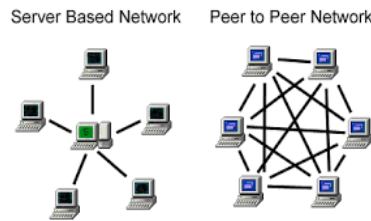


République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique Université
Abderrahmane Mira de Béjaïa
Faculté des Sciences Exactes
Département de Recherche Opérationnelle



Mémoire de Master

*Pour l'obtention du diplôme de Master en Recherche Opérationnelle
Option : Fiabilité et Evaluation des Performances des Réseaux*



Thème

Évaluation de performance des réseaux P2P avec les réseaux de Pétri

Réalisé par :

M^{elle} : BOUCHEBBAH MOUNIA

M^{elle} : HAMOUDI HASSIBA

Proposé par :

Promoteur : *Pr* D.AISSANI

Co-promoteur : *M^{elle}* Z. MORDJI

Promotion : 2014 – 2015

Remerciement

Nous tenons tout d'abord à remercier Dieu le tout puissant, qui nous a donné la force et la patience d'accomplir ce modeste travail. Nous souhaitons adresser nos remerciements les plus sincères aux personnes qui nous ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année universitaire.

Nous tenons à remercier très chaleureusement *M^r* D. AÏSSANI et *M^{elle}* Z. MORDJI qui nous ont permis de bénéficier de leur encadrement. Les conseils qu'ils nous ont prodigué, la patience, la confiance qu'ils nous ont témoignés ont été déterminants dans la réalisation de notre travail.

Nous tenons aussi à remercier *M^r* M. AMAD d'accepter de présider le jury de soutenance.

Nous remercions également, *M^r* L. IKHELEF et *M^{elle}* S. TOULOUM qui nous ont fait l'honneur d'accepter d'évaluer notre travail.

Nous n'oublions pas nos parents pour leur contribution, leur soutien et leur patience. Nous adressons nos plus sincères remerciements à tous nos proches et amis, qui nous ont toujours soutenu et encouragé au cours de la réalisation de ce mémoire.

Dédicaces

Je dédie ce modeste travail :

Aux êtres les plus chers au monde, mon père et ma mère qui ont sacrifié leurs vies pour ma réussite.

A mes chers frères et sœur.

A toute ma famille.

A tous mes amis (es).

Hassiba

Je dédie ce travail à :

Mes très chers parents qui m'ont toujours soutenu et pour lesquels les mots ne sont pas assez forts pour exprimer toutes les gratitudes et l'estime que je leur dois, que dieu les protège pour moi.

Mes très chers frères et sœur et a toute ma famille et mes amis (es).

Mounia

TABLE DES MATIÈRES

Introduction Générale	5
1 Généralités sur les réseaux <i>P2P</i>	7
1.1 Introduction	7
1.2 Définition du Peer to Peer	7
1.3 Client / Serveur V_S P2P	8
1.4 Les caractéristiques du Peer to Peer	9
1.5 Classification et comparaison des réseaux <i>P2P</i>	10
1.5.1 Architecture centralisée	11
1.5.2 Architecture décentralisée	13
1.5.3 Architecture hybride	17
1.5.4 Etude comparative des architectures <i>P2P</i>	18
1.6 Avantages et inconvénients du Peer to Peer	19
1.6.1 Avantages	19
1.6.2 Inconvénients	19
1.7 Domaine d'application des réseaux Peer to Peer	20
1.7.1 Calcul distribué	21
1.7.2 Diffusion de contenu	21
1.7.3 Programmes de messagerie	21
1.7.4 Streaming <i>P2P</i>	22
1.8 Conclusion	22

2	Modèles mathématiques appliqués aux réseaux <i>P2P</i>	23
2.1	Introduction	23
2.2	L'évaluation de performance	23
2.2.1	Rôle de l'évaluation des performances	24
2.2.2	Etapes de l'évaluation des performances	24
2.2.3	Types d'évaluation des performances	25
2.2.4	Méthodes d'évaluation des performances	25
2.3	Modèles analytiques	28
2.3.1	Chaîne de Markov	28
2.3.2	Système d'attente et Réseaux files d'attente	30
2.3.3	Réseau de Petri	35
2.4	Evaluation des indices de performances	41
2.5	Comparaison entre les modèles analytiques	43
2.6	Conclusion	44
3	Application d'un modèle analytique pour évaluer les performances d'un réseau <i>P2P</i>	45
3.1	Introduction	45
3.2	Description de l'architecture P2P : BitTorrent	46
3.3	Application à BitTorrent	47
3.4	La modélisation via les réseaux de Petri	48
3.4.1	L'analyse du modèle <i>R</i>	51
3.5	Généralisation du graphe de BitTorrent à <i>n</i> noeuds	51
3.6	Evaluation de performance de BitTorrent	52
3.6.1	Description du logiciel TimeNet	52
3.6.2	Application	53
3.7	Conclusion	55
	Conclusion Générale	57
	Bibliographie	59

TABLE DES FIGURES

1.1	Classification des réseaux P2P.	11
1.2	Architecture centralisée.	12
1.3	Architecture décentralisée.	13
1.4	Architecture hybride ou Super pair.	17
2.1	Etapas de l'évaluation de performances d'un système.	25
2.2	Méthode d'évaluation de performance.	26
2.3	Représentation graphique des éléments de RdP.	36
2.4	Transition immédiate et transition stochastique	39
2.5	Exemple de réseau de Petri stochastique généralisé	39
2.6	Représentation de modèle d'un serveur pour le calcul de la distribution des temps de transfert.	43
3.1	Topologie de BitTorrent.	47
3.2	Graphe formé par les noeuds.	48
3.3	Modélisation de l'architecture de BitTorrent via les RdPSG.	49
3.4	Graphe de BitTorrent à n noeuds.	52
3.5	La variation du temps moyen de réponse en fonction de λ	54

LISTE DES TABLEAUX

1.1	Comparaison des infrastructures client/serveur et <i>P2P</i>	9
1.2	Réseau pair-à-pair centralisé	18
1.3	Réseau pair-à-pair décentralisé structuré	18
1.4	Réseau pair-à-pair décentralisé non structuré	18
2.1	Tableau de comparaison entre les méthodes d'évaluation des performances	27
2.2	Quelques interprétations typiques de transitions et de places.	36
2.3	Comparaison entre les modèles analytiques	44
3.1	Tableau de comparaison entre les différents λ	54
3.2	Tableau de comparaison entre les différents taux de téléchargement $:\mu_i$	55

INTRODUCTION GÉNÉRALE

Un réseau informatique est un ensemble de machines interconnectées qui permet l'échange d'informations et le partage des ressources. Ce réseau est divisé en deux grandes catégories selon le moyen de connexion entre les machines : les réseaux filaires et les réseaux sans fil. Dans ces derniers réseaux, les terminaux sont interconnectés sans aucun moyen physique mais par des ondes radio.

Ces dernières années, les réseaux Peer-to-Peer (*P2P*) sont devenus très populaires. Cette popularité vient des bonnes caractéristiques offertes par ces systèmes, comme : la grande échelle, l'autonomie des noeuds et le contrôle décentralisé.

Ce type de réseaux offre une bonne opportunité pour répondre aux limites des systèmes basés sur le paradigme Client/serveur. Les réseaux *P2P* sont utilisés avec succès dans plusieurs domaines comme : le partage de fichiers, le partage de capacité de calcul et l'échange de messages instantanés ; grâce à leurs caractéristiques avantageuses, de nouveaux domaines se dirigent vers l'utilisation de ces réseaux dans de nouvelles applications.

Dans les réseaux *P2P*, les communications entre les pairs se font via les protocoles de communication. Cependant, il existe plusieurs protocoles de communications ; on peut citer Gnutella, BitTorrent, factTrack, eDonkey, etc. L'évaluation des performances de tels réseaux d'échange d'informations est un domaine très intéressant. Il permet d'estimer et d'évaluer certains critères, à savoir : la capacité de service, la puissance, la robustesse et la tolérance aux pannes. Pour cela, plusieurs modèles d'évaluation de performance sont mis en place pour étudier les comportements de tels réseaux et éventuellement mettre en place des configurations adéquates pour chacune des

utilisations voulues.

Les réseaux de Petri (RdP), ont été développés au début des années soixante par l'ingénieur allemand Carl Adam Petri dans sa thèse " Communication avec des automates " [20]. En 1981 le première livre sur les réseaux de Petri a été publié par J. Peterson. Plusieurs extensions ont été portées à la définition initiale des RdP. En effet, l'introduction de certains paramètres, caractéristiques ou critères particuliers aux systèmes à étudier, a conduit à la définition de plusieurs formalismes tels que : les réseaux de Petri stochastiques (RdPS), les réseaux de Petri stochastiques généralisés (RdPSG), les réseaux de Petri temporels, etc.

Dans le présent travail, nous allons modéliser un réseau $P2P$ particulier : BitTorrent qui est l'une des applications les plus populaires pour le partage de fichiers, et ce, grâce à son mode de fonctionnement qui permet de faire collaborer tous les clients pour télécharger un fichier donné. Pour cela, nous avons opté pour un modèle de réseaux de Petri stochastiques généralisés (RdPSG).

Ce mémoire est structuré de la manière suivante :

Une introduction où l'idée générale du sujet à traiter est exposée.

Le premier chapitre est consacré à une étude générale sur les réseaux $P2P$.

Dans le deuxième chapitre, nous décrivons les modèles mathématiques (analytiques) qui permettent d'étudier un réseau $P2P$. Ces modèles font partie de l'approche d'évaluation des performances (qui est l'objet du présent travail).

Le troisième chapitre, concerne le choix d'un modèle tout en ayant en vue la structure du réseau. Nous nous sommes penchés sur un type de réseau particulier : le BitTorrent. Cela nous permettra de donner l'évaluation des performances d'un réseau $P2P$ avec les réseaux de Petri stochastiques généralisés (RdPSG). La simulation a été réalisée en faisant usage du logiciel TimeNet.

Une conclusion générale viendra clore ce travail résumant les grands points qui ont été abordés dans ce mémoire, ainsi que les perspectives que l'on souhaite accomplir prochainement.

CHAPITRE 1

GÉNÉRALITÉS SUR LES RÉSEAUX *P2P*

1.1 Introduction

Ces dernières années, l'utilisation des réseaux est croissante, à cause des avantages comme l'échange de l'information et le partage des ressources. Ces réseaux ont été initialement fondés sur des architectures client-serveur.

Une telle architecture présente beaucoup d'inconvénients. L'existence d'un point central étant le plus évident. L'architecture *P2P* se pose comme une solution de rechange à l'architecture client-serveur en offrant une répartition du trafic et de la charge, une résistance aux fautes et l'anonymat.

Dans ce chapitre, on présente une vision globale sur les réseaux *P2P*, telle que sa définition, une comparaison avec l'architecture client-serveur, ses caractéristiques, sa classification, ses avantages et ses inconvénients, son domaine d'application.

1.2 Définition du Peer to Peer

An cours de la recherche bibliographique pour ce mémoire, nous avons trouvé plusieurs définitions du réseau *P2P*. Nous citons une définition suivante parmi celles que nous avons trouvé dans [16] et [22] respectivement :

Le terme "Peer-to-Peer", "poste à poste" ou "pair à pair" [16] en français (ou plus couramment *P2P*), désigne un modèle de réseau informatique dont les éléments (les noeuds ou "Peer") sont à la fois clients et serveurs lors des échanges. Ce modèle a permis une décentralisation des réseaux, en offrant une alternative aux traditionnelles architectures client/serveur ou des architectures N-tiers.

Comme leur nom l'indique [22], les Peer-to-Peer permettent d'établir des communications directes, d'égal à égal, entre les différents noeuds du réseau, qui peuvent alors échanger différents types d'informations sans passer par un serveur central.

A partir de cette définition, nous pouvons dire que la technologie *P2P* ne permet pas seulement le partage des ressources numériques (textes, sons, images, ..), mais elle permet également le partage des capacités de traitement de l'information et de l'espace de stockage (CPU, RAM. . .). Ainsi, l'une des spécificités les plus marquantes des réseaux *P2P* est que les échanges ont lieu directement entre les utilisateurs qui peuvent être contributeurs et consommateurs en même temps.

L'utilisation des réseaux Peer-to-Peer est aujourd'hui en pleine croissance à cause de leurs avantages et caractéristiques et leur propagation dans tous les domaines. Pour cela, plusieurs logiciels ont été développés.

1.3 Client / Serveur V_S P2P

Traditionnellement, l'échange de services entre les ordinateurs est fondé sur la technique client/serveur, selon cette architecture, il n'y a qu'une seule entité centrale très puissante, le serveur, et plusieurs entités généralement de puissances inférieures, les clients. Le serveur est le seul fournisseur des services aux clients. Un client consomme les services exécutés par le serveur, sans partager aucune de ses propres ressources [12].

L'architecture pair à pair se pose comme une solution de rechange à l'architecture client/serveur en offrant plusieurs avantages par rapport aux autres basés sur le paradigme client/serveur, le tableau 1.1 montre bien les différences entre les deux modèles.

Critère	Client/serveur	Modèle P2P
Gestion	Supervisé	Auto-organisé
Présence	Parmanante	Ah-doc
Accès aux ressources	Recherche	Découverte
Organisation	Héirarchique	Distribué
Mobilité	Statique	Mobile
Disponibilité	Dépendante du serveur	Indépendante des pairs
Nommage	RPC	Indépendant
Modèle de programmation	RPC	Asynchrone

TABLE 1.1 – Comparaison des infrastructures client/serveur et *P2P*

1.4 Les caractéristiques du Peer to Peer

Dans cette section, nous présentons les principales caractéristiques que présente le modèle Peer-to-Peer :

- **Décentralisation** : Le fait que chaque noeud gère ses propres ressources permet d'éviter la centralisation de contrôle. Un système *P2P* peut fonctionner sans avoir aucun besoin d'une administration centralisée ce qui permet d'éviter les goulots d'étranglements et d'augmenter la résistance du système face aux pannes et aux défaillances.
- **Passage à l'échelle** : Il s'agit de faire coopérer un grand nombre de noeuds (jusqu'à des milliers ou des millions) pour partager leurs ressources tout en maintenant une bonne performance du système. Cela signifie qu'un système *P2P* doit offrir des méthodes bien adaptées avec un environnement dans lequel il y a un grand volume de données à partager, un nombre important de messages à échanger entre un grand nombre de noeuds partageant leurs ressources via un réseau largement distribué.
- **L'auto-organisation** : Puisque les systèmes *P2P* sont souvent déployés sur l'Internet, la participation d'un nouveau noeud à un système *P2P* ne nécessite pas une infrastructure coûteuse. Il suffit d'avoir un point d'accès à l'Internet et de connaître un autre noeud déjà connecté pour se connecter au système. Un système *P2P* doit être un environnement ouvert ; c'est-à-dire, un utilisateur sur un noeud doit être capable de connecter son noeud au système sans avoir besoin de contacter une personne et sans avoir besoin de passer par une autorité centrale.
- **Autonomie des noeuds** : Chaque noeud gère ses ressources d'une façon autonome. Il décide

quelle partie de ses données à partager. Il peut se connecter ou/et se déconnecter à n'importe quel moment. Il possède également l'autonomie de gérer sa puissance de calcul et sa capacité de stockage.

