

MÉMOIRE DE STAGE DE MASTER

SPÉCIALITÉ : **Recherche en Informatique**
Mention : **Informatique, Mathématiques, Statistiques**

effectué au LIRMM/INFO

—
sous la direction de BAERT ANNE-ÉLISABETH ET BOUDET VINCENT

Stratégie de Recherche dans les réseaux Pairs à Pairs
PRE-VERSION

par

Leborgne Christian

Soutenu le 22 août 2006

Table des matières

1	Introduction	1
1.1	Position du problème	1
1.2	Motivation	1
2	Etat de l'art	3
2.1	Graphe aléatoire généralisé	3
2.1.1	Nombres de voisins et longueur de chemin	4
2.2	Théorie de la percolation	5
2.3	Marche aléatoire	7
2.3.1	Marche intelligente	7
3	Modélisation	9
3.1	Le modèle Pair-à-pair	9
3.1.1	Recherche par inondation	10
3.1.2	Gnutella	11
3.2	Le modèle	12
3.3	Processus de Branchement	14
3.3.1	Définition	14
3.3.2	Distribution du nombre de descendants par génération	14
3.3.3	Distribution du nombre total de descendants	15
4	Stratégie de recherche	17
4.1	Résultat	17
4.1.1	Choix du TTL	17
4.1.2	Inondation Probabiliste	17
4.1.3	Choix de p	19
4.1.4	Popularité d'une ressource	19
4.2	Exemple	20
4.2.1	Loi de Poisson	20
4.2.2	Loi de puissance	21
5	Perspectives	24
6	Conclusion	25
	Bibliographie	25

Table des figures

2.1	Percolation	5
2.2	Apparition de la composante connexe	6
3.1	Le réseau P2P	10
3.2	Le réseau P2P dynamique	10
3.3	Le graphe représentant le réseau P2P	13
3.4	L'arbre représentant la recherche	13
3.5	Processus de branchement	16
4.1	Inondation probabiliste	19

Abstract

Recent work on the structure of social networks and the internet has focussed attention on graphs with distributions of vertex degree that are significantly different from the Poisson degree distribution that have been widely studied in the past. In this paper we presents an analytical framework to study search strategies in large-scale decentralized unstructured peer-to-peer (P2P) networks. The peers comprising the P2P network and their application-level connections are modeled as Branching process whose simple and efficient analysis is accomplished using the generating function of the graph's degree distribution. The framework we defined allow the computation of several interesting performance such as the average number of messages sent throughout the P2P network and the probability that a query extinct. We apply our theory to some graphs such as the Poisson graph and the power-law graph.

Résumé

Des travaux récents sur la structure sociale des réseaux et sur internet se sont focalisés sur les graphes avec un degré de distribution similaire à une loi de Poisson ceux-ci étudiés largement. Dans ce papier, nous présentons une étude analytique des stratégies de recherche dans les réseaux pairs-à-pairs (P2P) non structurés. Les pairs dans le réseau P2P et leurs interconnexions sont modélisés par des processus de branchement dont leur analyse simple et efficace est accomplie en utilisant la fonction génératrice du degré de distribution du graphe. L'étude proposée permet d'analyser plusieurs résultats intéressants tel que le nombre moyen de messages envoyés sur le réseau et la probabilité d'extinction d'une requête de recherche. Nous avons appliqué notre théorie à quelques graphes tels que les graphes de poisson et les graphes possédant une loi de puissance.

Chapitre 1

Introduction

L'opération fondamentale dans la plupart des réseaux pairs-à-pairs (peers-to-peers) est de localiser les objets partagés parmi les noeuds du réseau de manière efficace. Dans les réseaux pairs-à-pairs non structurés (GNUTELLA) il est plutôt difficile de transmettre les requêtes d'une façon efficace jusqu'au noeud désiré. De ce fait, les techniques de recherches actuelles existantes dans les systèmes pairs-à-pairs ne sont pas très efficaces.

1.1 Position du problème

Le but de ce stage est d'établir un modèle analytique des réseaux P2P afin d'analyser et d'évaluer différentes stratégies de recherche.

Dans les réseaux P2P ainsi que dans les systèmes de calculs à grande échelle, il est difficile de déterminer quand et où va apparaître une nouvelle connexion. La modélisation de la topologie de ces systèmes est généralement basée sur des graphes aléatoires.

Lors d'une recherche de documents sur un système P2P, l'une des stratégies de recherche est d'inonder de requêtes le réseau (flooding algorithms). Le client envoie son message à tous ses voisins qui réitère eux-mêmes ce procédé jusqu'à ce que le message soit écarté (Time To Live) . Ainsi un noeud est susceptible de recevoir plusieurs fois cette requête. Ce nombre d'occurrences est étroitement corrélé à deux paramètres du graphe, le degré et le diamètre.

Dans ce stage, on cherchera tout d'abord à trouver un cadre analytique flexible qui permettra d'étudier et de proposer une stratégie de recherche efficace dans des systèmes P2P décentralisés. On s'intéressera ensuite à déterminer et à optimiser le nombre d'occurrences de réception du message sur un pair, ainsi que la taille du TTL . D'autres indices permettant d'optimiser les performances des stratégies de recherche pourront aussi être étudiés.

1.2 Motivation

La motivation principale de ce document est d'améliorer les performances de la recherche par inondation dans les réseaux pairs-à-pairs non structurés, tel que Gnutella. Nous présentons une modélisation des réseaux peer-to-peer (P2P) modélisés par des processus de branchement qui permettent d'appréhender certains résultats pratiques tels que le nombre de messages sur le réseau, la probabilité d'extinction de la recherche, la probabilité qu'une requête de recherche donne une réponse positive. De plus les réseaux P2P ont une profonde influence sur la manière de construire les nouvelles applications réseaux.

Le rapport est organisé comme ceci. Le chapitre 2 résume l'état de l'art sur ce problème. Le chapitre 3 détaille le modèle P2P et ensuite notre modèle qui nous servira à répondre aux ques-

tions principales le nombre de message sur le réseau, la probabilité qu'une requête de recherche obtienne une réponse positive ainsi que la probabilité d'avoir n sommets atteints. Le chapitre 4 met en avant quelques résultats ainsi que des modélisations possibles. Finalement, le chapitre 5 présente les conclusions ainsi que les futures perspectives du projet.

Chapitre 2

Etat de l'art

Depuis quelques années une émergence dans le domaine des techniques analytiques d'investigation des réseaux P2P a commencé à apparaître [13,14,15,16]. L'analyse des stratégies de recherche dans les réseaux P2P décentralisés et non structurés a été conduite dans [17] par la simulation pour analyser quelques variations de l'algorithme d'inondation.

Un certain nombre d'articles exploitent les graphes aléatoires pour modéliser et analyser les réseaux P2P. Le papier [10] définit un modèle de graphe aléatoire évoluant, pour étudier l'évolution des communautés P2P comme Gnutella ou Freenet ; ce modèle est très précis car les réseaux P2P sont très dynamiques mais leur technique permet seulement des résultats basiques.

2.1 Graphe aléatoire généralisé

L'article [9] est celui par lequel le stage a commencé, ce papier présente une étude analytique des stratégies de recherches dans les réseaux P2P décentralisés et non structurés. Le comportement du réseau P2P et de ses connexions sont modélisés par les graphes aléatoires généralisés (GRGs) dont l'analyse est accomplie en utilisant la fonction génératrice de la distribution du degré du graphe. Paul Erdős et Alfred Rényi [3] ont montré que la distribution du degré d'un sommet dans un graphe aléatoire est une Loi de Poisson. Malheureusement les graphes aléatoires ne sont pas représentatifs des réseaux d'aujourd'hui, en particulier dans les réseaux P2P, les papiers [4][5] présentent une évaluation quantitative du comportement des systèmes P2P basés sur un "crawler" pour extraire la topologie du réseau de Gnutella. Leur conclusion implique que le réseau Gnutella est très loin de ressembler à un graphe aléatoire possédant une Loi de Poisson.

