
Un classifieur du comportement des utilisateurs dans les applications pair-à-pair de streaming vidéo

Ihsan Ullah* — Grégory Bonnet** — Guillaume Doyen* — Dominique Gaïti*

* Université de Technologie de Troyes – UMR CNRS 6279 STMR / ERA
12, rue Marie Curie – 10000 TROYES – France
ullah@utt.fr, doyen@utt.fr, gaiti@utt.fr

** Université de Caen Basse-Normandie – UMR CNRS 6072 GREYC / MAD
Boulevard du Maréchal Juin BP 5186 – 14032 Caen Cedex – France
gregory.bonnet@unicaen.fr

RÉSUMÉ. Depuis quelques années, les applications de streaming vidéo pair à pair sont devenues de plus en plus populaires. Cependant, ces systèmes souffrent toujours de problèmes de performance du fait de la dépendance mutuelle des pairs pour la fourniture du contenu. De ce fait, le comportement individuel des utilisateurs qui les contrôlent influence directement la performance du service offert. L'étude du comportement des utilisateurs se présente alors comme une piste prometteuse pour définir des mécanismes de contrôle adaptatifs. Toutefois, la littérature ne propose que des modèles globaux qui considèrent des utilisateurs homogènes en comportement. Dans cet article, nous proposons un classifieur bayésien qui permet de rattacher un utilisateur à une classe de comportement. Ce classifieur est construit sur les relations de dépendances mesurées dans des implantations opérationnelles de systèmes de streaming vidéo pair à pair, et les classes de comportement que nous proposons sont issues de mesures individuelles. Afin de valider notre modèle, nous présentons les résultats de simulations effectuées sur un millier de pairs sur une période de cent jours. Enfin, nous montrons un exemple d'application de ce classifieur pour la construction des topologies virtuelles robustes à la dynamique du réseau.

ABSTRACT. P2P-based live video streaming has become popular in recent years. Nevertheless, these systems still suffer from performance problems due to the mutual dependancy of peers about content delivery: the individual behavior of the users that control the peers has a direct impact over the performance of the service delivery. For that reason, studying the user behavior is mandatory to design adaptive constrol mechanisms. However, the literature propose global models that only consider homogeneous users. In this paper, we propose a Bayesian classifier that allows to associate a video streaming application user to a given behavioral class. This classifier is based on the dependencies measured in the operational peer-to-peer video streaming implementations. The behavioral classes we proposed are based on individual measurements. In order to validate our model, we present simulation results with a thousand peers on a hundred-day period. Finally, we present an example of an application of our classifier to design virtual topologies that are churn resilient.

MOTS-CLÉS : applications multimédia, apprentissage automatique, comportement des utilisateurs, réseaux P2P.

KEY WORDS: machine learning, multimedia applications, P2P networks, user behavior.

1. Introduction

Depuis quelques années, les applications de *streaming* vidéo fondées sur une architecture pair à pair sont devenues de plus en plus populaires. D'une part et contrairement au *multicast* IP, ces applications ne nécessitent pas de modifications profondes de l'infrastructure du réseau. D'autre part et contrairement aux architectures de type client/serveur, ces applications réduisent le besoin de déployer de nouveaux serveurs à mesure que le nombre d'utilisateurs du réseau augmente. Une architecture pair à pair permet à des hôtes, appelés pairs, de s'auto-organiser au sein d'un réseau virtuel, appelé *overlay*. Les pairs de l'*overlay* partagent leur puissance de calcul et leur bande-passante pour mettre en cache et partager la diffusion du contenu vidéo. Toutefois, ces systèmes souffrent toujours de problèmes de performance, notamment en termes de délais à l'initialisation ou de qualité du flux reçu. De plus, comme les pairs dépendent les uns des autres, le comportement d'un pair a un effet direct sur le réseau. Par exemple, le départ d'un pair peut entraîner une rupture dans le flux diffusé, diminuant ainsi la qualité de service pour les autres pairs. Pour cette raison, l'étude du comportement des utilisateurs, et par extension des pairs, est nécessaire pour définir des mécanismes de contrôle adaptatifs.

Dans cette optique, d'importantes campagnes de mesures ont été réalisées par de nombreuses équipes de recherche afin d'identifier un comportement moyen qui peut être utilisé pour déduire un modèle général des utilisateurs. Toutefois, dans un réseau réel, le comportement individuel ne suit pas nécessairement le comportement global et les travaux proposés dans la littérature peinent à traiter cette question du fait de la difficulté à établir un lien entre les traces obtenues et des utilisateurs particuliers, et cela en raison de l'utilisation d'adresses IP et d'identifiants de pairs dynamiques. C'est pourquoi nous proposons dans cet article un classifieur bayésien qui permet de rattacher à un utilisateur de streaming vidéo pair à pair une classe de comportement individuel définie au préalable. Ce classifieur est construit à partir des relations de dépendances observées lors des campagnes de mesures effectuées sur des implantations opérationnelles de systèmes de *streaming* vidéo pair à pair. Les classes de comportement que nous utilisons sont, quant à elles, issues de personnages fictifs proposés dans le cadre du projet On-Demand¹. Nous validons notre modèle en mesurant sa précision à partir de simulations effectuées sur un millier de pairs, et sur une période de cent jours. À partir de ces résultats, nous proposons une application de notre modèle à la construction de topologies d'arbres de diffusion optimisées en fonction de la classe de comportement. Nous montrons que cette approche permet de diminuer de manière importante le taux de perturbation du service face à la dynamique du réseau, et par conséquent, de minimiser les opérations de maintenance.

