

# Rapport ARA : 1ère étude expérimentale

Titouan Polit, Mazigh Saoudi

3 janvier 2023

---

*Ce projet a pour but de réaliser une étude expérimentale à travers le simulateur  
Peer-Sim.*

---

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Description de ce que l'on teste . . . . .	3
1.2	Description de la plate-forme d'expérimentation . . . . .	3
<b>2</b>	<b>Paramètres de l'expérimentation</b>	<b>3</b>
<b>3</b>	<b>Métriques considérées</b>	<b>4</b>
3.1	Paramètres liés aux métriques . . . . .	5
<b>4</b>	<b>Mesure des métriques</b>	<b>6</b>
<b>5</b>	<b>Représentativité des résultats</b>	<b>6</b>
<b>6</b>	<b>Résultats</b>	<b>7</b>
6.1	Phase 1 : cas où la valeur de $\gamma$ est négligeable par rapport à celle de $\alpha$ . .	7
6.1.1	Estimations des résultats . . . . .	7
6.2	Phase 2 : cas où la valeur de $\gamma$ est comparable à celle de $\alpha$ . . . . .	8
6.2.1	Estimation des résultats . . . . .	8
6.3	Phase 3 : cas où la valeur de $\gamma$ est nettement supérieure à celle de $\alpha$ . . .	8
6.3.1	Estimation des résultats . . . . .	8
6.4	Analyse . . . . .	9
<b>7</b>	<b>Charge de la section critique</b>	<b>14</b>
<b>8</b>	<b>Conclusion</b>	<b>15</b>

<b>9 Références</b>	<b>15</b>
---------------------	-----------

# 1 Introduction

## 1.1 Description de ce que l'on teste

Dans ce rapport, nous allons présenter les résultats d'une étude expérimentale. Cette étude porte sur l'exécution, dans un environnement fiable, d'un protocole de verrouillage distribué. L'application sera constituée de plusieurs noeuds qui souhaitent accéder à une ressource partagée afin de pouvoir incrémenter un compteur. Cette application aura besoin d'un algorithme d'exclusion mutuelle distribué. Pour se faire, nous nous baserons sur l'implantation d'une variante de l'algorithme de Naimi Tréhel. Cet algorithme est basé sur la circulation d'un jeton. Seul le noeud qui possède le jeton peut entrer en section critique. De plus, l'unique jeton, contiendra des données qui ne sont accessibles que de manière exclusive, ici, il contiendra la valeur du compteur partagé.

## 1.2 Description de la plate-forme d'expérimentation

Pour réaliser notre étude expérimentale, nous allons utiliser PeerSim. PeerSim est un simulateur à événements discrets. Il est écrit en Java et fournit une api pour créer des simulations. L'architecture de ce simulateur est divisée en deux parties. La première regroupe le code décrivant et implantant le système à étudier et la seconde concentre le code permettant de contrôler la simulation. Pour lancer une simulation dans PeerSim, il faut rédiger un fichier de configuration, ce fichier de configuration contiendra les valeurs que l'on souhaite associer aux différents paramètres de l'expérimentation.

# 2 Paramètres de l'expérimentation

Afin de pouvoir quantifier la charge sur la section critique grâce aux métriques considérées, nous proposons d'utiliser 4 paramètres. Cependant, nous ne ferons varier que trois de ces paramètres. Le premier paramètre, noté  $N$  représente le nombre de noeuds, nous fixerons cette valeur à 10. Le second paramètre, noté  $\alpha$  est le temps moyen passé en section critique. Les trois valeurs prises par ce paramètre seront 400ms, 350ms et 10ms, nous détaillerons ce choix par la suite. Le troisième paramètre, noté  $\gamma$  représente le temps moyen pour transmettre un message entre deux noeuds. Les valeurs prises par ce paramètre seront 1ms, 350ms et 400ms, comme pour  $\alpha$  nous détaillerons ce choix par la suite. Il faut noter que, la vitesse de circulation du jeton dans l'application est directement reliée à  $\alpha$  et  $\gamma$ . Si le temps de transmission d'un message est très long, alors le jeton mettra du temps à passer d'un noeud à un autre, faisant donc diminuer le temps que passent les noeuds en section critique. En effet, si le jeton est en transit, aucun noeud n'est en section critique. De l'autre côté, si les noeuds restent très longtemps en section

critique, alors le jeton circulera peu. On aura donc moins d'accès à la section critique. On peut donc dégager trois cas de figure pour quantifier l'impact de ces deux paramètres sur le système :

- Le temps de transmission d'un message est négligeable par rapport au temps que passent les noeuds en section critique.
- Le temps de transmission d'un message est comparable au temps que passent les noeuds en section critique.
- Le temps de transmission d'un message est nettement supérieur au temps que passe les noeuds en section critique.

Le dernier paramètre, qui est celui que l'on fera varier tout au long de l'étude est  $\beta$ . Il représente le temps moyen qu'un noeud attend avant de redemander l'accès à la section critique. Nous ferons varier ce paramètre de **0ms** à **700ms** en augmentant sa valeur de **50ms** pour chaque test.

Pour terminer, dans PeerSim, nous devons choisir une durée de simulation et une graine pour la classe *Random* de java. Nous avons choisi de réaliser des simulations de **20 000ms**. Chaque simulation sera reproduite quatre fois pour chaque cas, en variant le **random.seed** avec les valeurs suivantes : 20, 784, 365, 874. Les valeurs que nous présenterons seront donc des moyennes basées sur les quatre simulation.

