Les applications pair-à-pair

Guillaume Roux

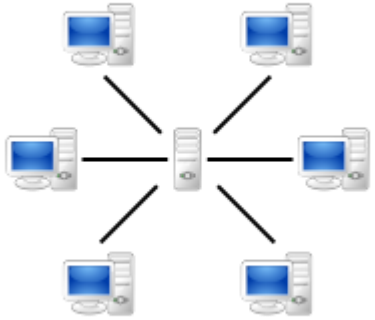
Sommaire

- Définition
- Architectures réseaux
- Les catégories de réseaux pair-à-pair
- Le réseau eDonkey
- Le protocole BitTorrent
- Bilan

Définition

Définition

- Pair-à-pair, peer to peer, P2P
- Plusieurs ordinateurs qui communiquent via un réseau
- On appelle *nœud* les postes connectés au réseau P2P
- Réseau dont les clients sont aussi les serveurs
- Chaque nœud nécessite un logiciel d'accès au réseau



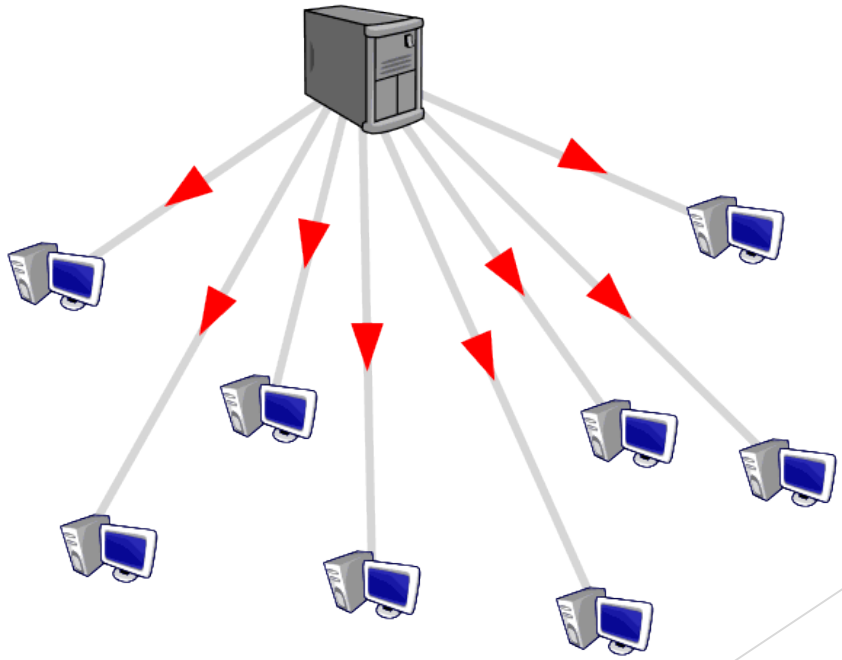
architecture client-serveur

architecture P2P

Architectures réseaux

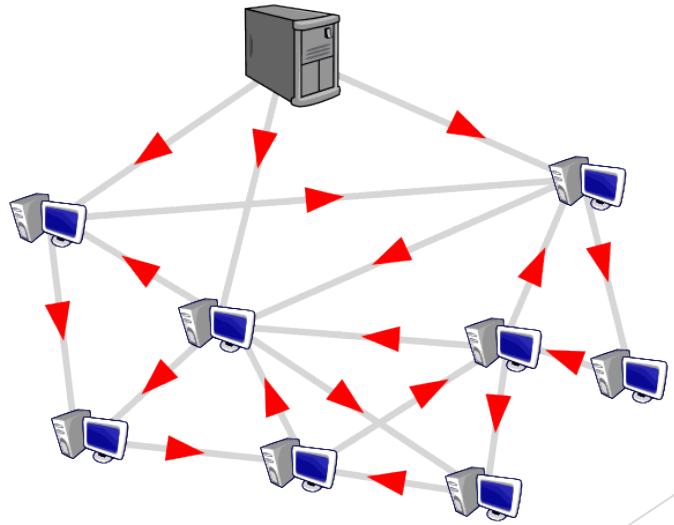
L’architecture client-serveur

- En client serveur, quand un fichier est populaire, tout le monde demande au serveur
- Coûteux en bande passante
- DOS en cas de forte demande



L’architecture P2P

- En client P2P, quand un fichier est populaire, tout le monde en dispose
- Le trafic est réparti sur tout les nœuds
- Très bon débit quand le fichier est connu
- Peu cher en infrastructure



L'architecture P2P
Les cas d'utilisation

- Le **partage de fichiers** avec eMule, BitTorrent
  - La **sauvegarde** avec BitTorrent Sync
- La **téléphonie sur internet** avec Skype
- La **messagerie instantanée** avec ICQ, AIM
- La **télévision** avec BitTorrent Live
- Les **moteurs de recherche** avec YaCy
- Le partage de **puissance de calcul** avec BOINC
  - En octobre 2013
    - 8,3 PFLOPS répartie sur 650 000 ordinateurs
    - Titan – Cray XK7 17,590 PFLOPS

