

Systemes Pair à Pair

P2P : Introduction

- ▶ Définition historique : Un système distribué où un nœud peut être à la fois client ou serveur
 - ▶ Plus vraiment de distinction entre les deux rôles
- ▶ Définition actuelle : Un système distribué utilisant les ressources localisées aux frontières d'Internet
- ▶ Premier système pair à pair connu: Napster
 - ▶ Chaque nœud était à la fois producteur et consommateur de fichiers mp3
- ▶ Propriétés
 - ▶ Décentralisation (Decentralised)
 - ▶ Dynamicité (Dynamic)
 - ▶ Auto configuration (Self configurable)
 - ▶ Auto cicatrisation (Self-healing)

Principe

- ▶ Le P2P de fichier est né du « besoin » de partager
 - ▶ Mettre à disposition des autres des fichiers
 - ▶ Extension du « cercle des amis »
- ▶ Chaque pair met à disposition des autres des fichiers
 - ▶ Les downloads se font de pair à pair
- ▶ Principalement utilisé à la limite (hors) de la légalité
- ▶ Mais fort potentiel
 - ▶ Distribution de fichiers sans serveur centralisé (exemple: patch dans WoW)
 - ▶ Récupération des espaces libres sur plusieurs machines dans un LAN
- ▶ Principaux problèmes
 - ▶ Recherche
 - ▶ Passage à l' échelle, tolérance aux pannes
 - ▶ Équilibre entre download et uploads
 - ▶ Pollution

Problèmes

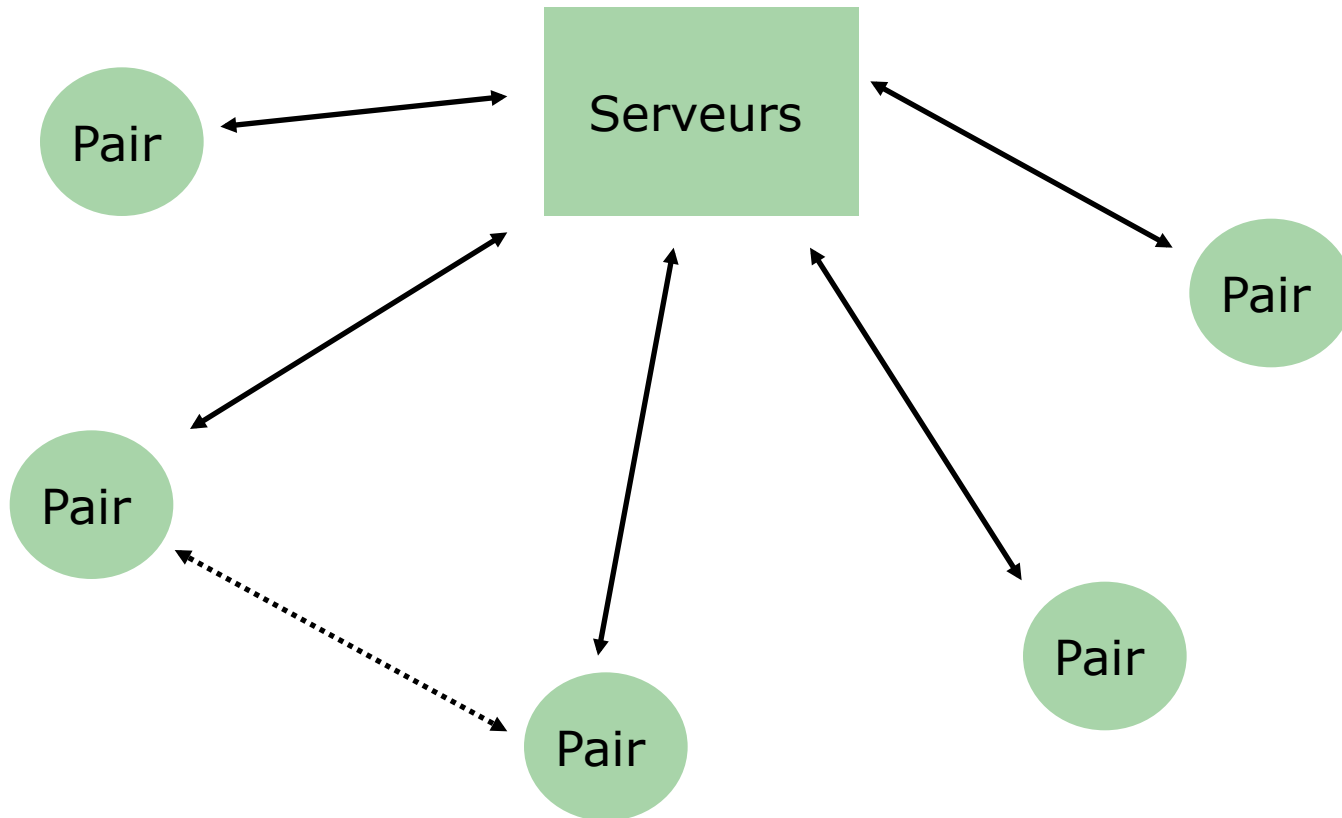
- ▶ Recherche
 - ▶ Mécanismes pour trouver le fichier recherché
 - ▶ Efficacité
 - ▶ Utilisation de la bande passante
 - ▶ Nombre de réponses
 - ▶ Temps
- ▶ Passage à l' échelle
 - ▶ Combien de pairs le réseau peut supporter en restant efficace
- ▶ Résistance aux pannes
 - ▶ Combien de pairs peuvent disparaître avant que le réseau ne soit plus efficace?
 - ▶ Liens avec la théorie de la percolation (étude à l' échelon global des effets d' actions locales)
- ▶ Équilibre down/up
 - ▶ Partage de fichiers, implique... partage :)
 - ▶ Comment inciter certains à partager?
- ▶ Pollution
 - ▶ Détecter les fichier corrompus
 - ▶ Les éliminer du réseau