### 3 Métriques considérées

Dans notre étude, nous allons considérer 4 métriques pour mesurer les performances de l'algorithme de Naimi-Tréhel. Nous identifierons dans un premier temps le nombre de messages applicatifs nécessaires à l'entrée en section critique d'un noeud. Dans ces messages applicatifs, nous retrouverons :

- les messages jetons : message envoyé par un noeud pour transmettre le jeton à un autre noeud
- les messages request : message envoyé par un noeud pour manifester son envie d'accéder à la section critique

Nous mesurerons le temps nécessaire moyen pour entrer en section critique, c'est à dire le temps que passe un noeud à l'état *requesting*. Cela nous permettra d'évaluer la vivacité de notre algorithme.

Nous nous intéresserons ensuite aux états dans lequel se trouve le jeton. Notamment aux pourcentages suivants (calculés par rapport au temps d'exécution total) :

- temps durant lequel le jeton est détenu par un noeud en section critique, noté **U**
- temps durant lequel le jeton est en transit entre deux noeuds, noté **T**
- temps durant lequel le jeton est détenu par un noeud n'étant pas en section critique, noté **N**

Nous avons identifié que le temps que passe le jeton dans ces trois états est directement relié à un ou plusieurs paramètres. Nous expliquons dans la section 3.1, la réflexion que nous avons utilisé pour estimer ensuite nos résultats.

Enfin, après avoir analyser les résultats des cas d'utilisation présentés précédemment, nous nous intéresserons à la valeur de  $\rho$ .  $\rho$  est une métrique permettant de calculer la charge sur notre section critique à partir de  $\alpha$ ,  $\beta$  et  $\gamma$ . La formule est présentée ci-dessous :

$$\rho = \frac{\alpha + \gamma}{\beta}$$

Plus la valeur de  $\rho$  sera élevée plus cela signifiera que la taille de la file d'attente pour entrer en section critique sera grande.

### 3.1 Paramètres liés aux métriques

Pour  $\beta$  : plus sa valeur se rapproche de 0 plus il y a de chances qu'un noeud sortant de section critique, y ré-entre sans perdre le jeton. Cette probabilité augmente d'autant plus lorsque le temps de transmission d'un message augmente et le temps d'exécution de la section critique diminue. En effet, le temps que la requête d'un autre noeud arrive, le noeud avec le jeton aura eu la possibilité de rentrer en section critique une nouvelle fois. On appellera ce phénomène **le dépassement**.

Pour T : si  $\gamma$  est faible, par rapport à  $\alpha$  et  $\beta$  qui sont assez grand pour limiter le nombre de fois le jeton est envoyé, de plus son temps de transit étant faible le transfert sera rapide. On aura donc peu d'envois de jetons et des envois rapides. Cela entraînera un faible pourcentage de temps passé dans l'état transit. Au contraire, si  $\gamma$  est important par rapport à  $\alpha$ ,  $\beta$  alors là les envois de jetons seront plus longs que la section critique elle-même, cela entraînera une augmentation du temps passé dans l'état en transit proportionnelle à la diminution du temps passé dans les autres états.

Pour U : si  $\alpha$  est faible, par rapport à  $\beta$  et  $\gamma$ , alors peu de noeuds seront dans l'état *requesting* car ils seront en attente pour faire leur demande d'entrée depuis leur état *tranquil*. Or, la section critique étant rapide, il y aura beaucoup de possibilité d'envois de jetons, cependant si personne ne demande la section critique alors le jeton restera en possession d'un noeud qui n'est pas en section critique. Ainsi, le temps passé dans l'état utilisé sera plutôt faible. Au contraire, si l'on considère  $\alpha$  plus important que  $\beta$  et  $\gamma$  alors il y aura beaucoup de processus en attente de la section critique et une section critique longue à exécuter. Le jeton sera donc très souvent détenu par un processus en section critique, faisant augmenter U et diminuer T et N.

Pour N : le jeton est dans l'état non utilisé si un noeud qui a terminé sa section critique ne trouve pas de noeud ayant demandé la section critique. Cela peut se produire lorsque la valeur de  $\beta$  est très élevé par rapport à la valeur de  $\alpha$ . En effet si la section

critique est très courte par rapport au temps d'attente avant de redemander la section critique alors la file d'attente sera souvent vide, entraînant une augmentation du taux de non-utilisation de la section critique.

## 4 Mesure des métriques

Pour réaliser nos mesures, nous avons utilisé un module de contrôle dans notre fichier de configuration PeerSim. Ce module de contrôle nous permet d'obtenir pour chaque noeud, à la fin d'une exécution, le nombre de messages request reçus (noté  $nb\_requests$ ), le nombre de messages jetons reçus (noté  $nb\_jetons$ ) et le nombre d'entrées en section critique (noté  $nb\_cs$ ). Pour avoir une mesure précise du temps d'attente (l'état *requesting*) nous avons utilisé une liste  $beta\_starting$  pour stoker les dates d'entrée dans l'état *tranquille*.

Nous utilisons ensuite des formules simples pour calculer les métriques qui nous intéressent.

- nombre de messages applicatifs :

$$nb\_mess = \sum_{i=0}^N (nb\_requests_i + nb\_jetons_i)$$

- nombre de messages applicatifs moyen par section critique :

$$msgPerCs = \frac{nb\_mess}{\sum_{i=0}^N nb\_cs_i}$$

- nombre de messages request par noeud :

$$reqCount = \frac{\sum_{i=0}^N nb\_requests}{\sum_{i=0}^N nb\_cs}$$

- temps passé en section critique par noeud :

$$time\_cs = nb\_cs * \alpha$$

- temps passé à ne rien faire par noeud (l'état tranquille) :

$$time\_idle = (taille(beta\_starting) - 1 + (End - betaF > \beta) ? End - betaF : 1) * \beta$$

où  $betaF$  est le dernier élément de la liste et  $End$  le temps total écoulé.