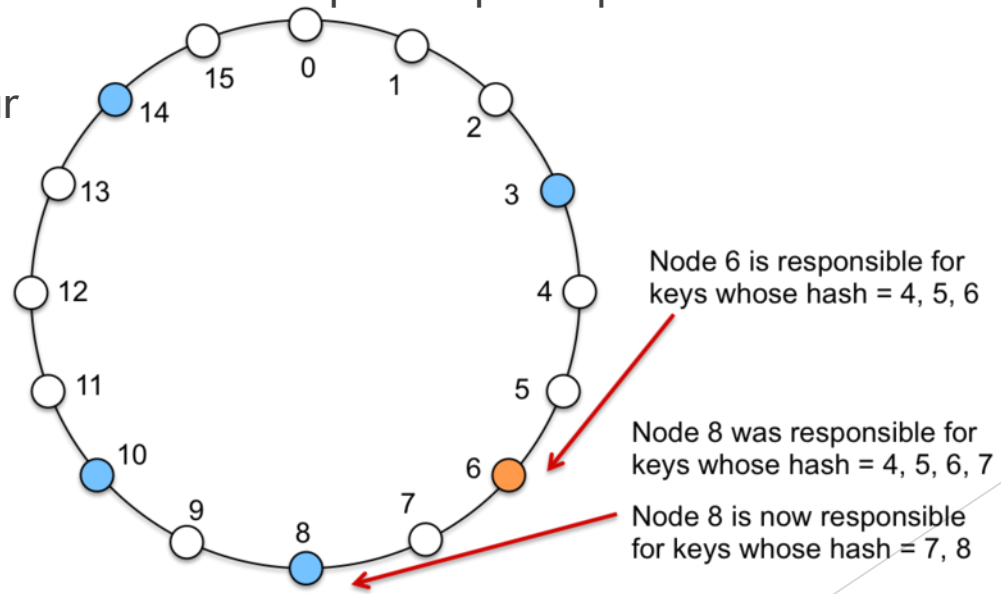
Les catégories de réseaux pair-à-pair

Les catégories de réseaux pair-à-pair

Structurés

- Basé sur l'établissement d'une DHT (Distributed Hash Table)
  - La DHT place les nouveau nœuds au sein du réseau
  - Chaque pair est responsable de l'indexation d'une partie spécifique du contenu du réseau
  - Identifié par un couple clé – valeur

- Quelques algorithmes structurés
  - CAN
  - Kademilia
  - Pastry



Les catégories de réseaux pair-à-pair

Non structurés

- Liens entre les nœuds sont établis de façon arbitraire
- Communication plus difficile à gérer
- Quelques protocoles
  - eDonkey
  - BitTorrent
  - Gnutella
  - Fastrack dont le client le plus connu est KaZaA

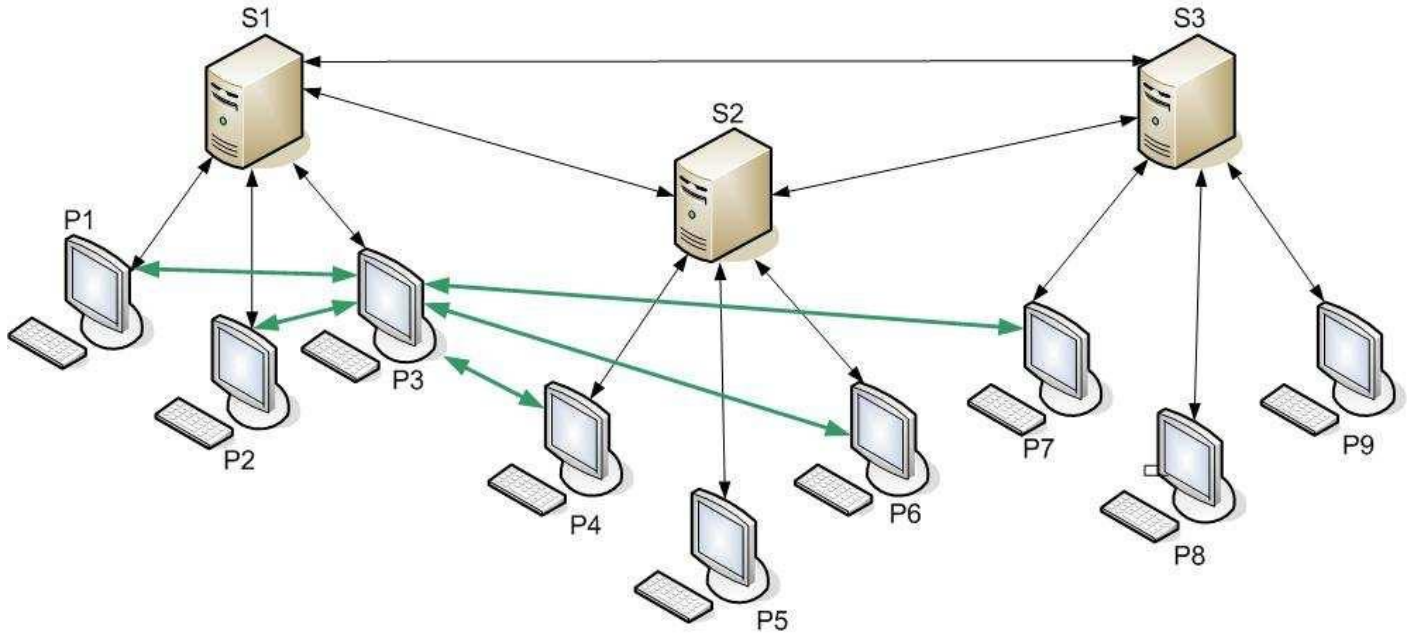
Le réseau eDonkey

Le réseau eDonkey

- Aussi connu sous le nom réseau eDonkey2000 ou eD2K
- Utilisé par eMule
- La partie serveur du réseau est Freeware mais les sources sont propriétaires
- Détrône FastTrack en 2004 puis se fait détrôner en 2009 par BitTorrent
- Fermeture du serveur Razorback2 le 21 février 2009