Napster

Napster

- ▶ Premier systèmes P2P
 - ▶ Inventé par Shawn Fanning en 1999
 - ▶ Utilise des serveurs centralisés pour l'indexation
 - ▶ Transferts de fichiers en P2P
- ▶ Vision utilisateur
 - ▶ Les serveurs sont transparents, tout se passe comme si le partage se faisait de pair à pair
- ▶ Pas de tolérance aux pannes
 - ▶ Un fichier était chargé complètement depuis un autre pair
 - ▶ Si disparition du pair, disparition du fichier
 - ▶ Dépendant des serveurs
 - ▶ Les détenteurs de droits ont fait fermer les serveurs, tuant le réseau
- ▶ Passage à l'échelle
 - ▶ Utilisation de serveurs multiples

Napster



↔ Offre/Recherche
↔ Téléchargement

Gnutella

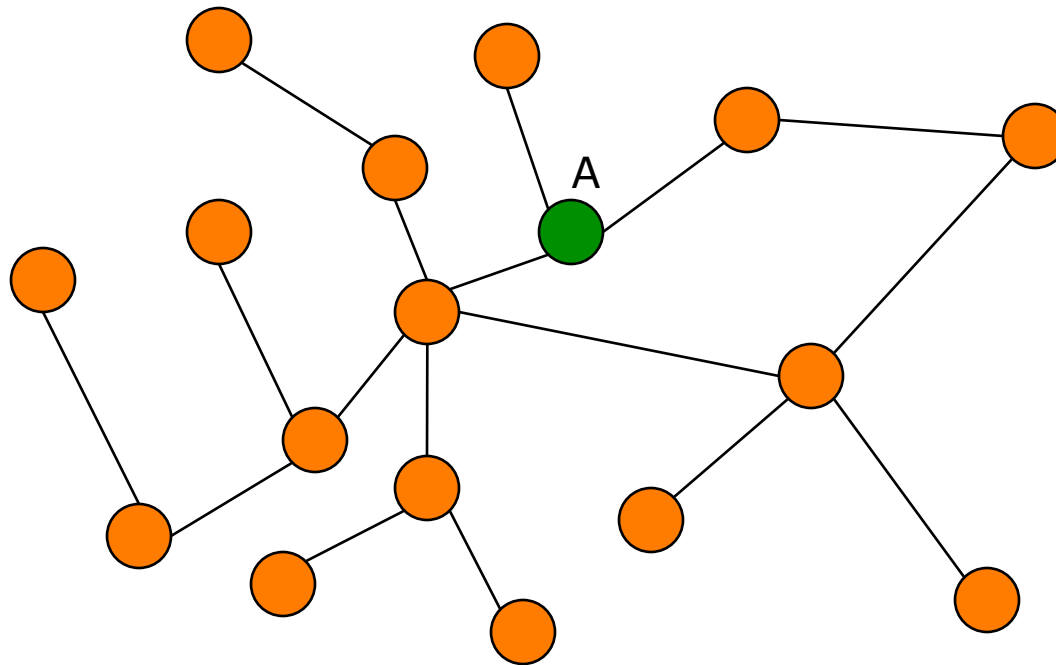
Gnutella

- ▶ Système Pair à Pair totalement décentralisé
- ▶ Gnutella est un protocole
 - ▶ Pour partager des fichiers sur un réseau (Internet)
 - ▶ Construit comme remplace décentralisé de Napster
- ▶ Construit pour être tolérant aux pannes
