

Matchmaking dans les jeux multijoueurs en ligne : étudier les traces utilisateurs pour améliorer l'expérience de jeu

Maxime Véron, Olivier Marin, Sébastien Monnet
Sorbonne Universités, UPMC Univ Paris 06, Équipe REGAL, LIP6, F-75005, Paris, France
CNRS, UMR_7606, LIP6, F-75005, Paris, France
Inria, Équipe REGAL, F-75005, Paris, France
Email : prénom.nom@lip6.fr

Résumé

L'acquisition et l'analyse de traces d'utilisation réelles issues de jeux vidéo permettent d'acquérir une connaissance approfondie des habitudes, des comportements et des attentes des joueurs. Cela constitue une étape cruciale pour la conception de services logiciels de qualité pour les jeux vidéo. Toutefois, les acteurs de cette industrie protègent jalousement de telles données afin de dissuader une éventuelle réutilisation par la concurrence et pour éviter la triche. Nous avons contourné ce problème en collectant, sur le serveur d'un jeu en vogue (League of Legends), des données publiques représentant plus de 28 millions de sessions.

Dans cet article, nous présentons notre base de données disponible librement en ligne. Nous détaillons ensuite l'analyse et les conclusions que nous tirons de ces données concernant les exigences d'un service essentiel pour les jeux multijoueurs en ligne : le matchmaking, ou mise en relation de joueurs.

1. Introduction

L'amélioration d'un service logiciel pour les jeux passe par une étude approfondie de son comportement lorsqu'il est utilisé par de vrais joueurs. Pour ce faire, il est crucial d'obtenir et analyser des données réelles. Or ces données sont très difficiles à obtenir car les développeurs de jeux veulent éviter leur réutilisation par la concurrence ou par des tricheurs potentiels.

Le *matchmaking* est un service essentiel pour les jeux multijoueurs en ligne (en anglais *Multiplayer Online Games* ou *MOGs*). Il permet la mise en relation de joueurs (*matchmaking*) pour les réunir dans une session partagée en réseau, et a un fort impact sur l'expérience de jeu. Cependant, la recherche en un temps minimal d'adversaires disponibles et de niveaux similaires est un problème particulièrement complexe. Les solutions actuelles favorisent généralement le temps de réponse, au détriment de la qualité du *matching* (mise en relation). En conséquence, des joueurs de bon niveau se retrouvent souvent confrontés à d'autres joueurs de faible niveau ; une situation qui ne satisfait aucun des acteurs concernés. En outre, les temps de réponse peuvent demeurer très longs, parfois plusieurs heures d'attente avant qu'une session de jeu ne puisse commencer.

Malgré cela, les producteurs de jeux vidéo préfèrent minimiser les enjeux liés à l'obsolescence des services tels que le matchmaking et misent généralement sur l'amélioration des graphismes et de la jouabilité pour rivaliser avec leur compétiteurs. La réutilisation de composants existants

leur permet ainsi de réduire leurs coûts de production. Dans ce contexte, et étant donné que l'industrie du jeu vidéo protège ses données utilisateur, il est particulièrement difficile d'argumenter en faveur d'une amélioration des services logiciels pour les jeux.

Dans cet article, nous nous concentrons sur l'acquisition de données issues de sessions de jeu réelles. Nous utilisons les données recueillies à partir d'un jeu en ligne à la mode [1] pour montrer que les matching incorrects et les temps de réponse sont des points cruciaux pour l'expérience de jeu.

Nos principales contributions sont les suivantes :

1. Nous décrivons notre ensemble de données, disponible librement, qui contient des informations relatives à plus de 28 millions de sessions de jeu [4].
2. A partir de ces données, nous livrons une analyse détaillée des principaux problèmes soulevés par le matchmaking.

2. Notre cas d'étude : le matchmaking dans League of Legends

Cette section décrit notre cas d'étude : le service de matchmaking du jeu League of Legends (LoL). LoL est un jeu populaire auprès d'une communauté vaste et très active, comptant plus de 12 millions de joueurs connectés par jour à travers le monde en 2012 [14]. Notre choix s'est porté naturellement sur ce jeu en ligne pour plusieurs raisons. Tout d'abord, son succès nous assure une abondance de traces utilisateurs. De plus, il s'agit d'un jeu compétitif où le classement est un élément central de satisfaction pour les joueurs. Le classement nous semble donc être une bonne métrique pour étudier d'une part le comportement des joueurs, et d'autre part la qualité de service en termes d'expérience de jeu : notamment en ce qui concerne la constitution des équipes. Enfin, LoL est basé sur des sessions de jeu relativement courtes : environ 34 minutes selon les développeurs du jeu [3]). Le service de matchmaking est donc fréquemment utilisé et joue un rôle important dans l'expérience de jeu globale.