- **Hétérogénéité** : A cause de l'autonomie de noeuds possédant des architectures matérielles et/ou logicielles hétérogènes, les systèmes *P2P* doivent posséder des techniques convenables pour résoudre les problèmes liés à l'hétérogénéité de ressources.
- **Dynamique** : A cause de l'autonomie de noeuds, chaque noeud peut quitter le système à n'importe quel moment ce qui fait disparaître ses ressources du système. De nouvelles ressources peuvent être ajoutées au système lors de la connexion de nouveaux noeuds. Alors, à cause de l'instabilité de noeuds, les systèmes *P2P* doivent être capables de gérer un grand nombre de ressources fortement variables. La sortie d'un noeud du système (ou la panne d'un noeud) ne doit pas mettre le système en échec. Elle doit être tolérée et avoir un "petit" impact sur la performance de tout le système.

1.5 Classification et comparaison des réseaux *P2P*

Trois grandes catégories de systèmes *P2P* comme il est présenté dans la FIGURE 1.1 suivante peuvent être identifiées : centralisé, décentralisé et hybride. La catégorie décentralisés peut encore être divisée en décentralisé structuré et décentralisé non structuré. La différence principale entre ces systèmes est le mécanisme utilisé pour rechercher des ressources dans le réseau *P2P* [21].

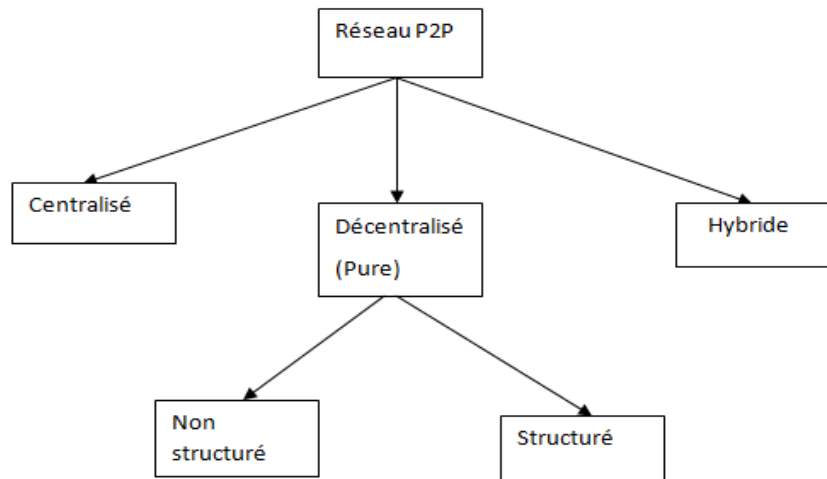


FIGURE 1.1 – Classification des réseaux P2P.

1.5.1 Architecture centralisée

Ce type d'architecture présente la première génération des réseaux *P2P*. Il repose sur un serveur central qui contient des métadonnées décrivant les ressources (spécialement des fichiers) partagées par les participants du réseau. Les utilisateurs se connectent au serveur afin de communiquer entre eux et d'échanger des fichiers.

Le rôle principal du serveur est la gestion de la recherche des ressources en identifiant les noeuds stockant les fichiers. Après la connexion au serveur et la découverte du noeud qui possède le fichier voulu, la communication et l'échange de ce fichier, entre le pair demandeur et le pair fournisseur, se fait de façon directe. La figure suivante présente cette architecture :

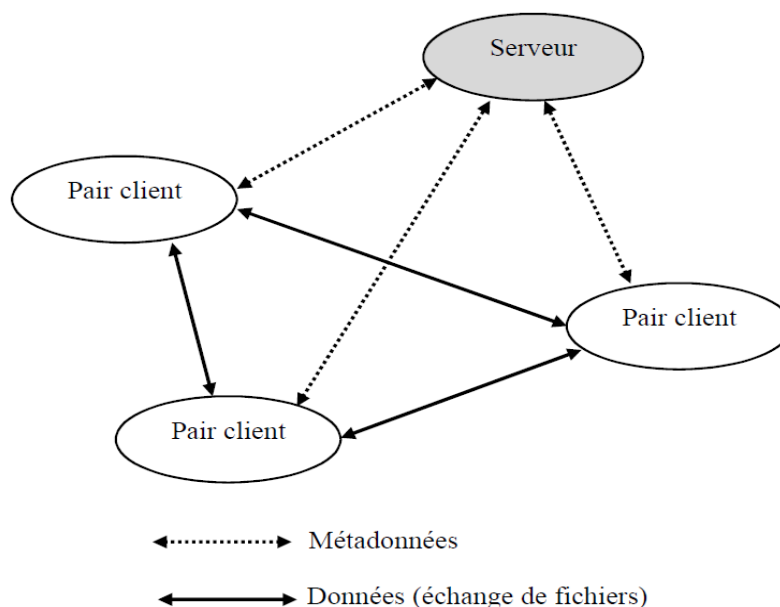


FIGURE 1.2 – Architecture centralisée.

Dans cette architecture (exemple de NAPSTER), chaque pair héberge un ensemble de fichiers qu'il partage avec les autres pairs du réseau. Alors que le serveur sauvegarde deux types de données :

- Une table qui dispose d'informations sur les pairs connectés (adresse IP, numéro du port, bande passante disponible ...).
- Une table qui recense tous les fichiers partagés par chaque pair, avec des métadonnées qui décrivent chaque fichier (nom du fichier, date de création, ...).

Les étapes suivies par un utilisateur sont les suivantes :

- L'utilisateur se connecte au serveur central et annonce les fichiers qu'il partage,
- S'il cherche un fichier, il envoie une requête au serveur,
- Le serveur effectue une recherche dans sa table d'index et retourne la liste des pairs qui disposent du fichier recherché,
- Enfin, le pair client ouvre une connexion directe avec un ou plusieurs pairs hébergeant le fichier recherché et le télécharge directement sans transiter par le serveur.

Le logiciel *P2P* qui utilise cette architecture est NAPSTER. C'est le premier programme *P2P* conçu pour le partage de fichiers, spécialement pour l'échange de fichiers audio/vidéo.

Notre objectif est de modéliser une architecture centralisée s'appelle BitTorrent quand va décrire sa description dans le chapitre 3.

1.5.2 Architecture décentralisée

Cette architecture est d'une nature plate où tous les noeuds du réseau remplissant les mêmes tâches. Ils jouent le rôle de serveur et de clients en même temps. Les pairs d'un tel réseau sont souvent appelés SERVENT (SERVEur+cliENT). Cette architecture ne disposant pas d'un serveur central, du fait qu'un pair quitte le réseau n'affecte pas le système. Nous pouvons désigner deux types d'architectures décentralisées : non structuré et structuré.

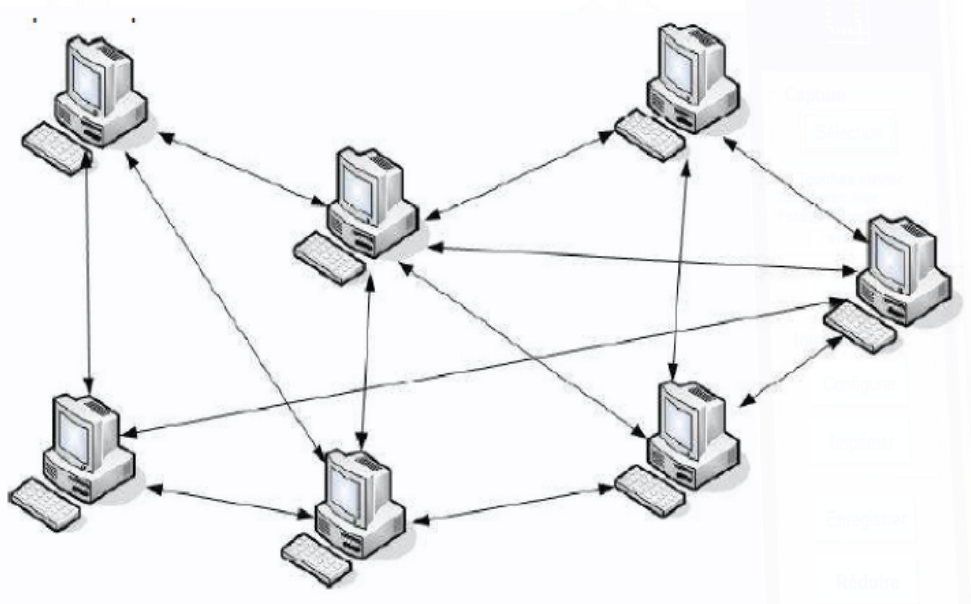


FIGURE 1.3 – Architecture décentralisée.

1.5.2.1 Architecture non-structurée

Cette classification est basée sur la notion de l'inondation qu'on lorsqu'un utilisateur peut découvrir une ressource, il transmette une requête qui contient le nom de la ressource vers les autres voisins jusqu'à atteindre le client qui dispose l'objet désiré, afin d'éviter l'inondation du réseau durant un temps trop long, le système associe à chaque requête un temporisateur TTL "Time To Live", la valeur attribuée au TTL est généralement 7. Lorsqu'il arrive à zéro, la requête n'est plus renvoyée.

L'inconvénient majeur de ce mécanisme, est que $TTL = 0$ avant le parcours de l'intégralité du réseau, ce qui peut aboutir à l'échec d'une recherche bien que l'objet désiré soit disponible sur le réseau *P2P*.

Ces architectures non structurées sont extrêmement résistantes aux nœuds entrant et sortant du système. Par contre, l'actuel mécanisme de recherche passe mal à l'échelle et génère une charge importante pour les participants au réseau. Cette méthode est utilisée dans le système : Gnutella et FreeNet.

Nous expliquons ici un exemple de réseau *P2P* décentralisé non structurés qu'est Gnutella qu'on a trouvé dans [21].

Gnutella

Gnutella a été le premier réseau *P2P* totalement décentralisé. Créé en mars 2000 par Justin Frankel et Tom Pepper, succédant à Napster, dont la centralisation présentait une forte faiblesse, Gnutella a tiré profit de cette expérience. Gnutella est un protocole ouvert, décentralisé pour des recherches distribuées sur une topologie plate de pairs.

Selon Gnutella, tous les pairs sont à la fois serveur et client. Ce protocole n'a pas de répertoire centralisé et n'a aucun contrôle sur la topologie ou l'emplacement des fichiers. Le réseau est formé avec des pairs joignant le réseau d'après certaines règles simples. L'emplacement des données n'est fondé sur aucune connaissance de la topologie. Pour localiser un objet, un client demande à ses voisins qui, eux-mêmes, demandent à leurs voisins. Ce système permet simplement l'entrée et la sortie des clients, mais le mécanisme utilisé passe mal à l'échelle et génère de fortes charges dans le réseau.

1.5.2.2 Architecture structurée

Les systèmes *P2P* décentralisés structurés mettent en oeuvre un algorithme de recherche. Cet algorithme est complètement déterministe, et les liens entre les pairs sont établis suivant des règles bien définies.

L'infrastructure de recherche fonctionne comme une table de hachage distribuée, ou DHT (Distributed Hash Table), implémentée à travers les pairs. Cette structure permet la découverte efficace des données par le biais de clés. Elle est particulièrement appropriée au développement de réseaux à grande échelle. Dans cette catégorie, on peut placer Chord, CAN (Content Adressable Network), Tapestry, Pastry, Kademia et Viceroy.

Une table de hachage distribué (DHT) hache premièrement le nom du fichier recherché. Pour cela, la DHT utilise une fonction de hachage standard -telles que SHA-1 ou MD5- sécurisée pour faire correspondre à chaque nom de fichier son « empreinte » sous forme d'une chaîne de bits de longueur fixe. Cette fonction de hachage garantit que pour deux ressources différentes les clés générées sont uniques. Ainsi, la clé permet d'identifier et de retrouver la ressource de manière fiable. Deuxièmement hache l'adresse IP est donne l'identifiant d'un utilisateur .Il s'agit ensuite de stocker de manière distribuée les couples (clé, identifiant) sur les noeuds du réseau pour qu'à chaque ressource du réseau soit associée l'adresse de l'utilisateur possédant la ressource. De la redondance dans le stockage est également introduite afin que le départ d'un noeud du système n'engendre pas la perte des métadonnées qu'il stocke et ne rende impossible l'accès à ces données.

Les DHT présentent de bonnes propriétés du fait de l'utilisation d'un modèle *P2P* pur : aucun pair ne joue de rôle particulier ou central, et chacun agit de manière strictement équivalente. Parmi les propriétés de DHT on trouve dans [21] les suivants :

Performance : Dans des conditions normales de fonctionnement, le nombre de sauts nécessaires est limité. Par exemple, dans une communauté de 106 pairs utilisant une base d'identifiants hexadécimale, la longueur moyenne d'une requête avoisine cinq sauts.

Passage à l'échelle (évolutivité) : Deux caractéristiques confèrent aux DHT de bonnes performances de passage à l'échelle. La première est liée au nombre moyen de sauts nécessaires au routage des requêtes, qui reste petit, même dans le cas de communautés comptant un grand nombre de participants. La seconde est relative aux tables de routage, qui restent elles aussi d'une taille raisonnable en regard du nombre de participants.

Fiabilité : L'utilisation d'un algorithme de découverte et de routage permet, pour une clé donnée, de déterminer le pair d'identifiant le plus proche. Dans des conditions statiques, une réponse négative à une requête signifie que la ressource requise n'est pas disponible dans la communauté.