 - ▶ “Gnutella can withstand a nuclear war and an army of lawyers”.
- ▶ Spécification ouvertes, pas de standardisations
 - ▶ Plein de clients plus ou moins compatibles
 - ▶ Ajout de fonctionnalités
- ▶ Les plus connus: BearShare, LimeWire, Gnucleus

Principes

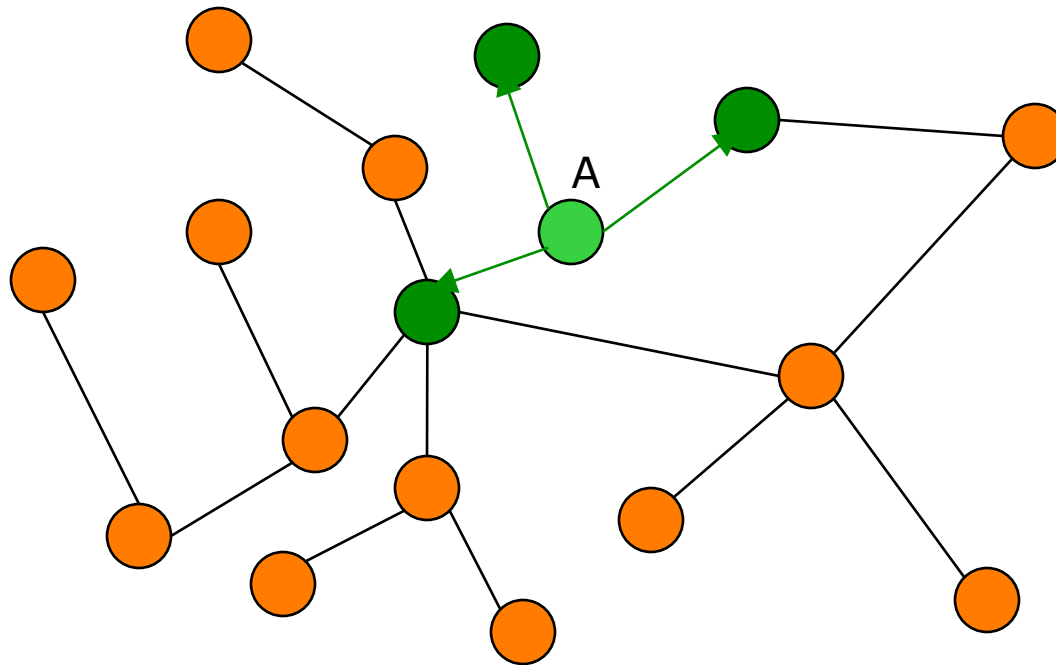
- ▶ Tous les nœuds sont égaux et ont 3 rôles
 - ▶ Producteur: partage de fichiers avec les autres nœuds
 - ▶ Consommateur : téléchargement de fichiers
 - ▶ Routeur: Maintient de l'infrastructure du réseau
- ▶ Chaque nœud a un ensemble (limité) de voisins avec lesquels il peut communiquer
 - ▶ Formation d'un réseau logique (overlay network)
 - ▶ Pas forcément adapté au réseau physique
- ▶ Les messages envoyés servent à
 - ▶ Maintenir l'infrastructure
 - ▶ Chercher/annoncer des fichiers
- ▶ Un nœud est obligatoirement un routeur, même si aucun fichier n'est partagé
- ▶ Le transfert de fichiers se fait entre deux nœuds via http
- ▶ Routage = inondation

Inondation (étape 1)