Plus formellement : Un graphe aléatoire est une collection de sommets V , avec des arêtes reliant les sommets par paire au hasard E . On définit donc $G = (V, E)$ un graphe aléatoire possédant $|V| = n$ sommets et une arête existe entre 2 sommets avec probabilité p et absente avec probabilité $1 - p$, ce modèle est le plus connu il est appelé $G(n, p)$.

Loi de Poisson : Soit λ un réel strictement positif. Une variable aléatoire X suit une loi de Poisson de paramètre λ , notée $\mathcal{P}(\lambda)$, si, et seulement si :

$$\forall k \in \mathbb{N}, \mathcal{P}(X = k) = \frac{\lambda^k}{k!} e^{-\lambda}$$

Donc les graphes aléatoires ne suffisent pas, ils vont introduire des graphes aléatoires généralisés, un graphe aléatoire généralisé représente une famille de graphes possédant N noeuds et ayant un degré distribué arbitrairement. Ceux-ci doivent être vus comme une moyenne sur la totalité des possibles graphes. Les travaux présentés permettent d'évaluer certains résultats tels que, le nombre moyen de

messages à travers le réseau P2P et la probabilité qu'une requête de recherche atteigne la ressource demandée. Leur approche est validée grâce à la simulation en montrant que la précision du modèle proposé s'améliore en fonction de la taille du réseau P2P.

L'article[8] rentre encore plus dans les détails. Leur approche est aussi basée sur les fonctions génératrices, dont la principale est la fonction génératrice de la distribution du degré du graphe $G_0(x)$. ils définissent :

$$G_0(x) = \sum_{k=0}^{\infty} p_k x^k$$

ou p_k est la probabilité qu'un noeud soit de degré k . On a $G_0(1) = 1$. Le degré moyen z est donné par :

$$z = \sum_k k p_k = G_0'(1)$$

de manière générale on a :

$$\sum_k k^n p_k = \left[\left(x \frac{d}{dx} \right)^n G_0(x) \right]_{x=1}$$

Si on choisit m sommets au hasard dans le graphe, alors la somme des distributions des degrés des sommets est $[G_0(x)]^m$, par exemple prenons le simple cas de deux sommets $[G_0(x)]^2$ on obtient :

$$\begin{aligned} [G_0(x)]^2 &= [\sum_k p_k x^k]^2 = \sum_{jk} p_j p_k x^{j+k} \\ &= p_0 p_0 x^0 + (p_0 p_1 + p_1 p_0) x^1 \\ &+ (p_0 p_2 + p_1 p_1 + p_2 p_0) x^2 \\ &+ (p_0 p_3 + p_1 p_2 + p_2 p_1 + p_3 p_0) x^3 + \dots \end{aligned}$$

Une relation importante ensuite qui est nécessaire pour la continuité de leur travaux, c'est la distribution du nombre de descendant des descendants du premier noeud choisi aléatoirement, cette distribution est générée par la fonction :

$$G_1(x) = \frac{G_0'(x)}{G_0'(1)} = \frac{1}{z} G_0'(x)$$

on peut l'écrire de cette manière grâce aux opérations données précédemment :

$$\sum_k p_j [G_1(x)]^k = G_0(G_1(x))$$

similairement la distribution du nombre de troisième voisins est donnée par $G_0(G_1(G_1(x)))$ etc... Donc la moyenne z_2 du nombre de second voisins est :

$$z_2 = \left[\frac{d}{dx} G_0(G_1(x)) \right]_{x=1} = G_0'(1) G_1'(1) = G_0''(1)$$

2.1.1 Nombres de voisins et longueur de chemin

Nous allons regarder maintenant le nombre de voisins qui sont à distance m du premier noeud choisit aléatoirement. Nous avons vu précédemment que le nombre de voisins à distance un et deux est donné respectivement par les fonctions $G_0(x)$ et $G_0(G_1(x))$, donc le nombre de voisins à distance m est donné par $G_0(G_1(G_1(\dots G_1(x) \dots)))$, avec $m-1$ itérations de la fonction G_1 . Si on défini

$G^{(m)}(x)$ la fonction génératrice des m^{eme} voisins, alors nous avons :

$$\begin{aligned} G^{(m)}(x) &= G_0(x) && \text{pour } m = 1 \\ &= G^{(m-1)}(x)(G_1(x)) && \text{pour } m \geq 2 \end{aligned}$$

Alors le nombre moyen z_m de m^{eme} voisins est

$$z_m = \frac{dG^{(m)}}{dx} \Big|_{x=1} = G_1'(1)G^{(m-1)}(1) = G_1'(1)z_{m-1}$$

Du fait de la condition initiale $z_1 = z = G_0'(1)$, on obtient :

$$z_m = [G_1'(1)]^{m-1}G_0'(1) = \left[\frac{z_2}{z_1}\right]^{m-1}z_1$$

De ce résultat on peut estimer la longueur λ du plus court chemin entre deux sommets du graphe.

Soit $1 + \sum_{m=1}^l z_m = N$ avec N le nombre de sommets du graphes, en utilisant l'équation précédente on obtient :

$$\lambda = \frac{\ln[(N-1)(z_2 - z_1) + z_1^2] - \ln(z_1^2)}{\ln(z_2/z_1)}$$

2.2 Théorie de la percolation

Dans l'article [11] il décrit la théorie de la percolation comme un système complexe. Pour introduire la théorie de la percolation il faut considérer une grille $(n \times n)$ dont seul les voisins directs peuvent communiquer (haut, bas, gauche, droite).

- Probabilité p que l'arête soit ouverte.
- Probabilité $1 - p$ que l'arête soit fermé.

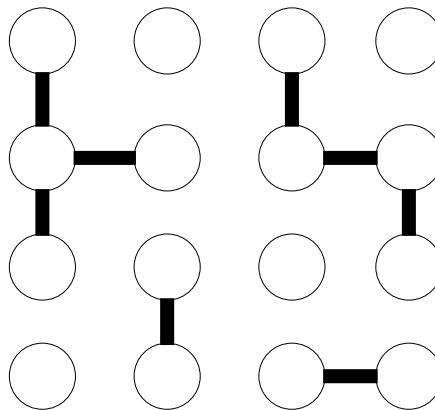


FIG. 2.1 – Percolation

En fonction de la valeur de p nous voyons se développer des composantes connexes. Plus p sera grand plus il y aura de composante connexe, la question qui se pose, quelle valeur de p faut il prendre pour qu'une seule composante connexe apparaisse ?

Cette valeur de p est appelé p_c , lorsque nous obtenons p_c nous avons l'apparition d'une composante connexe mais aussi nous pouvons dire qu'il y aura un minimum de chemin redondant entre les noeuds.

Pour trouver p_c nous devons émettre une hypothèse :(critère de transition)

Chaque noeud i connecté à un noeud j est aussi connecté à au moins un autre noeud, autrement les composantes connexes sont fragmentés.

C'est à dire la grande composante connexe commence à apparaître quans tous les noeuds ont un degré moyen k égal a 2.

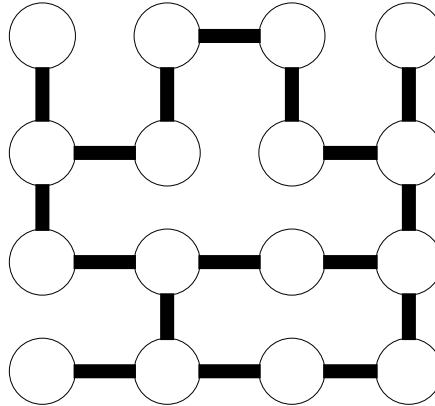


FIG. 2.2 – Apparition de la composante connexe

Soit $P(k)$ la probabilité que le noeud est k voisins (ouverts ou fermés).

Soit $P_e(k)$ la probabilité qu'un noeud possède k arête ouverte.

$$P_e(k) = \sum_{n=k}^{\infty} p^k C_n^k (1-p)^{n-k} P(n)$$

Ici nous voyons apparaître une loi binomial.

En mathématiques, une loi binomiale de paramètres n et p correspond au modèle suivant :

On renouvelle n fois de manière indépendante une épreuve de Bernoulli de paramètre p (expérience aléatoire à deux issues possibles, généralement dénommées respectivement "succès" et "échec", la probabilité d'un succès étant p , celle d'un échec étant $q = 1 - p$). On compte alors le nombre de succès obtenus à l'issue des n épreuves et on appelle X la variable aléatoire correspondant à ce nombre de succès.