Tolérance aux fautes : Du fait de l'absence de centralisation, qui exclut tout point central, les DHT présentent une bonne tolérance aux suppressions aléatoires de noeuds. Les requêtes peuvent être acheminées même si une partie des noeuds disparaît. Par contre, chaque noeud racine d'une ressource particulière s'apparente à un point central. Des mécanismes de redondance sont souvent mis en place pour éviter l'inaccessibilité d'une ressource présente dans une DHT. Un exemple de réseau *P2P* décentralisé structurés qu'est Chord qu'on a trouvé dans [11].

Chord

Chord organise son espace d'adressage suivant un anneau dont les 2^m adresses (ou identifiants, id) possibles sont ordonnées le long de sa circonférence. Chaque pair ainsi que chaque ressource possède un identifiant obtenu par une fonction de hachage SHA-1 (dans ce cas $m = 160$ bit) qui garantit une distribution homogène des ressources sur l'anneau Chord.

Chord se limite une fonction de routage, c'est à dire qu'étant donné un identifiant, Chord localise le pair responsable qui est celui possédant le plus petit identifiant supérieur ou égal à celui de la ressource. Quand un pair rejoint le réseau, il prend en charge une partie des identifiants attribués à son successeur direct, et quand il quitte le réseau, tous ses identifiants sont attribués à son successeur. Chord ne propose donc pas toutes les primitives de stockage / recherche / réplication pour opérer une DHT, et encore moins une application de partage de fichier fonctionnelle.

L'intérêt de Chord est de proposer une structure simple et efficace permettant de router les messages. Pour cela, chaque pair du réseau maintient une table de pointeurs contenant les informations «id, ip, port» pour un certain nombre de pairs. Plus cette liste est grande, plus elle est coûteuse à maintenir, mais plus le routage est rapide (le cas extrême étant de connaître tous les pairs du réseau, le routage ne nécessite alors qu'un message). Chord parvient à trouver un compromis entre la taille de la table de routage comportant $O(\log(N))$ entrées et le routage nécessitant $O(\log(N))$ messages. Pour cela, un pair, d'id `currentID`, choisit chaque pointeur de telle sorte que le pair sélectionné soit le seul représentant de l'intervalle $[(currentID + 2i) - (currentID + 2i + 1)]$ avec $i + 1 < m$. Un pair connaît donc un successeur représentant une portion de l'anneau qui est deux fois plus grande pour chaque nouvelle entrée.

Le routage est ensuite itératif, de proche en proche, chaque pair envoyant la requête au successeur connu le plus proche de l'identifiant recherché.

Les principales limites de Chord sont, d'une part sa topologie en anneau obligeant de parcourir l'ensemble de l'espace d'adressage dans le pire cas, et d'autre part, l'absence de primitives pour opérer une DHT. Ces faiblesses sont résolues par l'architecture Kademia.

1.5.3 Architecture hybride

Pratiquement, les noeuds d'un réseau *P2P* ont des capacités de stockage et de traitement caractérisées par une forte disparité ce qui approuve la création du modèle « hybride » qui est un couplage entre le mode *P2P* et l'architecture Client/serveur.

Les réseaux hybrides utilisent un nombre suffisant de serveurs (dits super-pairs) pour éviter le risque en cas de disparition de l'un d'eux. Chaque serveur présente un noeud central d'un groupe (cluster) qui comporte un ensemble de pairs organisés en *P2P*.

La figure suivante présente clairement l'architecture partiellement hybride :

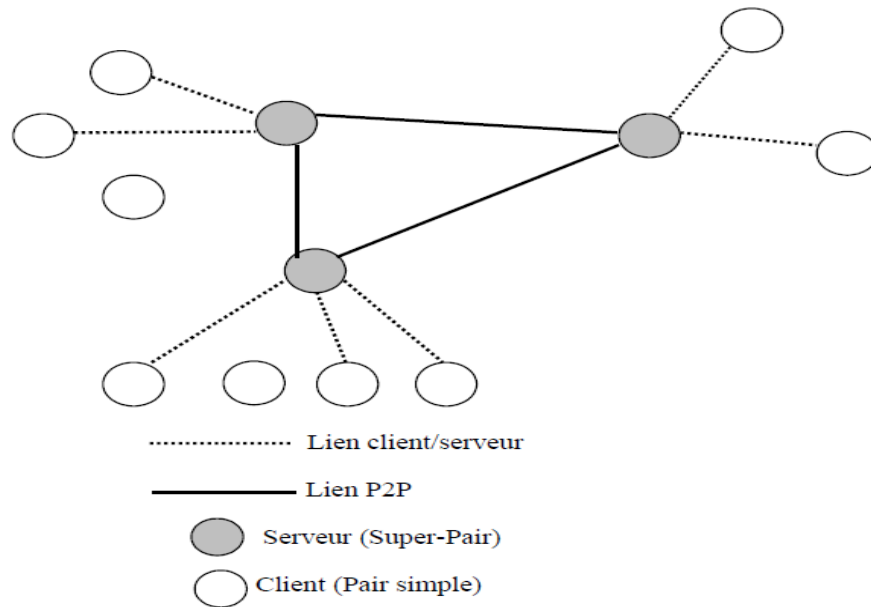


FIGURE 1.4 – Architecture hybride ou Super pair.

Nous distinguons deux types de réseaux hybrides :

Les hybrides statiques : les super-pairs sont manuellement désignés dès le début. Ces derniers peuvent jouer en même temps le rôle du client comme on le retrouve dans l'exemple du réseau eDonkey [10].

Les hybrides dynamiques : Dans certaines conditions, le logiciel client décide de transformer un noeud client en super-pair, comme dans l'exemple de Kazaa [10] .

1.5.4 Etude comparative des architectures P2P

Les trois tableaux représentent etude compérative des architecture P2P[4]

Avantage	Inconvénients
Recherches évaluées	Disponibilité du serveur
Charge minimale sur les pairs	Coût du serveur
Accès aux ressources	Confiance dans l'administration du serveur

TABLE 1.2 – Réseau pair-à-pair centralisé

Avantage	Inconvénients
Distribution complète	Recherches exactes uniquement
Passage l'échelle / Charge homogène	Structure à maintenir
Stockage des données dans les DHT	
Décentralisation	

TABLE 1.3 – Réseau pair-à-pair décentralisé structuré

Avantage	Inconvénients
Distribution Complete	Passage à l'échelle limité (inondation)
Décentralisation	Charge non-homogène des pairs (super pairs)
Recherches non-exactes	

TABLE 1.4 – Réseau pair-à-pair décentralisé non structuré

1.6 Avantages et inconvénients du Peer to Peer

Nous pouvons résumer les différents avantages et inconvénients des systèmes *P2P* comme suit :

1.6.1 Avantages

A partir de propriétés citées ci-dessus, nous pouvons affirmer que les systèmes *P2P* sont plus adaptés aux environnements qui possèdent de grandes quantités de ressources à partager.

- Eviter la création de goulots d'étranglement.
- Passer à l'échelle supérieure.
- Exploiter des ressources importantes distribuées sur un grand nombre de noeuds du réseau.
- Améliorer l'utilisation de la bande passante, des ressources de traitements et des espaces de stockage.
- Accélérer l'accomplissement des tâches en réduisant le temps de traitement à travers les liens directs entre les pairs.
- Eviter le seul point d'échec grâce à la distribution redondante des ressources et la décentralisation des systèmes *P2P*. Cette caractéristique offre à ce type de systèmes plus de fiabilité et de robustesse.
- Augmenter les performances du système en partageant la charge du travail et l'accroissement de l'autonomie et l'agrégation des ressources, ce qui augmente la performance des réseaux *P2P*.
- Permet aux utilisateurs de maintenir le contrôle de leurs ressources. En plus, ils peuvent joindre ou quitter le système à tout moment.

1.6.2 Inconvénients

Malgré les réseaux *P2P* sont très populaires et très utilisable aujourd'hui, mais ils ont des inconvénients, nous citons ici quelques une [7] :

- **Les problèmes de comportement des utilisateurs** : Les systèmes d'échanges de fichiers peer-to-peer peuvent facilement souffrir de l'anarchie générale. L'anonymat aidant, les membres sont parfois tentés d'avoir des comportements malveillants. Sans un minimum de contrôle, on peut voir apparaître des activités comme la diffusion de virus ou le freeloading.
- **Les problèmes de comportement des entreprises** :
 - **Une atteinte à la vie privée**

Notre vie privée n'est pas nécessairement protégée lorsque nous utilisons des outils peer-to-

peer. En effet, les adresses IP des utilisateurs peuvent être récupérées lorsque le logiciel est centralisé.

- **La pollution des réseaux**

Sans même annoncer d'ultimatum, plusieurs entreprises proposent de polluer les réseaux d'échange de musique entre particuliers en y incorporant des fichiers de moindre qualité ou incorrects. L'objectif est de rendre ces réseaux gratuits et anarchiques moins attractifs.

Une poignée de sociétés indépendantes développent en effet de nouvelles technologies destinées à polluer les réseaux. Leurs clients potentiels sont les industries de la musique et du cinéma, qui cherchent par tous les moyens à détourner les adeptes des téléchargements gratuits au profit de systèmes sécurisés et payants.

- **La propagande**

Nul n'est sans savoir que les systèmes peer-to-peer sont pour la plupart envahis par les bannières publicitaires même si à présent, des versions allégées des logiciels sont proposées sans publicité.

- **Les réseaux pair à pair restent imparfaits à cause de :**

- La confiance et le certificat
- L'anonymat
- La sécurité
- La performance

1.7 Domaine d'application des réseaux Peer to Peer

Depuis leur apparition, les réseaux *P2P* étaient considérés comme un système d'échange de fichiers. Jusqu'à maintenant, cette application présente le plus grand taux de trafic des différents réseaux *P2P*. C'est pour cette raison que le terme *P2P* peut parfois se confondre avec les systèmes d'échange de fichiers alors que ces derniers ne représentent qu'un cas de figure des types d'applications basés sur le paradigme *P2P*.

Les systèmes *P2P* sont utilisés dans plusieurs catégories d'applications qui peuvent être réparties en quatre classes [3] : calcul distribué, diffusion de contenu, programmes de messagerie, Streaming *P2P*.

1.7.1 Calcul distribué

Même si les ressources d'un ordinateur à part sont limitées, le regroupement de nombreux ordinateurs permet d'obtenir des performances théoriquement considérables. Ceci est l'un des objectifs des systèmes P2P largement illustré dans la littérature par l'exemple des mathématiciens (et physiciens) avec leurs calculs matriciels. Le but est de diviser un gros calcul, par exemple celui de l'inversion de la matrice en petits traitements plus ou moins indépendants que l'on pourrait répartir entre les pairs. Ce type d'applications est connu sous le nom de « GRID Computing ».

1.7.2 Diffusion de contenu

Le but de ce type d'applications est généralement le partage des fichiers qui présente l'objectif principal pour lequel les systèmes *P2P* ont vu le jour. Ces systèmes sont utilisés pour créer un réseau de noeuds permettant la recherche et le transfert des fichiers. Dans cette catégorie on retrouve des systèmes comme : Napster, Gnutella, KAZAA [10], Freenet [8] ...

Le partage des fichiers dans un réseau *P2P* est souvent implémenté dans le cas où des fichiers doivent être diffusés auprès d'un grand nombre de personnes (par exemple diffusion d'une émission vidéo ou audio), l'application *P2P* la plus adaptée à ce type d'application est BitTorrent [10].

Une deuxième fonctionnalité qui permet la diffusion de contenu est celle de stockage et de publication de contenu. Le but de ces systèmes est de créer un espace distribué qui permettra le stockage et la publication des données. Ces dernières seront utilisées par les pairs qui ont le privilège d'accès. Les exemples d'OceanStore [14] et Ivy [19] se distinguent dans cette catégorie.

1.7.3 Programmes de messagerie

De nos jours, il existe des services de messagerie électroniques basé sur le principe *P2P*, les utilisateurs peuvent envoyer et recevoir des e-mails d'une façon sécurisée, pas besoin d'un serveur central pour stocker temporairement les messages ce qui assure la confidentialité des correspondants, un système d'authentification et de cryptage est utilisé afin de protéger leur contenus, un bon exemple est JefTel.com. Des logiciels de messagerie instantanée peuvent aussi être vus comme *P2P*, on citera des exemples comme ICQ, AIM qui certes utilisent un serveur mais juste pour la résolution d'adresses (les adresses utilisées peuvent être des alternatives aux adresses IP).

1.7.4 Streaming *P2P*

Le streaming *P2P* est le fait de regarder en direct des flux produits et/ou relayés par d'autres pairs du réseau afin d'éviter ou du moins diminuer la congestion qui pourrait se produire sur les serveurs de téléchargement, tout se déroule entre les personnes qui veulent accéder au fichier, Swarmplayer, qui permet de lire des vidéos en streaming en utilisant BitTorrent, s'annonce comme une vraie révolution dans le domaine.

1.8 Conclusion

Nous avons présenté dans ce chapitre différents réseaux pair-à-pair, à savoir les réseaux *P2P* centralisés, hybrides et décentralisés non structurés et structurés. Nous avons donné des illustrations qui montrent leurs topologies respectives, les points forts et les inconvénients de chaque type. Nous en avons donné les objectifs et les caractéristiques principales. Par la suite, nous nous attacherons à décrire les modèles analytiques permettant d'évaluer et d'améliorer certaines performances de ces réseaux.

CHAPITRE 2

MODÈLES MATHÉMATIQUES APPLIQUÉS AUX RÉSEAUX *P2P*

2.1 Introduction

L'évaluation de performances d'un système est un domaine qui prend de plus en plus d'importance, il devient en effet inconcevable de construire un système (système informatique, système de production, ...) sans avoir auparavant mené une analyse de performance afin de construire un système adapté, conforme aux objectifs du cahier des charges. En d'autres termes, l'évaluation de performance consiste à déterminer les forces et les faiblesses d'un système en conception ou en exploitation sous environnement donné. Les modèles analytiques (réseaux de Petri, réseaux de files d'attente, chaîne de Markov, ...) constituent des outils performants pour la modélisation et l'évaluation des performances des réseaux *P2P*.

Dans ce chapitre, nous présentons les différentes notions liées à l'évaluation de performances, et les modèles mathématiques appliqués aux réseaux *P2P*.