Cet article est structuré comme suit. Nous présentons dans la section 2 les travaux relatifs à la modélisation du comportement des utilisateurs, et plus particulièrement l'apprentissage de comportements individuels. Ensuite, nous identifions dans la section 3 les principaux critères permettant d'identifier les types d'utilisateurs et proposons un réseau bayésien pour les classer. Nous identifions en section 4 plusieurs classes d'utilisateurs et, à partir de simulations, nous évaluons les capacités d'apprentissage de notre réseau bayésien. Afin de valider l'intérêt de notre approche, nous proposons en section 5 une application de la classification pour la construction de topologies stables et donnons des résultats de simulation montrant que notre classifieur permet de construire des arbres

1. http://redback.sics.se/projects/ondemand_ipstv

de diffusion plus robustes à la dynamique du réseau. Pour terminer, la section 6 conclue ce travail et donne les pistes que nous suivons pour nos travaux à venir.

2. Travaux relatifs

Depuis quelques années, de nombreuses campagnes de mesures à large échelle ont été réalisées sur des applications de *streaming* vidéo [BRA 99, ACH 04, VIL 05, YU 06, BRA 07, CHA 08b]. Le tableau 1 présente un panorama des modèles globaux qui ont été extraits de ces mesures. Tous s'accordent sur le fait que la population d'un réseau varie selon un cycle journalier [ACH 04, VIL 05, YU 06, CHA 08b] avec un pic en milieu de semaine [ACH 04] et une décroissance durant les week-ends [ACH 04, VIL 05]. Concernant les modèles d'arrivée, les lois exponentielles [ACH 04, CHA 08b] et les lois de Poisson [BRA 99, YU 06] sont les plus utilisées tandis que, pour les durées de sessions, les modèles se fondent sur des lois lognormales [BRA 99, YU 06, BRA 07] ou exponentielles [VIL 05, CHA 08b]. Concernant les modèles de popularité, toutes les études [ACH 04, VIL 05, YU 06, CHA 08b] s'accordent sur des lois de Zipf bien qu'elles ne modélisent qu'approximativement les valeurs extrêmes.

Ref.	Loi d'arrivée	Loi de durée de session	Loi de popularité
[BRA 99]	Poisson ($\lambda = 0.68$)	lognormale	non mesurée
[ACH 04]	exponentielle	exponentielle	Zipf ($\alpha = 0.27$)
[VIL 05]	non mesurée	lognormales ($\mu = (0.16, 0.2), \sigma = (0.06, 0.27)$)	Zipf ($\alpha = 0.667$)
[YU 06]	pseudo-Poisson ($\lambda = 17, N = 27$)	lognormale ($\mu = 2.2, \sigma = 27$)	Zipf
[BRA 07]	non mesurée	lognormale ($\mu = 4.835, \sigma = 1.704$)	normale ($\mu = 33.2, \sigma = 17.1$)
[CHA 08b]	combinaison d'exponentielles	exponentielle	Zipf

Tableau 1. Panorama des modèles globaux dans la littérature

Toutefois, ces travaux s'intéressent uniquement à la définition de modèles globaux d'utilisateurs et non pas à l'identification de leur comportement individuel qui peut être très éloigné de ce comportement moyen. En effet, l'enquête sociologique de [RUD 08] identifie des téléspectateurs ayant des habitudes très différentes. D'autres travaux se sont alors intéressés à l'identification de critères de comportements et à leur prédiction. [TAN 06] ont mesuré la stabilité des utilisateurs à la fois dans des architectures pair à pair mais aussi sur des architectures client-serveur. Ils ont alors identifié une corrélation positive entre le temps passé dans le système et la durée de session restante d'un utilisateur. Ils proposent un modèle de sélection de voisins tel que les pairs présents depuis le plus de temps dans le réseau sont préférés aux autres. [WAN 08] propose une méthode similaire pour identifier des pairs stables. Afin de minimiser l'effet de l'attrition, ils se servent ensuite de ces pairs comme ossature du réseau. Toutefois, ces deux approches ont tendance à considérer comme instables tous les pairs récemment arrivés dans le réseau, ce qui diminue la qualité de service de ces derniers, et favorise donc leur instabilité. [LIU 09a] ont alors mesuré et analysé l'effet de la qualité du flux sur la stabilité et la contribution en bande-passante des pairs. Ils observent que cette qualité présente

une corrélation positive avec ces deux critères. Ils proposent alors un modèle qui reste toutefois un modèle global et ne considère que des utilisateurs homogènes.