La variable aléatoire suit une loi de probabilité définie par :

$$p(k) = P(X = k) = C_n^k p^k q^{n-k}$$

Pour trouver p_c il faut utiliser le critère de transition de phase, celui-ci est donné par $\frac{\langle k^2 \rangle_e}{\langle k \rangle_e} = 2$. Ici $\langle k \rangle$ exprime la moyenne c'est à dire $\langle k \rangle = z = \sum_k kP(k) = G'_0(1)$.

$$\begin{aligned} \langle k \rangle_e &= \sum_{k=0}^{\infty} k \sum_{n=k}^{\infty} p^k C_n^k (1-p)^{n-k} P(n) \\ &= \sum_{n=0}^{\infty} P(n) \sum_{k=0}^n p^k C_n^k (1-p)^{n-k} k \\ &= p \sum_{n=0}^{\infty} n P(n) \\ &= p \langle k \rangle \end{aligned}$$

et :

$$\begin{aligned} \langle k^2 \rangle_e &= \sum_{k=0}^{\infty} k^2 \sum_{n=k}^{\infty} p^k C_n^k (1-p)^{n-k} P(n) \\ &= \sum_{n=0}^{\infty} P(n) \sum_{k=0}^n p^k C_n^k (1-p)^{n-k} k^2 \\ &= \sum_{n=0}^{\infty} P(n) (np(1-p) + n^2 p^2) \\ &= p^2 \langle k^2 \rangle + p(1-p) \langle k \rangle \end{aligned}$$

Donc on obtient :

$$p_c \frac{\langle k^2 \rangle}{\langle k \rangle} + (1-p_c) = 2 \Rightarrow p_c = \frac{1}{\alpha - 1}$$

Avec $\alpha = \frac{\langle k^2 \rangle}{\langle k \rangle}$ ou α représente le ratio entre le second moment et le premier du graphe.

Finalement, la théorie de la percolation nous a permis de calculer la valeur de p pour laquelle nous passons de plusieurs composantes connexes fragmentées à une seule ayant le moins de redondances dans les chemins possible.

2.3 Marche aléatoire

Le principe de la marche aléatoire consiste à ce qu'un noeud transmette le message à seulement un de ses voisins choisit aléatoirement (ou de façon intelligente) au lieu de tous ses voisins. Nous pouvons revenir à un phénomène de branchement si chaque noeud n'envoie pas à un seul voisin mais à plusieurs. Donc nous allons nous attacher à la façon intelligente de transmettre le message à un voisin.

2.3.1 Marche intelligente

Dans la marche intelligente ou Preferential Walk[20], chaque pair assigne une valeur de confiance à ses voisins. Au départ cette valeur est équiprobable entre tous les voisins, c'est à dire que si A possède 3 voisins B, C, D alors chacun a une valeur de confiance de 1/3, si par la suite le voisin C répond à la requête de A alors, A augmentera sa valeur de confiance et nous aurons B=1/4, C=1/2, D=1/4.

Chemin	valeur de confiance	nouvelle valeur
A à B	1/3	1/4
A à C	1/3	1/2
A à D	1/3	1/4

simulation

Pour valider leur méthode, elle a été comparé avec 2 autre Marches :

- la marche aléatoire, un pair transmet le message à un autre pair choisit aléatoirement.
- la marche sur le degré, un pair envoie le message a son voisin de plus grand degré.

Résultat

La marche intelligente obtient de meilleure résultat :

- Les pairs utilisant cette méthode ont plus de chance d'obtenir la ressource désirée.
- Le nombre de messages envoyés au pair possédant une petite valeur de confiance est réduit.
- On obtient un modèle d'apprentissage etant réévalué à chaque requête.

Chapitre 3

Modélisation

Pour ce chapitre, nous devons définir ce qu'est un réseau peer to peer.

3.1 Le modèle Pair-à-pair

Les réseaux Pair-à-pair (P2P) ont récemment émergé comme un nouveau paradigme pour construire des applications réseaux.

Ces derniers temps les mesures démontrent que les applications de partage de ressources P2P ont généré une réelle augmentation du trafic internet. Les informations récupérées depuis SD-NAP [1] prouvent que le trafic dominant observé par les fournisseurs d'accès internet est associé avec les applications de partage de ressources P2P et il est raisonnable de penser que ce trafic continuera d'augmenter. De plus, les réseaux P2P ont une influence sur la manière de créer des nouvelles applications réseaux. Les noeuds appelés peer dans un système P2P diffèrent de plusieurs façons par rapport à un noeud dans une application réseaux traditionnels client/serveur. Ceci est dû au fait que chaque peer peut travailler comme server ou client.

Dans les réseaux non structurés, décentralisés P2P chaque peer est responsable de maintenir un index des ressources qu'il peut distribuer avec les autres peers dans le réseau. Le manque d'une base de données centrales rend le processus de recherche plus difficile.

Pour représenter le réseau P2P, nous utiliserons la théorie des graphes, c'est à dire :

- chaque pair est représenté par un noeud du graphe.
- chaque liaison entre peers est représentée par une arête du graphe.

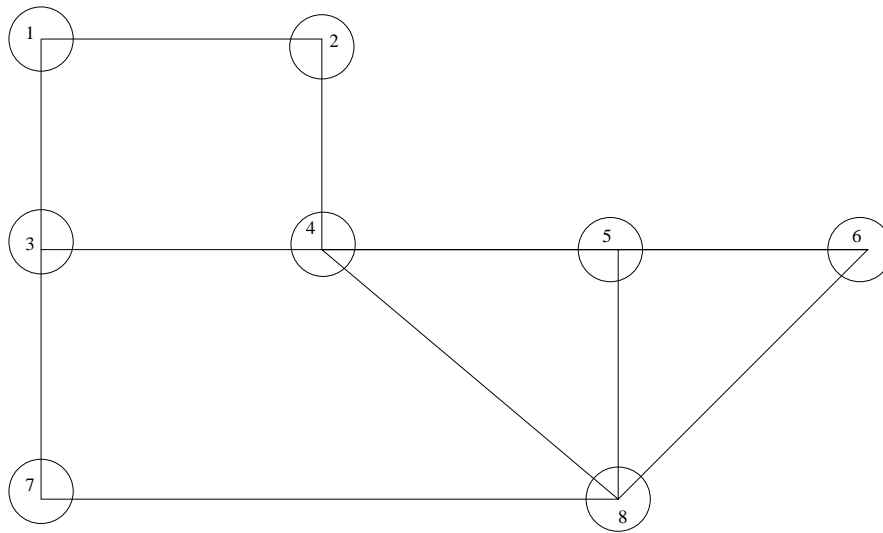


FIG. 3.1 – Le réseau P2P

Ainsi, nous possédons un jeu de N peers actifs correspondant à N noeuds d'un graphe et il existe une arête entre 2 peers si il y a une connexion entre eux deux.

Remarque dans les réseaux P2P la topologie est changeante, c'est à dire qu'un peer peut se connecter/déconnecter à tout moment.

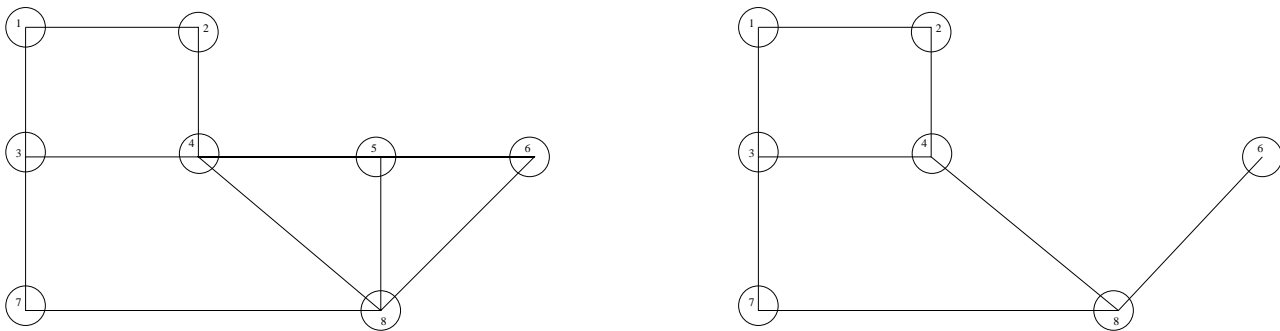


FIG. 3.2 – Le réseau P2P dynamique

Pour nos recherches, nous négligerons ce fait, en supposant que les opérations effectuées se feront dans un laps de temps inférieur au changement de la topologie. Maintenant, on peut considérer le réseau comme un graphe fini avec N peers actifs où les sommets représentent les applications P2P et les arêtes les connexions entre peers (voir Fig 3.1).