2.1. League of Legends : un aperçu

League of Legends est un jeu vidéo de type "multiplayer online battle arena" (MOBA) développé et édité par Riot Games. Des joueurs sont groupés en deux équipes comprenant cinq coéquipiers chacune. Les deux équipes s'affrontent dans une arène afin de détruire le bâtiment principal de l'équipe adverse appelé *nexus*.

Les serveurs de LoL définissent une *partie* comme étant la session de bagarre qui oppose les joueurs les uns contre les autres à l'intérieur de l'arène.

Trois facteurs sont essentiels à l'expérience de jeu : (i) les temps d'attente entre deux parties, (ii) la précision du matchmaking, et (iii) les temps de réponse du serveur en cours de partie. Avec un temps d'attente moyen de 90 secondes entre les sessions, les joueurs peuvent passer beaucoup de temps à attendre qu'une session commence (voir paragraphe 4.1). Cela est particulièrement vrai pour les joueurs très compétents : leur rareté rend plus difficile la recherche de 9 autres joueurs à leur mesure. Constituer une partie avec des joueurs de niveaux disparates réduit les temps d'attente de manière significative, mais cela peut avoir un fort impact négatif sur l'expérience de jeu. Les joueurs faibles se sentiront impuissants face à des adversaires beaucoup plus forts, tandis que les forts vont perdre du temps sur une session ennuyeuse avec peu ou pas de récompense à la clé. De manière orthogonale à ces questions, le temps de réponse du serveur est crucial dans ce jeu qui nécessite des réflexes extrêmement aiguisés. Les lags causés par les serveurs peuvent augmenter le ping jusqu'à 300%, ce qui entrave sérieusement l'expérience de jeu.



FIGURE 1 – Equivalence entre *Rang* et *Ligues*

LoL est un jeu compétitif où le classement prend une part importante, à la fois en termes de matchmaking et en termes de statut du joueur. Le *rang* définit le niveau de compétences du joueur. Obtenir un bon rang est le principal objectif des joueurs de LoL voulant s'investir dans le jeu. Une fois les joueurs classés, ils sont triés dans des catégories appelées *ligues*, et sous-catégorisés en *divisions*. Ce classement peut évoluer rapidement au fil du temps, il doit donc être calculé continuellement et c'est pourquoi la plupart des jeux utilisent le système de notation Elo. Le système de notation Elo [11] a été inventé comme une méthode améliorée de calcul des niveaux de compétences relatives des joueurs d'échecs. Aujourd'hui, de nombreux autres jeux, y compris les compétitions multi-joueurs, l'ont adapté pour leur propre usage. Par la suite, nous utilisons les termes classement Elo, ou MMR (abréviation de *MatchMaking Rating*) pour parler du Rang des joueurs.

La figure 1 décrit les équivalences entre les *Ligues* et le rang d'un joueur.

Les formules utilisées dans League of Legends pour les calculs de classement n'ont pas été divulguées publiquement. Cependant, la plupart des implémentations de classement partagent les mêmes bases héritées du système de notation Elo d'origine. Les développeurs de Riot Games fournissent cependant des informations intéressantes sur le matchmaking de LoL sur le site Web et les forums dédiés [6, 2]. A partir de ces informations et en analysant notre base de données, nous avons déduit les règles générales du matchmaking dans LoL et nous les décrivons ci-après.

2.2. Le Matchmaking dans League of Legends

Les joueurs de LoL peuvent rejoindre plusieurs types de parties, associées à différentes *files d'attente* sur les serveurs. Un groupe de personnes peut rejoindre la même *file* afin de jouer dans la même équipe : ils sont appelés des *premade*. Une *premade* peut comprendre de un à cinq joueurs. Les parties *normales* ne comptent pas dans le Classement officiel, tandis que les parties *classées* comptent et ne sont ouvertes qu'aux joueurs chevronnés (à partir du niveau 30). Notre étude se concentre principalement sur les parties *classées* car elles rassemblent les joueurs dont les statistiques sont les plus susceptibles d'être représentatives de la communauté de joueurs

assidus. De plus, un matchmaking équitable est plus critique pour les parties *classées* car elles constituent un enjeu réel pour les joueurs.

Selon les développeurs de Riot Games, le matchmaking dans LoL est basé sur : le classement du joueur, l'expérience de chaque joueur (nombre de matchs joués), et la taille de la premade.

