Calcul de scores de réputation sur un réseau P2P avec l'algorithme Eigentrust

Michel Chilowicz

26 janvier 2006

Utilité d'un système de confiance sur les réseaux P2P

Les réseaux pair à pair (P2P)

- Statut égalitaire de chaque nœud.
- Multiples applications envisageables : partage de fichiers, calcul distribué, sauvegarde, ...

Système de confiance

- Comportements indésirables de nœuds :
  - asymétrie de l'offre et de la demande (*freeriders*),
  - malhonnêteté en proposant des ressources altérées.
- Scores de réputation pour déterminer les nœuds indésirables.

Scores de réputation par Eigentrust

Algorithme distribué inspiré de *PageRank* pour la réputation des pages Web (Google).

Quantification de la réputation

Objectif

Attribution d'un score de réputation t_i à chaque nœud i du réseau

Comment ?

Maintenance d'un vecteur de notes locales de réputation chez chaque nœud.
Évaluation de chaque ressource apportée (point positif ou négatif), par exemple pour le partage de fichiers :

- Évaluation négative si fichier altéré ou trompeur.
- Évaluation positive sinon.

Matrice de notes locales de réputation

Matrice C

$c_{i,j}$ est la note locale de réputation attribuée par i à j.
Par exemple, $c_{i,j} = \max(\sum e^+ - \sum e^-)$.

Normalisation des $c_{i,j}$

Étape de normalisation : bornage des notes locales, somme des notes unitaire :

$$0 \leq c'_{i,j} = \frac{c_{i,j}}{\sum_j c_{i,j}} \leq 1$$

Réseau de réputation transitif

Problématique

Propagation des notes locales de réputation pour fabriquer une note globale.

Peut-on faire confiance dans les notes locales attribuées par un nœud de bonne réputation ?

- *Oui* pour Eigentrust : confiance dans les notes proportionnelle à la réputation.
- Dissociation de réputation et confiance de notation intéressante ? (*Web of Trust* de OpenPGP).

Intérêts des scores de réputation

- Choix optimal d'un nœud pour obtenir une ressource.
  - Choix déterministe (nœud de meilleure réputation).
  - Choix probabiliste (probabilité de choix pondérée par la réputation).
- Discrimination à l'accès d'une ressource (file de priorité).
- Isolement de nœuds malfaisants.
- Incitation à la non-propagation de fichiers altérés.

Super-nœuds

Constitution de super-nœuds (modèle Gnutella) :
limitation par un modèle de choix probabiliste de fournisseurs de ressource.