Afin de définir des modèles individuels, [HOR 09] proposent plutôt d'apprendre la contribution en bande-passante des pairs à l'aide d'une machine à vecteur support (SMV) mais ils ne considèrent aucun autre critère de comportement. Dans nos travaux précédents [ULL 09], nous avons proposé des estimateurs de la stabilité des pairs fondés sur une moyenne mobile exponentielle et la règle de Bayes. De plus, nous avons proposé un mécanisme proactif pour anticiper le départ de voisins et en sélectionner de nouveaux dynamiquement. Toutefois, ce travail ne se fonde que sur l'historique des pairs. C'est pourquoi, dans [ULL 10], nous avons proposé une modélisation fondée sur un réseau bayésien pour tenir compte de l'influence de tous les critères de comportements. Ce travail présente toutefois deux limites : il est difficile d'obtenir des estimations avec une granularité satisfaisante, et l'apprentissage du modèle de l'utilisateur nécessite de très nombreuses observations. Ainsi d'après nos précédents résultats de simulation, l'estimation précise du comportement d'un pair n'a lieu qu'après une période d'apprentissage d'environ 35 jours ce qui n'est pas satisfaisant.

Afin de pallier ces limites, nous proposons de non plus apprendre le modèle individuel des utilisateurs mais, sachant l'existence préalable d'un ensemble de modèles types, d'étiquetter chaque utilisateur par le modèle qui approxime au mieux son comportement.

3. Classification des utilisateurs

Dans un premier temps, nous identifions à partir de la littérature les métriques qui sont discriminantes pour la classification des utilisateurs. Sur cette base, nous proposons un classifieur fondé sur un réseau bayésien.

3.1. Identification des critères

Le principal critère d'identification d'un utilisateur est le temps qu'il passe sur un canal donné sous certaines conditions. Ce critère, appelé durée de session, est en effet la seule observation qu'un pair peut avoir d'un autre, et par conséquent la seule observation sur laquelle il est possible de se fonder pour identifier un pair. Dans la littérature, quatre métriques ont été identifiées comme présentant un facteur d'impact sur la durée de session :

1) **la qualité du flux** : [LIU 09a, LIU 09b] ont identifié une corrélation positive entre la durée de session et la qualité initiale du flux. Cette qualité de flux est évaluée par la taille du tampon de donnée reçu initialement par l'utilisateur. En effet, un utilisateur qui reçoit un tampon initial important a tendance à rester plus longtemps connecté au réseau car il est moins sujet à des interruptions de service liées à l'indisponibilité du flux ;

2) **la popularité** : [LIU 09a, HEI 07, LIU 09b] observent que les utilisateurs ont tendance à présenter une durée de session plus longue lorsqu'ils regardent un programme populaire, et inversement pour les programmes non populaires. Dans les applications de diffusion de contenu, la popularité est calculée en fonction de la population présente dans le réseau ;

3) **le type de programme** : [CHA 08a] observent des durées plus courtes pour les programmes d'information et de musique comparativement aux documentaires et programmes pour les enfants. Ces auteurs ont ainsi pu identifier trois types de contenu qui entraînent des comportements différents : fiction (films, séries, programmes pour enfant), réalité (informations, documents) et sport ;

4) **l'heure de la journée** : [LIU 09a] observent que la durée de session est fortement corrélée à l'heure de la journée. En effet, la fin d'un programme ou les pauses publicitaires – correspondant à des heures particulières de la journée – sont marquées par des départs soudains de pairs [CHA 08a, HEI 07]. De plus, la popularité est plus élevée durant le jour [QIU 09].

Ainsi, nous pouvons classifier le comportement d'un utilisateur en fonction de sa durée de session sachant les valeurs de ces quatre métriques. Pour cela, nous proposons d'utiliser un réseau bayésien.

3.2. Un classifieur bayésien

Le classifieur que nous proposons est représenté par le réseau bayésien sur la figure 1. Il possède six nœuds : un pour la durée de session, quatre pour les métriques précédemment identifiées et un représentant la classe de l'utilisateur. La durée de session présente un arc vers la classe d'utilisateur, et les quatre métriques présentent un arc vers la durée de session et vers la classe de l'utilisateur.