Le classement du joueur pèse plus car c'est le facteur qui a le plus d'impact sur le résultat de la partie. LoL essaie de faire correspondre les équipes aussi équitablement que possible : il calcule la moyenne de niveau des équipes, et utilise ensuite ces moyennes pour faire correspondre des équipes de même niveau dans le combat. Cette solution accélère le matchmaking d'une équipe entière, mais il peut en résulter un déséquilibre dans le jeu si deux joueurs aux classements très différents rejoignent la partie en premade. Un joueur moins qualifié est alors susceptible de faire face à un adversaire bien classé. Dans un contexte compétitif, ce genre de solution sacrifiant l'équité à la rapidité n'est pas satisfaisante.

Ajouter l'expérience du joueur (nombre total de parties jouées) comme critère de correspondance peut permettre d'atténuer ce problème, mais cela augmente aussi de 50 % le temps d'attente moyen. Cela montre l'impact direct de l'ajout d'un seul paramètre sur le temps d'attente moyen au moment du matchmaking.

Les *premades* peuvent augmenter le temps d'attente ; ils peuvent même induire des combats asymétriques. Si le système de recherche ne peut pas trouver une équipe adverse avec une premade de la même taille et du même niveau dans le temps imparti (environ 30 secondes pour une partie normale), il rassemble des joueurs seuls avec des rangs individuels supérieurs pour construire l'équipe adverse. Le rang supérieur de chaque joueur est censé compenser la moindre coordination dans l'équipe, cependant cela conduit généralement à une victoire des joueurs mieux classés.

3. Acquisition des données de League of Legends

Afin d'obtenir des données pour notre étude, nous avons récolté des informations publiques du serveur de League of Legends. L'ensemble des données récupérées sur un mois couvre plus de 28 millions de sessions de jeu. Nous publions ces données en ligne [4], et les mettons librement à disposition sous forme d'une base de données. Cette section décrit la nature des données que nous avons acquises ainsi que notre méthode de récupération.

3.1. Nature des données reçues

Nous distinguons trois catégories dans l'ensemble des données acquises :

La première catégorie, *informations de l'avatar*, regroupe des données qui caractérisent le statut d'un joueur dans le jeu. Cette catégorie est commune à de nombreux jeux tels que World of Warcraft ou Diablo 3, dans lesquels le but principal est l'évolution d'un avatar. Par exemple, les objets que l'avatar possède ou les dommages qu'il peut infliger sont des champs qui peuvent être trouvés dans tous les jeux de rôle ainsi que dans les jeux de type tir à la première personne (*First Person shooter* ou *FPS* en anglais). Cette catégorie est de loin la plus grande, représentant 43 champs de données sur 77.

La deuxième catégorie, *variables de maintenance*, rassemble le contenu qui est spécifique à League of Legends. Les variables de maintenance, telles que l'identifiant du joueur et l'horodatage, n'existent que pour aider les développeurs à maintenir le jeu. Les 13 champs de cette catégorie nous ont notamment servi à trier les joueurs dans notre base de données. Bien que ces données représentent du contenu spécifique à League of Legends, certaines d'entre elles permettent des analyses liées au business model de toute une catégorie d'entreprises de jeux vidéo. C'est le cas, par exemple, des données concernant les *skins* que les joueurs appliquent à leurs avatars,

des extensions graphiques achetées en échange d'argent réel dans LoL.

La troisième et dernière catégorie englobe les données relatives au *matchmaking*. Elle contient certains champs des catégories précédentes : aussi bien des informations précises sur les avatars, que des données sur la latence du réseau. Le champ *userServerPing* révèle la latence moyenne subie par un joueur durant un match. Le champ *timeInQueue* est également crucial pour notre analyse car il représente le temps d'attente avant qu'un joueur n'intègre une équipe avec d'autres joueurs. Nous utilisons également des données concernant les avatars dans notre analyse du matchmaking de LoL, comme le nombre d'avatars ennemis tués par le joueur (*kill*) et le nombre de fois où il est mort (*num_Death*) afin de détecter les parties mal équilibrées.

3.2. Services utilisés pour récupérer les données

Pour la récupération des données de LoL nous avons utilisé un code gratuit open source [5] pour emballer/déballer le contenu des messages échangés avec le serveur.

Les serveurs de LoL utilisent un paradigme d'*appel de fonction distante* pour traiter les demandes. Un message doit être construit de telle sorte qu'il appelle la bonne *fonction* du bon *service* et qu'il contienne les paramètres de la fonction.