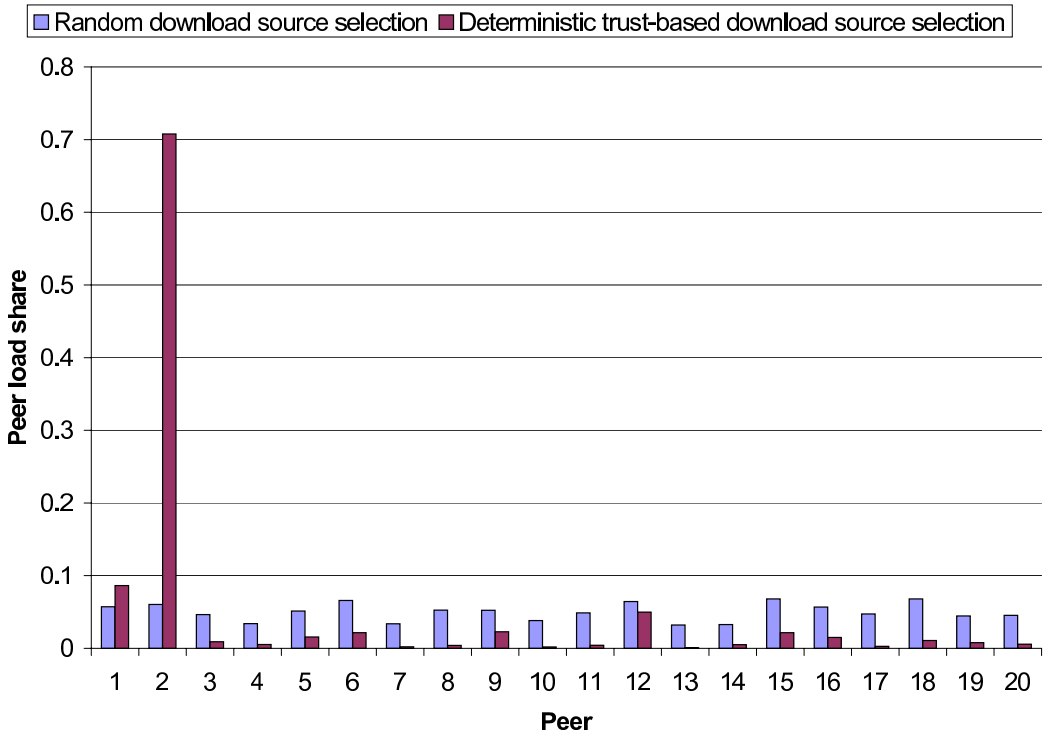


Figure: Distribution de charge sur un réseau P2P de 20 nœuds (modèle déterministe / probabiliste)

Le problème des nouveaux entrants

- Calcul du score de réputation des nouveaux entrants ?
  - Score nul ?
- Intégration des nouveaux entrants :
  - Allocation d'une probabilité de choix p_{ne}.
  - Problème : attaque de nouveaux entrants sans réputation exploitant la probabilité de choix.

Principe de *PageRank*

- Algorithme conçu par Larry Page et Serge Brin (Google).
- Note locale $c_{i,j}$ non nulle s'il existe un lien de i vers j.
- Problème des pages non-liantes (ignorées au début).

Équation récursive du *PageRank*

$$R_i = \sum_{j \in B_i} \frac{R_j}{|l_j|} \tag{1}$$

B_i: ensemble des pages avec lien vers i.

Le problème de la convergence

Convergence impossible

Si plusieurs pages s'auto-référencent :
augmentation du *PageRank* à chaque itération.

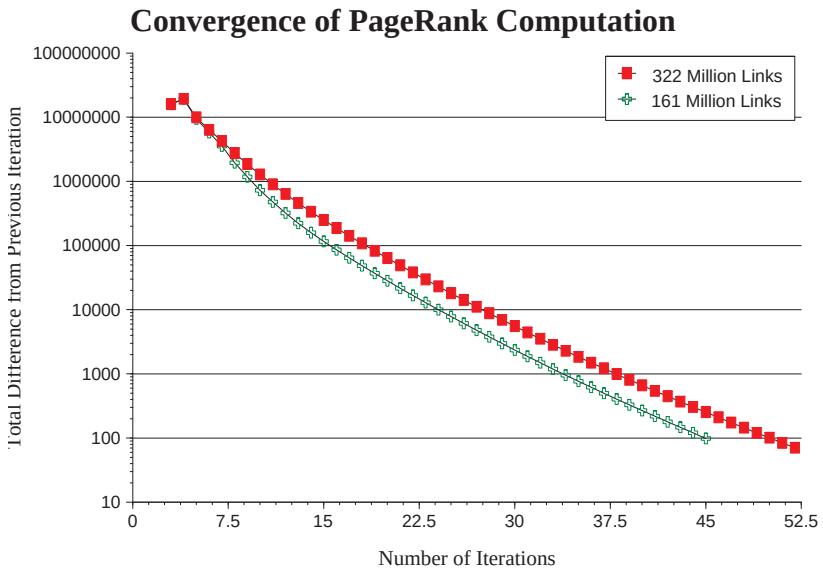
Solution : pré-réputation

- Utilisation d'un vecteur de pré-réputation $\overrightarrow{E}$:
  $E_i \neq 0$ si page très populaire.
- Pré-réputation attribuée par autorité centrale (Google).
- Modèle du surfeur aléatoire (parcours aléatoire du graphe du Web).