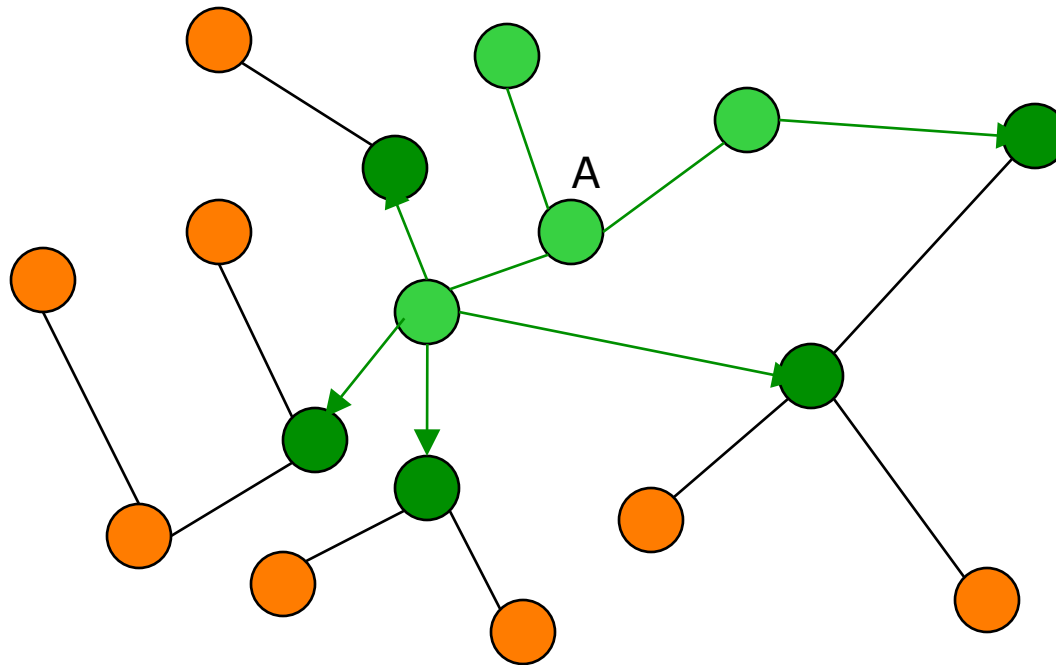
Le nœud A envoie un message de recherche, avec un TTL de 3

Inondation (étape 2)



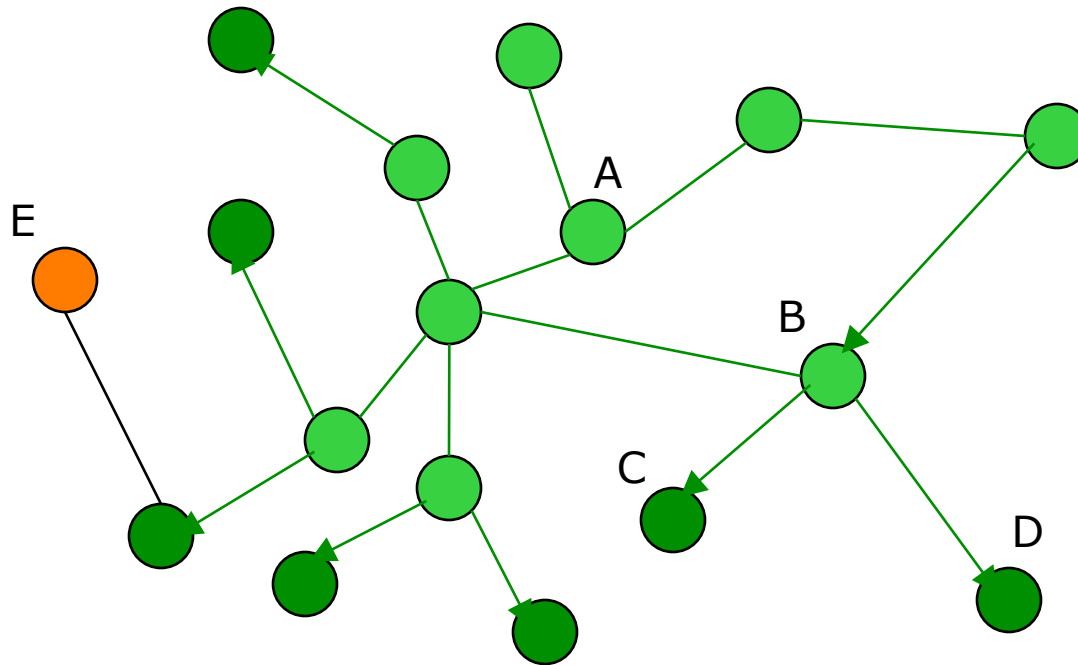
Le TTL est décrémenté par les autres clients, il vaut maintenant 2