Un exemple typique d'une demande est la suivante : `< " summonerService " " getSummonerByName " " darkKnight67 " >`.

Notre première tâche a consisté en l'examen de l'ensemble des services de LoL (plus d'une centaine), puis en l'identification de ceux associés aux statistiques de jeu. Parmi ces derniers, nous avons identifié deux services spécifiques : le premier gère l'historique récent des joueurs, et le second permet d'obtenir des statistiques sur une partie de jeu dont l'identifiant est connu. Plus précisément, en termes de requêtes au serveur de LoL, nous avons utilisé la fonction *getSummonerByName* du *summonerService* afin de traduire les *accountId* en ce que RiotGames appelle en interne les *summonerId*. Une fois ce *summonerId* obtenu, nous pouvons alors appeler *getRecentGames* sur le *playerStatsService* pour obtenir un tableau de parties à partir du serveur de LoL.

Chaque historique de joueur contient 10 parties ou moins, selon le nombre de matchs joués au cours des sept derniers jours.

Les serveurs de LoL traitent les données sous forme de paire clé/valeur mais les envoient à travers le réseau en données brutes non structurées. Une grande partie de notre travail a été de redéfinir une classe Java appropriée décrivant les statistiques de jeu, avec laquelle nous avons pu saisir tous les champs envoyés par le serveur lors de la demande de l'historique des parties. Une fois cette tâche réalisée, nous avons généré tous les *summonerId* possibles dans l'espace de nommage du serveur et les avons soumis pour obtenir les données d'historique. Nous avons ensuite stocké les résultats dans une base de données. Cela nous permet de réaliser des recherches efficaces et de conduire l'analyse présentée en section 4.

4. Analyse du système de matchmaking

Cette section présente notre analyse des performances du service de matchmaking de LoL à partir des données récoltées. Nous évaluons l'impact de ce service sur l'expérience de jeu en termes de temps d'attente avant d'obtenir une session, en termes de précision du matching, et enfin, en termes de temps de réponse du serveur.

4.1. L'influence du rang sur les temps d'attente

Notre première évaluation, représentée par la ligne continue de la figure 2, porte sur la répartition des joueurs de LoL en fonction de leur classement. Cette information est précieuse pour

la conception et l'évaluation de nouvelles solutions de matchmaking, notamment par corrélation entre les compétences des joueurs et la qualité du service. Dans LoL, la valeur initiale du classement pour les débutants est de 1200, ce qui explique pourquoi il y a une grande majorité de joueurs autour de cette valeur. Un autre résultat prévisible est le nombre de joueurs qui diminue au fur et à mesure que le classement augmente. Monter dans le classement requiert de la pratique et des compétences ; donc les joueurs très bien classés sont plus rares. Une conséquence immédiate de cette distribution est que la mise en relation des joueurs les mieux classés est plus difficile à effectuer, et il est logique qu'elle prenne plus de temps.

La figure 2 illustre cette question à travers la ligne pointillée qui représente le temps moyen d'attente en fonction du classement des joueurs. Au-dessus du rang 1700, le temps moyen d'attente augmente de façon exponentielle. Il atteint jusqu'à 45 minutes pour les plus hauts rangs, mais nous n'avons pas inclus cette valeur dans la figure afin de préserver sa lisibilité ; en effet, la grande majorité des temps d'attente sont inférieurs à 100 secondes. Dans les forums dédiés à LoL, les développeurs de Riot Games affirment qu'ils visent un temps d'attente moyen qui ne dépasserait pas 30 secondes. Nos observations montrent que le temps d'attente moyen pour la grande majorité des joueurs est proche de 90 secondes. On pourrait considérer que 90 secondes est une durée acceptable. Cependant d'autres calculs montrent que sur 825000 demandes de parties classées émanant de joueurs de tous rangs, 65 259 ont subi plus de 5 minutes d'attente avant d'être mis en relation. Cela équivaut à 7,9 % des matchs qui ne démarrent pas dans un délai acceptable.

Nous tirons une conclusion importante de ces observations : les vives critiques émises par les joueurs de LoL en ce qui concerne le système de matchmaking sont justifiées. Il y a bien un problème de temps d'attente dans deux cas distincts : lorsque la mise en relation concerne des classements de joueurs associés soit à des effectifs très importants, soit à des effectifs en très petit nombre. Les deux courbes présentent des maxima simultanés autour de la valeur Elo de 1200. Cela démontre l'effet de goulot d'étranglement du service de matchmaking face à un très grand nombre simultané de requêtes. Les données permettant cette observation couvrent plus d'un mois d'observations ; il est donc peu vraisemblable que cette conclusion soit induite par des problèmes temporaires de serveur.

