

Compte rendu du projet de simulation

Raphaël Duprat 21305317, Quentin Buot 21302101

May 25, 2020

1 Introduction

Ce projet vise à simuler une architecture de réseau en anneau. Des packets sont envoyés depuis des stations situées sur l'anneau, en font le tour complet, puis sont supprimés une fois revenus à la station émettrice. L'intérêt de la simulation est de permettre de concevoir efficacement une telle structure.

2 Implémentation

Pour l'implémentation, nous avons choisit le langage C car cela nous semblait la méthode la plus directe de travailler sur le simulateur, n'ayant pas absolument besoin de concepts objets comme offrent le langage C++. De plus, pour une simulation, la performance est un souci primordial et en cas de problème il aurait été plus facile d'améliorer notre programme sans de lourds et coûteux concepts OO non obligatoires pour le bon fonctionnement du simulateur.

Le programme est découpé en 3 fichiers principaux :

- `evenements.c` : contient les divers evenements applicables aux packets (création, insertion, suppression)
- `expo.c` : gère l'aléatoire, c'est à dire la lecture et création de la fonction de répartition, ainsi que la génération des valeurs.
- `simulateur.c` : contient le main du programme, lance les initialisations et la boucle de simulation. La fonction principale de ce fichier est "simulateur", qui initialise les structures utilisées et récupère les données permettant de dessiner nos courbes.

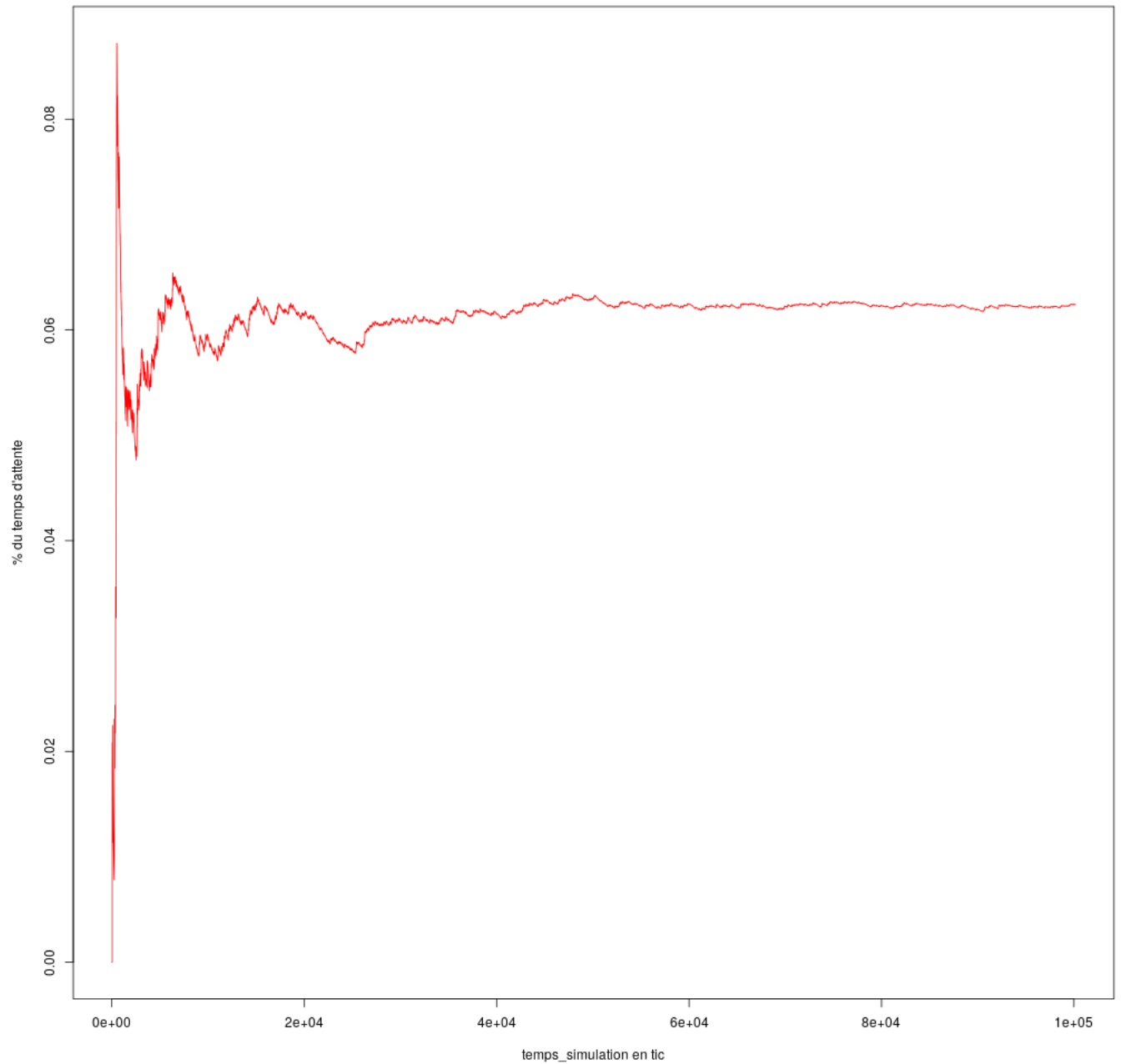
Une boucle de simulation fonctionne de la manière suivante :

