5 sum of the file.
-name	The filename of a string(string)
-piece length	Number of bytes in each piece (integer), commonly $2^{18} = 256\text{KB}$
-pieces	String consisting of the concatenation of all 20-byte SHA1 hash values, one per piece.(raw binary encoded)
Announce	The announce URL of the tracker
Announce-list (optional)	This is an extension to the official specification, which is also backwards compatible. This key is used to implement lists of backup trackers.
Creation date (optional)	The creation time of the torrent, in standard Unix epoch format (integer seconds since 1-Jan-1970 00:00:00 UTC)
Comment (optional)	Free form text comments.(string)
Created by (optional)	Name and version of the program used to create.

d8:announce35:https://torrent.ubuntu.com/announce13:announce-list135:https://
torrent.ubuntu.com/announcee140:https://ipv6.torrent.ubuntu.com/announceee7:comment29:Ubuntu
CD releases.ubuntu.com10:created by13:mktorrent 1.113:creation
datei1634219565e4:infod6:lengthi3116482560e4:name30:ubuntu-21.10-desktop-amd64.iso12:piece
lengthi262144e6:pieces237780:

announce	https://torrent.ubuntu.com/announce
announce-list	https://torrent.ubuntu.com/announce https://ipv6.torrent.ubuntu.com/announce
comment	Ubuntu CD releases.ubuntu.com
created by	mktorrent 1.1
creation date	1634219565
info	Length = 3116482560 Name = ubuntu-21.10-desktop-amd64.iso piece length = 262144 Pieces = 237780

d8:announce35:https://torrent.ubuntu.com/announce13:announce-list135:https://
torrent.ubuntu.com/announcee140:https://ipv6.torrent.ubuntu.com/announceee7:comment29:Ubuntu
CD releases.ubuntu.com10:created by13:mktorrent 1.113:creation
datei1634219565e4:infod6:lengthi3116482560e4:name30:ubuntu-21.10-desktop-amd64.iso12:piece
lengthi262144e6:pieces237780:

announce	https://torrent.ubuntu.com/announce
announce-list	https://torrent.ubuntu.com/announce https://ipv6.torrent.ubuntu.com/announce
comment	Ubuntu CD releases.ubuntu.com
created by	mktorrent 1.1
creation date	1634219565
info	Length = 3116482560 Name = ubuntu-21.10-desktop-amd64.iso piece length = 262144 Pieces = 237780

announce	https://torrent.ubuntu.com/announce
announce-list	https://torrent.ubuntu.com/announce https://ipv6.torrent.ubuntu.com/announce
comment	Ubuntu CD releases.ubuntu.com
created by	mktorrent 1.1
creation date	1634219565
info	Length = 3116482560 Name = ubuntu-21.10-desktop-amd64.iso piece length = 262144 Pieces = 237780

(Ex)

3116482560	=	3.11648256
Byte		Gigabyte

(Ex)

announce	https://torrent.ubuntu.com/announce
announce-list	https://torrent.ubuntu.com/announce https://ipv6.torrent.ubuntu.com/announce
comment	Ubuntu CD releases.ubuntu.com
created by	mktorrent 1.1
creation date	1634219565
info	

Format	Seconds
GMT	Thu Oct 14 2021 13:52:45 GMT+0000
Your Time Zone	Thu Oct 14 2021 14:52:45 GMT+0100 (GMT+01:00)
Relative	6 months ago

Enter a Date & Time

Year	Month	Day	Hour (24 hour)	Minutes	Seconds
<input type="text" value="2022"/>	<input type="text" value="04"/>	<input type="text" value="18"/>	<input type="text" value="04"/>	<input type="text" value="19"/>	<input type="text" value="45"/>

Connexion au Tracker

Tracker HTTP Protocol (Request)

Obtenir des informations concernant les clients connectés

Envoi d'une requête HTTP avec les paramètres :

peer_id : identifiant du client

info_hash : le code SHA1 du dictionnaire "info"

ip : adresse IP du client

port : port du client

SHA-1(hello)=aaf4c61ddcc5e8a2dabede0f3b482cd9aea9434d

SHA-1(Hello)=f7ff9e8b7bb2e09b70935a5d785e0cc5d9d0abf0

Réponse du Tracker

Tracker HTTP Protocol (Response)

Encodée au format BEncoding

Dictionnaire contenant :

interval : Intervalle à attendre avant de contacter le tracker

peers : liste des clients connectés

peer_id : identifiant d'un client

ip : adresse IP d'un client

port : port d'un client

Les clients sont renvoyés de manière aléatoire

Communication inter clients

Communication inter clients

Basé sur le protocole TCP

Notion d'état :