Etude du fonctionnement du réseau eDonkey

- P2P hybride
- Dispose de nombreux serveurs pouvant être considérés comme super-peers
- Partage des parties dont on dispose tout en continuant le téléchargement



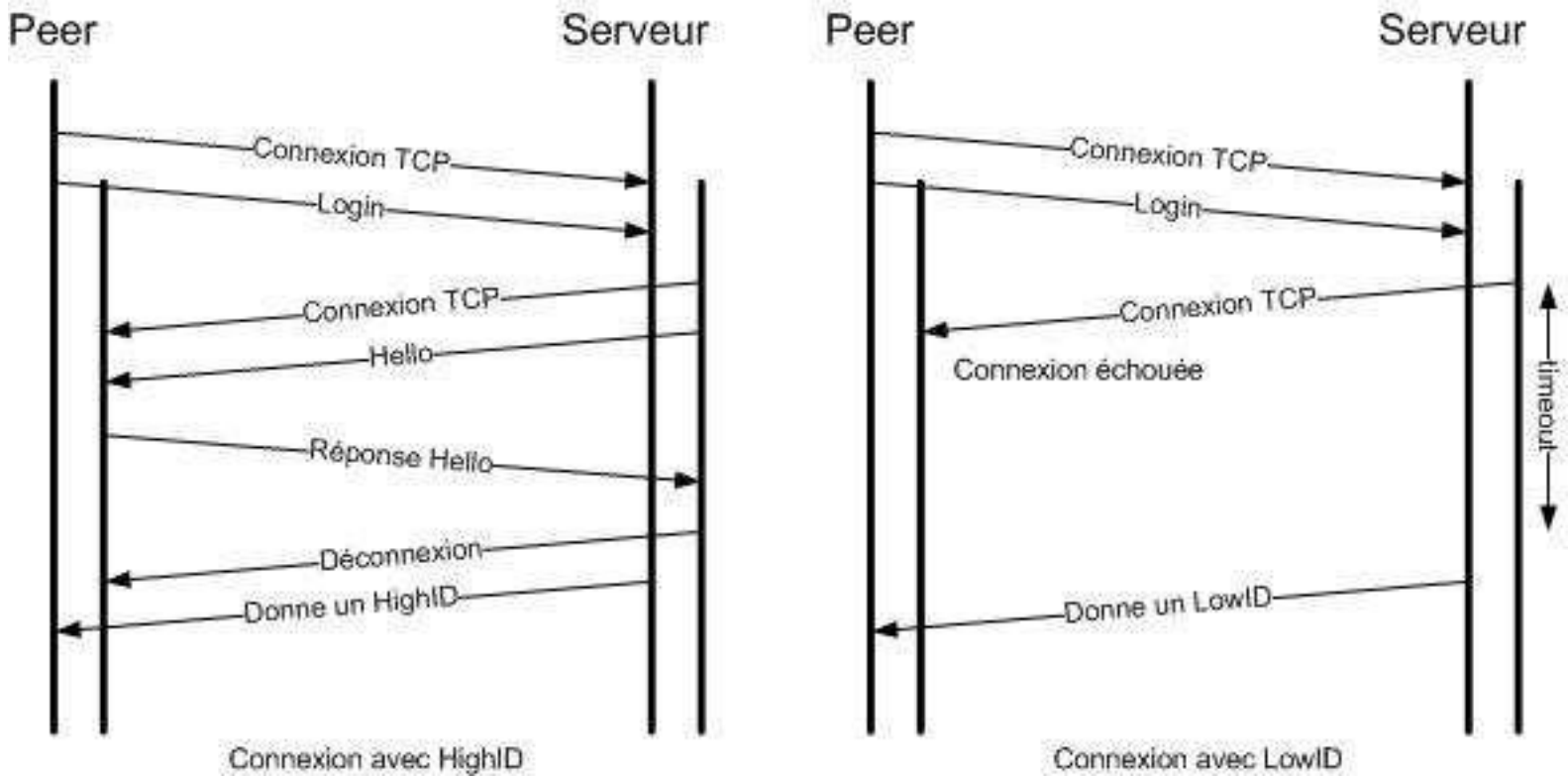
Rôle d’un serveur eDonkey

- Point d’entrée du réseau
- Gère l’indexation des données
- Permet les recherches
- Effectue la mise en relation des utilisateurs pour commencer les échanges