4.2. Impact de la distance du matching sur l'expérience de jeu

Notre prochaine évaluation s'intéresse à l'impact de la distance de la mise en relation sur l'expérience de jeu. Une mauvaise mise en relation, c'est-à-dire contre des adversaires avec des niveaux de compétence très différents, se traduit généralement par une séance pénible. Un joueur surclassé connaîtra une demi-heure de défaite sévère, tandis qu'un joueur d'un niveau très supérieur aux autres devra s'impliquer dans une partie rébarbative sans même en retirer le moindre bénéfice en termes de classement. Il en résulte parfois qu'un joueur quitte tout simplement le jeu. Il est important de garder à l'esprit, cependant, que les défections sont rares dans LoL car elles conduisent souvent à un bannissement du jeu.

Pour les besoins de notre étude, nous avons d'abord calculé un coefficient associé à la fréquence des défections de jeu ; une valeur de 1 correspond au plus grand nombre de défections rencontrées pour une distance de mise en relation donnée. La figure 3 montre la corrélation entre ces résultats et le temps d'attente moyen. Les deux courbes sont inversement proportionnelles, ce qui signifie qu'une mise en relation rapide entraîne une différence significative de classement et conduit donc souvent à une défection. Les deux pics de défection observés avant 300 et 600 correspondent aux distances de classement entre les différentes *ligues* de LoL. En d'autres termes, les joueurs devant faire face à des adversaires dans une ligue différente de la leur sont plus susceptibles de faire défection. C'est un argument fort **contre** le fait d'opter pour une mise

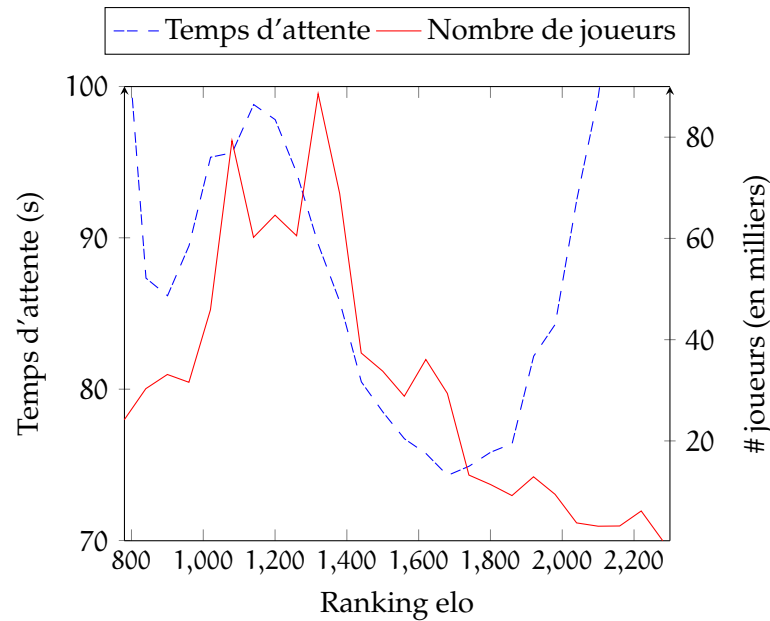


FIGURE 2 – Distribution des rangs des joueurs ainsi que leur temps d'attente moyens respectifs.

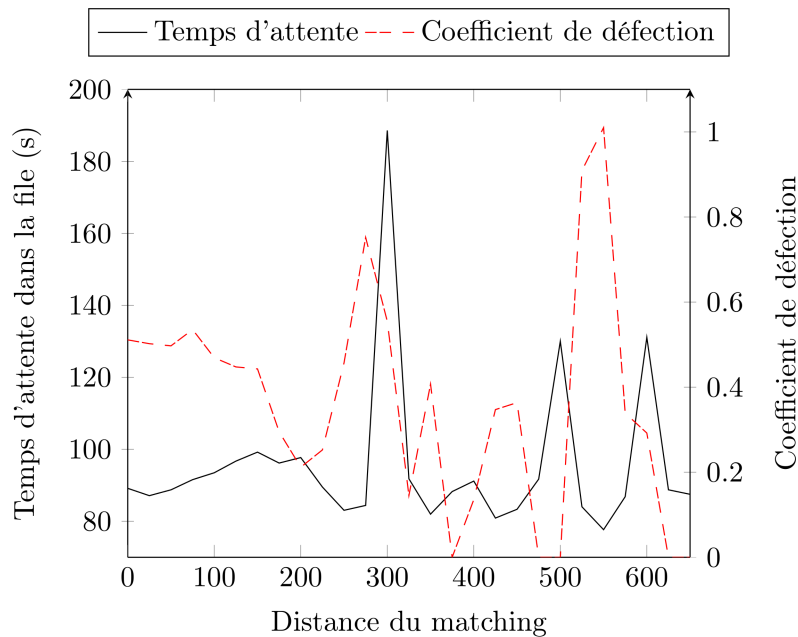


FIGURE 3 – Distance du matching et fréquence de défection.

en relation moins précise dans le but d'accélérer le processus.