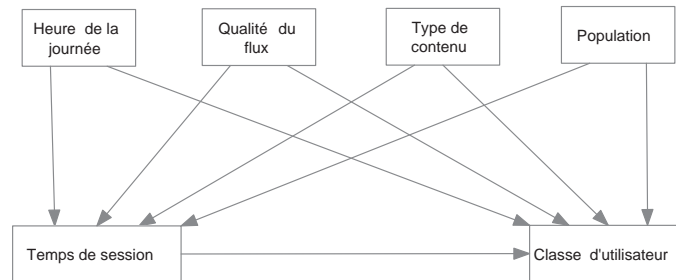


Figure 1. Diagramme du classifieur bayésien

Nous considérons chaque variable représentée par un nœud comme discrète. L'heure de la journée est discrétisée en 24 états, un pour chaque heure. Ce choix fait suite à des travaux précédents [ULL 11] qui montrent qu'une discrétisation par intervalle d'une heure assure un compromis efficace entre précision et vitesse d'apprentissage. Le type de contenu est, lui, discrétisé en 3 états : fiction, réalité et sports. La qualité du flux et la popularité sont toute deux discrétisées comme suit : le domaine de définition de la variable considérée est divisé selon un ordre total en cinq classe d'équivalence correspondant à un intervalle couvrant un cinquième du domaine de définition de la variable. Ceci peut être exprimé par l'équation 1 où S est la valeur de l'état calculé, V_{courant} est la popularité ou la qualité actuelle du canal tandis que V_{max} en est la valeur maximale.

$$S = \left\lceil 5 \cdot \frac{V_{\text{courant}}}{V_{\text{max}}} \right\rceil \quad (1)$$

Paramètre	Valeur
Durée de la simulation	60 jours (apprentissage) 40 jours (validation)
Population	1000 pairs
Contenu	fiction, réalité, sports
Durée d'un programme	2 heures
Algorithme d'inférence	Arbre de jonction
Algorithme d'apprentissage	Maximum de vraisemblance

Tableau 2. Paramètres de la simulation

La durée de session est, elle, discrétisée en 100 états d'une granularité d'une minute afin de couvrir toute les durées de session possibles. En dernier lieu, la classe de l'utilisateur est un nœud comprenant autant d'état que de catégories d'utilisateurs identifiées dans l'application cible. Dans notre cas, six classes d'utilisateurs ont été définies et sont présentées dans la section suivante.

4. Évaluation

Afin de valider notre proposition, nous avons réalisé nos simulations à l'aide de Matlab et de la boîte à outils Bayes Net Toolbox. Les paramètres de nos simulations sont résumés dans la table 2. Nous avons simulée pendant 100 jours une communauté de 1000 pairs entrant et quittant un réseau mono-canal dont le type de contenu change toutes les 2 heures. Nous utilisons alors le classifieur bayésien pour prédire la classe d'utilisateur d'un pair donné. Durant les 60 premiers jours, le classifieur réalise un apprentissage supervisé non bruité : toutes les variables sont observables et, à chaque observation, le réseau bayésien met à jour ses tables de probabilités conditionnelles. À partir du 61^e jour, la variable de classe d'utilisateur est cachée et le classifieur prédit cette dernière comme étant celle qui maximise la probabilité d'être sachant les valeurs des autres variables. Avant de présenter les résultats proprement dit, nous présentons plus en détail la manière dont nous simulons les utilisateurs sur le réseau.

4.1. Simulation des utilisateurs

Malgré l'existence de modèles globaux d'utilisateurs, il est difficile d'en dériver des modèles individuels. Dans le cadre du projet On-Demand qui a pour objectif d'améliorer la qualité de service sur les réseaux de diffusion de contenu, [RUD 08] ont identifié 6 archétypes de téléspectateurs (appelés persona) à partir d'une enquête sociologique. Ce travail présente une description qualitative, résumée sur la table 3, de modèles individuels d'utilisateurs que nous considérons comme des classes d'utilisateurs.

Afin de pouvoir simuler ces modèles, nous avons proposé dans nos travaux précédents [BON 11] un modèle semi-markovien non-homogène de ces personas. Dans un tel processus, l'état d'un utilisateur (en ligne ou hors ligne) à un instant donné dépend non seulement de son état à l'instant précédent comme pour tout modèle markovien mais aussi du temps qu'il a passé dans l'état courant et du temps global. Nous avons validé cette modélisation en confrontant une population de pairs générés aléatoirement par notre modèle aux résultats de mesures obtenus sur des systèmes réels et montré

Persona Paramètre	Johnatan (J)	Emma (E)	Stephan (S)
Âge	17	25	33
Intérêts	Sports et séries	Sans préférences	Nouvelles et sports
Temps de présence par jour	2 - 3 h.	1.5 h.	1.5 h. (Δ important)
Habitudes	Soirée et nuit	Nuit	Midi et soirée
Catégorie	Étudiant	Commerçant	Cadre
	Anna (A)	Peter (P)	Ellen (L)
Âge	46	58	69
Intérêts	Séries	Nouvelles et reportages	Variété et reportages
Temps de présence par jour	1.5 h.	1.5 h.	2 h.
Habitudes	Après-midi et soirée	Midi et soirée	Sans habitudes
Catégorie	Femme au foyer	Professeur	Retraitée