Équation corrigée

$$R'_i = cR_i + (1 - c)E_i \tag{2}$$

Convergence expérimentale et complexité de *PageRank*



Complexité

$O(m \log(n))$

m : nombre de liens

$\log(n)$: nombre d'itérations (n : nombre de pages)

L'algorithme Eigentrust

Formule basique

- Score global t_i du nœud i :

$$t_i = \sum_j t_j c_{j,i}$$

- Équation matricielle :

$$\overrightarrow{t} = C^T \overrightarrow{t}$$

Calcul pratique

Trouver le vecteur propre principal de C^T ?

$$(\overrightarrow{t^n}) : \begin{cases} \overrightarrow{t^0} = (\frac{1}{m})_{1 \leq i \leq m} \\ \overrightarrow{t^{k+1}} = C^T \overrightarrow{t^k} \end{cases}$$

Convergence de t

Cliques de nœuds malhonnêtes

nœuds malhonnêtes s'attribuant mutuellement une réputation (évaluation positives échangées).
Réputation non distribuée à l'extérieur.
$\longrightarrow$ Augmentation de la réputation des nœuds lors des itérations.

Utilisation d'un vecteur de pré-réputation $\overrightarrow{p}$

Pré-réputation accordée à certains nœuds de référence (fondateurs du réseau). Nouvelle formule matricielle :

$$\overrightarrow{t} = (1 - a)C^T \overrightarrow{t} + a\overrightarrow{p}$$

Complexité de Eigentrust

Convergence expérimentale

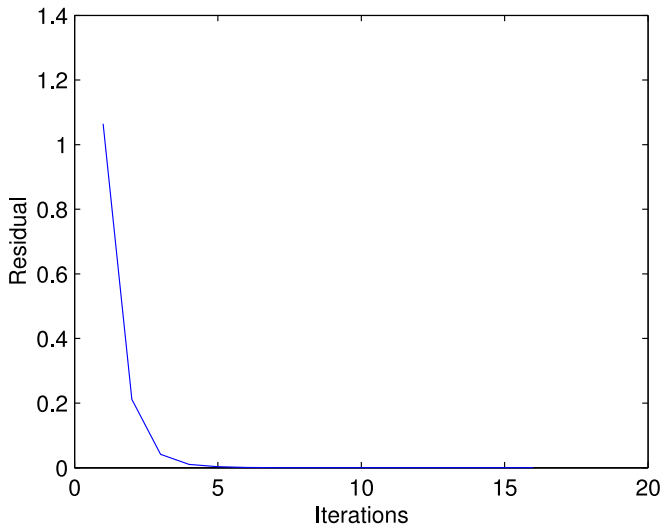


Figure: Convergence de Eigentrust

Complexité

n: nombre de nœuds sur le réseau

m: nombre de relations entre nœuds

Complexité : $O(iter(n)m)$

Calcul naïf distribué de Eigentrust

Principe

Chaque nœud calcul son propre score global de réputation :
$\longrightarrow$ tricherie possible.

Algorithme naïf distribué Eigentrust

Définitions

- A_i : nœuds ayant demandé des ressources à i.
- B_i : nœuds ayant fourni des ressources à i.

Algorithme pour chaque nœud i

- Demander aux nœuds $j \in A_i$ $t_j^0 = p_j$
- Répéter
  - Calculer $t_i^{k+1} = (1-a)\sum_{1 \leq j \leq n} t_j^k c_{j,i} + a p_i$
  - Envoyer $c_{i,j} t_i^{k+1}$ aux nœuds $j \in B_i$
  - Calculer $\delta = |t_i^{k+1} - t_i^k|$
  - Réceptionner pour $j \in A_i$ les valeurs $c_{j,i} t_j^{k+1}$
- Jusqu'à convergence $(\delta < \epsilon)$

Version sécurisée du calcul distribuée

- Le nœud i ne calcule plus lui-même t_i.
- Utilisation d'un système de DHT (CAN, Chord, Kademlia, ...) pour lier identifiant de nœud et nœud gestionnaire de la réputation.
- Redondance pour limiter les effets de nœuds malveillants : calcul à la moyenne ou à la majorité.

Attaques et statistiques avec Eigentrust

- Réalisation de simulations avec plusieurs scénarios d'attaque.
- Reprise des résultats de Kamber, Schlosser et Garcia-Molina (pas de vérification).