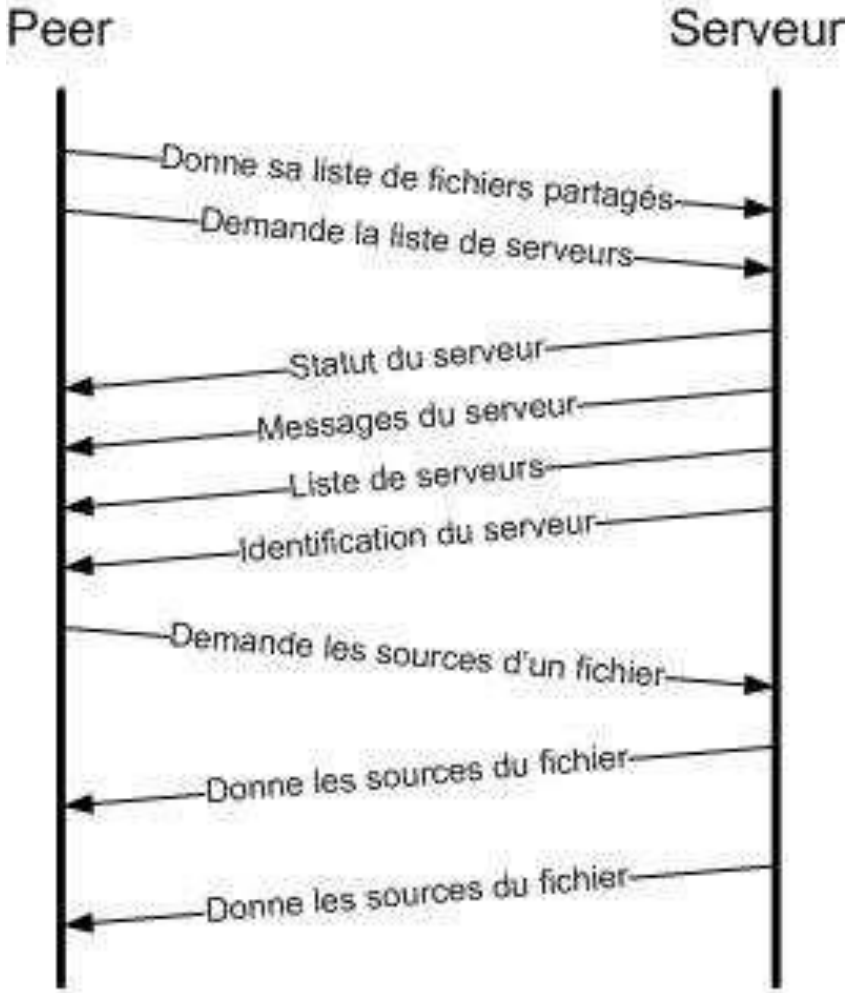
Les clients eDonkey

1. Connexion à un serveur
2. Emettre des requêtes de recherche de fichier
3. Demande les sources possibles pour un fichier demandé
4. Télécharge les fichiers demandés sur les sources
5. Vérifier l'intégrité et la corruption des fichiers
6. Partages avec d'autres clients
7. Gestion de la file d'attente des clients