2.2 L'évaluation de performance

L'évaluation des performances d'un système s'avère indispensable dès l'instant où l'on souhaite obtenir les performances de celui-ci et que l'on ne peut pas effectuer des mesures directes sur le système *P2P*.

2.2.1 Rôle de l'évaluation des performances

Le rôle essentiel de toute application informatique est d'accomplir les fonctions pour lesquelles elle a été conçue, et d'offrir une performance adaptée à un coût raisonnable. La performance d'une application est donc un facteur clé de son succès. L'évaluation des performances peut se faire à deux niveaux :

En conception :le système n'existe pas encore et il s'agit de le créer en respectant le cahier des charges.

En exploitation :le système existe mais on souhaite le tester ou le modifier de telle manière à améliorer son fonctionnement.

2.2.2 Etapes de l'évaluation des performances

Les différentes étapes d'évaluation de performances sont :

- A) Substituer le système à un modèle que l'on pourra résoudre soit par des méthodes analytiques (mathématiques) soit par des simulations.
- B) Appliquer une méthode de résolution ou de simulation (au modèle donné); l'objectif étant d'obtenir les performances du système (temps moyen d'attente, temps moyen de réponse...).
- C) Analyser les résultats obtenus par une méthode quelconque.

Le dimensionnement du système sera obtenu en exploitant les résultats par l'intermédiaire des scénarios et en modifiant éventuellement certains éléments du modèle.

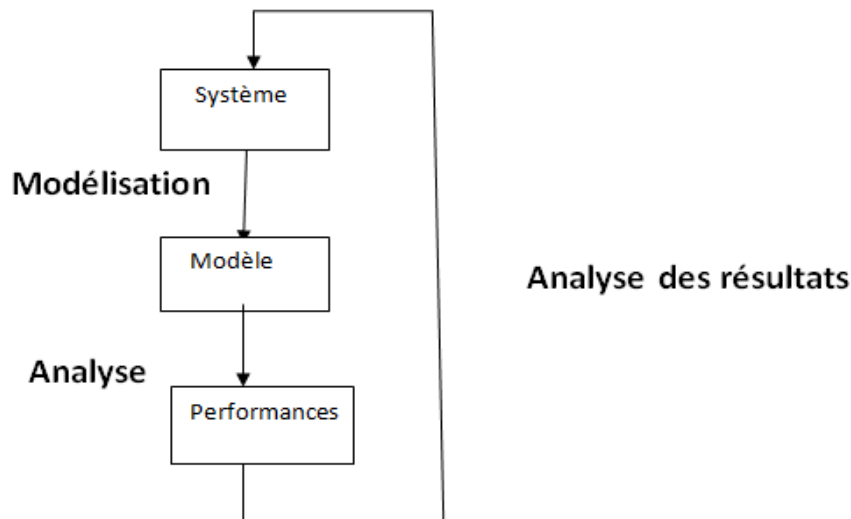


FIGURE 2.1 – Etapes de l'évaluation de performances d'un système.

2.2.3 Types d'évaluation des performances

Il existe deux approches d'évaluation pour un système, l'approche qualitative et l'approche quantitative.

L'approche qualitative : consiste à définir les propriétés structurelles et comportementales du système. Il s'agit de vérifier quelques propriétés qualitatives ; l'absence de blocage, l'exclusion mutuelle, le comportement fini ou borné, etc.

L'approche quantitative : concerne le calcul des mesures que l'on veut effectuer sur un système informatique permettant de quantifier ces performances : débit, temps moyen de réponse, taux d'utilisation de ses ressources, etc [4].

2.2.4 Méthodes d'évaluation des performances

Les méthodes de l'évaluation de performances des systèmes informatiques peuvent être classées en trois catégories, l'obtention des mesures directes sur le système, les méthodes analytiques et la simulation.

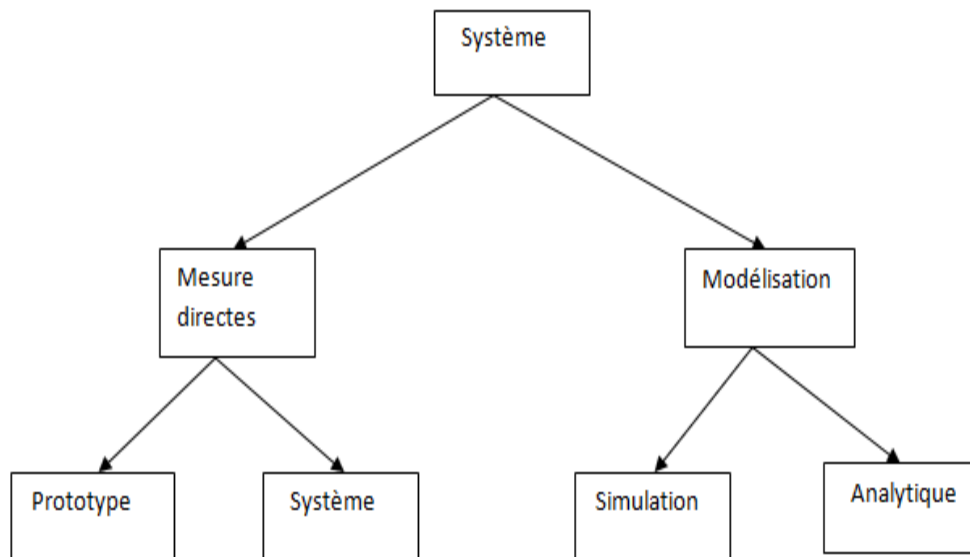


FIGURE 2.2 – Méthode d'évaluation de performance.

2.2.4.1 Les mesures directes

Cette méthode est la seule à pouvoir offrir l'image réelle de l'état d'un système réel en tenant compte de toutes ses caractéristiques. Mais elle a des inconvénients dans le fait que les résultats obtenus à un moment donné ne permettent pas toujours de prévoir le comportement du système dans d'autres conditions et générer les coûts elles sont souvent très difficiles à appliquer.

2.2.4.2 La modélisation

Il est souvent difficile d'étudier un système réel dans un environnement opérationnel ; pour y remédier on représente le fonctionnement du système de manière approchée en s'appuyant sur des outils adéquats ; cette représentation par un modèle abstrait est appelée "Modélisation", on distingue deux méthodes :

Les méthodes analytiques

Ces méthodes, peu coûteuses, consistent à analyser numériquement le système après l'avoir traduit en un modèle mathématique. Ces méthodes sont parfois complexes ; voire impossibles de les utiliser pour certains systèmes. Pour y remédier, on considère des hypothèses pour simplifier l'étape de modélisation en permettant un comportement approché et rendre l'évaluation numérique possible, mais ces hypothèses peuvent toucher à la fidélité du système réel. Plusieurs outils mathématiques permettent l'évaluation numérique, on cite :

- Les chaînes de Markov (temps discret et continu) pour l'accès aux ressources ;

- Les algèbres des processus stochastiques pour les systèmes complexes ;
- Les réseaux de files d’attente pour les réseaux ;
- Les réseaux de Petri pour la synchronisation en temps réel.

La simulation

C’est une méthode très utilisée pour l’évaluation de performances. Elle consiste à observer le comportement d’un modèle simplifié du système réel à l’aide d’un programme de simulation adéquat qui donnera pour résultat des graphes faciles à analyser et à interpréter. Cette méthode s’approche plus du modèle réel par rapport aux méthodes analytiques, et elle est utilisée lorsque l’évaluation par mesures directes devient très coûteuse [24].

2.2.4.3 Tableau de comparaison entre les méthodes d’évaluation des performances

Le tableau ci dessous présente une comparaison entre les différentes techniques d’évaluation de performance.

Critère	Technique analytique	Technique de simulation	Technique de mesures
Etape	N’importe	N’importe	Nécessité l’existence d’un système réel ou prototype
Temps requis	Peu	Moyen	Variable
Outils	Des analystes	Langages de programmation simulateurs	Les instrumentations moniteurs
Précision	Haut	Moyen	Variable
Compromis l’évaluation	Facile	Facile	Difficile
Coût	Faible	Moyen	Haut

TABLE 2.1 – Tableau de comparaison entre les méthodes d’évaluation des performances

2.3 Modèles analytiques

Pour explorer les performances des réseaux *P2P* beaucoup de modèles analytiques (réseaux de Petri, réseaux de files d'attente, chaînes de Markov, ...) ont été développés dans littérature.

Généralement, chaque modèle évalue un certain exemple pour donner un aperçu sur les performances d'un système *P2P*.

2.3.1 Chaîne de Markov

Les chaîne de Markov [4] sont des processus stochastiques très utilisés dans l'évaluation de performance des systèmes *P2P* de nature discrète plus précisément des systèmes dynamiques. Ce type de processus permet souvent de caractériser les états stationnaires du système, c'est-à-dire les états possibles du système à un instant choisi "aléatoirement" loin dans le temps. Les probabilités d'état stationnaire du système représentent aussi la proportion de temps que le système a passé dans chacun de ces états au cours d'une très longue observation du système. Si les probabilités sont bien définies, on dit que le système admet un état stationnaire. Le système tend alors vers cet état au bout d'un certain temps.

2.3.1.1 Description

Il y'a deux types de chaîne de Markov.

Chaîne de Markov à Temps Discret (CMTD)

Soit $\{(X_n), n=0,1,2,\dots\}$ un processus stochastique à temps discret. Le processus stochastique $\{(X_n), n=0,1,2,\dots\}$ est une chaîne de Markov à temps discret, si pour tout $n = 0, 1, 2, \dots$ et quelque soient les $(n + 1)$ états $i_0, i_1, \dots, i_{n-1}, i, j$ de E tel que :

$$P[X(n + 1) = j, X(n) = i, \dots, X(0) = i_0] = 0 \quad (2.1)$$

La propriété de Markov est vérifiée :

$$P[X(n + 1) = j/X(n) = i, \dots, X(0)] = i_n = P[X(n + 1) = j/X(n) = i] = p_{ij} \quad (2.2)$$

La probabilité $p_{ij}(n)$ s'appelle la probabilité de transition de l'état i vers l'état j à l'instant n . La propriété de Markov signifie que l'état futur de la chaîne à l'instant $n + 1$ ne dépend que de son état présent à l'instant n , et non des états passés par lesquels elle est passée.

Une chaîne de Markov $\{(X_n), n=0,1,2,\dots\}$ est dite homogène si pour tout $n=0,1,\dots$ pour tout

états i, j de E , la probabilité

$$P[X(n+1) = j / X(n) = i] = p_{ij}, \quad (2.3)$$

est indépendante de l'instant n . La probabilité p_{ij} s'appelle la probabilité de transition E_e l'état i vers l'état j .

La matrice $P = (p_{ij}) / i, j \in E$, dont les éléments de probabilités de transition p_{ij} lorsque i et j décrivent l'espace d'états E , p s'appelle la matrice des transitions.

$$P = \begin{pmatrix} p_{00} & p_{01} & \dots & p_{0j} & \dots \\ p_{10} & p_{11} & \dots & p_{1j} & \dots \\ \vdots & \dots & \dots & \dots & \dots \\ p_{i0} & p_{i1} & \dots & p_{ij} & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

Une chaîne de Markov à temp discret est totalement définie par sa matrice des transition.

Chaîne de Markov à Temps Continu (CMTC)

Un processus stochastique $\{ X(t), t \geq 0 \}$ est dit chaîne de Markov à temps continu, s'il vérifie les trois conditions suivantes :

1. L'espace d'états S est dénombrable ;
2. Le temps d'observation est de nature continue ;
3. Le processus $X(t)$ vérifie la propriété de Markov ;

Une chaîne de Markov homogène à temps continu est définie par ses probabilités de transition :

$$P[X(t_n) = j / X(t_{n-1}) = i_{n-1}, \dots, X(t_0) = i_0] = P[X(t_n) = j / X(t_{n-1}) = i_{n-1}], t_0 < t_1 < \dots < t_n, n \in \mathbb{N}$$

Une chaîne de Markov homogène à temps continu est définie par ses probabilités de transition :

$$p_{ij} = P [X(s+t) = j / X(s) = i] \quad s \geq 0. \quad (2.5)$$

C'est à dire que les probabilités $P[X(t_n) = j / X(t_{n-1})=i]$ ne dépendent pas des instants d'observations t_n et t_{n-1} , mais uniquement de la durée $t_n - t_{n-1}$ qui sépare les deux observations [2].

2.3.1.2 Exemple pratique : le réseau BitTorrent

Les auteurs de [23] ont proposé un modèle *P2P* avec une architecture centralisée non structurée BitTorrent. Ils ont calculé numériquement la distribution stationnaire pour ce modèle par une chaîne de Markov.

Les auteurs ont démontré, qu'un système *P2P* dans l'état d'équilibre (stationnaire) présente des performances assez bonnes lorsque la vitesse à laquelle les pairs quittent le système est assez lente. Les fichiers avec une charge élevée (c'est-à-dire plus populaires) peuvent améliorer les performances moyennes par rapport à celles de téléchargement des fichiers à faible charge (moins populaire).

D'autres données empiriques suggèrent que le système *P2P* en régime stationnaire aura probablement de bonnes performances, comme dans le régime transitoire.

Une part importante de capacité de service est fournie par les pairs qui sont en même temps en train de télécharger. Il est cependant utile d'inciter ces derniers à rester dans le système. En effet, ils ont constatés que si le taux des pairs qui quittent le système est faible (par exemple, γ/μ est petit) ceci améliore la charge offerte (fichier plus populaire), ainsi que le rendement moyen par rapport aux fichiers moins populaires. Cette caractéristique est en pratique très souhaitable, car elle réalise de meilleures performances avec des charges de service plus élevées.

Avantage

Les auteurs ont réalisé cette étude d'évaluation des performances pour un système de BitTorrent par un calcul numérique.

Inconvénients

Un résultat approximatif.

2.3.2 Système d'attente et Réseaux files d'attente

2.3.2.1 Description

Les files d'attente peuvent être considérés comme un phénomène caractéristique de la vie contemporaine. On les rencontre dans les domaines d'activité les plus divers (guichet de poste, trafic routier, central téléphonique, atelier de réparation,...).

L'étude mathématique des phénomènes d'attente constitue un champ d'application important des processus stochastiques. On parle de phénomènes d'attente chaque fois que certaines unités appelées " client " se présentent d'une manière aléatoire à des " stations " afin de recevoir un service dont la durée est généralement aléatoire.