- 1) Chaque station reçoit une date d'arrivée aléatoire du prochain packet, si elle n'en a pas.
- 2) Pour chaque station, un packet arrive si la date d'arrivée = 0. Cela augmente de 1 la file de ladite station.
- 3) Toutes les stations tentent d'émettre, ainsi un packet qui vient d'arriver sera immédiatement envoyé dans l'anneau. Un packet est représenté par une valeur égale à l'id de la station. Une absence de packet est notée -1.
- 4) L'anneau tourne, tous les packets passent dans la case suivante
- 5) Pour chaque station, on vérifie si la case de l'anneau correspondante à la station contient un packet dont la valeur est égale à l'id de la station, si oui, on le supprime.
- 6) on augmente de 1 le tic d'horloge, on diminue de 1 l'arrivée des prochains packets, puis on retourne à l'étape (1)

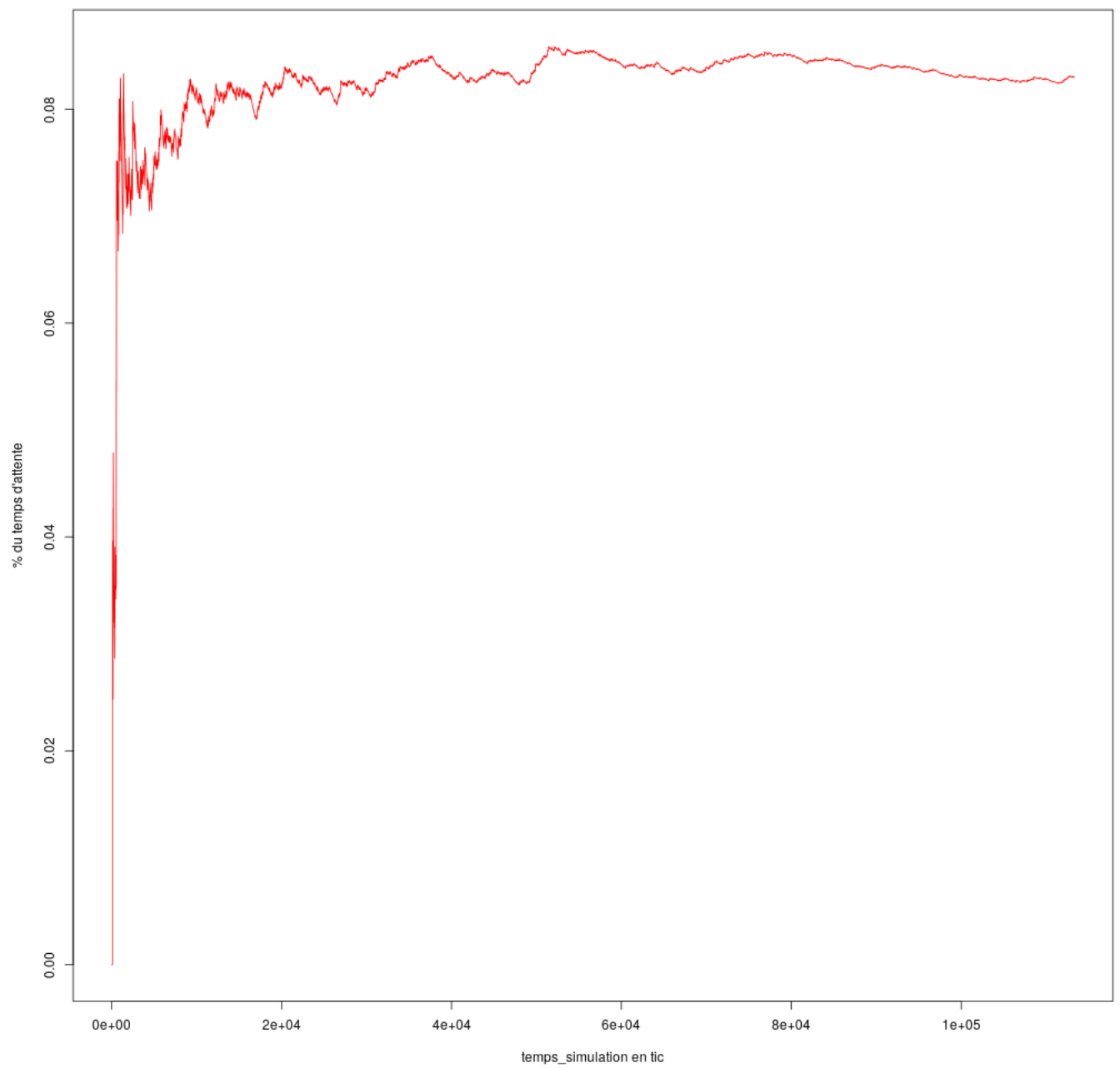
3 Résultats

3.1 Courbes

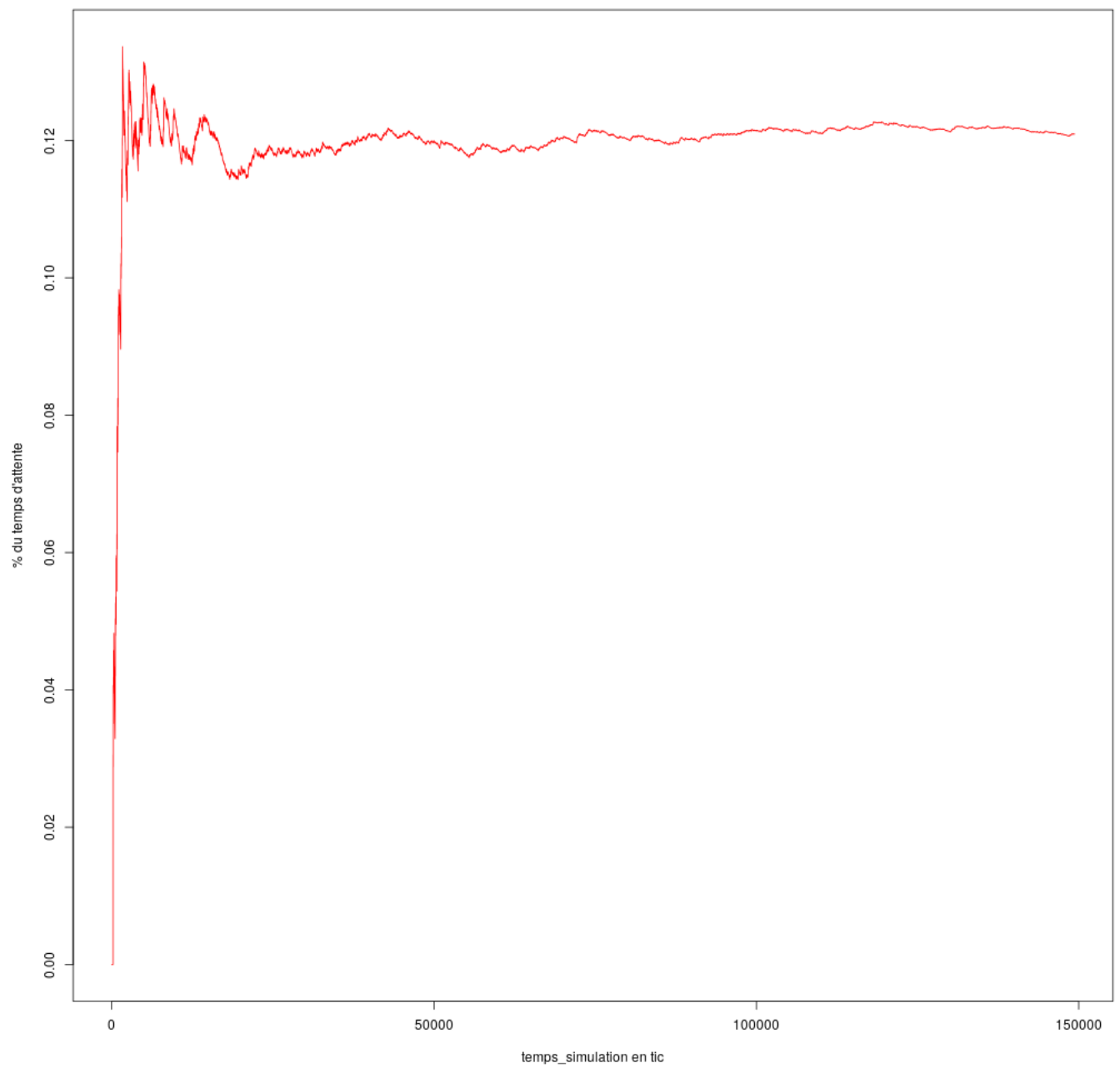
Pour toutes les courbes du temps d'attente, nous avons choisi appliqué la méthode du manchon avec $\text{EPSILON} = 1\text{e-}5$ et une longueur de 1000.