Tableau 3. Les six classes d'utilisateurs définies par [RUD 08]

Classe réelle	# cas	Classe prédite						# erreur	% erreur
		J	E	S	A	P	L		
J	32890	30128	645	381	619	359	758	2762	8,3977
E	26783	957	24488	714	205	198	221	2295	8,5689
S	34898	767	3277	26686	459	2146	1563	8212	23,5314
A	42497	743	906	558	39030	303	957	3467	8,1582
P	30928	476	631	1695	1375	26174	577	4754	15,3712
L	20817	436	219	730	1920	532	16980	3837	18,4321

Tableau 4. Résultats : matrice de confusion et erreur de classification

l'adéquation de ces deux approches. Ici, chaque persona correspond ainsi à une classe d'utilisateur pour notre classifieur bayésien, et chaque utilisateur est associé à un processus semi-markovien qui simule le persona.

4.2. Premiers résultats

Les résultats sont présentés sur la table 4. La matrice de confusion ainsi que le nombre d'erreurs de classification sont donnés. Afin de clarifier ces résultats, l'histogramme illustrant ces erreurs est donné en figure 2. Nous pouvons remarquer que pour la moitié des classes d'utilisateurs, l'erreur reste en-dessous de 10%. Cependant, elle peut atteindre jusqu'à 23,5% pour les autres. Ainsi, les résultats varient fortement en fonction de la classe considérée. La principale raison de cette erreur vient du fait que le classifieur doit nécessairement classer un pair dans une et unique classe, quelle que soit l'information qu'il possède. Il est des cas où la distribution de probabilité entre les classes tend vers une distribution uniforme, empêchant de donner ainsi une prédiction pertinente car certaines classes sont semblables entre elles.

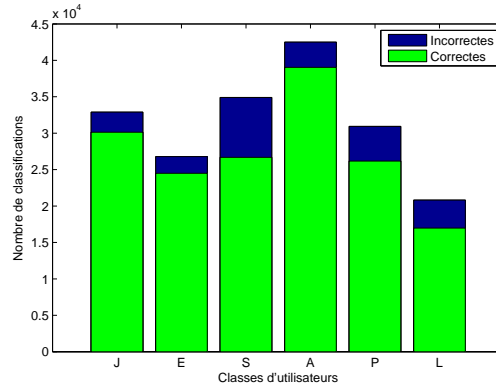


Figure 2. Classifications selon les différents personas

4.3. Amélioration de la précision

Afin de palier les limites de cette classification initiale, nous proposons deux approches pour en améliorer la précision : accepter de ne pas classer un utilisateur et tenir compte d'un historique.

4.3.1. Une classe inconnue

Pour les applications où la précision prime sur le fait de classer tous les utilisateurs, décider entre deux classes n'est pas nécessairement pertinent. Par exemple, nous pouvons considérer un protocole de construction d'une ossature de réseau à partir d'utilisateurs stables pour lequel il suffit d'identifier un petit nombre d'utilisateurs pour avoir un cœur stable. Nous proposons d'améliorer le classifieur pour de telles applications en introduisant une nouvelle classe d'utilisateur, appelée *inconnue*, et en fixant un seuil de précision dans l'intervalle $[0, 1]$. Si la probabilité d'appartenance à la classe prédite est en-dessous de ce seuil alors le classifieur indique que la classe est inconnue. Nous avons simulé plusieurs niveaux de seuil, et indiquons en table 5 l'erreur pour chaque classe et le nombre d'inconnus. Nous pouvons alors remarquer que l'erreur pour chaque classe décroît au fur et à mesure que le seuil augmente, au prix d'une augmentation du nombre d'inconnus. Nous pouvons nous servir de cette approche pour fixer un compromis entre précision et classification totale.

4.3.2. Historique des classifications

Dans les simulations précédentes, nous avons classé les utilisateurs à partir d'une unique observation. Nous pouvons aussi améliorer la précision en considérant l'historique des classifications. Si nous considérons que nous n'avons aucune connaissance sur la distribution *a priori* des classes, nous utilisons une distribution uniforme entre les classes et la probabilité qu'un utilisateur U_i appartienne à une classe C_j est donnée par l'équation 2 où O est l'ensemble des précédentes classifications et α le nombre de classifications où $U_i \in C_j$. Il est aussi possible de considérer une distribution *a priori* connues. Dans [BON 11], nous avons considéré que la distribution des utilisateurs suivait les statistiques de l'INSEE en fonction de leur tranche d'âge.