3.1.1 Recherche par inondation

La stratégie classique de recherche correspond à l'inondation. Pour parfaire celle-ci lorsque qu'un peer envoie une requête de recherche alors elle est transmise à tous ses voisins dans le réseaux, ces mêmes voisins font suivre à tous leurs voisins jusqu'à la distance max_ttl .

Voici un algorithme simple de l'inondation :

- Un noeud cherchant une ressource initie la requête de recherche.
- Il fixe le Time To Live (TTL).
- Il envoie la requête à tous ses voisins.
- Chaque noeud qui reçoit la requête décrémente le TTL d'un et,

- Fait suivre la requête à tous ses voisins.
- La requête continue jusqu'à ce que l'objet soit atteint ou que le TTL = 0.

Définition :

Le champ Time To Live (TTL), temps de vie.

Il s'agit d'une donnée placée au niveau de l'entête du paquet IP qui indique le nombre de routeurs traversés. C'est une valeur (définie par l'application qui utilise la couche IP, les systèmes Unix la fixent généralement à 64) qui est décrémentée à chaque passage d'un routeur. Dès que cette valeur atteint 0, le paquet est détruit par le routeur.

Ceci évite qu'un paquet boucle à l'infini s'il existe un problème au niveau de son routage, et a aussi permis l'utilisation d'outils traceroute.

Les stratégies de recherche dans ces réseaux P2P ont besoin d'être implantées avec précaution, car il ne faut ni trop saturer le réseau ni obtenir aucune réponse lors de la recherche. Avec l'approche par inondation un noeud peut recevoir la requête plusieurs fois de différents noeuds. Un problème soulevé est la valeur du TTL qui est acquise sans se préoccuper de la taille actuelle du réseau P2P.

3.1.2 Gnutella

Gnutella est un protocole informatique décentralisé de recherche et de transfert de fichiers peer-to-peer (aussi appelé P2P). Il a été imaginé en 2000 par Tom Pepper et Justin Frankel alors programmeurs pour la société Nullsoft, qui a également édité WinAmp. Le protocole Gnutella continue d'évoluer permettant en outre une diminution sensible de la bande passante utilisée. Parmi ses améliorations, on peut citer la gestion des ultrapeers, GUESS et GWebCache qui permet de manière décentralisée à des clients Gnutella de connaître d'autres machines auxquelles se connecter.

Gnutella v0.4

Dans Gnutella v0.4, chaque poste agit comme un serveur et un client et est nommé *servent* (contraction de serveur et client).

Chaque *servent* est connecté à un ensemble de voisins. Pour lancer une recherche, un *servent* interroge tous ses voisins en leur envoyant un message de recherche. Ses voisins font de même avec leurs propres voisins. Un champ TTL (Time To Live) est associé au message de recherche pour comptabiliser le nombre de retransmissions restantes. Quand la valeur du TTL est nulle, le message n'est plus renvoyé. Cette méthode de propagation est appelée inondation.

Les *servents* ayant des fichiers qui répondent à la requête renvoient leur réponse (nom du fichier + leur adresse IP) au voisin qui leur a retransmis la requête. La réponse remonte ainsi de proche en proche jusqu'au *servent* qui a initié la requête.

Le *servent* initiateur de la requête va ensuite choisir les fichiers à télécharger en envoyant directement une requête de téléchargement au *servent* qui possède le fichier.

Cependant cette inondation est coûteuse en bande passante et les recherches sont plus lentes que dans les réseaux centralisés (Napster).

Gnutella v0.6

Ultrapeers

Ce qui conduit, les développeurs de Limewire à proposer en 2001 le système d'ultrapeers. La gestion des ultrapeer (ou superpeer) est incluse dans la version 0.6 du protocole Gnutella. Les *servents* sont

divisés en deux catégories : les ultrapeers (ou superpeer) qui sont des serveurs stables ayant une bonne connectivité et les clients (ou noeuds feuilles). Les leafs sont rattachés à 3 ultrapeers et les ultrapeers admettent entre 30 à 45 clients et 30 ultrapeers. Les ultrapeers indexent le contenu des clients auxquels ils sont rattachés et répondent aux requêtes de recherche en incluant les documents de leurs clients.

Les requêtes de recherches reçues par les ultrapeers ne sont retransmises qu'aux autres ultrapeers. Ainsi un client interrogeant un ultrapeer peut rechercher sur 30 autres clients avec une seule requête. Si la requête est retransmise aux 30 autres ultrapeers auxquels ils sont connectés, sa recherche atteint le contenu de 900 clients... Autre point non négligeable, les clients ne reçoivent plus de requêtes de recherche, on dit qu'ils sont protégés (shielded) par leur ultrapeer. Cela permet en outre aux utilisateurs connectés à Internet en bas débit de pouvoir utiliser Gnutella sans diminuer leurs performances.

Caractéristique

D'après [18] en fixant le TTL à 7 on parcourt 95% du réseau.

Le réseau GNUTELLA suit une loi de puissance pour décrire la distribution du degré.

$$P(k) = Ck^{-t} e^{-\frac{k}{v}}$$

ou C , t , et v sont des constantes et,

C : facteur de normalisation

$e^{-\frac{k}{v}}$: facteur de coupure exponentiel requis pour modéliser les réseaux.

Pour Gnutella, l'exposant de la loi de puissance t est estimé au plus bas à 1,4 et au plus haut à 2,3 et v est compris entre 100 et 1000.

Voici notre modèle utilisé dans le papier

3.2 Le modèle

Nous devons émettre plusieurs hypothèses.

Hypothèse 1 : Lors d'une requête de recherche, la probabilité des messages transmis dans le réseau P2P passent par un noeud déjà parcouru est négligeable du fait que le réseau est très grand.

Hypothèse 2 : Le temps d'une opération est court de ce fait le réseau ne change pas, c'est à dire qu'il n'y a aucune déconnexion/connexion sur le réseau.

Donc nous possédons un graphe $G = (V, E)$ où V représente les sommets/peers et E représente les arêtes/connexion entre peers.

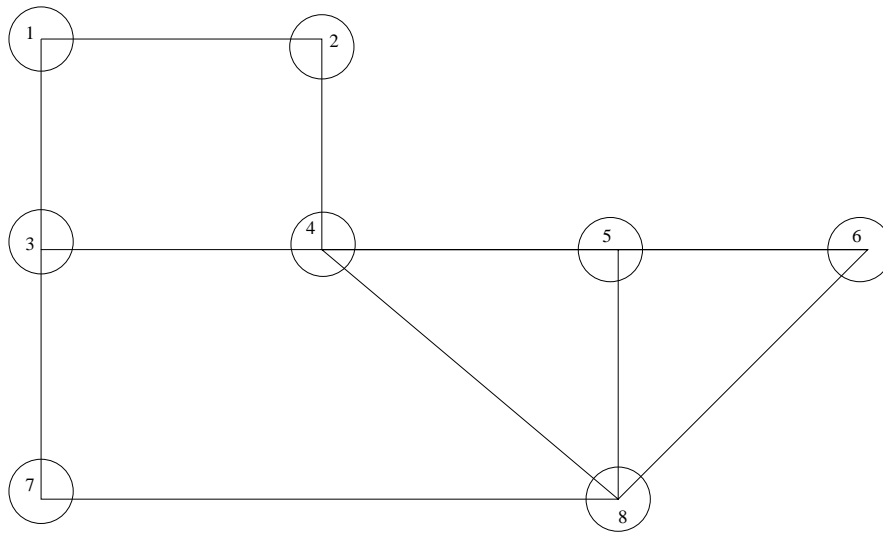


FIG. 3.3 – Le graphe représentant le réseau P2P

Du fait de l'hypothèse (1) lors d'une requête de recherche, celle-ci peut se représenter sous forme d'arbre.