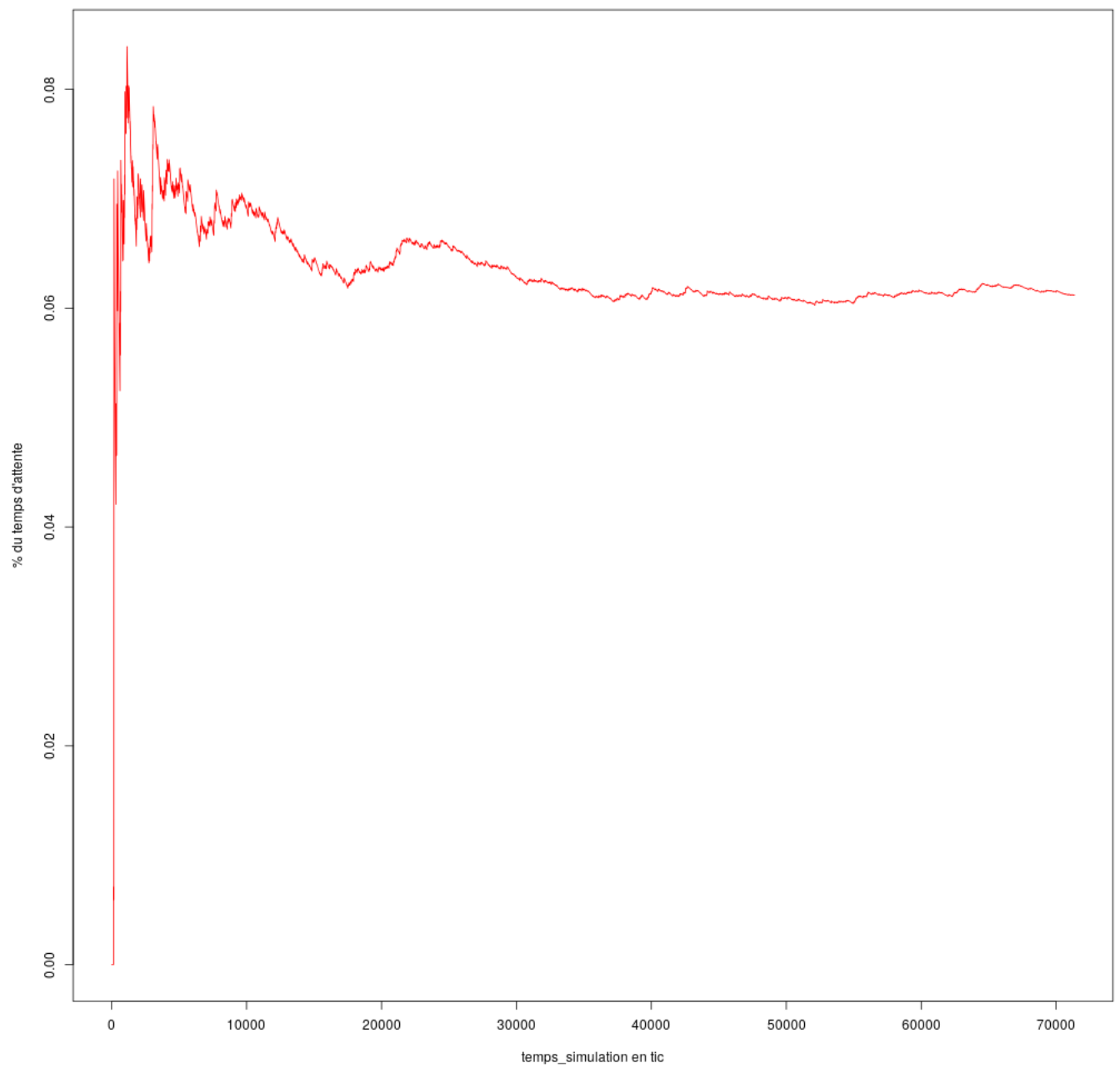
Cette courbe représente le temps d'attente en pourcentage du temps de simulation pour la station 1 et avec $K = 20$ stations. On remarque que la simulation se stabilise à 6% de temps d'attente pour environ 100000 itérations.