Seuil	Erreur pour chaque classe						Inconnue (%)
	<i>J</i>	<i>E</i>	<i>S</i>	<i>A</i>	<i>P</i>	<i>L</i>	
0,2	8,4424	8,1918	23,0397	7,8177	14,9615	17,8321	0,5111
0,3	8,2823	7,9772	22,8758	7,6250	14,7406	17,7039	1,3145
0,4	6,8627	7,5310	21,8501	7,3008	14,1242	16,5759	3,5474
0,5	5,8305	6,3165	18,8363	6,4280	11,7597	13,8887	8,4406
0,6	4,0089	5,8304	8,0827	4,4150	9,2384	12,4707	17,8081
0,7	2,9120	4,2268	6,4121	3,9540	7,9906	12,0455	22,7278
0,8	1,7908	3,6179	4,4427	2,7311	2,4767	7,4937	34,8954
0,9	0,7829	2,3012	2,4788	0,7595	0,8710	2,3440	51,9683

Tableau 5. Évolution de l'erreur sous différents seuils

$$f(\phi_j) = \frac{\alpha_j + 1}{|O| + 6} \quad (2)$$

La figure 3 illustre le gain en précision en fonction de la taille de l'historique sans connaissance au préalable et avec une distribution connue *a priori*. Dans les deux cas, l'augmentation de la taille de l'historique diminue l'erreur observée avec au mieux une erreur moyenne d'environ 5% sur l'ensemble des classes pour un historique de 20 observations. Au delà, le gain est minimal. Si l'utilisation de connaissance pour déterminer la distribution *a priori* est plus efficace sur des petites tailles d'historique, les deux approches donnent des résultats équivalents après 15 observations.

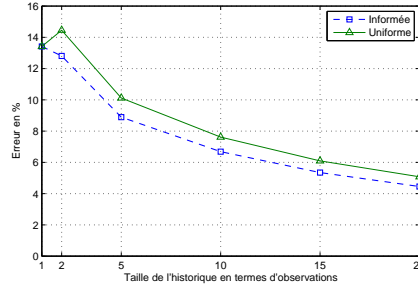


Figure 3. Utilisation de l'historique

5. Application : construction de topologies contrôlées

Une des applications de notre classifieur est la construction de topologies de diffusion stables. Nous nous sommes intéressés aux arbres de diffusion car ces topologies sont efficaces en termes de temps de délai de diffusion tout en minimisant le coût de communication comparativement aux topologies maillées. Toutefois, elles sont sensibles à la dynamique du réseau : le départ d'un pair interrompt potentiellement le service dans l'ensemble de son sous-arbre. Ainsi, construire un arbre structuré en fonction de la stabilité des pairs permet d'améliorer sensiblement ses performances.

Nous avons défini le scénario suivant. Un ensemble de 1000 utilisateurs entrent et quittent dynamiquement un réseau de *streaming* vidéo sur une période de 10 jours. Chaque utilisateur est associé à une classe présentée en section 4.1, définissant alors son comportement individuel. Ils reçoivent le contenu en rejoignant un arbre de diffusion dont la racine (source) est fixe et dont le degré est 5. Le degré de chaque autre nœud varie dans $[1; 5]$. Nous considérons alors trois cas : (1) l'algorithme place les pairs dans l'arbre sans tenir compte de leur classe. Il s'agit du comportement standard d'un algorithme construisant une topologie aléatoire ; (2) l'algorithme estime tout d'abord la classe des pairs avant de les placer dans l'arbre. À partir de la classe estimée d'un pair, nous pouvons estimer son temps de session et les pairs les plus stables sont placés au plus proche de la racine tandis que les moins stables sont placés au plus proche des feuilles. Ce processus est réitéré toutes les cinq minutes pour estimer à nouveau le temps de session des pairs et les réordonner en fonction de leur date de départ estimée. Nous construisons ainsi une topologie contrôlée ; (3) nous considérons le cas optimal (intitulé *idéal* sur la figure 4.a.) où la prédiction est donnée par un oracle. Dans ce cas, aucune erreur de classification n'est présente.