Modèle de réseau

Interconnexion

Modèle Gnutella : interconnexion de nœuds selon loi de puissance.
Utilisation d'un autre modèle ?

Acheminement des requêtes

Acheminement par propagation de proche en proche avec TTL.
Utilisation d'un DHT ?

Le modèle de distribution des contenus

- Ressources classées en catégories (20).
- Chaque nœud = 1 catégorie.
- Ressources de popularité suivant une distribution de Zipf.
- Requête = (catégorie, popularité)

La simulation d'exécution

Cycle de demande

1. i transmet une demande, est inactif, stoppé ou ne répond pas.
2. Après émission d'une demande, attente de la liste des nœuds possédant le fichier.
3. Téléchargement de fichier (choix probabiliste par réputation).
4. Évaluation du fichier.
5. Retour en 3 si le fichier est altéré.

Menaces et simulation

Les menaces

Introduction de nœuds malhonnêtes, individuels ou en communauté avec :

- Envoi volontaire de ressources non-altérées avec probabilité $p_{na} < 1$.
- Comportement d'évaluation impropre.

Métrique de mesure d'efficacité

Proportion de fichiers altérés téléchargés.

Nœuds malhonnêtes individuels (1)

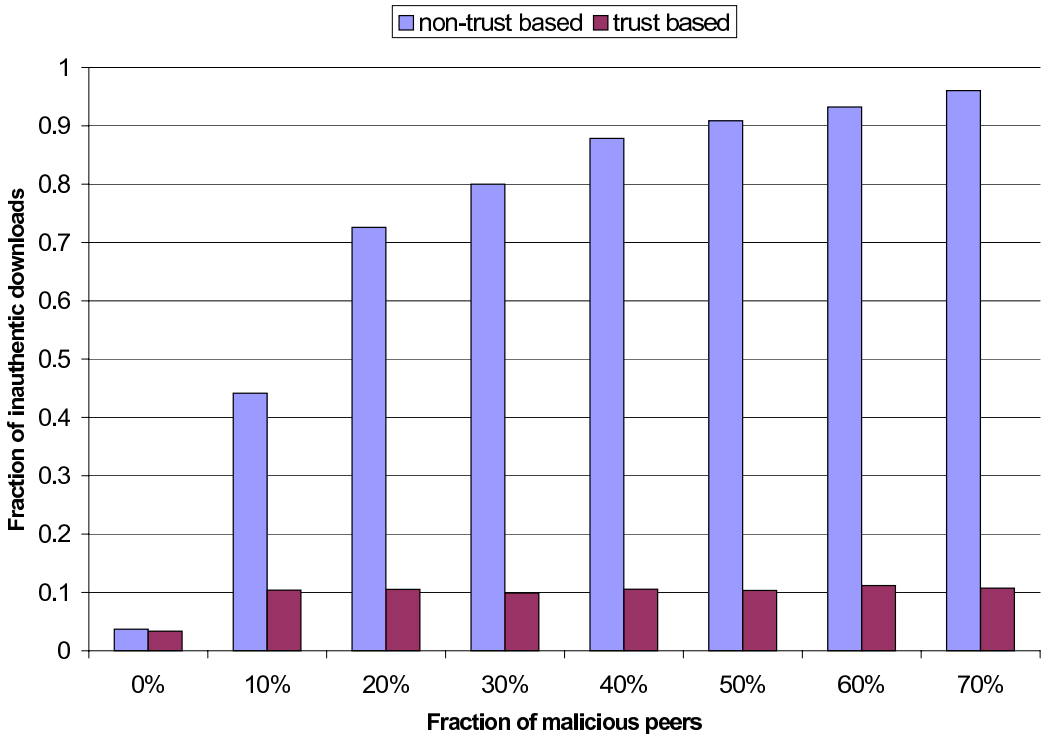


Figure: Réduction de la proportion de téléchargements altérés par Eigentrust lors d'une menace de nœuds malhonnêtes agissant individuellement

Nœuds malhonnêtes individuels (2)

- Efficacité de Eigentrust contre cette attaque.
- Si la proportion de nœuds malhonnêtes est trop importante, calcul Eigentrust non fiable
  - Amélioration par redondance des calculs.