Processus d'arrivée

L'arrivée des clients à la station sera décrite à l'aide d'un processus stochastique de comptage $N(t)$, $t \geq 0$ (On notera indifféremment N_t) ou bien $N(t)$. Si A_n désigne la variable aléatoire mesurant l'instant d'arrivée du n^{ieme} client dans le système, on aura ainsi :

$A_0 = 0$ (par convention) et $A_n = \inf t, N_t = n$ Si T_n désigne la variable aléatoire mesurant le temps séparant l'arrivée du $(n - 1)^{ieme}$ client et du n^{ieme} client, on a alors : $T_n = A_n - A_{n-1}$.

Temps de service

La distribution du temps de service la plus simple à étudier est la distribution exponentielle. Cependant, la propriété " sans mémoire " de la loi exponentielle fait que celle-ci n'est généralement pas très réaliste pour modéliser les phénomènes réels. On est donc souvent obligé de recourir à d'autres distributions de service.

Structure et discipline de la file

Nombre de serveurs : Une station peut disposer de plusieurs serveurs en parallèle. Soit C le nombre de serveurs. Dès qu'un client arrive à la station, soit il y a un serveur libre et le client entre instantanément en service, soit tous les serveurs sont occupés et le client se place dans la file en attente de libération d'un des serveurs. La plupart du temps les serveurs sont supposés identiques (ils possèdent donc la même distribution) et indépendants les uns des autres.

Capacité de la file : La capacité de la file à accueillir des clients en attente de service peut être finie ou infinie. Soit K la capacité de la file (incluant le ou les clients en service). Une file à capacité illimitée vérifie $K = +\infty$. Lorsque la capacité de la file est limitée et qu'un client arrive alors que cette dernière est pleine, le client est perdu.

Discipline de service : La discipline de service détermine l'ordre dans lequel les clients sont rangés dans la file et y sont retirés pour recevoir un service. Les disciplines les plus courantes sont :

- **FIFO** (first in, first out) ou FCFS (first come first served) PAPS (premier arrivée, premier servi) : c'est la file standard dans laquelle les clients sont servis dans leur ordre d'arrivée.
- **LIFO** (last in, first out) ou LCFS (last come, first served) ou DAPS (dernier arrive, premier servi). Cela correspond à une pile, dans laquelle le dernier client arrive (donc posé sur la pile) sera le premier traité (retiré de la pile).

- **FIRO** (first in, Random out) les clients sont servis de manière aléatoire.

Notation de Kendall

La notation de Kendall permet de décrire le système à files d'attente par une suite de paramètres. Pour identifier un système à files d'attentes, le formalisme suivant a été proposé et est unanimement adopté : $A/B/n/K/N$

A : La première lettre identifie la loi du processus des arrivées, B : la seconde le processus des services, avec dans les deux cas les conventions :

M : loi "sans mémoire" (arrivées poissonniennes, service exponentiel) ;

D : loi déterministe ;

Ek : loi "Erlang-k"

Hk : loi "hyper exponentielle" d'ordre k ;

GI : loi générale, les variables successives étant indépendantes ;

G : loi générale, sans hypothèse d'indépendance.

n : Le chiffre qui suit donne le nombre de serveurs,

K et N : les lettres qui suivent (facultatives) identifient la taille de la file d'attente et la taille de la population source (si ces valeurs sont omises, elles sont supposées infinies).

Quelques mesures de performance :

L'analyse théorique d'un modèle de files d'attente a pour objet de saisir quantitativement et qualitativement le fonctionnement du système en question. Pour cela il faut définir les critères et les mesures susceptibles d'atteindre cet objectif afin d'être capable de déterminer à l'avance les performances du système, les effets d'un changement ou encore d'identifier les paramètres les plus sensibles. Pour un système constitué d'une seule file d'attente, les principales mesures de performances sont.

1. N_s : Le nombre moyen de clients dans le système,
2. T_s : Le temps moyen de séjour dans le système,
3. N_q : Le nombre moyen de clients en attente,
4. W_S Le temps d'attente moyen d'un client dans le système,
5. W_q : Le temps d'attente moyen d'un client dans la file,
6. Le taux d'occupation $\rho = \lambda/\mu$ d'un ou des serveurs.

Ses facteurs sont liés de la manière suivante :

- $N_s = \lambda W_S$
- $N_q = \lambda W_q$

- $W_s = W_q + 1/\mu$
- $N_s = \lambda/\mu + N_q$

Il est important de remarquer que ces grandeurs sont aléatoires, leur espérance et distribution dépendent a priori de l'instant t où elles sont considérées. Nous devrions donc parler du nombre de clients dans le système à l'instant t .

Ce n'est que sous l'hypothèse de stationnarité du système que ces mesures de performance deviennent indépendantes du temps lors de l'étude d'une file, on cherche à estimer l'espérance en régime stationnaire des grandeurs précédentes.

Modèle d'attente Markovien

Les modèles markoviens de files d'attente sont des systèmes où les deux quantités stochastiques principales " le temps des inter-arrivées " et " la durée de service " sont des variables aléatoires indépendantes, exponentiellement distribuées. La propriété " sans mémoire " de la loi exponentielle facilite l'analyse de ces modèles.

Système M/M/1

Pour le système d'attente le plus simple, M/M/1, le flux des arrivées est poissonnien, de paramètre λ et la durée de service est exponentielle, de paramètre μ . La capacité d'attente est illimitée, la discipline de service est FIFO et il y a une seule station de service.

La condition de stabilité (équilibre) de ce système est : $\rho = \lambda/\mu < 1$.

Système M/M/ $+\infty$: On considère un système composé d'un nombre illimité de serveurs indépendants les uns des autres. Dès qu'un client arrive, il rentre instantanément en service. Cette file particulière ne comporte donc pas de file d'attente. Le processus d'arrivée des clients est poissonnien de taux λ et les temps des services sont exponentiels de taux μ_i , $i = 1, \infty$.

Système M/M/1/K : Le modèle M/M/1/K correspond au cas capacité de K clients, K représente le nombre maximum de clients dans le système, c'est-à-dire en attente et en service.

Si un client arrive et trouve devant lui K client dans le système celui la sera perdu, donc la probabilité de perte est égale à la probabilité de trouver K clients dans le système au régime stationnaire, on a : $p_{perte} = \rho^k(1-\rho) / 1-\rho^{k+1}$

Modèle d'attente non-Markovien

En s'écartant de l'hypothèse d'exponentialité de l'une de deux quantités stochastiques " le temps des inter-arrivées " et " les durées de service " ou en introduisant des paramètres supplémentaires spécifiques au n'aura plus de processus Markovien. Ce qui rend l'analyse du modèle analytique très délicate, voire impossible. C'est pourquoi, souvent on se ramène à un processus Markovien.

Système G/G/1 : Pour d'écrire l'évaluation du système G/G/1, on a besoin de caractériser,

en plus du nombre de clients dans le système, le temps déjà passé dans le serveur pour le client en service et le temps déjà écoulé depuis l'arrivée du dernier client.

Système M/G/1 : Une file M/G/1 est caractérisée par un processus d'arrivée poissonnien et par des temps de service indépendants et identiquement distribués selon une loi de probabilités " quelconque " (mais ne prenant que des valeurs non négatives).

L'évaluation du nombre de clients dans une file ne peut plus être modélisée par une chaîne de Markov à temps continu car si, à un instant quelconque $t \geq 0$, X_t clients sont présents, l'évaluation future de la variable X_t dépend non seulement de la valeur de X_t mais également du temps de service déjà alloué au client occupant le serveur.

Les réseaux de files d'attentes

Un réseau de file d'attente est un ensemble fini s_1, s_2, \dots, s_n de systèmes de file d'attente. Le séjour d'un client dans le système consiste à parcourir une partie ou l'ensemble de tous les systèmes s_i ($i=1, \dots, n$), selon un itinéraire déterministe ou aléatoire. En d'autres termes, un réseau de files d'attente est composé d'un ensemble de stations de service et d'un ensemble de clients. Ce système est caractérisé par :

- Les processus d'arrivées des clients au réseau.
- Les temps de service des clients aux stations.
- Le cheminement des clients d'une station à une autre.
- Les disciplines de service à chaque station.

Les différents types de réseaux

On distingue deux types de réseaux :

Réseaux ouverts : Ces réseaux possèdent une ou plusieurs entrées de l'extérieur. Le nombre de clients pouvant se trouver à un instant donné dans un réseau ouvert n'est pas limité.

Réseaux fermés : Ces réseaux ne possèdent pas d'entrées l'extérieur, le nombre de clients dans ce cas est fini constant. Pour un mécanisme de routage probabiliste, on définit p_{ij} la probabilité qu'un client qui quitte la station i , se rende à la station j .

2.3.2.2 Exemple d'application : HPM

Les auteurs [18] ont proposé un modèle *P2P* avec une architecture décentralisée structurée. Ce modèle hiérarchique Peer-to-Peer (HPM) est organisé comme un ensemble d'anneaux hiérarchiques composées des nœuds. Ils ont modélisé les nœuds relais avec un réseau de files d'attente de type GI / G / 1 et chaque pair est modélisé avec une file d'attente M / G / 1 / K. Le paramètre de performance, obtenu est le temps total de téléchargement.

Etapes de l'évaluation de performance de G/G/1

Etape 1 : analyse de l'interaction entre les noeuds relais et les réseaux ;

Etape 2 : évaluation de performance des mesures à chaque noeuds relais ;

Etape 3 : évaluation de performance des mesures pour l'ensemble du réseau.

L'évaluation de performance de M/G/1/K/

Les auteurs ont calculé le temps de téléchargement des pairs, tel que le temps total de téléchargement est égal au temps de téléchargement des noeuds relais plus le temps de téléchargement des pairs.

Avantage

Etude mathématique.

Inconvénient

Un résultat approximatif.

2.3.3 Réseau de Petri

2.3.3.1 Description

Les réseaux de Petri(RdP) ont été introduits par C.A. Petri pour la modélisation des systèmes concurrents. Un des avantages clé de cet outil est la façon naturelle dans laquelle de nombreux aspects fondamentaux, à savoir l'aspect mathématique et l'aspect conceptuel, des systèmes concurrents sont identifiés. Ceci a contribué à l'élaboration d'une théorie riche des systèmes concurrents basés sur les RdP. Leur facilité de modélisation conceptuelle, due à une notation graphique facile à comprendre, a par ailleurs fait des RdP l'outil de choix dans de nombreuses applications. Les RdPs sont très populaires dans le domaine de gestion des processus métiers (BPM) en raison de la variété des processus de flux de contrôle qu'ils peuvent modéliser.

Réseau Petri généralisé

Définition informellement

Un RdP [13] est un graphe orienté biparti comprenant deux sortes de nœuds : des places et des transitions (Figure 2.3). Ce graphe est constitué de telle sorte que les arcs du graphe ne peuvent relier que des places aux transitions ou des transitions aux places.

On représente graphiquement les places par des cercles et les transitions par des barres. Les places servent à représenter les états du système modélisé, tandis que les transitions représentent les changements d'état ou les événements.

Quelques interprétations typiques des transitions et leurs places d'entrées et de sorties sont présentées dans le tableau suivant.

Place d'entrée	Transition	Place de sortie
Pré-conditions	Événement	Post-condition
Données d'entrée	Traitement	Données de sorties
Signaux d'entrée	Processeur	Signaux de sorties
Ressources demandées	Tâches	Ressources libérées
Condition	Clauses logique	Conclusion
Buffers	Processeur	Buffers

TABLE 2.2 – Quelques interprétations typiques de transitions et de places.

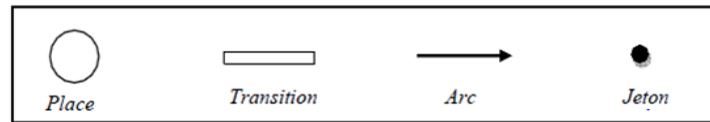


FIGURE 2.3 – Représentation graphique des éléments de RdP.

À chaque arc est associé un nombre entier strictement positif appelé poids de l'arc. Lorsque le poids n'est pas signalé, il est égal à "1" par défaut. Le RdP dont tous ses arcs sont de poids "1" est appelé RdP ordinaire. Dans le cas où les arcs peuvent avoir des poids supérieurs à "1", il s'agit de RdP généralisé.

Définition formelle

D'après [6], un réseau de Petri est constitué d'un graphe orienté biparti valué $\langle P, T, Pre, Post \rangle$ avec :

- $P = \{p_1, p_2, \dots, p_n\}$, est un ensemble fini de places.
- $T = \{t_1, t_2, \dots, t_m\}$, est un ensemble fini de transitions.

Ces deux ensembles formant les sommets du RdP, avec $P \cap T = \emptyset$

- $Pre : P * T \rightarrow \mathbb{N}$, est l'application d'incidence avant, correspondant aux arcs directs reliant les places aux transitions.
- $Post : P * T \rightarrow \mathbb{N}$, est l'application d'incidence arrière, correspondant aux arcs directs reliant les transitions aux places.

Lorsque Pre et $Post$ prennent leurs valeurs dans l'ensemble $\{0, 1\}$, le réseau Petri est un réseau ordinaire.

En utilisant une notation matricielle, les matrices d'incidence avant et arrière peuvent être res-

pectivement définies comme suit :

- $Pre = [\delta_{ij}](1 \leq i \leq n, 1 \leq j \leq m)$ avec $[\delta_{ij}] = Pre(p_i, t_j)$,
- $Post = [\varepsilon_{ij}](1 \leq i \leq n, 1 \leq j \leq m)$ avec $[\varepsilon_{ij}] = Post(p_i, t_j)$,

Ainsi, la matrice d'incidence du réseau $W = [w_{ij}](1 \leq i \leq n, 1 \leq j \leq m)$ avec $Post(p_i, t_j) - Pre(p_i, t_j)$, est obtenue.

Nous adopterons par la suite la notation suivante :

- ${}^o t$ (resp. t^o) représentera l'ensemble $p \in P / Per(p, t) \neq 0$ (resp. $p \in P / Post(p, t) \neq 0$) correspondant à l'ensemble des places d'entrée (resp. sortie) de la transition t .
- ${}^o p$ (resp. p^o) représentera l'ensemble $p \in P / Post(p, t) \neq 0$ (resp. $t \in T / Post(p, t) \neq 0$) correspondant à l'ensemble des places d'entrée (resp. sortie) de la transition p .