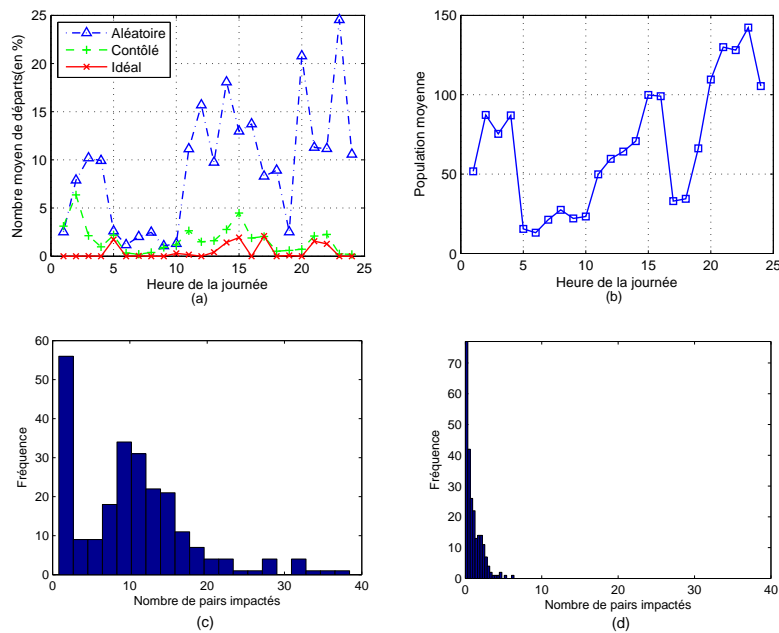


Figure 4. Résultats : (a) perturbation du réseau ; (b) utilisateurs en ligne ; (c) distribution de la perturbation sur une topologie aléatoire ; (d) sur une topologie contrôlée

Nous utilisons notre réseau Bayésien pour estimer le temps de session des pairs. En effet, ce modèle permet d'estimer n'importe laquelle de ces variables, dont fait partie le temps de session. Il s'agit donc d'un processus qui se fait en deux temps : (1) nous classifions le pair après une observation, c'est-à-dire après que le pair a déjà joint au moins une fois le réseau ; (2) nous faisons l'hypothèse que toutes les autres variables, à l'exception du temps de session, sont connues. Cette

hypothèse est réaliste car l'heure de la journée et la qualité du flux peuvent être mesurées directement chez le pair alors que la popularité du flux, définie par la population instantanée d'une communauté, peut être estimée par un protocole d'aggregation décentralisé [MAK 09]. Afin de comparer chaque approche, nous nous intéressons au nombre de pairs subissant une interruption de service à un instant donné suite au départ d'un autre pair. La figure 4.a représente le taux de perturbation du réseau sur une fenêtre de 24 heures mis en regard de la population du réseau sur la même période (figure 4.b). Dans le cas aléatoire, le taux moyen de perturbation est de 10,23 nœuds par minute tandis qu'il est de 1,03 nœuds par minutes pour la topologie contrôlée. Nous remarquons aussi que l'approche contrôlée présente des résultats comparables au cas idéal et qu'au fur et à mesure de la journée, sa performance tends vers le cas idéal en apprenant le comportement des utilisateurs. La variation du taux de perturbation quant à elle reste faible dans le cas d'une topologie contrôlée. En effet, les figures 4.c et 4.d indiquent la fréquence du nombre de pairs impactés par un départ. Nous pouvons remarquer que la topologie aléatoire (figure 4.c) présente une distribution bimodale et comporte des variations importantes, contrairement à la topologie contrôlée (figure 4.d) qui indique clairement que la majorité des changements n'implique que les feuilles de l'arbre. Au vu de ces premiers résultats, notre topologie contrôlée permet une amélioration significative de la stabilité du réseau.

6. Conclusions et travaux futurs

Modéliser de manière précise les utilisateurs d'une application pair à pair de *streaming* vidéo ouvre de nombreuses perspectives, depuis son intégration dans des outils de simulations à la gestion de ressources pour les fournisseurs de services en passant par l'amélioration de la qualité de service. Toutefois, les modèles globaux proposés dans la littérature s'éloignent du comportement individuel des utilisateurs, ce qui en réduit la portée. Dans cet article, nous proposons un classifieur bayésien permettant d'associer à chaque utilisateur une classe de comportement, et nous l'instancions sur six classes d'utilisateurs définis à l'aide de la méthodologie des personas. Nous montrons que le classifieur fournit un résultat assez satisfaisant sauf pour les classes dont le comportement est diffus. Aussi, nous introduisons deux méthodes pour l'améliorer. La première repose sur un seuil de précision à fixer ; à son niveau le plus élevé, seule la moitié des classes est prédite mais avec une précision très satisfaisante. La seconde méthode considère l'historique des prédictions dans la classification. Afin d'illustrer une des applications de notre classifieur, nous avons simulé un arbre de diffusion qu'un fournisseur de service peut contrôler en plaçant les utilisateurs appartenant aux classes les plus stables vers la racine. Cette approche permet de sensiblement diminuer le taux de perturbation du réseau comparé à des mécanismes qui ne tiennent pas compte du comportement des utilisateurs. Ce travail doit toutefois être étendu. Tout d'abord, la méthode de classification elle-même peut être améliorée notamment en combinant l'introduction de seuils de classification avec la prise en compte de l'historique. Cette combinaison permettrait d'obtenir des observations de meilleure qualité et ainsi améliorer la précision de la classification avec toutefois un coût plus important sur le nombre d'observations nécessaires à la classification. Ensuite, dans cet article nous avons seulement évalué le gain potentiel à utiliser la classification pour construire des topologies stables. Les algorithmes et protocoles qui supportent cette application sont encore à définir et à évaluer en termes de performance et de coût par le biais d'expérimentations. Il serait particulièrement intéressant de définir un algorithme de placement décentralisé permettant aux pairs de s'auto-organiser et de non plus se reposer sur un fournisseur de service.