4.3. Impact de la latence sur l'expérience de jeu

Même si Sheldon et al. [18] montrent que la latence a peu d'importance dans les jeux de stratégie en temps réel tels que Warcraft 3, nous avons voulu vérifier si cela s'avère également vrai pour les MOBAs. Nous avons donc évalué l'impact de la latence sur le gameplay de LoL.

Pour commencer, nous avons étudié la distribution des temps de latence au sein de la population de joueurs ; la figure 4 présente nos résultats. Le graphique montre que le ping reste entre 30ms et 50ms pour une grande majorité des utilisateurs. Si cette valeur moyenne est plus qu'acceptable, le nombre de parties pour lesquelles nous observons des pings supérieurs à 100ms ne l'est pas. Nous en avons trouvé plus de 1,2 millions, ce qui représente 7% du nombre total des parties étudiées.

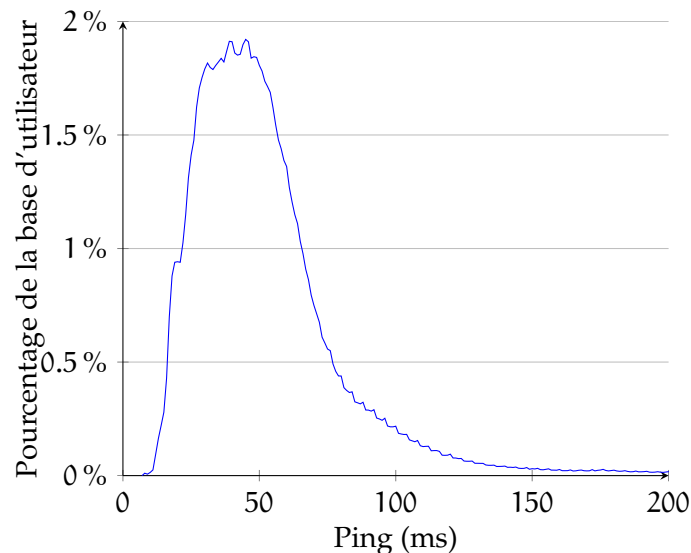


FIGURE 4 – Distribution des pings des utilisateurs de League of Legends.

Pour évaluer l'effet de la latence sur le gameplay, nous avons voulu utiliser une métrique plus précise que le reclassement (évolution du classement) des joueurs au terme d'une partie au cours de laquelle on observe des lags. Ce reclassement est un résultat final qui, s'il est important vis-à-vis de l'appréciation générale du jeu, ne rend pas vraiment compte de l'expérience de chaque joueur au cours d'une partie. En outre, l'issue d'une partie impacte toute une équipe tandis que les lags affectent des joueurs isolés. Par conséquent, nous avons choisi une métrique associée à la performance du joueur en cours de partie : le *kill death assist ratio* ou *KDA*. Il est possible de mourir, de tuer ou d'aider à tuer les avatars ennemis plusieurs fois dans une session de LoL. Par conséquent, le ratio entre tuer/aider et mourir donne un bon aperçu de la performance d'un joueur au cours d'une partie. La formule utilisée pour calculer le KDA est la suivante :

$$kda = \frac{assists+kills}{\max(deaths,1)}$$

Un KDA inférieur à 1 reflète une mauvaise performance d'un joueur.

La courbe de la figure 5 illustre la relation entre le KDA et le ping. Le KDA diminuant de façon constante alors que la durée du ping augmente, jusqu'à diviser le KDA maximum par un facteur de 8, tend à prouver que la latence gêne fortement la jouabilité.

Un autre résultat vient souligner l'impact négatif de la latence : nous avons constaté que parmi les abandons de partie, 16% viennent de joueurs qui subissent une latence supérieure à 100 ms. Etant donné que les lags de plus de 100 ms affectent 7% de l'ensemble des joueurs, il est légitime de penser qu'une latence élevée contribue à faire quitter prématurément une partie.