- temps nécessaire moyen pour obtenir la section critique pour un processus :

$$time\_getCS = Tempstotald'excution - time\_cs - time\_idle$$

- temps moyen nécessaire pour obtenir la section critique :

$$waiting\_time\_avg = time\_getCS / N$$

## 5 Représentativité des résultats

Dans cette première expérimentation, nous nous concentrons sur la mesure des métriques dans des conditions fiables. Aucune panne ne peut survenir dans le système. De plus, les noeuds demandent la section critique périodiquement. Ils possèdent tous la même

période, cela donne à l'application une exécution qui ne reflète pas au mieux la réalité. Cependant, nous allons changer la valeur de la période de demande de section critique dans trois scénarios différents. Les résultats que nous obtiendrons nous permettront d'avoir une vue globale de comment se comporte l'application. Ainsi en croisant ces résultats nous pourrons nous approcher d'une étude plus réaliste puisque le choix des trois scénarios et des métriques est très bien adapté à l'étude que l'on fait de notre système : quantifier l'impact de la charge sur une application de verrouillage distribué.

## 6 Résultats

### 6.1 Phase 1 : cas où la valeur de $\gamma$ est négligeable par rapport à celle de $\alpha$

Ce cas représente un scénario où le temps de transmission d'un message est très faible, le jeton passera donc très peu de temps en transit, il sera donc très majoritairement possédé par un noeud. D'un autre côté, les noeuds garderont le jeton longtemps. Nous avons choisi les valeurs suivantes pour ce cas :  $\gamma = 1\text{ms}$  et  $\alpha = 400\text{ms}$ .

#### 6.1.1 Estimations des résultats

Étant donné que les noeuds restent longtemps en possession du jeton, on observera une baisse du nombre de messages request par noeud en comparaison avec le cas où la valeur d' $\alpha$  est fortement diminuée. Majoritairement dû au fait qu'il y aura beaucoup moins de messages request intermédiaire. En effet, le transport étant quasi instantané et le temps à l'état tranquille court, les noeuds auront tendance à envoyer leurs requêtes au possesseur du jeton, autrement dit les noeuds sont quasiment tous le temps à jour sur la racine de l'arbre des demandeurs.

Nous utiliserons ici les réflexions présentées dans la section 3.1. Le jeton passera la majorité de son temps dans l'état 'utilisé' plutôt que dans l'état 'en transit'. En effet  $\gamma$  étant très faible, le jeton passera peu de temps en transit.

Quant à l'effet de  $\beta$ , avec des valeurs faibles, les noeuds passeront plus de temps dans l'état *requesting* que dans l'état tranquille. Cela induit donc que la file d'attente pour accéder à la section critique aura peu de chances d'être vide. Ainsi le jeton passera de noeud en noeud sans perdre de temps dans l'état non utilisé. Plus la valeur de  $\beta$  augmentera, plus le temps que passent les noeuds dans l'état tranquille augmentera, faisant diminuer la congestion à l'entrée de la section critique et ainsi augmenter le temps que passe le jeton dans l'état non utilisé, autrement dit les chances qu'un noeud n'utilisant pas son jeton et qu'aucun autre ne le demande augmentent.

## 6.2 Phase 2 : cas où la valeur de $\gamma$ est comparable à celle de $\alpha$

Ce cas représente un scénario où le temps de transmission d'un message est quasiment équivalent au temps qu'un noeud passe dans sa section critique. Le jeton passera donc autant de temps en transit qu'en utilisation. Pour ce scénario nous avons choisi les valeurs suivantes :  $\gamma = 350\text{ms}$  et  $\alpha = 350\text{ms}$

### 6.2.1 Estimation des résultats

Avec un temps de transit comparable au temps d'exécution de la section critique, les noeuds passeront plus de temps à l'état *requesting* et généreront moins de messages request que dans le premier cas présenté, en section 6.1. L'augmentation de la valeur de  $\beta$  aura pour effet de diminuer la charge au niveau de la file d'attente d'entrée en section critique. En effet, les noeuds devront attendre plus longtemps avant de refaire une demande d'entrée.

Pour l'état du jeton, nous pensons obtenir des pourcentages proches entre le temps que passe le jeton dans l'état *utilisé* et dans l'état *en transit*. L'état non utilisé dépendra principalement de  $\beta$ . En effet avec un  $\beta$  plus important les noeuds utiliseront moins le jeton ce qui amènera à une diminution du temps passé dans l'état *utilisé* et augmentera proportionnellement le temps passé dans l'état *non utilisé*. Le temps passé en transit ne changera que si  $\beta$  a réduit le nombre de section critique exécutées.

## 6.3 Phase 3 : cas où la valeur de $\gamma$ est nettement supérieure à celle de $\alpha$

Ce dernier cas présente un scénario où le temps de transmission d'un message est nettement plus important que le temps d'exécution de la section critique. Le jeton passera donc plus de temps en transit qu'en utilisation. Pour ce scénario nous avons choisi les valeurs suivantes :  $\gamma = 400\text{ms}$  et  $\alpha = 10\text{ms}$

### 6.3.1 Estimation des résultats

Dans ce cas d'étude, le temps d'exécution de la section critique étant très faible, les noeuds seront majoritairement dans l'état *tranquille* ou *requesting*. Les messages prenant beaucoup de temps à arriver, il sera assez probable que le noeud récepteur (le **last** chez les autres) aura eu le temps de ré-entrer la section critique si le temps  $\beta$  était court ou d'avoir retransmit le jeton vers un autre. Ce dernier engendrera plusieurs messages request intermédiaires qui étaient moins présents dans les autres cas.