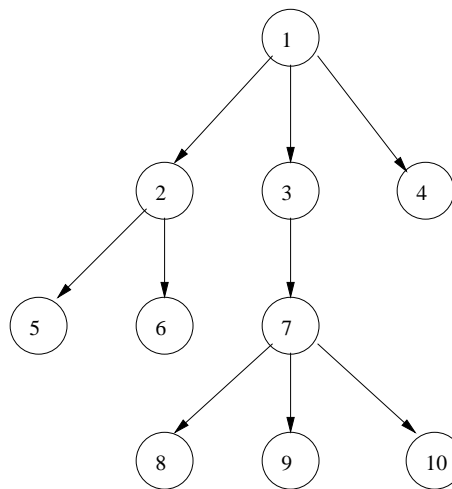


FIG. 3.4 – L'arbre représentant la recherche

Arbres

On nomme arbre un graphe connexe non orienté et acyclique (sa forme évoque en effet la ramification des branches d'un arbre). On distingue deux types de sommets dans un arbre

- les feuilles dont le degré est 1 ;
- les noeuds internes dont le degré est supérieur à 1.

D'autres définitions équivalentes sont possibles. Par exemple :

- basée sur le nombre d'arêtes : Un arbre est un graphe connexe non orienté possédant $n - 1$ arêtes.

On peut démontrer qu'il y a $n^{(n-2)}$ arbres numérotés à n sommets, c'est le nombre de Cayley (aussi $n^{(n-1)}$ arbre enracinés).

Arborescences

Une arborescence est un graphe orienté acyclique connexe où chaque sommet est de degré entrant au plus 1. Dans ce cas, un seul sommet est de degré entrant 0, il est appelé racine de l'arborescence.

Grâce aux hypothèses nous obtenons donc une recherche en forme d'arbre, nous allons pouvoir utiliser les processus de branchement pour modéliser nos travaux.

3.3 Processus de Branchement

3.3.1 Définition

Les processus de branchement sont une classe particulière de chaînes de Markov. Ils sont utiles dans la modélisation de processus naturel pouvant être vus comme des arbres aléatoires. Ces arbres modélisent également le déroulement de certains algorithmes ou de certains programmes parallèles.

Définition :

En mathématiques, une chaîne de Markov est un processus stochastique possédant la propriété markovienne. Dans un tel processus, la prédiction du futur à partir du présent ne nécessite pas la connaissance du passé. Elles ont pris le nom de leur découvreur, Andrei Markov.

Une chaîne de Markov en temps discret est une séquence X_1, X_2, X_3, \dots de variables aléatoires. L'ensemble de leurs valeurs possibles est appelé l'espace d'états, la valeur X_n étant l'état du processus au moment n

Si la distribution de probabilité conditionnelle de X_{n+1} sur les états passés est une fonction de X_n seul, alors :

$$P(X_{n+1} = x | X_0, X_1, X_2, \dots, X_n) = P(X_{n+1} = x | X_n).$$

où x est un état quelconque du processus. L'identité ci-dessus reconnaît la probabilité markovienne.

Il y a de nombreuses variantes. Dans sa version la plus simple (processus de Galton-Watson), historiquement, les premiers arbres aléatoires considérés ont été les arbres généalogiques : à la génération zéro il y a un individu. On se donne une suite (p_k) dans $[0, 1]$ telle que $\sum_{k \geq 0} p_k = 1$. p_k représente la probabilité qu'un individu ait k enfants.

Plus formellement le processus se décrit de la manière suivante.

Soit X une distribution discrète à valeurs dans \mathbb{N} , de moyenne m et de fonction génératrice α telle que $\alpha(z) = \sum_{k \geq 0} p_k z^k$. On considère que dans une population, un individu peut donner naissance à un nombre aléatoire de descendants directs, répartis selon la distribution de X . Les descendances (directes) d'individus distincts sont stochastiquement indépendantes.

3.3.2 Distribution du nombre de descendants par génération

On suppose qu'il y a un individu unique au départ, et on dénombre ses descendants par générations. Soit $Z(n)$ le nombre de descendants à la génération n on s'aperçoit facilement que le processus $\{Z(n), n \in \mathbb{N}\}$ est une chaîne de Markov à valeurs dans \mathbb{N} .

Les fonctions génératrices des $Z(n)$ vérifient :

$$\begin{aligned}
F_{n+1}(z) = \mathbb{E}(z^{Z(n+1)}) &= \sum_{k=0}^{\infty} \mathbb{P}(Z(n) = k) \mathbb{E}(z^{Z(n+1)} | Z(n) = k) \\
&= \sum_{k=0}^{\infty} \mathbb{P}(Z(n) = k) \mathbb{E}(z^{\sum_{j=1}^k X_j}) \\
&= \sum_{k=0}^{\infty} \mathbb{P}(Z(n) = k) \alpha(z)^k \\
&= F_n(\alpha(z)).
\end{aligned}$$

en conséquence, par récurrence :

$$F_n(z) = \alpha \circ \alpha \circ \dots \circ \alpha(z) = \alpha^{(n)}(z), \text{ et } F_0(z) = 1$$

En dérivant l'équation ci-dessus par rapport à z , on en déduit une récurrence qui conduit aux premiers moments :

$$\mathbb{E}(Z(n)) = m^n$$

3.3.3 Distribution du nombre total de descendants

Soit $T(n) = \sum_{m=0}^n Z(m)$ le nombre total de descendants dans l'arbre jusqu'à la génération n . On s'intéresse à la distribution limite de la variable aléatoire $T(n)$ quand $n \rightarrow \infty$. Cette limite existe car la suite $T(n)$ est stochastiquement croissante. Mais elle peut être "impropre", c'est-à-dire donner une probabilité non nulle à l'infini.

$$\text{Pour } m \neq 1, \\ T(n) = \sum_{i=0}^n Z(i)$$

$$\begin{aligned}
\mathbb{E}(T(n)) &= \mathbb{E}(\sum_{i=0}^n Z(i)) \\
&= \sum_{i=0}^n \mathbb{E}(Z(i)) \\
&= \sum_{i=0}^n m^i
\end{aligned}$$

$$\mathbb{E}(T(n)) = \frac{m^{n+1} - 1}{m - 1}$$

Par conséquent, si n croît, alors :

- si $m \geq 1$, le nombre moyen de descendants tend vers l'infini, ce qui suggère que l'arbre devient infini avec une probabilité non nulle, ou bien il reste fini mais avec une taille d'espérance infinie, on dit que le processus de branchement est surcritique (critique quand $m = 1$).
- si $m < 1$, le nombre moyen de descendants converge vers $1/(1 - m)$, ce qui implique que l'arbre se termine avec probabilité 1, on dit que le processus de branchement est sous-critique.

En résumé voici une figure qui représente :

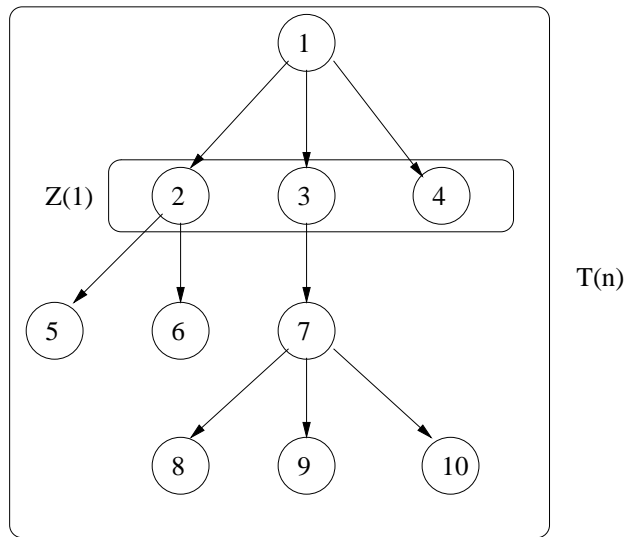


FIG. 3.5 – Processus de branchement

Chapitre 4

Stratégie de recherche

4.1 Résultat

Nous allons voir dans cette section plusieurs méthodes pour réduire le nombre de messages sur le réseau pair à pair.