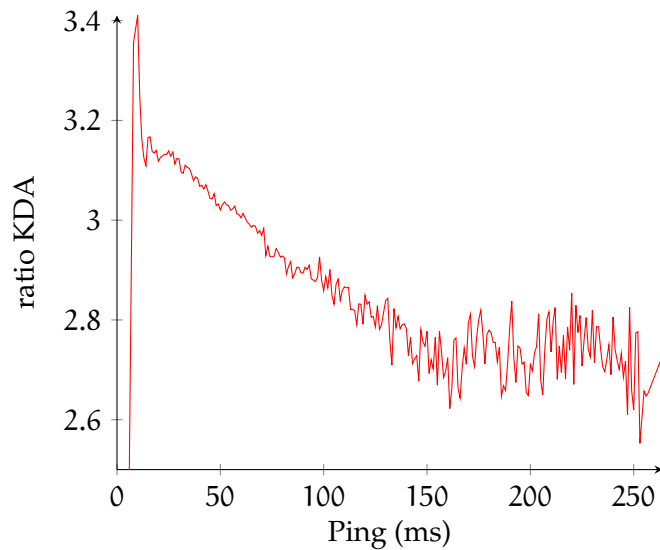


FIGURE 5 – Impact de la latence sur la performance des joueurs.

Ces deux résultats sont proches de ceux de [9], qui étudie les effets de la latence dans les FPS. Nous concluons donc, qu'à l'instar des FPS, la latence a un fort impact sur la qualité de l'expérience de jeu dans les MOBAs.

Une conséquence de cet impact avéré est qu'il est important d'inclure les valeurs de ping comme critère de matchmaking afin d'améliorer l'expérience de jeu. Toutefois, l'inclusion d'un critère supplémentaire augmentera forcément le temps de réponse, et le service de matchmaking actuel de LoL ne répond déjà pas à temps avec le *rank* pour unique critère.

Notre conclusion générale, basée sur les traces obtenues et sur les évaluations décrites dans cette section, est que le service de matchmaking gagnerait grandement à être amélioré. Les améliorations les plus évidentes étant : (i) la réduction des temps d'attente, (ii) la prise en compte plus stricte des classements et (iii) l'intégration de critères supplémentaires lors de la mise en relation afin de mieux satisfaire les joueurs.

5. Travaux connexes

La littérature sur les jeux distribués est abondante. A notre connaissance, toutefois, le présent article est le premier à rassembler des données d'utilisateurs réels en termes de matchmaking dans le but de les analyser.

La plupart des publications sur des traces de jeux distribués concernent soit des travaux sur la méthode de récupération des traces [12, 21], soit des travaux sur des aspects très spécifiques des jeux comme les positions ou les mouvements des avatar au fil du temps [19, 8, 13].

Certaines études proposent des approches pour améliorer le système de matchmaking. Chen et al. [9] montrent que la latence a un fort impact sur la durée des sessions de jeu. Les résultats de cette étude sont en corrélation avec ceux de l'analyse de notre base de données dans laquelle nous observons un nombre croissant de joueurs se déconnectant en cas de forte latence. En conséquence, de nombreux travaux proposent de prendre la latence en compte dans les services de matchmaking. Agarwal et Lorch proposent une solution de matchmaking utilisant la latence [7]; Zander et al. [20] suggèrent également d'améliorer le matchmaking en ajoutant

plus de propriétés (par exemple la latence notamment). Dans [10], les auteurs proposent un système de matchmaking amélioré, encore une fois en ajoutant de plus amples informations lors de la sélection du match.

Pourtant, notre étude montre que les systèmes de matchmaking actuels dans des jeux populaires comme League of Legends ne suffisent pas à répondre à temps aux joueurs avec peu de critères. Il nous semble important de concevoir un système de matchmaking multi-critères efficace.

D'autres publications, comme [17, 16, 15], se concentrent sur la conception de systèmes de matchmaking distribués. Ils manquent cependant de données concrètes venant d'un vrai jeu / d'une vraie application pour valider leurs approches. Ces solutions pourraient prendre avantage de notre base de données disponible publiquement.

6. Conclusion

L'étude et l'analyse des traces utilisateurs sont indispensables pour améliorer tout service logiciel, notamment ceux qui servent de support aux jeux vidéos en ligne. Cependant, les compagnies créatrices de jeu ne veulent pas rendre ces données publiques. Dans cet article, nous montrons comment recueillir une grande quantité d'information disponible publiquement sur les serveurs d'un jeu en ligne très populaire : *League of Legends*. Nous analysons également cette information pour tirer des conclusions sur la façon d'améliorer un service essentiel pour les jeux multi-joueurs en ligne, le matchmaking.