Quand on a  $\beta = 0$  comme expliqué en section 3.1 on remarquera une baisse importante dans la moyenne des messages applicatifs par section critique. Vu que  $\gamma$  est bien



supérieur à  $\beta + \alpha$ , les requêtes des noeuds non possesseurs de jeton prendront plus de temps qu'il en faut au propriétaire pour renouveler sa demande. Ainsi le noeud possesseur du jeton pourra rentrer en section critique au moins une fois sans émettre de message applicatif.

Pour ce qui est de l'état du jeton, on notera une forte augmentation du temps passé dans l'état en transit et un temps d'utilisation moindre. Par contre, avec une valeur de  $\beta$  plus importante, on notera une diminution du temps d'utilisation au profit du temps où il n'est pas utilisé.

## 6.4 Analyse

Grâce aux figures qui vont suivre, nous allons pouvoir vérifier ou non, les estimations que nous avons faites.

On peut observer dans la fig 1 que dans les trois cas on obtient des résultats plutôt constant peu importe le cas d'étude. On observe cependant que dans le 3ème cas, quand  $\beta = 0$  avec  $\gamma$  très élevé et  $\alpha$  très bas, le nombre de messages par section critique est inférieur à 2. Cela est dû au phénomène de dépassement que nous avons présenté dans la section 3.1

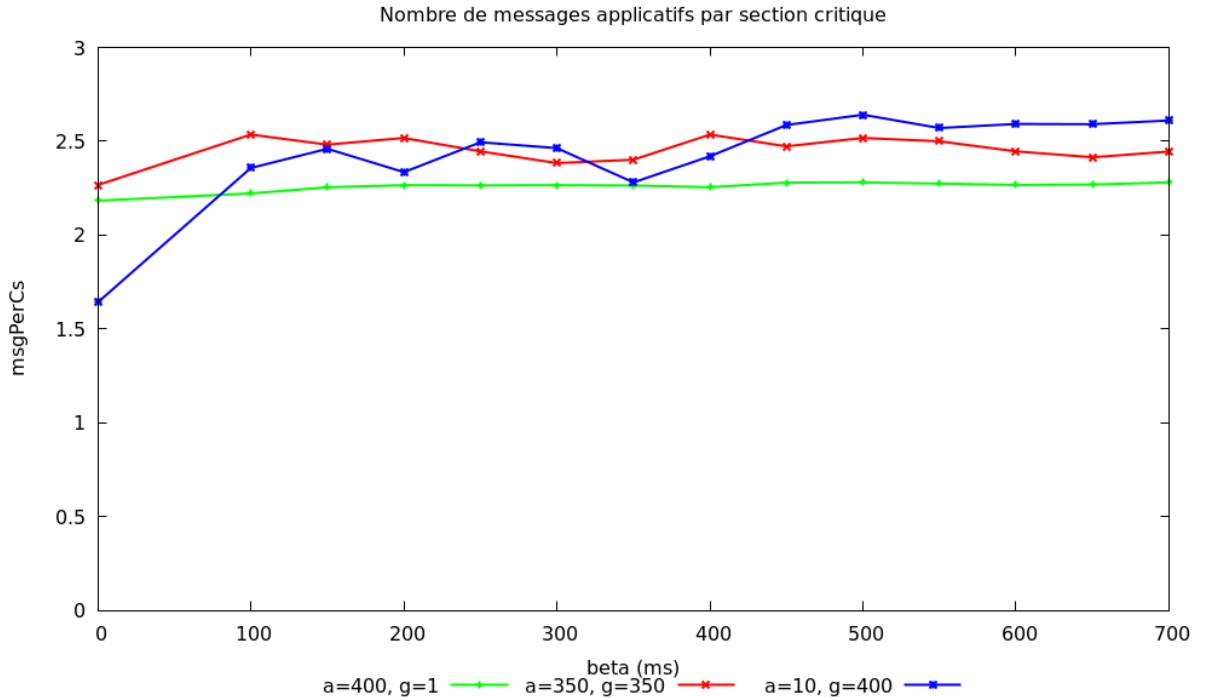


FIGURE 1 – Nombre de messages applicatifs par section critique pour les trois cas d'étude

On peut observer dans la fig 2 que dans le cas 2, on obtient moins de requêtes que dans le cas 1. Cela est dû à la durée plus importante que passent les noeuds dans l'état

*requesting* à cause des valeurs élevées de  $\alpha$  et  $\gamma$ . Par ailleurs, on constate des plus grandes valeurs dans le cas 3. Cependant, cela n'est pas dû à un nombre plus élevé de requêtes mais plutôt à un nombre plus élevé de messages request relayés. En effet, les noeuds peuvent se retrouver plus éloignés de la racine à cause de la valeur importante de  $\gamma$  par rapport à  $\alpha$ . A cause de cet écart, les changements de last peuvent avoir lieu durant l'envoi d'un message request par un noeud, induisant donc une augmentation des messages request intermédiaires.