Connexion à un serveur

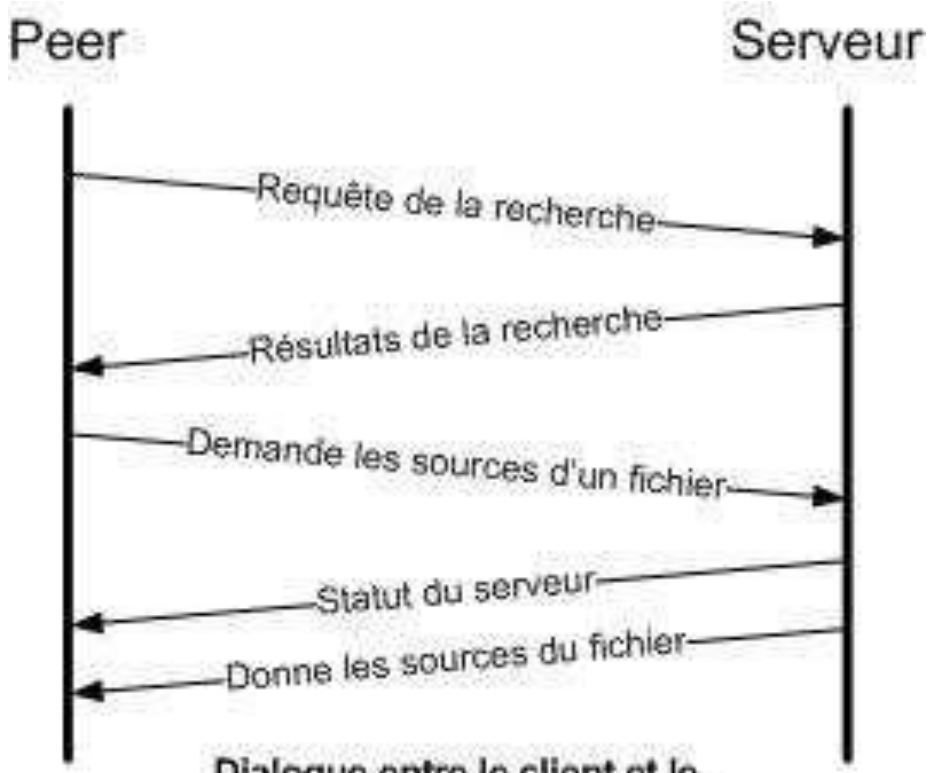


Connexion à un serveur



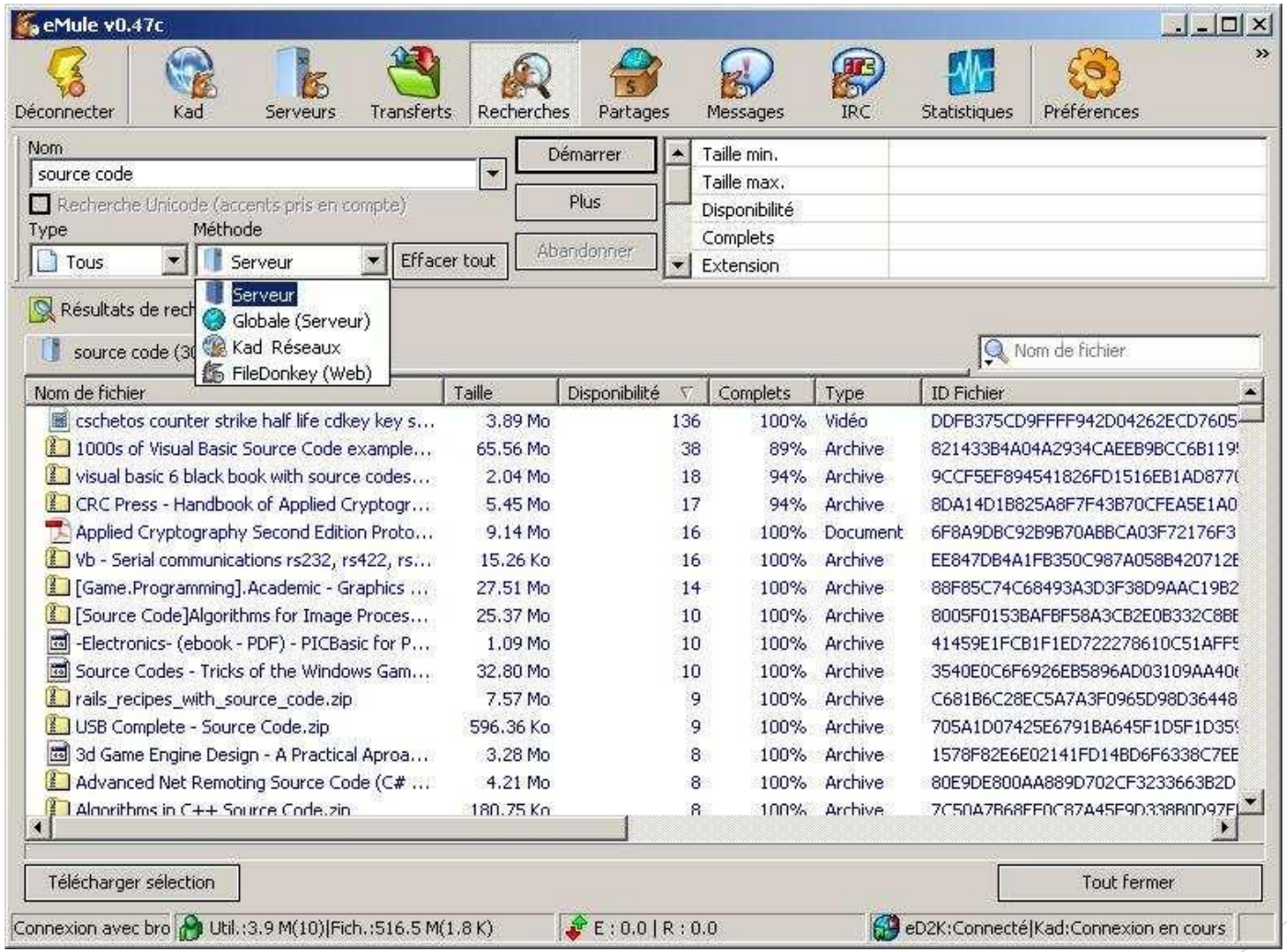
Début de dialogue entre le client et le serveur