En perspective du travail présenté dans cet article, nous travaillons actuellement sur un système de matchmaking entièrement distribué, basé sur le paradigme pair à pair. La base de données que nous avons acquise constitue un outil précieux pour concevoir et tester notre système avec des données d'application réelles. Nous visons à offrir une meilleure précision, mais aussi de meilleurs temps de réponse que les solutions de matchmaking existantes. Notre approche vise à proposer un matchmaking plus efficace, tout en permettant d'élargir les critères de mise en relation.

Nous pensons que notre base de données peut aider à concevoir des solutions logicielles pour les jeux du futur. Un spectre considérable d'études pourrait utiliser cette mine d'information, et nous pensons que la collecte automatique de ce type d'information sur les jeux récents pourrait grandement améliorer l'avenir des solutions de jeux.

Bibliographie

1. League of legends.
2. League of legends community - general discussion - matchmaking changes [1-16-2013].
3. League of legends community forums - average game length.
4. League of legends game traces database.
5. lolrtmpscient - a lightweight rtmp client in java for communicating with league of legends.
6. MatchMaking in league of legends.
7. Agarwal (S.) et Lorch (J. R.). – Matchmaking for online games and other latency-sensitive p2p systems. In : *Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication*. pp. 315–326. – New York, NY, USA, 2009.
8. Armitage (G.). – An experimental estimation of latency sensitivity in multiplayer quake 3. In : *Networks, 2003. ICON2003. The 11th IEEE International Conference on*, pp. 137–141.
9. Chen (K.-T.), Huang (P.) et Lei (C.-L.). – How sensitive are online gamers to network quality ? *Commun. ACM*, vol. 49, n11, novembre 2006, pp. 34–38.

10. Delalleau (O.), Contal (E.), Thibodeau-Laufer (E.), Ferrari (R.), Bengio (Y.) et Zhang (F.). – Beyond skill rating : Advanced matchmaking in ghost recon online. *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 4, n3, 2012, pp. 167–177.
11. Elo (A. E.). – *The rating of chessplayers, past and present*. – New York, Arco Pub., 1978.
12. Guo (Y.) et Iosup (A.). – The game trace archive. In : *Proceedings of the 11th Annual Workshop on Network and Systems Support for Games*. pp. 4 :1–4 :6. – Piscataway, NJ, USA, 2012.
13. Kinicki (J.) et Claypool (M.). – Traffic analysis of avatars in second life. In : *Proceedings of the 18th International Workshop on Network and Operating Systems Support for Digital Audio and Video*. pp. 69–74. – New York, NY, USA, 2008.
14. MacManus (C.). – League of legends the "world's most played video game". – CNET News, October 2012.
15. Manweiler (J.), Agarwal (S.), Zhang (M.), Roy Choudhury (R.) et Bahl (P.). – Switchboard : A matchmaking system for multiplayer mobile games. In : *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*. pp. 71–84. – New York, NY, USA, 2011.
16. Ogston (E.) et Vassiliadis (S.). – Local distributed agent matchmaking. In : *In Proc. of the 9th International Conference on Cooperative Information Systems*, pp. 67–79.
17. Shafran (V.), Kaminka (G.), Kraus (S.) et Goldman (C. V.). – Towards bidirectional distributed matchmaking. In : *Proceedings of the 7th int. joint conference on Autonomous agents and multiagent systems*, pp. 1437–1440.
18. Sheldon (N.), Girard (E.), Borg (S.), Claypool (M.) et Agu (E.). – The effect of latency on user performance in warcraft iii. In : *Proceedings of the 2nd workshop on Network and system support for games*. pp. 3–14. – New York, NY, USA, 2003.
19. Tan (S. A.), Lau (W.) et Loh (A.). – Networked game mobility model for first-person-shooter games. In : *Proceedings of 4th ACM SIGCOMM workshop on Network and system support for games*. pp. 1–9. – New York, NY, USA, 2005.
20. Zander (S.), Leeder (I.) et Armitage (G.). – Achieving fairness in multiplayer network games through automated latency balancing. In : *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*. pp. 117–124. – New York, NY, USA, 2005.
21. Zhang (B.), Iosup (A.), Garbacki (P.) et Pouwelse (J.). – A unified format for traces of peer-to-peer systems. In : *Proceedings of the 1st ACM workshop on Large-Scale system and application performance*. pp. 27–34. – New York, NY, USA, 2009.