Ces notions peuvent aussi être généralisées à des ensembles de sommets, soit $X \subseteq (P \cup T)$, alors l'ensemble $({}^o X)$ sera défini par ${}^o X = X \cup {}^o \alpha$

La composante statique est complétée par un marquage initial :

$M_o : P \rightarrow \mathbb{N}$, $M_o(p)$ est le nombre de marques contenues initialement dans la place p .

Graphe de Marquage

Une fois définie sa structure de base (avec un marquage initial), un RdP peut être exécuté en suivant un comportement précis. On choisit à Chaque étape une transition franchissable et on la tire ce qui a pour effet de modifier l'état du processus.

• Définition 1

L'ensemble de tous les marquages accessibles, $Acc(R, M_o)$ est défini par l'ensemble contenant tous les états accessible tel que :

- $M_o \in Acc$;

- Si : $(M \in Acc)$ et $M(t_i) > M'$ alors $(M' \in Acc)$

$Acc(R, M_o)$ définit tous les marquages accessibles que le système peut effectivement prendre.

D'une façon formelle $Acc = M \exists \sigma \in T^*, tq : M_o \xrightarrow{\sigma} M$.

• Définition 2

Le graphe des marquages du réseau $G(R, M_o)$ est défini par un graphe orienté dont les sommets sont étiquetés par les marquages de $Acc(R, M_o)$ et dont les arcs sont étiquetés par des transition de $L(R, M_o)$

Tel que : Pour $M, M' \in Acc$, il existe un arc $M \xrightarrow{t} M'$ ssi $M(t) > M'$.

Le graphe G est obtenu comme suit :

- Pour chaque marquage M obtenu à partir de M_o , trouver toutes les transitions franchissables t_i ;
- Pour chaque transition t_i , trouver son marquage M' ;
- Construire le nouveau nœud s'il est différent de celui déjà obtenu, puis ajouter l'arc correspondant au marquage actuel vers le prochain ;
- Continuer à développer tant que des marquages et des transitions n'ont pas été encore considérés.

L'ensemble d'accessibilité est l'ensemble des étiquettes des sommets du graphe et le langage du réseau est l'ensemble des étiquettes des chemins du graphe issus du sommet initial.

Les Réseaux de Petri stochastiques

Les réseaux de Petri stochastiques (RdPS) sont surtout utilisés pour modéliser les phénomènes de file d'attente dans le domaine de l'informatique et dans les réseaux de communications. Ce sont des modèles formels qui permettent à la fois des analyses qualitatives et quantitatives .

Définition d'un RdPS

Dans un RdP temporisé, une durée fixe est associée à chaque place ou transition du réseau. On obtient des modèles qui sont bien adaptés pour étudier des systèmes où les durées opératoires sont fixes. En revanche, certains phénomènes sont imprévisibles et ne peuvent plus être traités avec ce type de modèles. Dans ces cas, chaque durée est modélisée par une variable aléatoire. Nous obtenons alors un RdPS, dans lequel un temps aléatoire est associé au franchissement d'une transition. La méthode d'analyse couramment employée consiste à construire le graphe des marquages du RdP sous-jacent au RdPS et à étiqueter chaque arc par un taux de franchissement. Le comportement aléatoire du RdPS est alors identique à celui d'une chaîne de Markov.

Les Réseaux de Petri stochastiques généralisés

Définition d'un RdPSG

Dans les réseaux de Petri stochastiques, toutes les transitions sont associées à une temporisation selon une distribution de loi exponentielle. Les réseaux de Petri stochastiques généralisés ont été introduits par Ajmone [5], [17] afin de limiter certaines restrictions.

Cette classe de réseau autorise deux types de transitions :

- Les transitions temporisées basées sur des distributions exponentielles.
- Les transitions déterministes à temporisation nulle (transition immédiate). Ces transitions sont franchies immédiatement dès lors qu'elles sont sensibilisées.

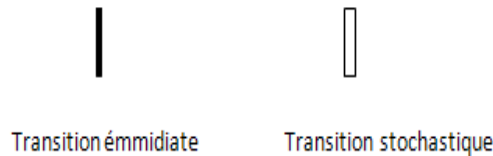


FIGURE 2.4 – Transition immédiate et transition stochastique

Les transitions immédiates permettent notamment de modéliser les synchronisations, ou bien encore d’approximer des temporisations très faibles par rapport aux transitions stochastiques. Graphiquement, les transitions immédiates sont généralement représentées par un trait noir, et les transitions temporisées par un rectangle blanc.

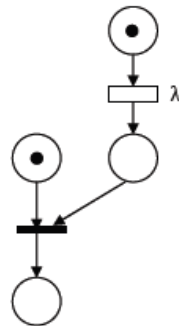


FIGURE 2.5 – Exemple de réseau de Petri stochastique généralisé

L’existence de transitions franchies immédiatement a pour conséquence de faire coexister deux types d’états :

1. **Les états tangibles** : les transitions sensibilisées sont des transitions temporisées.
2. **Les états évanescents (ou virtuels ou instantanés)** : il existe au moins une transition sensibilisée qui est immédiate. Celle-ci est alors franchi immédiatement. Le temps de séjour dans cet état est donc nul.

Dans un marquage où plusieurs transitions sont sensibilisées, les transitions immédiates sont obligatoirement tirées avant les transitions temporisées. on définit ainsi deux types de marquages : les marquages *tangible* dans lesquels seules des transitions temporisées sont sensibilisées, et les marquages *évanescents* dans lequel au moins une transition immédiate est sensibilisée.

La définition formelle d’un RdPSG est donnée par la définition ci-après.

Un RdPSG est un 8 – uplet $\langle P, T, Pre, Post, Inh, pri, W, M_0 \rangle$ où

- P est l'ensemble fini des places ;
- T est l'ensemble fini des transitions temporisées et des transitions immédiates ;
- Pre, Post et $\text{Inh} : P \times T \longrightarrow \mathbb{N}$; sont les applications d'incidence avant, d'incidence arrière et d'inhibition respectivement ;
- $\text{pri} : T \longrightarrow \{0, 1\}$; est la fonction de priorité qui associe à chaque transition temporisée la valeur 0 et à chaque transition immédiate la valeur 1 ;
- $W : T \longrightarrow \mathbb{R}^+$; est une fonction qui associe à chaque transition temporisée un taux de franchissement et à chaque transition immédiate un poids ;
- $M_0 : P \longrightarrow \mathbb{N}$; est le marquage initial du réseau.

Analyse des RdPSG

L'analyse des RdPSG est basée sur deux aspects essentiels [15]

Analyse qualitative : consiste à vérifier ses propriétés : la vivacité, la bornitude, les états d'accueil, etc.

Analyse quantitative : Cette partie d'analyse consiste à calculer les probabilités stationnaires et les indices de performance. Elle est basée sur la chaîne de Markov associée au RdPSG. Cette chaîne peut être construite à partir du graphe des marquages accessibles de la manière suivante :

la première étape consiste à éliminer tous les états évanescents. À partir de ce nouveau réseau, on cherche le graphe des marquages relatif aux états tangibles. Le graphe de Markov possède alors la même structure que le graphe des marquages. Il ne reste plus qu'à étiqueter les arcs par les taux de franchissement dépendants des taux de transition du réseau de Petri et du marquage. On obtient le générateur infinitésimal Q de cette CTMC qui est alors une matrice carrée de dimension $(r \times r)$ (r le nombre fini de marquages tangibles du RdPSG) qui regroupe l'ensemble des taux de transition d'un marquage vers un autre.

La distribution de probabilité $\pi = (\pi_1, \pi_2, \dots, \pi_n)$ à l'état stationnaire sur les marquages tangibles peut alors être obtenue par la résolution de système d'équation linéaire suivant :

$$\begin{cases} \pi Q = 0 \\ \sum_{i=1}^r \pi_i = 1 \end{cases}$$

Proposition :

Le graphe de marquage d'un RdPSG est isomorphe à une chaîne de Markov à temps continue.

Théorème :

Un RdPSG bornée et tel que son graphe des marquages accessibles est fortement connexe est

ergodique.

Théorème : Un RdPSG bornée et ergodique s'il admet le marquage initial comme état d'accueil.

Remarque : Le processus stochastique engendré par un RdPSG borné avec le marquage initial comme état d'accueil, peut être classé comme un processus semi-markovien à temps continu, à espace d'états fini, stationnaire et irréductible.

Méthode d'analyse

Les RdP disposent de nombreux outils théoriques permettant l'analyse des différentes propriétés, trois grandes approches qui se complètent mutuellement ont été développées :

- la première est basée sur l'utilisation des graphes et arborescences de marquage ;
- la seconde fait appel à l'algèbre linéaire ;
- et la troisième se base sur la réduction de RdP.

2.4 Evaluation des indices de performances

Une fois le modèle est obtenu, on vérifie ses propriétés qualitatives pour déduire son ergodicité pour faire de l'analyse quantitative. Si le modèle est ergodique, alors la distribution de probabilité des marquages à l'état stationnaire existe et elle est unique [15].

Plusieurs indices de performances peuvent être calculés. Parmi ces indices les plus importants, on cite :

★ **Fréquence moyenne de franchissement d'une transition :** On appelle fréquence moyenne (débit moyen) de franchissement d'une transition t_i , le nombre moyen de tirs de t_i en une unité du temps. Elle est calculée par

$$\bar{\lambda}(t_j) = \sum_{M_j \in E(t_i)} \lambda_i(M_j)\pi_j; \quad (2.6)$$

où :

$E(t_i)$ est l'ensemble des marquages où la transition t_i est franchissable.

$\lambda(M_j)$ est le taux de franchissement de t_i en M_j .

★ **Nombre moyen de marques dans une place :** Le nombre moyen de marques dans une place p est calculé en appliquant la formule :

$$n(p) = \sum_{i: M_i \in E} M_i(p)\pi_i; \quad (2.7)$$

où :

$M_i(p)$ est le nombre de jetons dans la place p pour le marquage M_i .

E est l'ensemble des marquages accessibles.

★ **Le temps moyen de séjour d'une marque dans un sous-réseau** : Le délai moyen qu'un jeton passe dans une partie S (sous-réseau) d'un RdPSG à l'état stationnaire, peut être calculé en utilisant la formule de Little :

$$E[T] = \frac{E[N]}{E[\beta]}; \quad (2.8)$$

où :

$E[T]$: est le nombre moyen de jetons dans S .

$E[\beta]$: est le taux d'arrivée effectif des jetons dans S .

2.4.0.2 Exemple d'application : le réseau Gnutella

Les auteurs [5] ont proposé un modèle $P2P$ avec une architecture décentralisée non structurée Gnutella qui est un protocole ouvert, pour des recherches distribuées sur une topologie plate de pairs. Dans ce travail, ils ont modélisés ce système avec un réseau de Petri stochastique généralisée c'est un cas fluide.

Les auteurs représentent également les téléchargements simultanés interférence par d'autres clients dont l'effet est d'introduire les fluctuations de la bande passante disponible pour le classement de client, la figure suivante représente un modèle d'un serveur pour le calcul de la distribution des temps de transfert.

Ils se sont concentrés uniquement sur les transferts de ressources entre clients et les serveurs de la bande passante, mais l'analyse peut être étendue à la bande passante de n'importe quel poste.

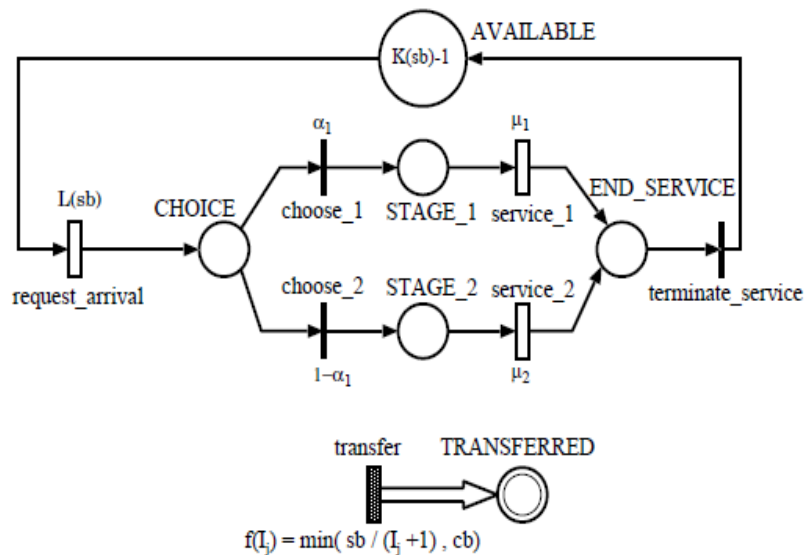


FIGURE 2.6 – Représentation de modèle d’un serveur pour le calcul de la distribution des temps de transfert.

Avantage

- Modélisation graphique.
- Représentations faciles à comprendre.
- Facile à simuler.

Inconvénients

- Exposition combinatoire.

2.5 Comparaison entre les modèles analytiques

Référence	Modèle	Architecture	Point fort	Point faible
[23]	Chaîne de Markov	BitTorrent	-Calcul numérique	-Résultat approximatif
[18]	File d'attente	HPM	-Etude mathématique -Modélisation des phénomènes réels	-Résultat approximatif
[9]	Réseau de Petri	Gnutella	-Modélisation graphique -Formulation mathématique -Représentation facile à comprendre -Résultat numérique -Facile à simuler	-Exposition combinatoire

TABLE 2.3 – Comparaison entre les modèles analytiques

2.6 Conclusion

Nous nous sommes penchés dans ce chapitre sur quelques modèles analytiques puissants pour la modélisation des réseaux *P2P*. Nous en avons montré les paramètres et les caractéristiques. Chaque outil correspond à une situation précise, le choix du modèle dépend de plusieurs paramètres (exemple : la performance qu'on cherche à quantifier).

Dans le chapitre suivant, nous appliquerons un modèle analytique de réseau de Petri pour modéliser et évaluer les performances d'un réseau *P2P*.

CHAPITRE 3

APPLICATION D'UN MODÈLE ANALYTIQUE POUR ÉVALUER LES PERFORMANCES D'UN RÉSEAU *P2P*

3.1 Introduction

Comme nous avons vu dans le chapitre précédent, divers modèles analytiques sont proposés pour évaluer les performances des réseaux *P2P*. L'objectif principal de ce chapitre est de présenter notre contribution dans le cadre de ce mémoire qui consiste à modéliser l'architecture *P2P* (BitTorrent) à l'aide des réseaux Petri stochastique généralisé.