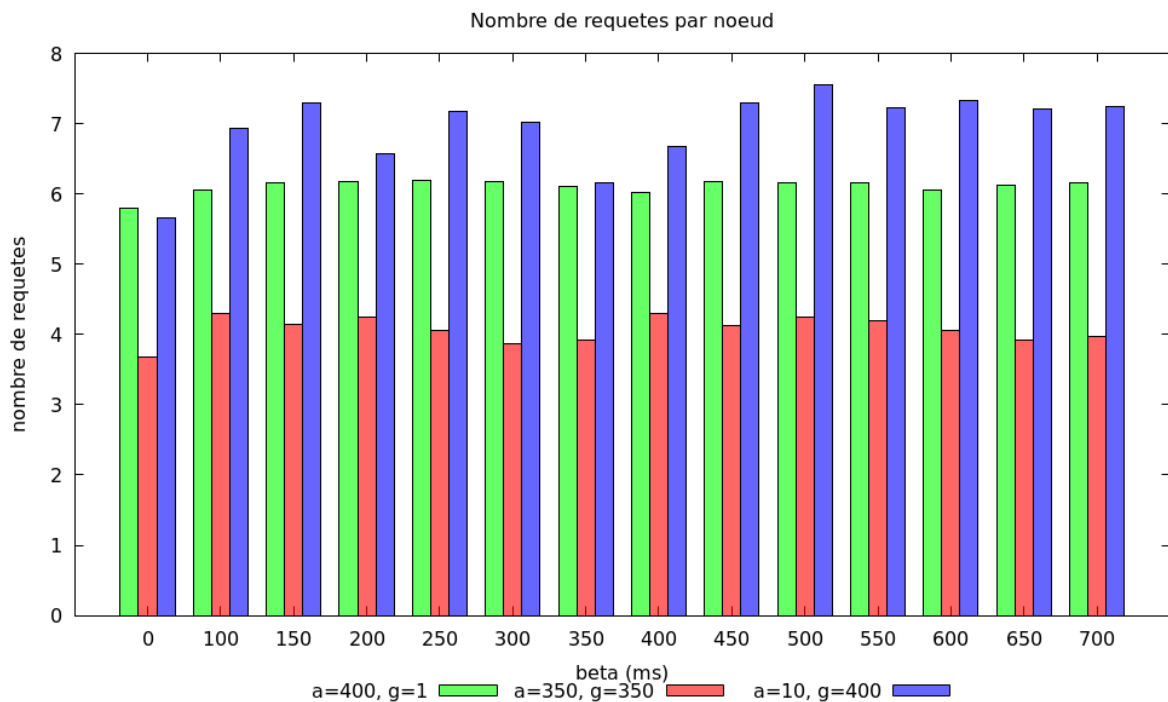


FIGURE 2 – Nombre de requêtes par noeud pour les trois cas d'études

Dans la fig 3, le temps d'attente a une diminution linéaire par rapport à l'augmentation de  $\beta$  3. On peut aussi observer que jusqu'à  $\beta = 450$ , les noeuds attendent plus dans le cas 3 que dans les deux autres cas. Cependant, cela change après puisque les valeurs dans le cas 2 deviennent plus élevées. Cela est directement relié à la valeur de  $\beta$ . En effet, celle-ci étant élevée, il y a moins de requêtes donc moins de messages envoyés. Ainsi,  $\alpha$  devient plus déterminante dans le calcul du temps d'attente.