7. Bibliographie

- [ACH 04] ACHARYA S., SMITH B., « Characterizing user access to videos on the world wide web », *Lecture Notes in Computer Science*, vol. 2720, 2004, p. 375–384.
- [BON 11] BONNET G., ULLAH I., DOYEN G., FILLATRE L., NIKIVOROV I., GAÏTI D., « A semi-markovian individual model of users for P2P video streaming applications », *Proceedings of the 4th NTMS*, 2011.
- [BRA 99] BRANCH P., EGAN G., TONKIN B., « Modeling interactive behaviour of a video based multimedia system », *Proceedings of the IEEE International Conference on Communications*, 1999, p. 978-982.
- [BRA 07] BRAMPTON A., MACQUIRE A., RAI I., NICHOLAS J.-P., MATHY L., FRY M., « Characterising user interactivity for sports video-on-demand », *Proceedings of the 17th NOSSDAV*, 2007.
- [CHA 08a] CHA M., RODRIGUEZ P., CROWCROFT J., MOON S., AMATRIAIN X., « Watching television over an IP network », *Proceedings of the 8th IMC*, 2008, p. 71–84.
- [CHA 08b] CHANG B., DAI L., CUI Y., XUE Y., « On feasibility of P2P on-demand streaming via empirical VoD user behavior analysis », *Proceedings of the 28th ICDCS*, 2008, p. 7–11.
- [HEI 07] HEI X., LIANG C., LIANG J., LIU Y., ROSS K. W., « A measurement study of a large-scale P2P IPTV system », *IEEE Transactions on Multimedia*, vol. 9, n° 8, 2007, p. 1672–1687.
- [HOR 09] HOROVITZ S., DOLEV D., « Collabrium : active traffic pattern prediction for boosting P2P collaboration », *Proceedings of the 18th WETICE*, 2009, p. 116–121.
- [LIU 09a] LIU Z., WU C., LI B., ZHAO S., « Distilling superior peers in large-scale P2P streaming systems », *Proceedings of the 28th INFOCOM*, 2009, p. 82–90.
- [LIU 09b] LIU Z., WU C., LI B., ZHAO S., « Why are peers less stable in unpopular P2P streaming channels ? », *Networking*, 2009, p. 274–286.
- [MAK 09] MAKHLOUFI R., BONNET G., DOYEN G., GAÏTI D., « Decentralized aggregation protocols in peer-to-peer networks : a survey », *Proceedings of the 4th MACE*, 2009, p. 111-116.
- [QIU 09] QIU T., GE Z., LEE S., WANG J., XU J., ZHAO Q., « Modeling user activities in a large IPTV system », *9th ACM SIGCOMM Conference on Internet Measurement*, 2009, p. 430–441.
- [RUD 08] RUDSTROM A., SJOLINDER M., « Capturing TV user behaviour in fictional character descriptions », rapport, October 2008, SICS.
- [TAN 06] TANG Y., SUN L., LUO J.-G., ZHONG Y., « Characterizing user behavior to improve quality of streaming service over P2P networks », *Proceedings of the 7th PRCM*, 2006, p. 175–184.
- [ULL 09] ULLAH I., BONNET G., DOYEN G., GAÏTI D., « Improving performance of ALM systems with Bayesian estimation of peers dynamics », *Proceedings of the 12th MMNS*, 2009, p. 157–169.
- [ULL 10] ULLAH I., BONNET G., DOYEN G., GAÏTI D., « Modeling user behavior in P2P live video streaming systems through a Bayesian network », *Proceedings of the 4th AIMS*, 2010, p. 2–13.
- [ULL 11] ULLAH I., DOYEN G., BONNET G., GAÏTI D., « User behavior anticipation in P2P live video streaming systems through a Bayesian network », *Proceedings of the 12th IM*, 2011, page to appear.
- [VIL 05] VILAS M., PANEDA X.-G., GARCIA R., MELENDI D., GARCIA V.-G., « User behavior analysis of a video-on-demand service with a wide variety of subjects and lengths », *Proceedings of the 31st EUROMICRO Conference on Software Engineering and Advanced Applications*, 2005, p. 330-337.
- [WAN 08] WANG F., LIU J., XIONG Y., « Stable peers : existence, importance and application in Peer-to-Peer live video streaming », *Proceedings of the 27th INFOCOM*, 2008, p. 1364–1372.
- [YU 06] YU H., ZHENG D., ZHAO B. Y., ZHENG W., « Understanding user behavior in large-scale video-on-demand systems », *Operating Systems Review*, vol. 40, n° 4, 2006, p. 333–344.