Recherche de fichiers

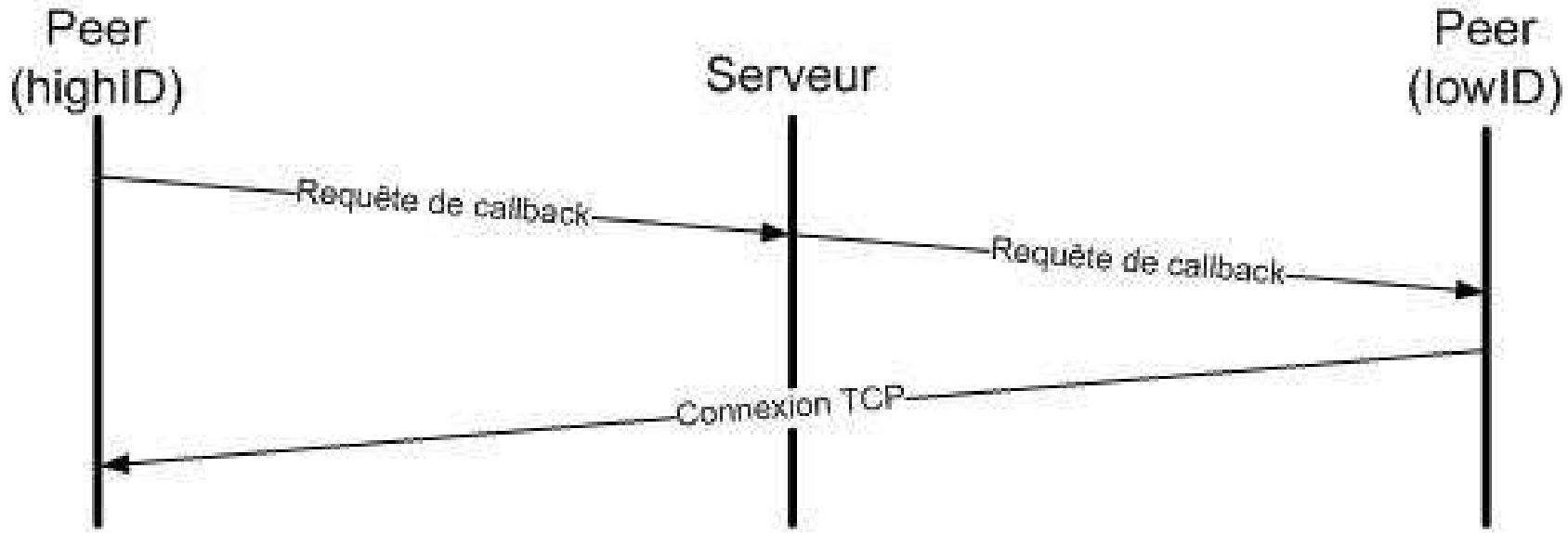


Dialogue entre le client et le serveur lors d'une recherche

Recherche de fichiers



Mécanisme de callBack (ou de rappel)



Mécanisme de callback pour les lowID

Gestion de la corruption

- Intelligent Corruption Handling (ICH)
  - Chaque partie de fichier (9,28MO) → MD4
  - En cas de corruption ICH reprend uniquement des blocs de 128Ko
- File ID ou File Hash
  - Représente de manière unique un fichier
  - Est issu d'un hachage du contenu du fichier
  - Hash sur 128 bits
- Le partage est fait uniquement quand la partie est correcte

Gestion de la corruption

- Advanced Intelligent Corruption Handling (AICH)
- SHA1 sur des blocs de 180Ko
  - Plus fin
  - Partage plus rapide
  - Taille du hash : 160 bits
- Stockage des hash dans un fichier (known2.met) qui est lu à la demande
  - Important en mémoire

User ID (ou User Hash)

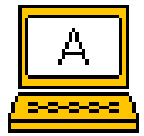
- Crédit dans le but d'encourager les utilisateurs au partage de fichiers
- Nombre sur 128 bits (16 octets)

```
void CPreferences::CreateUserHash()
{
   for (int i = 0; i < 8; i++)
   {
      uint16 random = GetRandomUInt16();
      memcpy(&userhash[i*2], &random, 2);
   }

   // mark as emule client. that will be need in later version
   userhash[5] = 14;
   userhash[14] = 111;
}
```

Echange du User ID

1

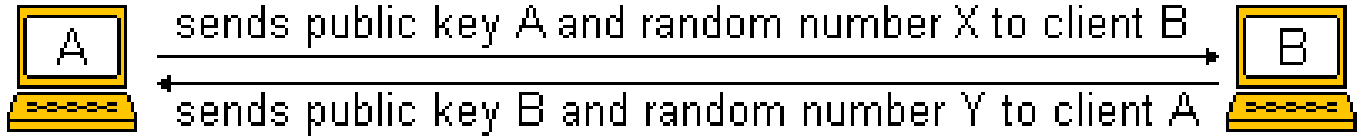


emule-project.net

- creates a 384-bit RSA key
- this key is stored in the "cryptkey.dat"

2

emule-project.net



- stores public key in "clients.met"

- stores public key in "clients.met"

Echange du User ID

3

- creates a signature of random number Y
(and public key B) with his private key A

4

- Checks if the signature if from random number Y (and public key B)
and fits to his public key A
- If everything is OK, the identification was successfull.

La file d’attente

- La file d’attente est de taille finie
  - Drop si la file est pleine
- Système de crédit mis en place
  - Ratio (données émises / données reçues)
- Vos crédits sont sauvegardés par le pair qui accorde ce crédit