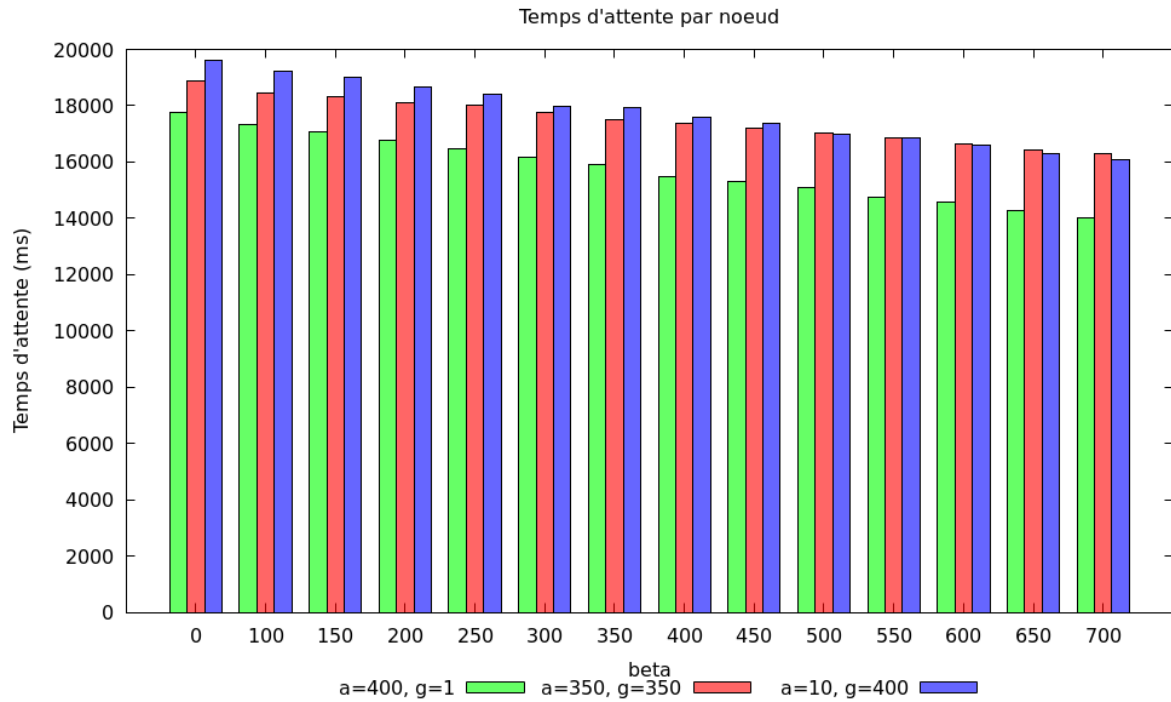
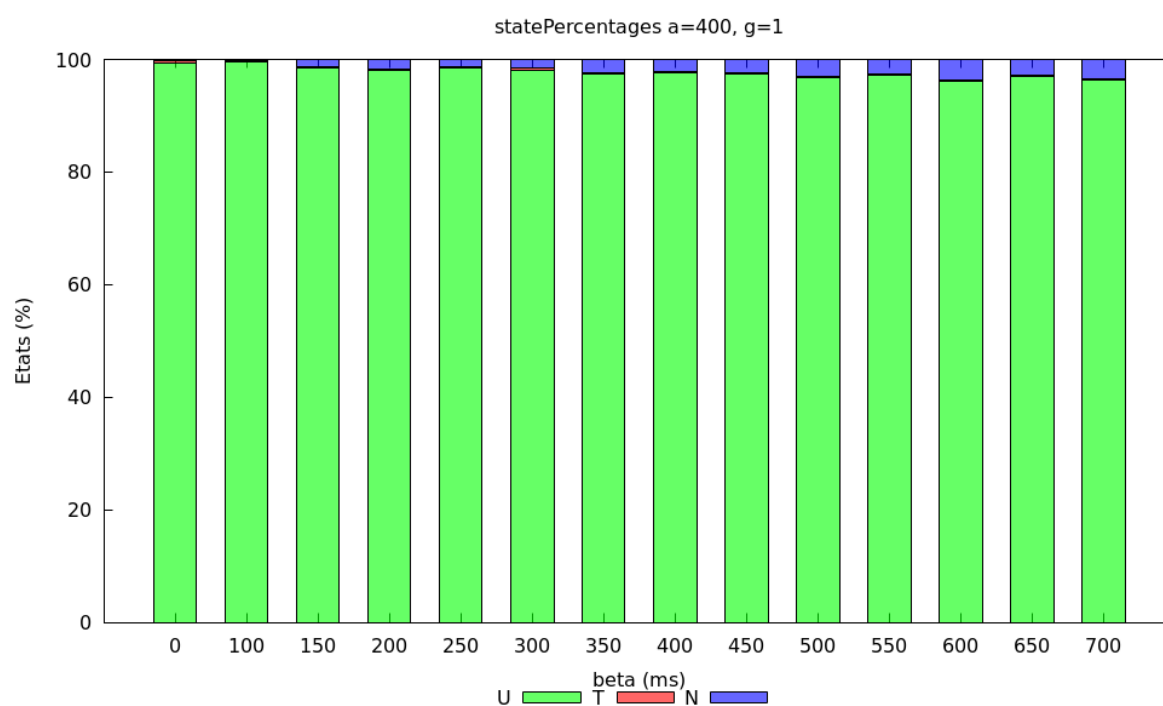
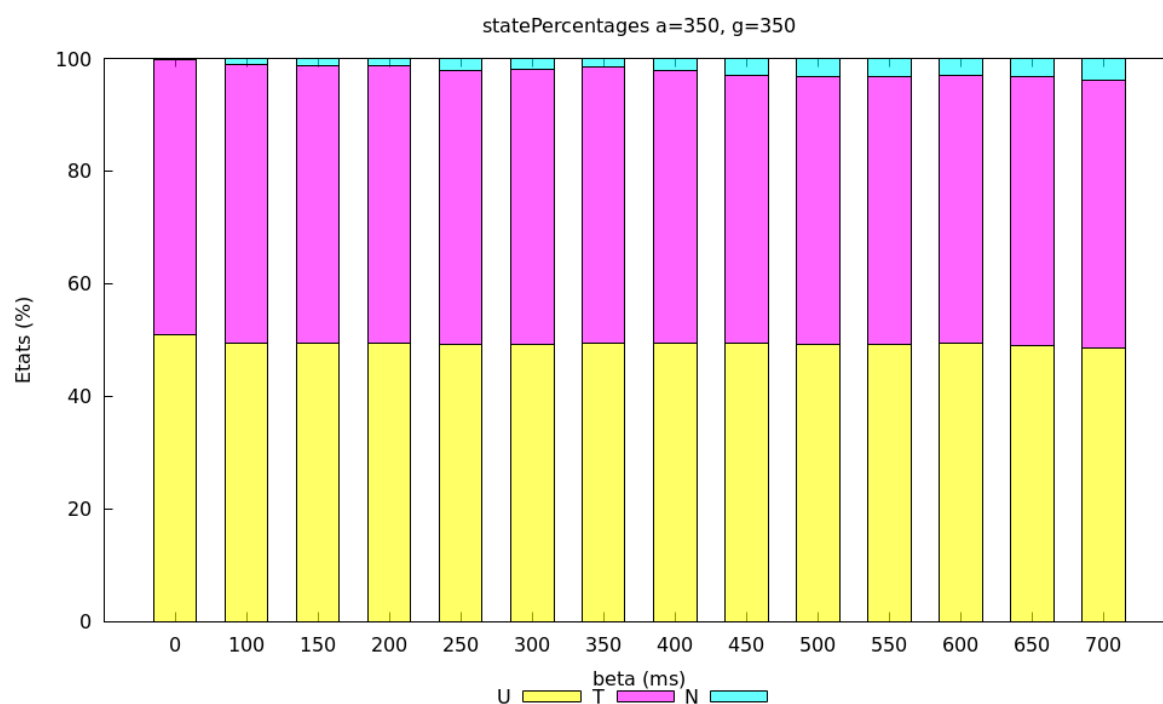