"Choked" : toutes les requêtes seront annulées

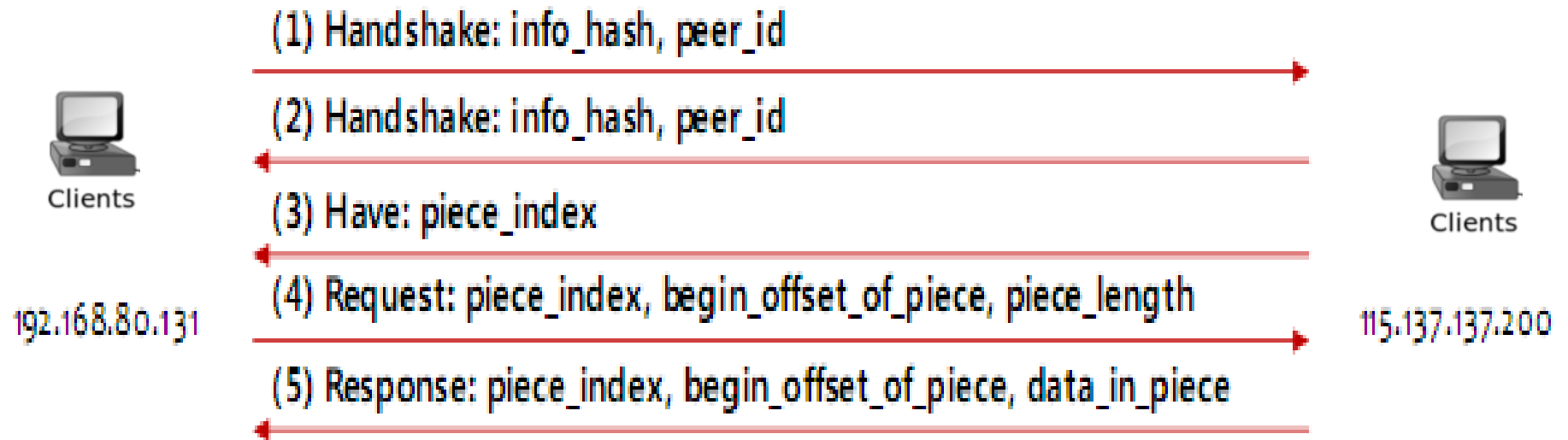
"Interested" : si le client est intéressé

Handshake

info_hash : Code SHA1 du dictionnaire info

peer_id : Identifiant du client (20 octets)

Communication inter clients



Algorithmes

Encouragement au téléchargement

Un client envoie en priorité aux clients chez qui il télécharge le plus rapidement.

Algorithmes

Rarest First

Envoi en priorité les parties les plus rares.

Avantages / Inconvénients

Avantages

- Pas de files d'attente
- Protocole très rapide
- Diminution de la charge des serveurs

Avantages / Inconvénients

Inconvénients

- Pas de fonction de recherche
- Pas de support des clients derrière un Firewall ou un réseau NAT
- Démarrage lent d'un téléchargement

	Client/Serveur	Peer to Peer
De base	Il y a un serveur spécifique et des clients spécifiques connectés au serveur.	Le client et le serveur font le même travail. chaque nœud agit en tant que client et serveur.
Service	Le client demande le service et le serveur offre le service.	Chaque nœud peut demander des services et peut également fournir des services.
La stabilité	Modèle Client-Serveur est plus	Peer-to Peer souffre si le nombre de pairs augmente dans le système.

La stabilité	Modèle Client-Serveur est plus stable et évolutif.	Peer-to Peer souffre si le nombre de pairs augmente dans le système.
Le coût	Le client-serveur est coûteux à implémenter.	Peer-to-peer sont moins chers à mettre en œuvre.
Coté Serveur	Lorsque plusieurs clients demandent les services simultanément, un serveur peut être encombré.	Comme les services sont fournis par plusieurs serveurs répartis dans le système peer-to-peer, un serveur n'est pas encombré.

Le coût	Le client-serveur est coûteux à implémenter.	Peer-to-peer sont moins chers à mettre en œuvre.
Coté Serveur	Lorsque plusieurs clients demandent les services simultanément, un serveur peut être encombré.	Comme les services sont fournis par plusieurs serveurs répartis dans le système peer-to-peer, un serveur n'est pas encombré.
Les données	Les données sont stockées dans un serveur centralisé.	Chaque pair a ses propres données.