Les problèmes du protocole eDonkey

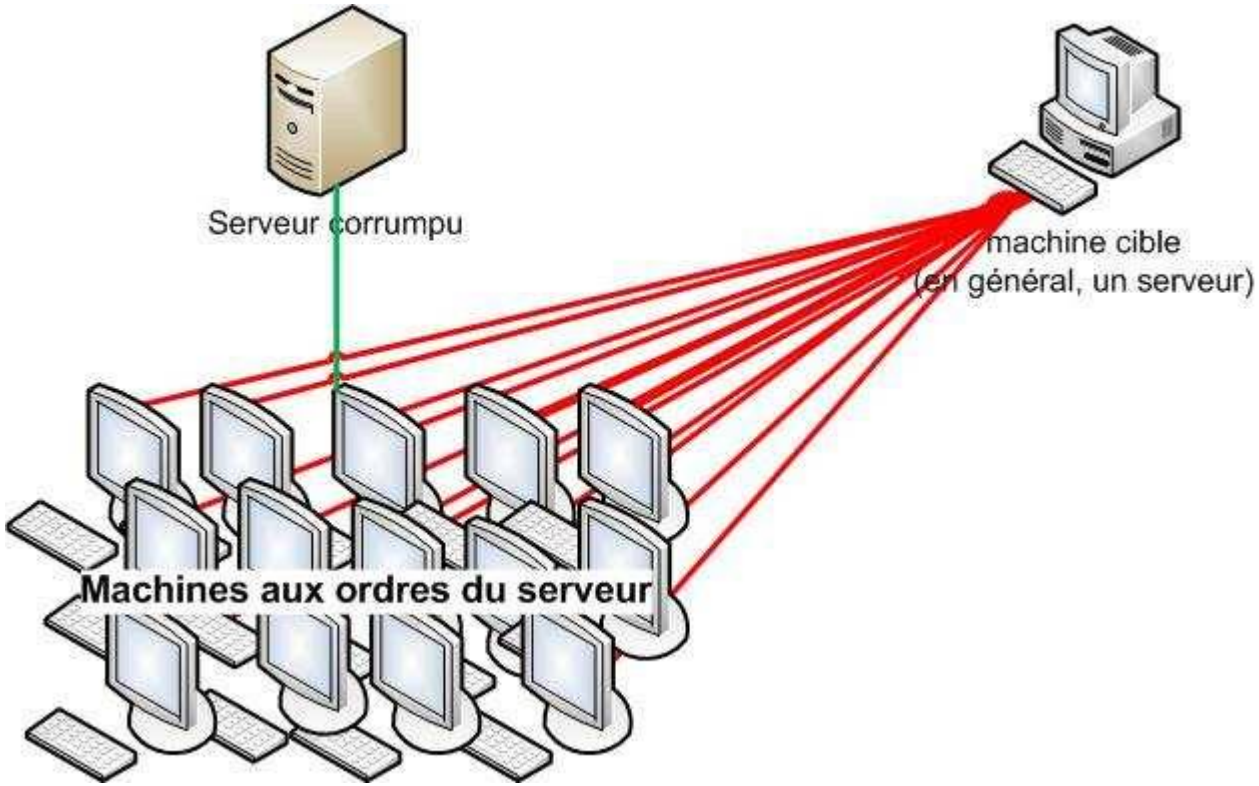
- L'utilisation de NAT sans PAT
- Les extensions non respectueuses du protocole
  - eDonkeyBot sur eDonkey2000 → limité par l'arrivée d'eMule
- Redirection de trafic à cause d'un serveur peu fiable
  - Création d'un DDOS facilement
- Filtrage d'eDonkey et obfuscation du protocole
  - Souvent les ports par défaut (4661 – 4665)
  - Premier octet de l'en-tête : 0xE3

Les problèmes du protocole eDonkey

- Limitation du nombre de fichiers partagés
  - 120 max
- Code source des serveurs fermés
- Problèmes de serveurs
  - Fermeture du serveur