Nous commencerons par une description générale sur l'architecture de BitTorrent puis le modèle de Petri stochastique généralisé d'évaluation des performances par une analyse qualitative du modèle. On termine ce chapitre par un simulation de modèle *R* avec un simulateur de Petri TimeNet.

3.2 Description de l'architecture P2P : BitTorrent

BitTorrent [1] est un protocole de partage de fichiers. En fait, c'est un système Peer to Peer comme eDonkey. Les fichiers sont échangés par morceaux entre les utilisateurs. Ils sont identifiés par une URL (exemple : `http://www.levien.com/mandrake9.1.torrent`) afin de s'adapter parfaitement à la navigation Internet. Pour réunir les informations et les utilisateurs de BitTorrent, il existe des serveurs centraux, appelés tracker. Ces trackers permettent de coordonner le téléchargement des pièces de chaque fichier aux utilisateurs du programme. Tous les utilisateurs participent au partage en recevant (download), mais aussi en envoyant (upload) des parties du fichier. Ce fonctionnement permet d'optimiser le transfert des données et de limiter la charge sur le fichier source.

Les caractéristiques de BitTorrent sont

- Son objectif étant uniquement la distribution de contenu (le plus efficacement possible et en limitant la bande passante du fournisseur initial) et non pas la localisation de ce dernier, BitTorrent ne contient pas de fonctionnalité pour trouver un fichier, comme dans les logiciels peer to peer classiques.
- Il traite les gros fichiers comme un certain nombre de morceaux et chaque morceau peut être téléchargé d'une source différente.
- Il contient divers mécanismes pour assurer son efficacité, dont notamment un mécanisme qui récompense les clients qui envoient des données aux autres [4].

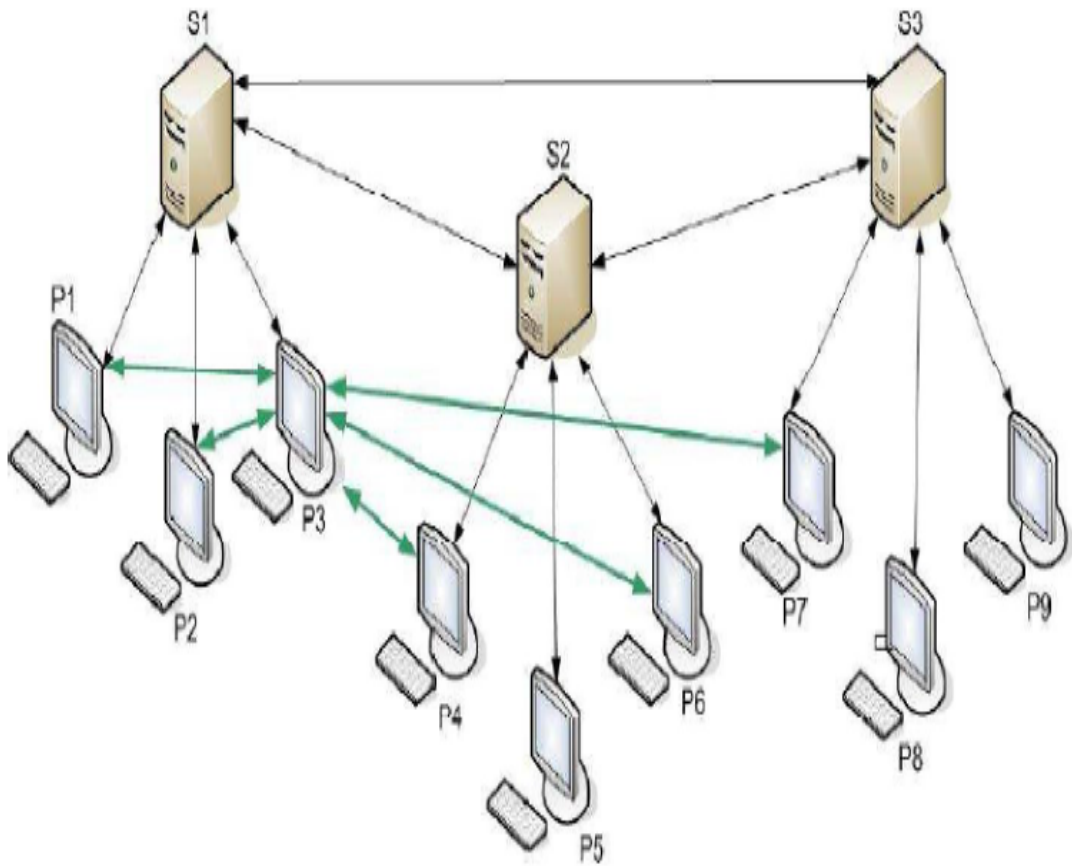


FIGURE 3.1 – Topologie de BitTorrent.

3.3 Application à BitTorrent

A la différence de beaucoup d'autres protocoles Peer to Peer, dans BitTorrent, les clients (noeuds) ne font pas partie d'un réseau global (comprenant tous les utilisateurs du réseau), mais sont plutôt regroupés par fichier : chaque fichier distribué verra son propre groupe d'utilisateurs. Chacun de ces groupes sera organisé autour d'un composant central. Le tracker, qui sert en quelque sorte d'annuaire dynamique de clients intéressés par le téléchargement de ce fichier. Après avoir été organisés sous une topologie donnée, les noeuds peuvent commencer le téléchargement des paquets, sachant qu'un client peut être associé à plusieurs téléchargements à la fois.

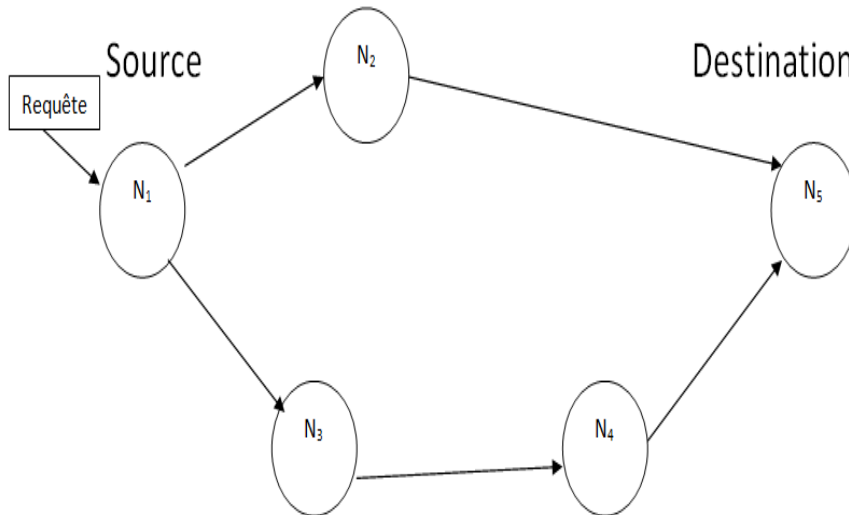


FIGURE 3.2 – Graphe formé par les noeuds.

Nous considérons, pour notre étude l'exemple ci-dessus, un réseau BitTorrent constitué de cinq noeuds collaborant au téléchargement d'un fichier (sous forme d'un ensemble de paquets de même taille) reçu par le noeud 1, qui le fait transférer aux autres noeuds intéressés. Nous allons considérer que les noeuds qui reçoivent les paquets du noeud 1 acceptent de collaborer en les transmettant aux autres. Donc, on dit qu'un paquet ne circule plus dans le réseau quand il est transféré à tous les noeuds. Pour cela, nous allons suivre le transfert des paquets entre les noeuds jusqu'à ce qu'ils atteignent le dernier noeud à savoir le noeud 5, qui se trouve dans la dernière position.

3.4 La modélisation via les réseaux de Petri

Parmi les différentes classes des réseaux de Petri, nous avons privilégié les réseaux de Petri stochastiques généralisés (RdPSG) principalement pour modéliser l'architecture de BitTorrent parce qu'ils permettent de fournir des résultats en régime permanent, qui donneront directement les probabilités stationnaires du système. De plus, des outils prennent en considération l'aspect aléatoire des paramètres du système : l'arrivée des requêtes suit une loi de Poisson et les durées de services (le téléchargement de fichier) sont exponentiellement distribuées.

Définition du modèle R

Soit RdP le réseau de Petri associé à l'architecture de BitTorrent défini par : R c'est un réseau de Petri Stochastique Généralisé.

Les hypothèses du modèle R

Les hypothèses du modèle sont définis comme suit :

- Le nombre de requêtes est illimité.
- Les arrivées des requêtes sont indépendantes les unes des autres suivant un processus de Poisson de paramètre λ .
- Le service de téléchargement des requêtes s'effectue au niveau du chaque client, et les durées de services suivent une loi exponentielle de paramètres : $\mu_1, \mu_2, \mu_3, \mu_4$ tel que :
 μ_i : le taux de téléchargement.

Les valeurs de taux d'arrivées et taux des services sont exprimées en paquets par millisecondes.

Le modèle R de BitTorrent

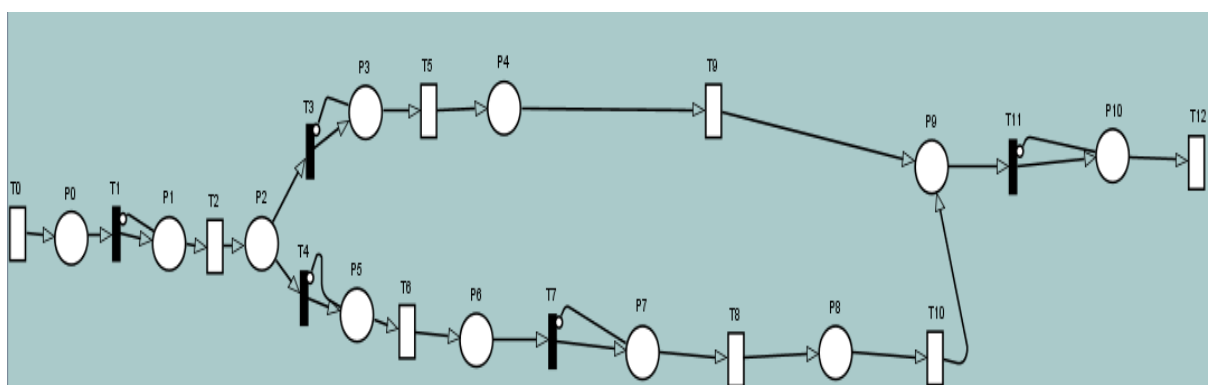


FIGURE 3.3 – Modélisation de l'architecture de BitTorrent via les RdPSG.

Les places

Les places	Description
p_0	La file d'attente des requêtes de client 1.
p_1	Téléchargement des requête.
p_2	La place choix.
p_3	La file d'attente des requêtes de client 2.
p_4	Téléchargement des requête.
p_5	La file d'attente des requêtes de client 3.
p_6	Téléchargement des requêtes .
p_7	La file d'attente des requêtes des clients 4.
p_8	Téléchargement des requêtes.
p_9	La file d'attente des requêtes de client 5.
p_{10}	Téléchargement des requêtes.

Les transitions

Les transitions	Description	Type
t_0	L'arrivée des requêtes	stochastique
t_1	Le traitement des requêtes	immédiat
t_2	Début du service de client 1	stochastique
t_3	Le traitement des requêtes	immédiat
t_4	Le traitement des requête	immédiat
t_5	Début du service de client 2	stochastique
t_6	Début du service de client 3	stochastique
t_7	Le traitement des requêtes	immédiat
t_8	Début du service de client 4	stochastique
t_9	Le traitement des requêtes	stochastique
t_{10}	Le traitement des requêtes	stochastique
t_{11}	Le traitement des requêtes	immédiat
t_{12}	Fin de téléchargement des requêtes	stochastique

3.4.1 L'analyse du modèle R

3.4.1.1 L'analyse qualitative

Le modèle de RdP, R admet comme propriétés qualitatives :

- La bornitude : toutes les places du réseau ne sont pas bornées donc on déduit que le RdP n'est pas borné.
- La vivacité : toutes les transitions sont vivantes donc le RdP est vivant.
- La propriété de graphe d'événement : Le RdP n'est pas un graphe d'événement.
- Les états de blocage : le RdP ne possède aucun état de blocage.
- La connectivité : le RdP est connexe.
- La réinitialisabilité : le RdP n'est pas réinitialisable.

3.4.1.2 L'analyse quantitative

Cette analyse quantitative consiste à étudier le graphe des marquages et les indices de performances associés au modèle R . Puisque ce dernier n'est pas borné et non réinitialisable donc il n'est pas ergodique, et l'évaluation des performances à l'état stationnaire est impossible.

Donc nous proposons une simulation avec le logiciel TimeNet dans la suite de ce chapitre .

3.5 Généralisation du graphe de BitTorrent à n noeuds

BitTorrent (ou simplement BT) est un réseau $P2P$ qui permet le partage des fichiers à travers Internet. L'efficacité de ce réseau augmente lorsqu'il y a beaucoup de clients connectés, car plus il y a de clients qui téléchargent, plus il y a de clients qui partagent, et il n'est pas nécessaire d'attendre dans une file virtuelle pour commencer à télécharger.

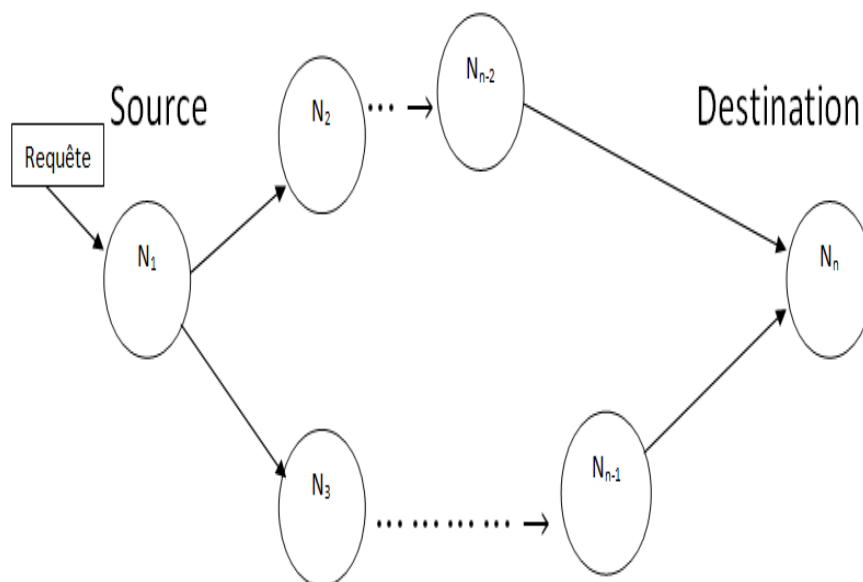


FIGURE 3.4 – Graphe de BitTorrent à n noeuds.