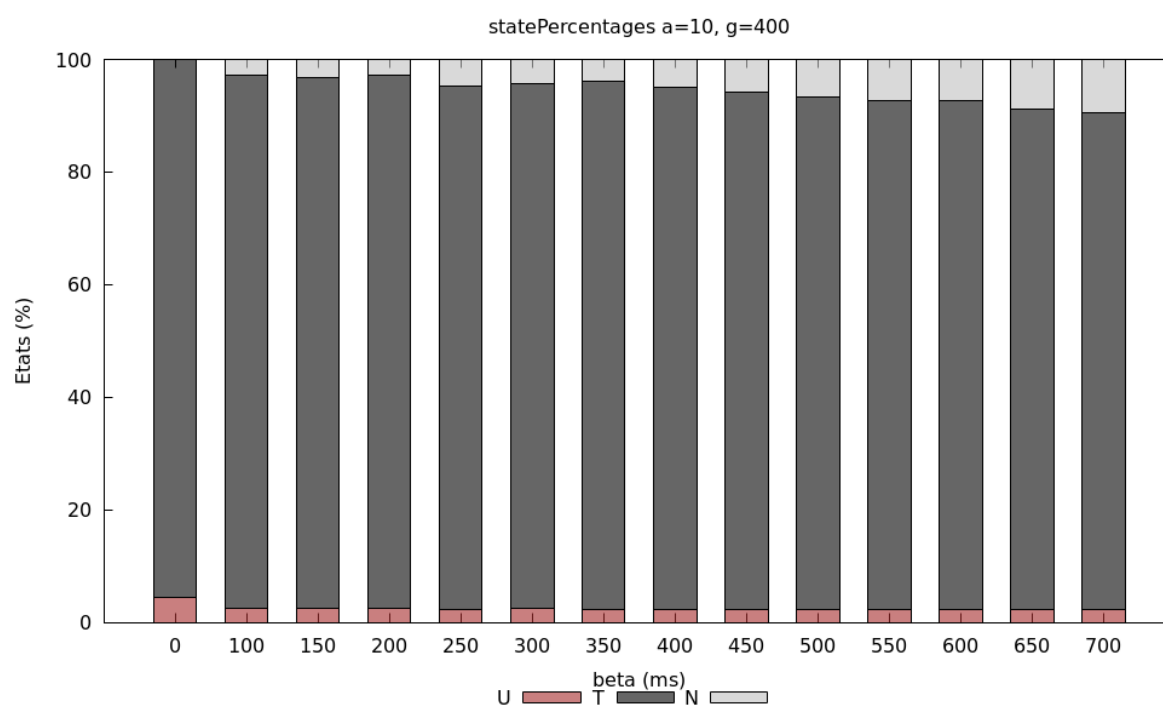
FIGURE 3 – Temps d’attente par noeud dans les trois cas d’études

La valeur de  $\alpha$  étant élevée nous avons imaginé que le jeton serait très largement utilisé. C’est en effet ce que nous pouvons observer sur la figure 4. On observe néanmoins une augmentation du temps passé dans l’état non-utilisé lorsque la valeur de  $\beta$  augmente. Cela est normal, car on avait supposé que la taille de la file d’attente serait plus petite et qu’il y aurait donc moins de chance de trouver un noeud intéressé par la section critique.

Dans le second cas, présenté dans la figure 5, deux états se dégagent. On constate que ces deux états, *utilisé* et *en transit* sont quasi égaux. C’est une nouvelle fois, le comportement que nous avons imaginé et qui est logique puisque les valeurs de  $\alpha$  et  $\gamma$  sont égales. On observe par ailleurs le même phénomène que dans le scénario 1 pour le temps passé dans l’état non utilisé qui croît lorsque  $\beta$  augmente.

Dans le troisième et dernier scénario où  $\gamma$  est nettement supérieure à  $\alpha$ , on observe sur la figure 4 que le jeton passe l’énorme majorité de son temps en transit. De nouveau, le temps que passe le jeton sans être utilisé augmente avec  $\beta$ . Compte tenu que ce phénomène persiste dans les trois scénarios, nous espérons observer que  $\rho$  diminue lorsque  $\beta$  augmente car cela signifierait que la file d’attente est plus petite lorsque les valeurs de  $\beta$  s’élèvent.

FIGURE 4 – Pourcentages des états du jeton aux cas  $\alpha=400$   $\gamma=1$ FIGURE 5 – Pourcentages des états du jeton aux cas  $\alpha=350$   $\gamma=350$

FIGURE 6 – Pourcentages des états du jeton aux cas  $\alpha=10$   $\gamma=400$

## 7 Charge de la section critique

Grâce aux trois cas présentés dans les parties précédentes, nous pouvons maintenant utiliser le ratio  $\rho$  représentant la charge sur la section critique. Nous allons donc comparer en fonction des trois scénarios présentés précédemment la valeur de  $\rho$ . Cela nous permettra de déduire dans quel cas d'utilisation notre algorithme est le plus mit en difficulté.