Tout d'abord une méthode basée sur la valeur de TTL, et ensuite nous verrons comment réduire le nombre de messages avec une approche probabiliste.

4.1.1 Choix du TTL

Un des problèmes avec l'inondation est de choisir la bonne valeur du TTL pour ne pas trop saturer le réseau. Le TTL est normalement fixé sans tenir compte du nombre actuel de noeuds. Pour résoudre ce problème on peut donner à chaque noeud l'opportunité d'estimer la valeur appropriée pour son TTL grâce à des informations locales. En exploitant les résultats de Newman[8] qui donnent une manière formelle d'estimer la longueur λ du plus court chemin entre 2 sommets aléatoires.

Voici l'équation :

$$\lambda = \frac{\ln[(N-1)(z_2 - z_1) + z_1^2] - \ln(z_1^2)}{\ln(z_2/z_1)}$$

avec,

- N le nombre moyen de noeuds dans le réseau.
- z_1 nombre de voisins à distance 1.
- z_2 nombre de voisins à distance 2.

Chaque noeud peut estimer le nombre de voisins à distance 1 et 2 en envoyant au début des paquets avec TTL = 1 et TTL = 2. Ainsi on peut obtenir λ avant le démarrage de la requête de recherche. Il est évident que cette proposition pour le choix du TTL améliore considérablement la stratégie d'inondation classique.

4.1.2 Inondation Probabiliste

L'inondation probabiliste, comme son nom l'indique va inonder le réseau mais seulement une partie, certains noeud ne seront pas empruntés. soit $P(n)$ la probabilité qu'un noeud ait n voisins

(degré n).

soit p la probabilité de transmettre le message d'un noeud à un voisin, p est appelée le facteur probabiliste, car c'est lui qui va fixer la portion de réseau à inonder, plus ou moins grande.

On cherche à savoir qu'elle est la probabilité d'avoir n_k noeuds à l'étage k , pour ceci nous devons procéder étape par étape (ou étage par étage) afin d'obtenir la formule exacte.

On obtient donc la probabilité que l'origine envoie à n_0 voisins :

$$P_e^{(0)}(n_0) = \sum_{h=n_0}^{\infty} P(h) C_h^{n_0} p^{n_0} (1-p)^{h-n_0}$$

Pour que l'origine envoie à n_0 voisins il faut que $P(h) \geq n_0$ c'est à dire que l'origine ait au moins n_0 voisins, d'où le démarrage de la somme à $h = n_0$ ensuite parmi c'est h il faut en choisir n_0 (arrangement), ensuite il faut que les n_0 transmettent chacun le message p^{n_0} et que les $h - n_0$ ne le transmettent pas $(1-p)^{h-n_0}$

la probabilité que les n_0 premiers voisins envoient à n_1 voisins c'est à dire que nous descendons d'un étage dans le branchement :

$$P_e^{(1)}(n_1) = P_e^{(0)}(n_0) \times \prod_{i=1}^{n_0} \sum_{h=n_{1,i}}^{\infty} P(h_i) C_{h_i}^{n_{1,i}} p^{n_{1,i}} (1-p)^{h-n_{1,i}}$$

avec $n_1 = \sum_1^{n_0} n_{1,i}$ c'est à dire chaque $n_{1,i}$ est le nombre de descendants d'un fils de n_0 exemple $n_{1,0}$ est le nombre de fils du premier descendant de l'origine. Et pour avoir changé d'étage il faut avoir parcouru le premier étage d'où le $P_e^{(0)}(n_0)$

Continuons :

La probabilité que les n_1 second voisins envoient à n_2 voisins (second étage) :

$$P_e^{(2)}(n_2) = P_e^{(1)}(n_1) \times \prod_{i=1}^{n_1} \sum_{h=n_{2,i}}^{\infty} P(h_i) C_{h_i}^{n_{2,i}} p^{n_{2,i}} (1-p)^{h-n_{2,i}}$$

Donc la probabilité que les n_{k-1} ($k-1$)^{me} voisins envoient à n_k voisins est

$$\begin{aligned} P_e^{(k)}(n_k) &= P_e^{(k-1)}(n_{k-1}) \times \prod_{i=1}^{n_{k-1}} \sum_{h=n_{k,i}}^{\infty} P(h_i) C_{h_i}^{n_{k,i}} p^{n_{k,i}} (1-p)^{h-n_{k,i}} \\ &= P_e^{(0)}(n_0) \prod_{j=0}^{k-1} \prod_{i=1}^{n_j} \sum_{h=n_{j+1,i}}^{\infty} P(h_i) C_{h_i}^{n_{j+1,i}} p^{n_{j+1,i}} (1-p)^{h-n_{j+1,i}} \end{aligned}$$

Voici une figure résumant les différentes variables.

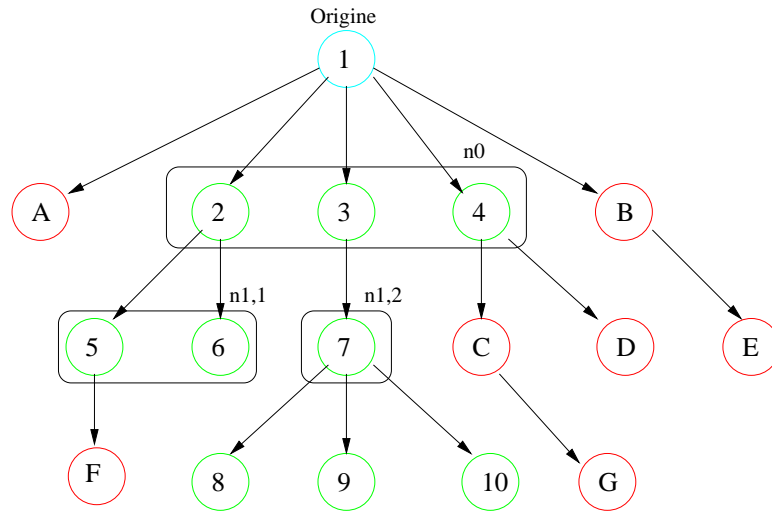


FIG. 4.1 – Inondation probabiliste

L'origine en bleue représente aussi l'initiateur de la requête de recherche, les sommets en rouge sont les sommets non parcourus par la requête, les sommets verts du premier étage appartiennent à n_0 ensuite nous voyons les descendants du premier descendant de n_0 c'est à dire $n_{1,1}$ puis les descendants du deuxième sommet de n_0 c'est $n_{1,2}$, avec ces éléments il est facile de comprendre la suite.

4.1.3 Choix de p

Pour réduire le nombre de messages dans le réseau pair à pair, il faut choisir p de manière intelligente. c'est à dire :

- si p est trop proche de 1 alors nous retombons dans le cas de l'inondation.
- si p est trop petit alors nous ne pouvons pas atteindre tous les pairs du réseau on dit que le réseau se déconnecte.

On appelle p_c la probabilité critique, cela signifie qu'à partir de cette valeur, on obtient un réseau connexe, c'est à dire que n'importe quel noeud du réseau peut être atteint.

il a été montré dans [11] que en dessinant la courbe p_c en fonction de ν indice de coupure de Gnutella, la valeur de p_c avoisine 0, 1. cela veut dire que nous pouvons réduire le nombre de messages de 90% en moyenne et quand même avoir une chance d'atteindre tous les noeuds du réseau.

4.1.4 Popularité d'une ressource

Dans cette partie, nous cherchons à évaluer si une ressource se trouve sur r_i pairs du réseau à l'étage i .

Hypothèse : Les ressources sont uniformément distribuées parmi les peers.

Soit β la probabilité que la ressource se trouve sur un peer.