3.6 Evaluation de performance de BitTorrent

3.6.1 Description du logiciel TimeNet

Le logiciel TimeNet (TimeNet Evaluation Tool) [25] est une boîte à outil graphique et interactive qui permet la modélisation et l'analyse qualitative et quantitative des systèmes en utilisant les réseaux Petri stochastiques généralisé (RdPSG) et réseaux Petri stochastiques généralisé colorés (RdPSGC).

TimeNet a été développé par "Real-Time System" et le groupe de "Robotique" de technique Universitat à Berlin (Allemagne). La première version de cet outil a été développée en 1991. Le projet a été motivé par la nécessité d'un logiciel puissant pour une évaluation efficace des systèmes de RdPSG.

TimeNet fournit une interface graphique conviviale. Il est particulièrement adapté à l'analyse des réseaux de Petri déterministe, stochastique et stochastique généralisé.

La version TimeNet 4.0 est une version disponible depuis 2007 dans laquelle de nouveaux algorithmes ont été ajoutés. Elle comprend une interface graphique qui intègre les différents formalismes des réseaux de Petri et des algorithmes d'analyse générique en Java [25]. Elle prend en charge une nouvelle classe de réseaux Petri stochastiques colorés (SCPN) qui permet la conception de systèmes complexes d'une manière compacte. Une simulation standard à événements discrets

à été mise en oeuvre pour l'évaluation de performance des modèles RdPSC.

3.6.2 Application

Les paramètres du modèle R

Pour des valeurs données de taux d'arrivée et taux des services, on calcule le paramètre de performance : le temps moyen de réponse à une requête lancé par un client.

$$\lambda = 3$$

$$\mu_1 = 25$$

$$\mu_2 = 20$$

$$\mu_3 = 14$$

$$\mu_4 = 11$$

Quelques indices de performances lié à R

★ Nombre moyen des requêtes dans le système :

$$n_{mc} = 4$$

★ Le taux moyen effectif des arrivées :

$$\tau_{me} = 19$$

★ Le temps moyen de réponse :

$$T_{mr} = \frac{n_{mc}}{\tau_{me}} = \frac{4}{19} = 0.2105$$

Analyse de performance

• Lorsque on fixe les valeurs du taux de téléchargement $\mu_1, \mu_3, \mu_3, \mu_4$, on fait varier le taux d'arrivée λ . On obtient le tableau suivant :

λ	n_{mc}	τ_{me}	T_{mr}
5	5	17	0.2941
7	11	15	0.7333
9	12	9	1.3333
10	15	8	1.875

TABLE 3.1 – Tableau de comparaison entre les différents λ

L'implémentation des résultats à permis d'obtenir :

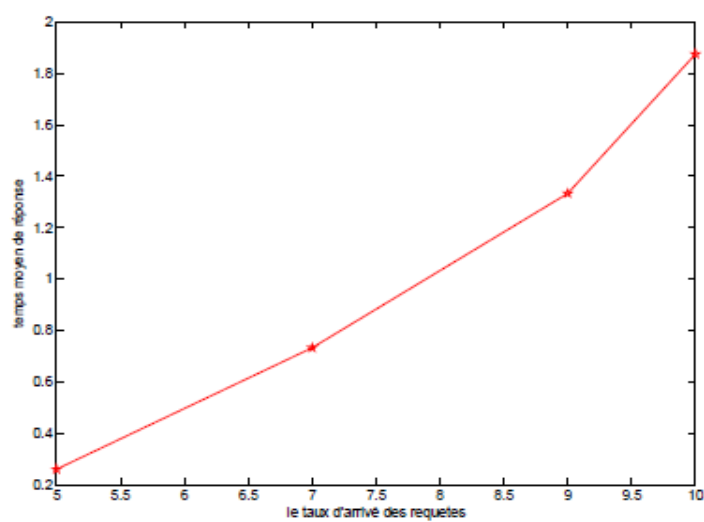


FIGURE 3.5 – La variation du temps moyen de réponse en fonction de λ .

Interprétation des résultats :

Le graphe 3.5 représente la variation du temps moyen de réponse (T_{mr}) en fonction du taux d'arrivée λ variant entre 5 et 10 d'un paquet dans le réseau de Petri.

On voit à partir du ce graphe que le temps moyen de réponse d'une requête s'accroît avec l'augmentation du flot entrant, ce qui montre que plus il y a des requêtes dans le réseau, plus le temps de téléchargement s'accroît.

Analyse de performance

• Lorsque on fixe la valeur du taux d'arrivée $\lambda = 3$, on fait varier à la fois les taux de téléchargement $\mu_1, \mu_2, \mu_3, \mu_4$, on obtient le tableau suivant :

μ_1	μ_2	μ_3	μ_4	n_{mc}	τ_{me}	T_{mr}
29	25	19	16	18	7	2.5714
20	14	11	7	3	26	0.1153

TABLE 3.2 – Tableau de comparaison entre les différents taux de téléchargement : μ_i

Interprétation des résultats :

Le tableau 3.2 représente la variation du temps moyen de réponse (T_{mr}) en fonction du taux de téléchargement μ_i .

On voit à partir du ce tableau que le temps moyen de réponse d'une requête diminue avec la diminution du taux de téléchargement (service) du chaque μ_i ce qui montre que plus qu'il n' y a pas de requête dans le service plus que le temps moyen de réponse diminue.

Et que le temps moyen de réponse d'une requête s'accroît avec l'augmentation du taux de téléchargement μ_i ce qui montre que plus il y a de requête dans le réseau de Petri, plus le temps de moyen de réponse s'accroît.

3.7 Conclusion

Nous avons présenté dans ce chapitre un modèle analytique pour l'évaluation des performances d'un type de réseau pair-à-pair particulier : le BitTorrent. Son architecture qui regroupe les noeuds par le fichier auquel ils s'intéressent, permettant ainsi de limiter le nombre de noeuds dans le réseau. Nous avons appliqué les réseaux de Petri stochastique généralisé, pour l'analyse qualitative de ce

réseau.

Ensuite, nous avons appliqué une autre approche pour l'évaluation des performances du réseau étudié plus haut : la simulation avec le logiciel TimeNet.

Nous avons obtenu, quelques indices de performances lié à R , et une comparaison des délais moyens de réponse en fonction du taux d'arrivée λ , et fonction des taux de service μ_i telle que $i=1,4$.

CONCLUSION GÉNÉRALE

Le *P2P* est une nouvelle technologie réseau et ses applications sont variées, nous citons à titre d'exemple : le multicast applicatif, la signalisation, le parallélisme, le partage de fichiers, la téléphonie IP, la messagerie instantanée, les moteurs de recherche, etc. Ce type de réseau est généralement caractérisé par une bonne scalabilité (passage à l'échelle) et une très grande dynamique (churn rate). Les mesures de leurs performances sont généralement basées sur les méthodes de simulation avec des logiciels tels que NS-2, P2PSim, OpenNet, etc. D'où le manque de modèle de validation des résultats de simulation.

Les modèles analytiques (réseaux de Petri, réseaux de files d'attente, chaînes de Markov...) constituent des outils très performants pour la modélisation et l'évaluation des performances des réseaux *P2P*. Leurs adaptations pour les grands réseaux dynamiques, où la mobilité est un aspect très important nous semble une approche très prometteuse. Ceci permettra en particulier de mesurer plusieurs facteurs de performance.

Dans le cadre de ce travail, nous avons donné une brève description du système Peer to Peer (P2P). Nous avons défini les différents types selon des critères de classification spécifiques. En particulier, nous avons présenté les différents modèles analytiques proposés pour l'évaluation des performances des systèmes *P2P*.

Nous nous sommes intéressés à l'évaluation de performance de l'architecture centralisée non structurée BitTorrent et notre but est de le modéliser en utilisant les réseaux de Petri stochastiques généralisés (RdPSG).

Comme le graphe obtenu par RdPSG est non ergodique, nous avons utilisé la simulation avec

TimeNet pour évaluer la métriques de performances qui est le temps moyen de réponse d'une requête lancée par un client.

Comme perspective, nous envisageons de généraliser le réseau à n noeuds et d'étudier ses performances.

BIBLIOGRAPHIE

- [1] " bittorent <http://bitconjurer.org.bittorent>".
- [2] D. Aissani. "*Cours de files d'attente*". Département de Recherche Opérationnelle, Université A/Mira de Béjaia, 2013.
- [3] D. Alexandre and E. Samuel. "*Peer-To-Peer*". Université Claude Bernard Lyon 1, 2006.
- [4] D. Arkoub and Y. Krouri. "*Modèles analytiques pour l'évaluation des performances d'un réseau pair a pair (BitTorrent)*". Université A/Mira de Béjaia, 2013.
- [5] Y. Atamna. "*Réseaux de Petri Temporisés Stochastiques Classiques et Bien Formés. Définition, Analyse et Application aux Systèmes Distribuée Temps Réel*". Université de Paul Sabatier, 1994.
- [6] P. Bonhomme. "*Reseaux de Petri P-temporels : contribution à la commande robuste*". Université de Savoie, 2001.
- [7] N. Budan, B. Tedeschi, and S. Vaubourg. "*Nouvelles Technologies Réseaux Les réseaux Peer-to-Peer*". Thèse Ingénieur Informatique Réseaux, 2003.
- [8] I. Clarke, S. G. Miller, T.W. Hong, O. Sandberg, and B. Wiley. (40-49), 2002.
- [9] R. Gaeta, M. Gribaudo, D. Manini, and M. Ereno. "fluid stochastic petri nets for computing transfer time distributions in peer-to-peer file sharing applications". *Dipartimento di Informatica Università di Torino Corso Svizzera, 185, 1049 Torino, Italia*, 2005.
- [10] M. Gharzouli. "*Composition des Web Services Sémantiques dans les systèmes Peer-to-Peer*". 2011.
- [11] D. Guillaume, D. Juliette, and Kh. Rida. "*Approche Collaborative pour la détection d'attaques dans les réseaux Pair à Pair*". Groupement -GIS 3SGS-, livrable1, Loria-Inria, 2010.

- [12] H. Hafi. "*Protocole pour la sécurité des réseaux sans fil Peer to Peer*". Université Kasdi Merbah – Ouargla.
- [13] H. Haiouni. "*Approche mixte de modélisation par Réseaux de Petri et SMA*". Ecole Doctorale en Informatique de l'Est Pôle de Constantine, 2010.
- [14] J. Kubiawicz, D. Bindel, Y. Chen, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao. "*Oceanstore : An architecture for global scale persistent storage*". In Proceedings of Acm Asplos, 2000.
- [15] P. Lezard. "*Etudes des réseaux de Petri Stochastiques, DEA Mathématique appliquées*". Université de Paul Sabatier, 1995.
- [16] P. Marlier. "*Sécurité du Peer-to-Peer*". www.labo-asso.com, 2007.
- [17] J. Marsan, A. M. Balbo, and G. Conte. "a class of generalized stochastic petri nets for the performances of multiprocessor systems". May1984.
- [18] Z. Mordji, M. Amad, and D. Aissani. "derivation of a queuing network model for structured p2p architectures". *Vecos 2014, Bejaia*, (76-84), 2014.
- [19] A. Muthitacharoen, R. Morris, T.M. Gil, and B. Chen. "*Ivy : A read/write Peer to Peer file system*". In Proceedings of 5th Symposium on Operating Systems Design and Implementation, 2002.
- [20] C.A. Petri. "*Kommunikation mit automaten. Phd dissertatio, Institut fur Instrumentelle Mathematik*". Number 65-377. University of Bonn, West Germany, 1962.
- [21] G. Pujolle. "*Les Réseaux*". Eyrolles, Paris, France, 1128, 2008.
- [22] A. S. Theotokis and D. Spinellis. "*A survey of Peer-to-Peer content distribution technologies*". ACM Computing Surveys, December 2004.
- [23] X. Yang and G. Veciana. "service capacity of peer to peer networks". *Department of Electrical and Computer Engineering The University of Texas at Austin*, 2004.
- [24] D. Zefouni. "*Evaluation des performances d'un réseau de files d'attente en TANDEM*". Université A/Mira de Béjaia, 2002.
- [25] A. Zimmermann and M. Knoke. "timenet 4.0 a software tool for the performability evaluation with stochastic and colored petri nets user manual armin zimmermann and michael knoke technische universit at berlin real-time". *Technische Universit at Berlin, Real-Time Systems and Robotics Group*, August 2007.

Résumé

Les réseaux *P2P* sont utilisés avec succès dans plusieurs domaines comme : le partage de fichiers, le partage de capacité de calcul et l'échange de messages instantanés, . Les modèles analytiques (réseaux de Petri, réseaux de files d'attente, les chaînes de Markov, etc) constituent des outils très performants pour la modélisation et l'évaluation des performances des réseaux P2P. Leurs adaptations pour les grands réseaux dynamiques, où la mobilité est un aspect très important nous semble une approche très prometteuse. Ceci permettra en particulier de mesurer plusieurs facteurs de performance. Dans ce mémoire, nous avons étudié l'évaluation de performance de l'architecture centralisée non structurée BitTorrent, le modélisé en utilisant les réseaux de Petri stochastiques généralisés par approche de simulation avec le logiciel TimeNet.

Mots clés : Réseaux P2P, modélisation, évaluation des performances, simulation, BitTorrent, réseau de Petri stochastique généralisé, TimeNet.

Abstract

P2P networks are successfully used in several areas such as : sharing files , calculation capacity sharing and exchanging instant messages. the models Analytical (Petri nets, queuing networks, Markov chains , etc.) are powerful tools for modeling and performance evaluation P2P networks. Their adaptations for large dynamic networks , where mobility is a very important aspect seems a very promising approach. This will especially to measure several performance factors.

In this brief, we studied the performance evaluation of the centralized architecture unstructured BitTorrent, the modele using stochastic Petri nets by generalized approach TimeNet simulation with the software ..

Keywords : : P2P networks, modeling, performance evaluation, simulation, BitTorrent, generalized stochastic Petri net , TimeNet .