Attaque de nœuds malhonnêtes organisés

- Notes locales maximales attribuées aux autres nœuds malhonnêtes connus.
- Envoi de fichiers altérés.

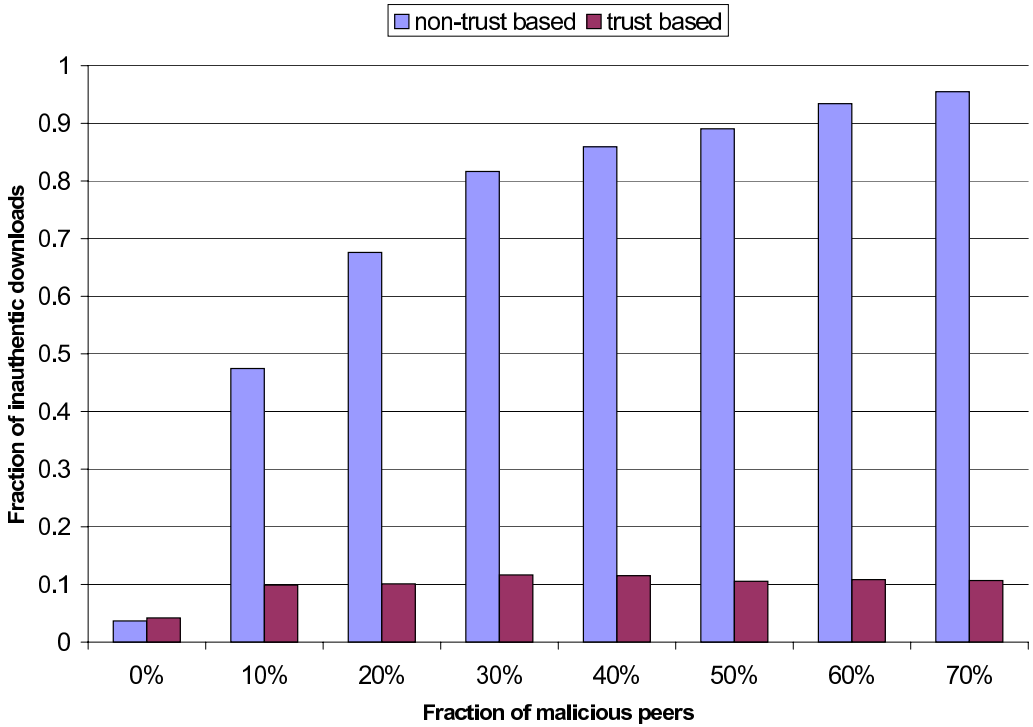
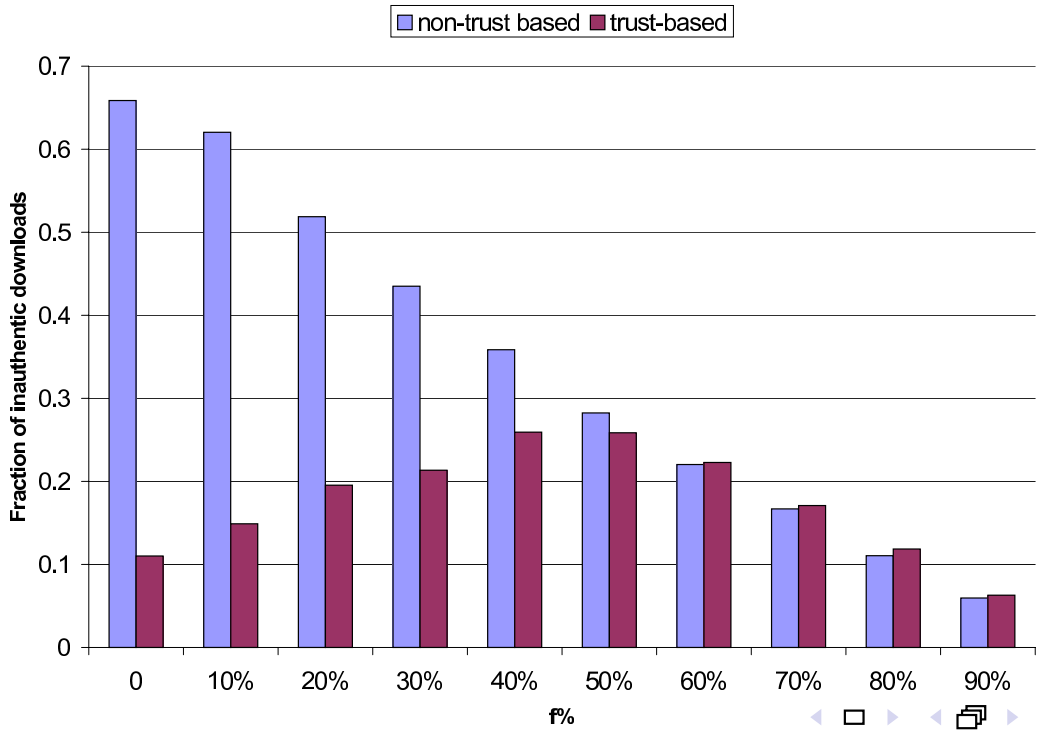