Tout d'abord, on, constate que peu importe le cas dans lequel on se situe, la valeur de  $\rho$  diminue lorsque  $\beta$  augmente. Cela signifie donc que la taille de la file d'attente diminue lorsque la fréquence de demande d'entrée en section critique diminue. Cela influence donc bien le temps que passe le jeton dans l'état non-utilisé, comme nous l'avions observé précédemment.

On constate aussi que peu importe le scénario, la charge sur notre section critique est la même. Celle-ci ne dépend donc que de la fréquence à laquelle les noeuds vont initier une requête d'accès, ce qui est logique.

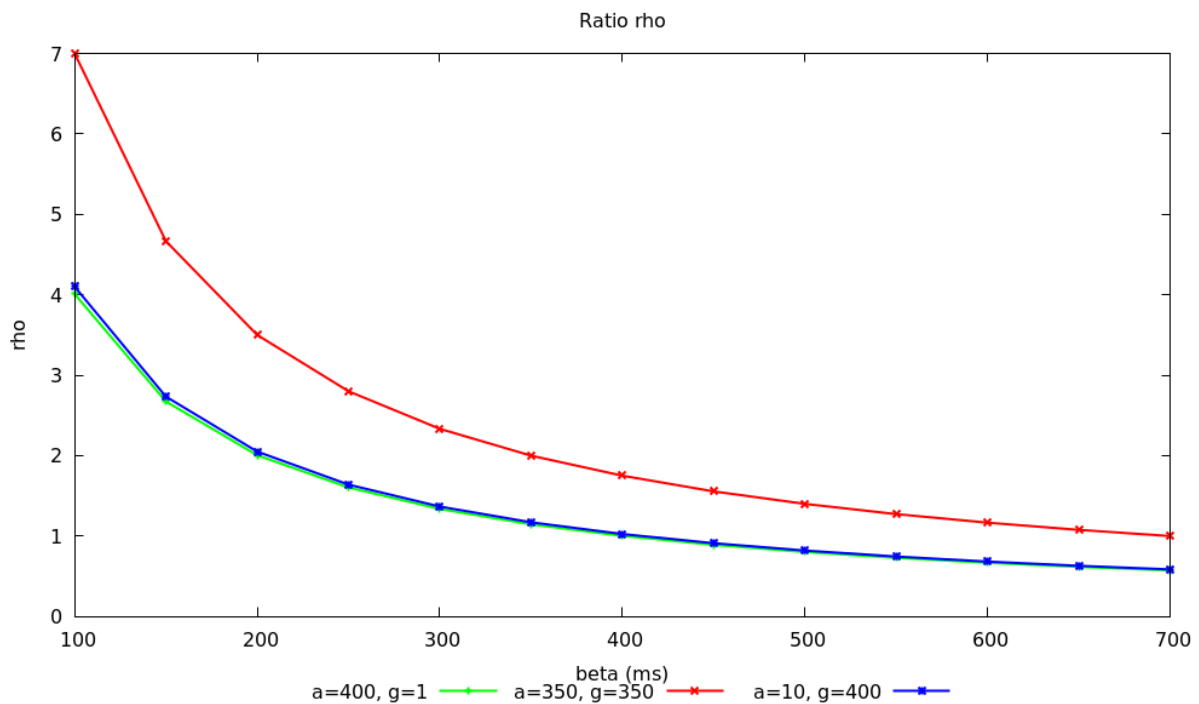


FIGURE 7 – Le ratio  $\rho$  par rapport à  $\beta$

## 8 Conclusion

Durant cette étude expérimentale, nous avons pu observer quels étaient les effets des différents paramètres sur les différentes métriques que nous avons décidé d'étudier.

Nous avons donc pu observer que  $\beta$  est un paramètre important pour gérer la charge sur notre section critique, plus sa valeur augmente et donc la fréquence de demande d'accès, plus la charge sera élevée. Ce sera donc un paramètre dont il faudrait se soucier si l'on souhaite maîtriser la charge sur une section critique dans une application utilisant cette variante de l'algorithme de Naimi-Tréhel.

Lorsque l'on s'est placé dans le scénario 1, avec une durée d'exécution de section critique bien plus élevée que la durée de transfert des messages, on a pu observer que les temps d'attente étaient moins longs que dans le troisième scénario. Nous avons donc identifié que cela était dû aux requêtes des nœuds éloignés, en effet leur requête devaient être relayées afin d'atteindre le propriétaire du jeton.

Lorsque  $\alpha$  et  $\gamma$  avaient des valeurs semblables, on a constaté que la probabilité de requête relayée était moindre. Le facteur qui fait la plus grosse différence quant au temps d'attente et au nombre de requêtes émises par rapport aux autres cas était  $(\alpha + \gamma)$ . La somme étant la plus élevée des trois cas, la charge sur la section critique est bien plus importante, ce qui cause un temps d'attente plus important et un nombre de requêtes initiées plus petit.

Nous avons aussi mis en évidence que lorsque  $\alpha + \beta \approx 0$ , il existait un phénomène de dépassement entre les nœuds. Cependant, celui-ci n'est pas dérangeant car il n'induit aucune famine dans l'algorithme.

## 9 Références

- [1] M. Naimi, M. Trehel, and A. Arnold. A log (N) distributed mutual exclusion algorithm based on path reversal. *Journal of Parallel and Distributed Computing*, 34(1) :1–13, 10 April 1996.
- [2] Mohamed Naimi and Michel Trehel. How to detect a failure and regenerate the jeton in the log(n) distributed algorithm for mutual exclusion. *Lecture Notes In Computer Science LNCS*, 312 :155–166, 1987.