Inondation (étape 3)



Le TTL est à nouveau décrétementé, il vaut 1

Inondation (étape 4)



Le nœud B reçoit le message pour la seconde fois, il ne le retransmet pas
Le TTL vaut 0, la transmission s'arrête, le nœud E ne reçoit pas le message

Problèmes

- ▶ L' inondation ne passe pas à l' échelle
 - ▶ Quelle que soit la taille du réseau, un nœud est limité par son horizon (en général TTL=7)
- ▶ La recherche de fichiers se fait par nom
 - ▶ Deux fichiers aux noms différents peuvent avoir le même contenu
 - ▶ Très sensible à la pollution
- ▶ Les messages d' architecture constituent une part importante du trafic de chaque nœud
- ▶ Les noeuds sont identifiés par IP et port
 - ▶ Impossible de retrouver un nœud qui a subit une panne temporaire
- ▶ Protocole de download primitif (http)
 - ▶ Source unique
 - ▶ Pas de download en parallèle

Quelques faits

- ▶ La popularité des recherches suit la même loi que les pages webs
 - ▶ Les technologies de cache ont de bons résultats
 - ▶ Mettre en cache les résultats de recherches pendant 30 min réduit le trafic de 50%
- ▶ Free riding (leechers)
 - ▶ 66% des nœuds ne partagent aucun fichier
 - ▶ 73% des nœuds partagent moins de 10 fichiers
 - ▶ 37% des fichiers sont partagés par 1% des nœuds
 - ▶ Encore très proche du client/serveur
- ▶ Certains nœuds prennent une grande importance dans le réseau
 - ▶ Ils partagent beaucoup de fichiers
 - ▶ Ils ont beaucoup de connexions
- ▶ Rejoint une structuration à la Napster

Le monde est petit

- ▶ En 1967, Stanley Milgram effectue une expérience
 - ▶ Il écrit 160 lettres qu'il donne au hasard à des gens d'Omaha, Nebraska
 - ▶ Il leur demande de les faire parvenir à un courtier de New-York
 - ▶ Interdiction d'utiliser la poste, juste passer la lettre de main en main, à des connaissances bien choisies pour se rapprocher du courtier
- ▶ À la fin, 42 lettres sont arrivées à destination
- ▶ En moyenne, les lettres sont passées entre les mains de seulement 5.5 personnes
- ▶ Cela montre que le réseau social des États-Unis (200 millions de personnes) a un diamètre d'environ 6 personnes
 - ▶ Certaines personnes ont beaucoup de relations, ce sont des ponts entre les communautés
 - ▶ Dans l'expérience de Milgram, 25% des lettres sont passées par le même marchand, 50% par les mêmes trois personnes
- ▶ Un petit nombre de ponts suffit à réduire énormément le nombre de sauts: small world effect

Edonkey/Emule

Edonkey

- ▶ Similaire à Napster
 - ▶ Serveurs utilisés pour l'indexation
 - ▶ Transfert direct entre les clients
- ▶ Données identifiées par des valeurs de hachage
- ▶ Clients sont identifiés par des IDs
- ▶ Chaque client est connecté à un seul serveur en TCP/IP
 - ▶ Connexions multiples en UDP

Edonkey

- ▶ Le contenu à partager est divisé en morceaux (chunk)
- ▶ Un client peut télécharger des morceaux en parallèle de plusieurs clients
- ▶ Une fois téléchargé, l'intégrité d'un morceau est vérifiée
 - ▶ Utilisation des informations de hachage
- ▶ Chaque morceau téléchargé est automatiquement mis en partage
 - ▶ Obligatoire par le logiciel, rien dans le protocole

Téléchargement

- ▶ Le client A envoie une requête à un serveur
- ▶ Le serveur répond avec une liste d' ID de clients
- ▶ Le client A peut contacter les clients pour télécharger les morceaux