Figure: nœuds malhonnêtes organisés en clique et distribuant des fichiers altérés.

Tentative de camouflage de nœuds malhonnêtes organisés

Camouflage

Distribuer également des fichiers non-altérés pour augmenter sa réputation :
maximisation de la proportion de fichiers altérés sur le réseau pour $p_{na} = 0,5$.



Nœuds espions

Un nœud espion :

- émet des fichiers non-altérés,
- mais attribue des notes locales non-nulles uniquement à des nœuds malhonnêtes.

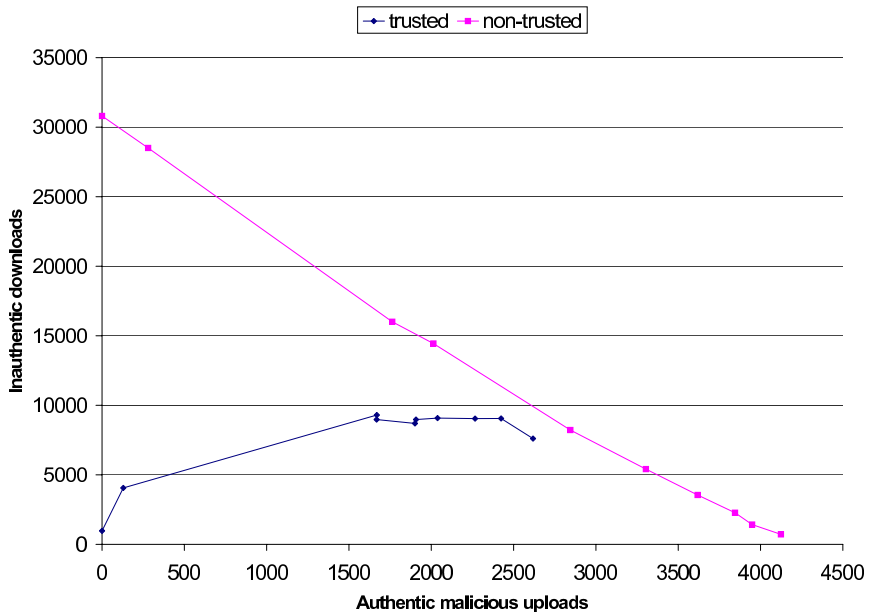


Figure: Simulation avec 63 nœuds honnêtes, 3 pré-réputés, 40 nœuds malhonnêtes en deux groupes d'effectif variable

Cas d'attaque virale

Caractérisation de l'attaque

Pourcentage faible de fichiers vérolés transmis.

Comportement de Eigentrust

Inefficace.

En conclusion...

- *Eigentrust* similaire à *PageRank*
  - Avantages : adapté pour des topologies en loi de puissance (Gnutella), algorithme très étudié.
  - Inconvénient : pour d'autres topologies ?
- Subjectivité de la notion d'altération d'une ressource.
- Système de réputation : regroupement de nœuds par valeurs communes.
- Autorités de confiances ultimes (pré-réputation) ?