On sait que la probabilité que l'origine envoie a n_0 voisins est :

$$P_e^{(0)}(n_0) = \sum_{h=n_0}^{\infty} P(h) C_h^{n_0} p^{n_0} (1-p)^{h-n_0}$$

donc nous avons n_0 voisins, nous cherchons à savoir si la ressource se trouve sur r_0 de ces n_0 voisins alors on obtient la probabilité suivante :

$$Q_e^{(0)}(n_0, r_0) = P_e^{(0)}(n_0) \beta^{r_0} (1 - \beta)^{n_0 - r_0} C_{r_0}^{n_0} \text{ avec } 1 \leq r_0 \leq n_0$$

Tout d'abord il faut envoyer le message à n_0 voisins ($P_e^{(0)}(n_0)$), il faut que r_0 possède la ressource β^{r_0} et que le reste n'ait pas la ressource ($(1 - \beta)^{n_0 - r_0}$).

Supposons maintenant que la ressource ne se trouvait pas sur ces n_0 voisins nous allons devoir descendre d'un étage, donc nous supposons que les n_0 voisins envoient à n_1 voisins et parmi ces n_1 , seul r_1 possèdent la ressource. On obtient :

$$\begin{aligned} Q_e^{(1)}(n_1, r_1) &= P_e^{(0)}(n_0) \times \beta^0 (1 - \beta)^{n_0 - 0} C_0^{n_0} \times P_e^{(1)}(n_1) \times \beta^{r_1} (1 - \beta)^{n_1 - r_1} C_{r_1}^{n_1} \\ &= Q_e^{(0)}(n_0, 0) \times P_e^{(1)}(n_1) \times \beta^{r_1} (1 - \beta)^{n_1 - r_1} C_{r_1}^{n_1} \\ &\text{avec } 1 \leq r_1 \leq n_1 \end{aligned}$$

Donc la probabilité qu'il y est r_k noeuds possédant la ressource parmi les n_k noeuds finaux est :

$$\begin{aligned} Q_e^{(k)}(n_k, r_k) &= P_e^{(k-1)}(n_{k-1}) \times \beta^0 (1 - \beta)^{n_{k-1} - 0} C_0^{n_{k-1}} \times P_e^{(k)}(n_k) \times \beta^{r_k} (1 - \beta)^{n_k - r_k} C_{r_k}^{n_k} \\ &= Q_e^{(k-1)}(n_{k-1}, 0) \times P_e^{(k)}(n_k) \times \beta^{r_k} (1 - \beta)^{n_k - r_k} C_{r_k}^{n_k} \\ &= \prod_{j=0}^k P_e^{(j)}(n_j) (1 - \beta)^{n_j - r_j} \times \beta^{r_k} C_{r_k}^{n_k} \\ &\text{avec } \forall j \neq k, r_j = 0 \text{ et } 1 \leq r_k \leq n_k \end{aligned}$$

4.2 Exemple

Nous avons besoin des fonctions génératrices pour aller plus loin dans les résultats. Soit $G_0(x)$ la fonction génératrice pour la probabilité de distribution du degré d'un noeud.

$$G_0(x) = \sum_{k=0}^{\infty} P(k) x^k$$

on sait que la moyenne du degré de distribution m d'un noeud est donnée par :

$$m = \sum_k k P(k) = G'_0(1)$$

Pour modéliser les recherches, nous allons utiliser plusieurs lois.

4.2.1 Loi de Poisson

Nous possédons un graphe avec une loi de distribution du degré est binomial ou dite de Poisson, c'est à dire la probabilité de l'existence d'une arête entre 2 sommets est donnée par $\alpha = z/N$.

Nous devons utiliser la loi de Poisson pour fixer le degré des noeuds.

$$\begin{aligned} G_0(x) &= \sum_{k=0}^N C_n^k \alpha^k (1 - \alpha)^{N-k} x^k \\ &= (1 - \alpha + \alpha x)^N = e^{z(x-1)} \end{aligned}$$

on sait que $G'_0(1) = z$ c'est le degré moyen d'un noeud et que :

$$P(k) = z^k e^{-z} / k!$$

donc dans les processus de branchement on obtient :

$$P_e(k) = pz^k e^{-z} / k!$$

p étant la probabilité de branchement (transmettre le message).

Du coup il est évident que le degré moyen dans les processus de branchement vaut zp .
Soit $Z(n)$ le nombre de descendants à la génération n , on obtient $\mathbb{E}(Z(n)) = (zp)^n$.

Nombre de messages

Soit $T(n) = \sum_{m=0}^n Z(m)$ le nombre total de messages donc :

$$T(n) = \sum_{i=0}^n Z(i)$$

$$\mathbb{E}(T(n)) = \mathbb{E}(\sum_{i=0}^n Z(i))$$

$$= \sum_{i=0}^n \mathbb{E}(Z(i))$$

$$= \sum_{i=0}^n (zp)^i$$

$$\mathbb{E}(T(n)) = \frac{(zp)^{n+1} - 1}{(zp) - 1}$$

Extinction de l'arbre

Par conséquent, si n croît, alors :

- si $zp \geq 1$, le nombre moyen de descendants tend vers l'infini, ce qui suggère que l'arbre devient infini avec une probabilité non nulle, ou bien il reste fini mais avec une taille d'espérance infinie, on dit que le processus de branchement est surcritique (critique quand $zp = 1$).
- si $zp < 1$, le nombre moyen de descendants converge vers $1/(1 - zp)$, ce qui implique que l'arbre se termine avec probabilité 1, on dit que le processus de branchement est sous-critique.

Plus court chemin entre 2 sommets

Dans le cas des graphes possédant une loi de Poisson on a $z_1 = z$, $z_2 = z^2$, donc $\lambda = \ln N / \ln z$.

4.2.2 Loi de puissance

La loi de puissance sert à modéliser le graphe du web ou du moins beaucoup de ses propriétés. Notamment, je rappelle que GNUTELLA possède une loi de puissance dont l'exposant de la loi de puissance t est estimé au plus bas à 1,4 et au plus haut à 2,3 et v est compris entre 100 et 1000. La probabilité qu'un noeud est k voisins est donnée par cette formule :

$$P(k) = Ck^{-t} e^{\frac{-k}{v}} = \frac{k^{-t} e^{-k/v}}{L_{i_t}(e^{(-1/v)})}$$

donc dans les processus de branchement,

$$P_e(k) = pCk^{-t} e^{-\frac{k}{v}} = p \frac{k^{-t} e^{-k/v}}{L_{i_t}(e^{(-1/v)})}$$

avec $L_{i_n}(x) = \sum_{j=1}^{\infty} \frac{x^j}{j^n}$ on a :

$$G_0(x) = \frac{L_{i_t}(xe^{(-1/v)})}{L_{i_t}(e^{(-1/v)})}$$

et :

$$m = G'_0(1) = \frac{L_{i_{t-1}}(e^{(-1/v)})}{L_{i_t}(e^{(-1/v)})}$$

il est facile de voir que le degré moyen dans les processus de branchement vaut

$$p \frac{L_{i_{t-1}}(e^{(-1/v)})}{L_{i_t}(e^{(-1/v)})}$$

Soit $Z(n)$ le nombre de descendants à la génération n , on obtient $\mathbb{E}(Z(n)) = (p \frac{L_{i_{t-1}}(e^{(-1/v)})}{L_{i_t}(e^{(-1/v)})})^n$.