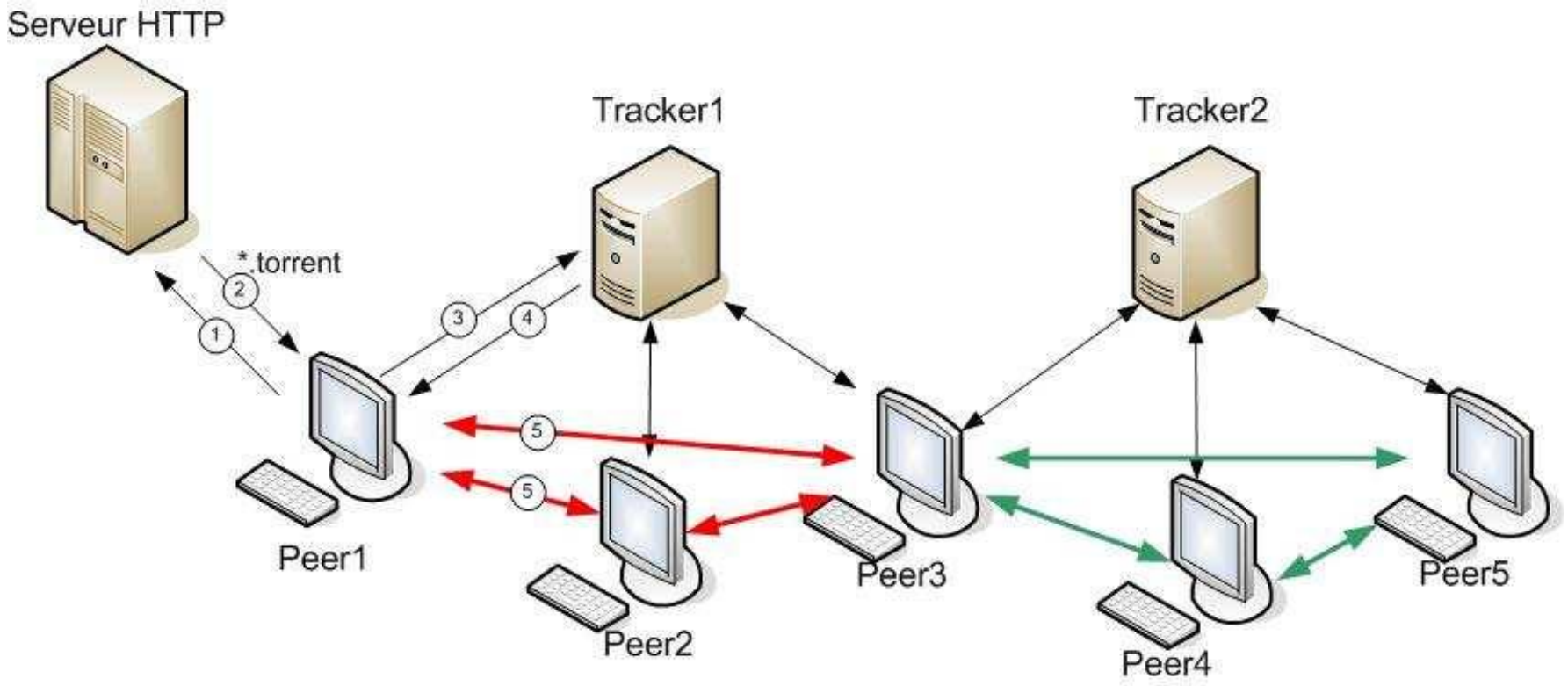
Les problèmes du protocole eDonkey

- Attaque par déni de service



Le protocole BitTorrent

L’architecture BitTorrent



Terminologie

- Torrent
  - Fichier codé en ASCII
  - .torrent
  - Identifie les partages
- Tracker
  - Serveur qui met en relation les pairs
  - « Annuaire »
- Seeder
- Leecher

Structure d’un fichier Torrent

- Announce : adresse du tracker
- Length : taille du fichier en octets
- Name : nom du fichier
- Piece lenght : taille de chaque partie
- Pieces : « … » concaténation du code sha-1 sur 160 bits des fragments

d8:announce34:http://seedy.mine.nu:6969/announce18:azureus_propertiesd17:dht_backup_enablei1ee7:comment0:13:comment.utf-80:10:created by15:Azureus/2.5.0.013:creation datei1164774262e8:encoding5:UTF-84:infod4:ed2k16:0_ẽ o_Ì_C‘_“×8ôÐ6:lengthi179998720e4:name47:Black_Lagoon_-_20.avi10:name.utf-847:Black_Lagoon_-_20.avi12:piece lengthi393216e6:pieces9160:‰_@¡ö—y%¿›a-rò\ etc.

Téléchargement d’un fichier

- Connexion au tracker en HTTP(S)

- Envoi des infos à l’aide de la requête GET
  - Info_hash : Le hash SHA1 du fichier
  - Peer_id : ID sur 20 bits du client généré au démarrage
  - Port : le numéro de port utilisé par le client
  - Adresse IP : optionnel car facilement trouvable
  - Nombre d’adresses de pairs : par défaut, une liste de max. 50 pairs
  - Les statistiques du client : nombre de bits envoyés et reçus

Téléchargement d’un fichier

Le tracker répond

- La liste des seeder et leecher

- Pour chaque pairs
  - L’ID du pair
  - Son adresse IP
  - Le port d’écoute

- A intervalle régulier le client envoie des statistiques au tracker

Téléchargement des fichiers

- Multisourcing
- Annonces à chaque fin de téléchargement de partie
- Téléchargement des parties rares en priorité
- Pas de file d'attente
  - Algorithme ressemblant au « tit-for-tat »
  - « Coopération-réciprocité-pardon » – « donnant-donnant »

Algorithme de partage
Optimistic unchoking

- « J'envoie des données au personnes qui partagent avec moi »
- Un client qui refuse d'envoyer des données est dit « **choke** »
- Un client qui partage est dit : « **unchoke** »
- Le client évalue les autres pairs toutes les 10 secondes
- Super-seeding
  - Mentir aux autres clients

Les problèmes du protocole BitTorrent

- Les fichiers truqués
  - Poison peers
- Failles de sécurité dans le code source
- IP visible
- Utilisation des ressources réseaux

Bilan

Les avantages du P2P

- La répartition de charge
  - Echanges gérés directement par les pairs
  - Plus de soucis de répartition de la charge
- La capacité de stockage
  - Chaque nœud dispose d'une infime partie des données du réseau
  - Tout ce qui est annoncé par le nœud est partagé
- Le calcul distribué
  - Pour les utilisateurs moyens (usage bureautique) seulement 20% du CPU est utilisé

Les avantages du P2P

Pour les entreprises :

- Outils de travail collaboratif
  - Logiciel Groove racheté par Microsoft
- Résistance au panne
  - Sauvegardes croisées
  - Rapidité de propagation
- Extensibilité
  - Auto-configuration des pairs

Mes sources

Cours de Fabien Mathieu

http://gang.inria.fr/~fmathieu/index.php/Main/Teaching

Wikipédia

http://fr.wikipedia.org/wiki/Pair_à_pair

http://en.wikipedia.org/wiki/EDonkey_network

Youtube

http://www.youtube.com/watch?v=WqQRQz_XYg4 (Pastry DHT)

La sécurité des réseaux P2P – Patrick MARLIER

www.laboskopia.com/download/Securite_des_reseaux_P2P.pdf

Merci de votre attention

Des questions ?