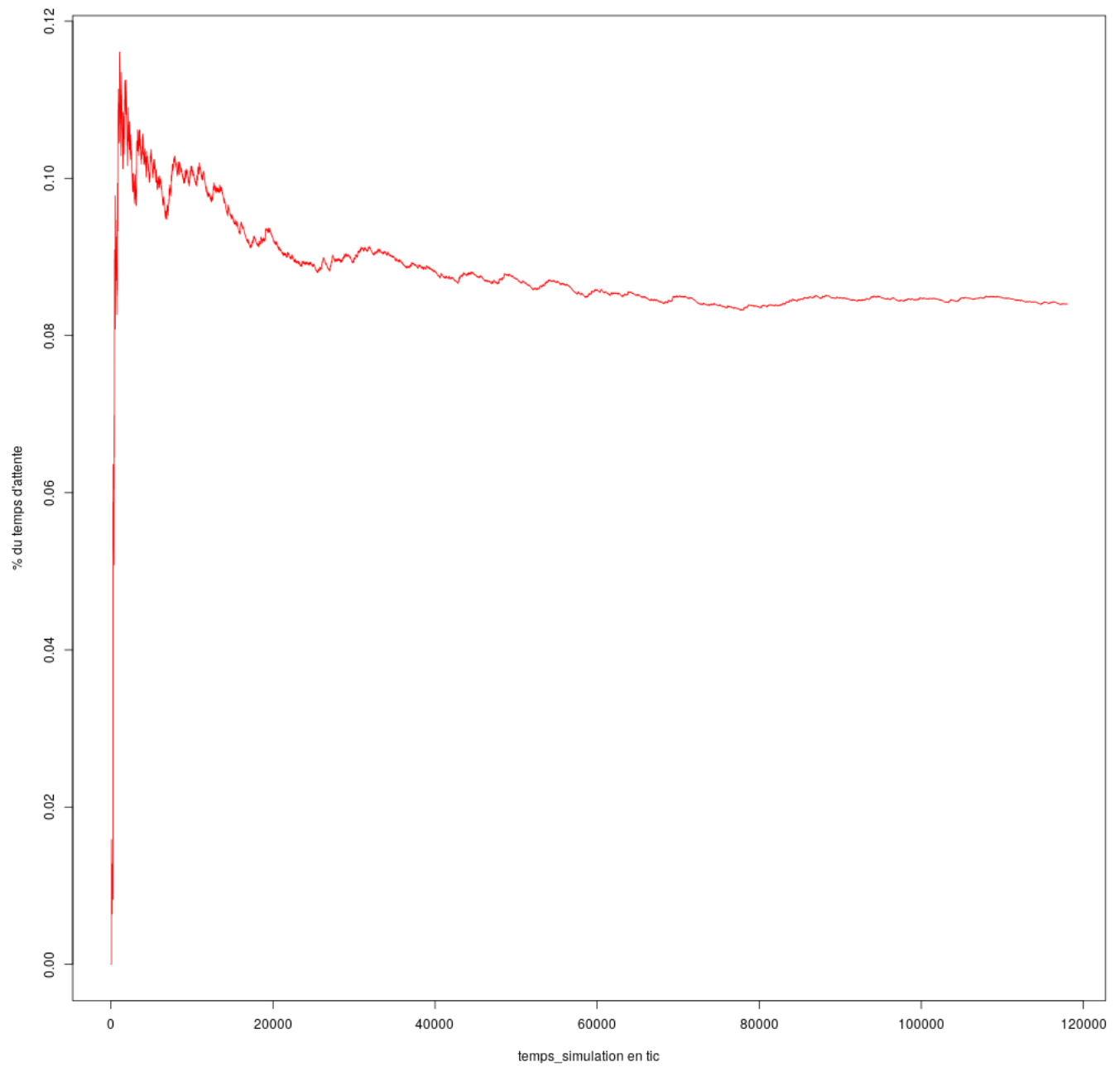
Cette courbe représente le temps d'attente en pourcentage du temps de simulation pour la station 1 et avec $K = 22$ stations. On remarque que la simulation se stabilise à 8% de temps d'attente pour environ 100000 itérations.