Nombre de messages

Soit $T(n) = \sum_{m=0}^n Z(m)$ le nombre total de messages donc :

$$\begin{aligned} T(n) &= \sum_{i=0}^n Z(i) \\ \mathbb{E}(T(n)) &= \mathbb{E}(\sum_{i=0}^n Z(i)) \\ &= \sum_{i=0}^n \mathbb{E}(Z(i)) \\ &= \sum_{i=0}^n (p \frac{L_{i_{t-1}}(e^{(-1/v)})}{L_{i_t}(e^{(-1/v)})})^i \\ \mathbb{E}(T(n)) &= \frac{(p \frac{L_{i_{t-1}}(e^{(-1/v)})}{L_{i_t}(e^{(-1/v)})})^{n+1} - 1}{(p \frac{L_{i_{t-1}}(e^{(-1/v)})}{L_{i_t}(e^{(-1/v)})}) - 1} \end{aligned}$$

Extinction de l'arbre

Par conséquent, si n croît, alors :

- si $p \frac{L_{i_{t-1}}(e^{(-1/v)})}{L_{i_t}(e^{(-1/v)})} \geq 1$, le nombre moyen de descendants tend vers l'infini, ce qui suggère que l'arbre devient infini avec une probabilité non nulle, ou bien il reste fini mais avec une taille d'espérance infinie, on dit que le processus de branchement est surcritique (critique quand $p \frac{L_{i_{t-1}}(e^{(-1/v)})}{L_{i_t}(e^{(-1/v)})} = 1$).
- si $p \frac{L_{i_{t-1}}(e^{(-1/v)})}{L_{i_t}(e^{(-1/v)})} < 1$, le nombre moyen de descendants converge vers $1/(1 - p \frac{L_{i_{t-1}}(e^{(-1/v)})}{L_{i_t}(e^{(-1/v)})})$, ce qui implique que l'arbre se termine avec probabilité 1, on dit que le processus de branchement est sous-critique.

Plus court chemin entre 2 sommets

$$\lambda = \frac{\ln N + \ln[L_i(e^{-1/v})/L_{i-1}(e^{-1/v})]}{\ln[L_{i-2}(e^{-1/v})/L_{i-1}(e^{-1/v}) - 1]} + 1$$

Chapitre 5

Perspectives

Les travaux présentés peuvent être étendus afin de pousser les résultats plus loin.

- Nous avons supposé que les ressources étaient distribuées uniformément parmi les pairs, il doit être possible d'enlever cette hypothèse en la remplaçant par un scénario plus réaliste telle que la probabilité que la ressource soit possédée par un pair dépende de son degré .
- La grande hypothèse nous ayant permis de créer un réseau P2P statique, pourrait être enlevée, ainsi on pourrait modéliser le réseau avec des pairs non-actifs et d'autres actifs, cela impliquerait que la probabilité p de transmettre le message à un voisin serait nulle pour les pairs inactifs.
- La probabilité qu'un pair transmette le message à un voisin est égale à p , il serait envisageable de faire varier p de plusieurs manières pour obtenir des résultats intéressants, de cette manière on ferait varier p en fonction du degré du noeud actuel, ou de sa distance par rapport au pair initiateur.
- Il doit être possible d'ajouter la probabilité de retomber sur un noeud déjà parcouru, de toute manière dans Gnutella les requêtes de recherche possèdent une ID unique donc si elles repassent par un noeud n'est pas renvoyé elle sont détruites.

Il serait intéressant aussi de s'interroger sur l'article [19] car, ils ont montré grâce à leur crawler que le réseau Gnutella ne peut se représenter par une loi de puissance comme utilisé dans beaucoup d'études, cependant ils dénotent certaines propriétés des petits mondes. Il pourrait être utile d'explorer cette branche.

Chapitre 6

Conclusion

Nous avons défini une étude analytique pour les stratégies de recherche dans les réseaux pairs-à-pairs décentralisés et non structurés tel que Gnutella. Le réseau P2P est modélisé avec les processus de branchement (Chap 3.3) grâce à l'hypothèse 1 du modèle, celle-ci nous a permis de générer plusieurs résultats en moyenne comme le nombre de messages sur le réseau et la popularité de la ressource. Nous avons appliqué notre modèle à 2 réseaux P2P différents, les réseaux à loi de Poisson et ceux avec une loi de puissance.

Nous avons aussi vu comment calculer certaines probabilités telle que :

- La probabilité du nombre de descendants par étage appelé $P_e^{(k)}(n_k)$
- La probabilité que la ressource recherchée se trouve sur un certain nombre de noeuds $Q_e^{(k)}(n_k, r_k)$.

De ce fait on peut savoir si le p choisit pour la transmission du message n'impliquera pas une saturation du réseau trop importante ou si il permettra d'atteindre nos objectifs.

Bibliographie

- [1] Top applications (bytes) for subinterface : Sd-nap traffic, Technical report, CAIDA workload analysis of SD-NAP data, 2002. [urlhttp ://www.caida.org/http ://caida.org/](http://www.caida.org/http://caida.org/).
- [2] M.E.J. Newman, S.H. Strogatz, D.J. Watts, Random Graphs with arbitrary degree distribution and their applications, *Phys. Rev.E64* (2001) 026118.
- [3] P.Erdős, A.Rényi, On the Evolution of Random Graphs, *Publicationes Mathematicae*, vol 6, 574-617, 1959.
- [4] M. Ripeanu, Peer-to-Peer architecture case study : Gnutella network, in : Proceedings of the IEEE First International Conference on Peer-to-Peer Computing (P2P2001), Linkoping, Sweden, August, 2001.
- [5] M. Ripeanu, I. Foster, A. Iamnitchi, Mapping the gnutella network : properties of large-scale peer-to-peer systems and implications for system design, *IEEE internet comput.* 6 (1) (2002) 50-57.
- [6] M.E.J. Newman, The spread of epidemic disease on networks, *Phys. Rev. E66* (2002) 016128.
- [7] M.E.J. Newman, The structure and function of networks, *Comput. Phys. Commun.* 147 (2002) 40-45.
- [8] M.E.J. Newman, S.H. Strogatz, D.J. Watts, Random Graphs with arbitrary degree distribution and their applications, *Phys. Rev.E64* (2001) 026118.
- [9] R.Gaeta, G.Balbo, S.Bruell, M.Gribaudo, M.Sereno, A simple analytical framework to analyze search strategies in large-scale peer-to-peer networks, *Performance Evaluation* 62 (2005) 1-16.
- [10] Krishna Kant, An Analytic Model for Peer to Peer File Sharing Networks, in : Proceedings of the International Communications Conference, Anchorage, AL, USA, May, 2003.
- [11] Franoush Banaei-Kashani and Cyrus Shahabi, Critically-based Analysis and Design of Peer-to-Peer Networks as “Complex Systems”, December, 2003
- [12] Herbert S. Wilf, *Generatingfunctionology*, 2nd Edition, Academic Press, London (1994).
- [13] R.Gaeta, M. Gribaudo, D.manini, M.Sereno. Analysis of resource transfers in P2P file sharing., in : Proceedings of the fluid models. Perform Eval., in press ; doi : 10.1016/j.peva.2005.01.001.
- [14] B. Yang, H. Garcia-Molina, Comparing hybrid P2P systems, in : Proceedings of the Seventh International Conference on Very Large Data Bases (VLDB01), Rome, Italy, 2001.
- [15] M. Feldman, L. Kevin, J. Chuang, I. Stoica, Quantifying disincentives in P2P networks, in : Proceedings of the First workshop on economics of p2p systems, Berkeley, CA, USA, June, 2003.
- [16] Z. Ge, D.R. Figueiredo, S. Jaiswal, J. Kurose, D. Towsley, Modeling P2P file sharing system., in : Proceedings of the INFOCOM 2003, San Francisco, USA, April, 2003.
- [17] E.P. Elias Leontiadis, Vassilios V. Dimakopoulos, Cache updates in a P2P network of mobile agents, in : Proceedings of the Fourth IEEE P2P 2004, Zurich, Switzerland, August, IEEE Comp. Soc. Press, 2004.

- [18] Matei Ripeanu and Ian Foster. Mapping the Gnutella Network : Macroscopic Properties of Large-Scale Peer-to-Peer Systems. In Proceedings of the 1st International Workshop on Peer-to-Peer Systems, March 2002. <http://citeseer.ist.psu.edu/article/ripeanu02mapping.html>
- [19] Daniel Stutzbach, Reaz Rejaie, Characterizing Today's Gnutella Topology, Technical Report CIS-TR-04-02, Departement of computer science, University of Oregon, Dec. 7,2004
- [20] Hai Zhuge, Xue Chen, Xiaoping Sun, Preferential Walk : Towards efficient and scalable Search in Unstructured Peer-to-Peer Networks, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100080, China Mai, 2005.