Limitations

- ▶ Un client ne voit que les fichiers disponibles sur le serveur auquel il est connecté
 - ▶ Requêtes possibles vers d'autres serveurs en UDP
 - ▶ Aucune garantie de réponse, et augmente la charge des serveurs
- ▶ Toujours des free-riders
 - ▶ Le ratio down/up est géré par le client
 - ▶ Très facile à contourner avec un client « maison » (mldonkey) ou du traffic-shapping
- ▶ Les pairs sont très faciles à identifier
 - ▶ L'adresse IP est très facile à retrouver avec l'ID
 - ▶ Liste des partages de chaque client disponible (mais peut être interdit)

Bittorrent

Bittorrent

- ▶ Système P2P avec serveurs (trackers)
- ▶ Les trackers ne font que coordonner les clients
 - ▶ Aucune connaissance des données distribuées
 - ▶ Peu gourmand en ressources
- ▶ Les données distribuées sont décrites dans des fichiers .torrent contenant
 - ▶ Le nom du fichier
 - ▶ Sa taille
 - ▶ Des informations de hachage
 - ▶ L' URL du tracker
- ▶ Pas de mécanismes de recherche
- ▶ Le protocole essaie de forcer le partage
 - ▶ Probablement le plus réussi actuellement

Bittorrent

- ▶ Pour télécharger, un client donne au tracker
 - ▶ Des informations sur ce qu'il recherche
 - ▶ Ses ports disponibles
- ▶ Le tracker retourne une liste de pairs ayant le même intérêt
- ▶ Les clients peuvent alors se connecter entre eux
- ▶ Les données sont divisées en morceaux (typiquement 250KB)
- ▶ Les morceaux peuvent être téléchargés chez n'importe quel pair
- ▶ Pour démarrer, un pair spécial ayant le contenu complet est nécessaire : la graine (seed)

Connexions

- ▶ Les connexions entre pairs ont plusieurs états
 - ▶ interested : indique qu' on veut downloader quelque chose
 - ▶ choked : indique qu' on refuse d' uploader
- ▶ Sert à maintenir des informations entres les pairs
- ▶ Utilisé pour décider du sens d' écoulement des informations
- ▶ Un pair download ssi
 - ▶ Il a indiqué interested
 - ▶ Il n' est pas choked par l' autre pair
- ▶ Un pair upload ssi
 - ▶ Il n' a pas choked l' autre pair
 - ▶ Si l' autre pair est interested
- ▶ Il est impératif de maintenir ces 2 états à jour
 - ▶ Un pair choked indiquera quand même qu' il est interested

Connexions

- ▶ Un connexion est marquée interested si le pair a des pièces qui nous intéressent
- ▶ Un pair peut temporairement refuser d'uploader vers un autre pair: il l'étouffe (choke)
 - ▶ Sert à indiquer qu'on ne veut pas envoyer de données, si par exemple on ne download rien
 - ▶ Un client choke toujours une seed (la seed ne veut pas recevoir de données)
 - ▶ Utile pour maintenir des connexions ouvertes sans rien envoyer
- ▶ Les décisions d'étouffement sont prises toutes les 10 secondes
 - ▶ Évite de basculer trop rapidement entre choked/unchoked
- ▶ On dé étouffe les 4 pairs qui ont le meilleur upload (i.e envoient des fichiers vers vous avec le meilleur débit) et sont interested
 - ▶ Permet de maximiser le download
- ▶ Les pairs qui ont un meilleur upload mais ne sont pas interested sont dé étouffés
 - ▶ Si ils deviennent interested ils remplacent le(s) plus mauvais des 4 précédents

Bittorent

- ▶ Recherche:
 - ▶ Utilisation de serveurs web recensant des fichiers torrent
 - ▶ Modération, élimination de la pollution
- ▶ Certains trackers forcent un ratio download/upload
 - ▶ Si en dessous du ratio, alors un pair ne peut plus downloader
- ▶ Rend moins facile la collecte d'informations
 - ▶ Seuls le tracker et le serveur web ont des informations globales
 - ▶ Il faut avoir le fichier .torrent et donc avoir une cible
 - ▶ Mais étant donné un tracker, très facile d'avoir la liste des pairs
- ▶ Modèle économique
 - ▶ Chaque pièce à une valeur dans le protocole
 - ▶ Une pièce rare sera demandée par beaucoup de clients, et donc permettra d'en downloader d'autres
 - ▶ Une pièce très répandue ne sera jamais needed