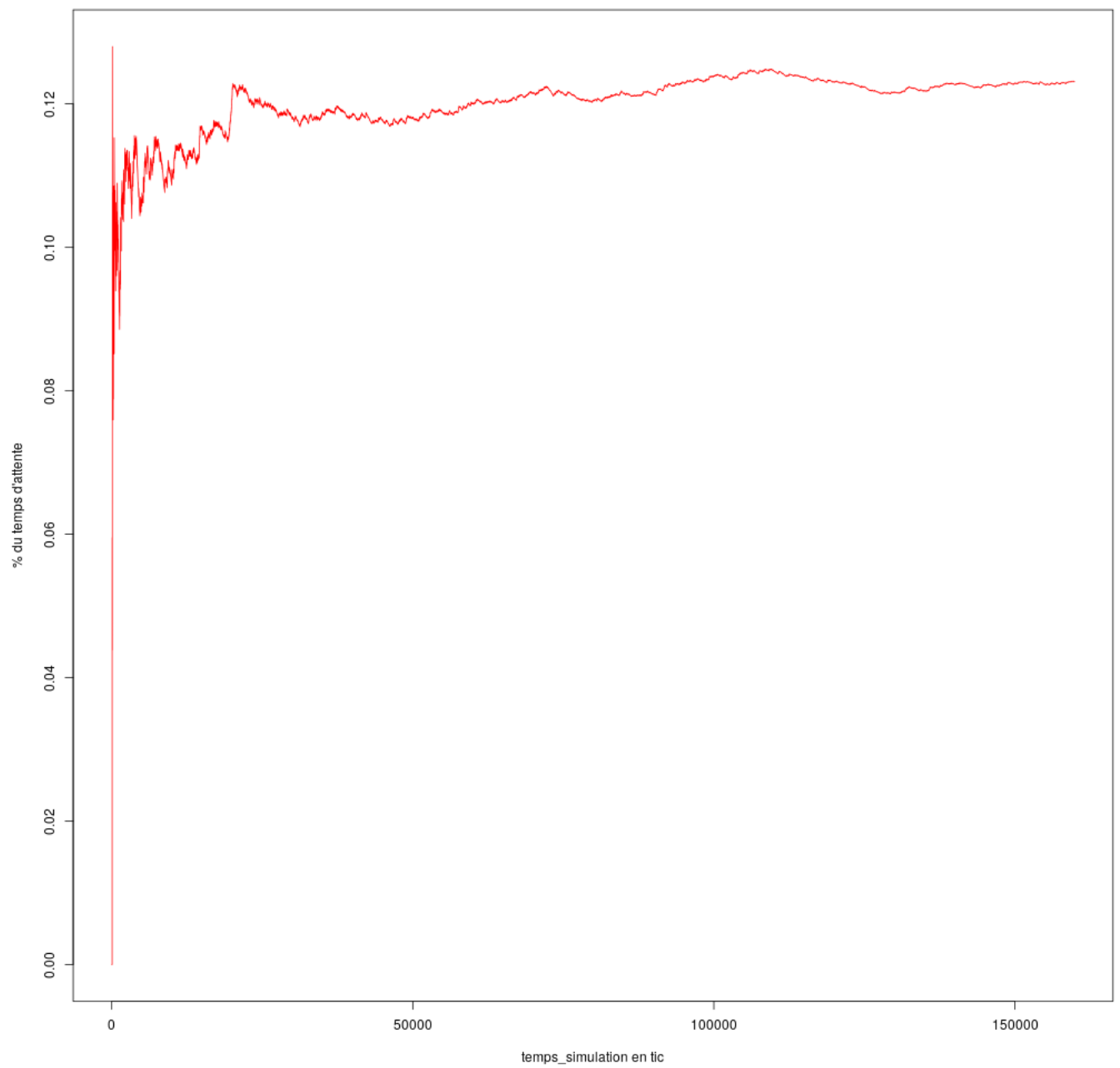
Cette courbe représente le temps d'attente en pourcentage du temps de simulation pour la station 1 et avec $K = 24$ stations. On remarque que la simulation se stabilise à 12% de temps d'attente pour environ 150000 itérations.



Cette courbe représente le temps d'attente en pourcentage du temps de simulation pour la station 10 et avec $K = 20$ stations. On remarque que la simulation se stabilise à 7% de temps d'attente pour environ 70000 itérations.

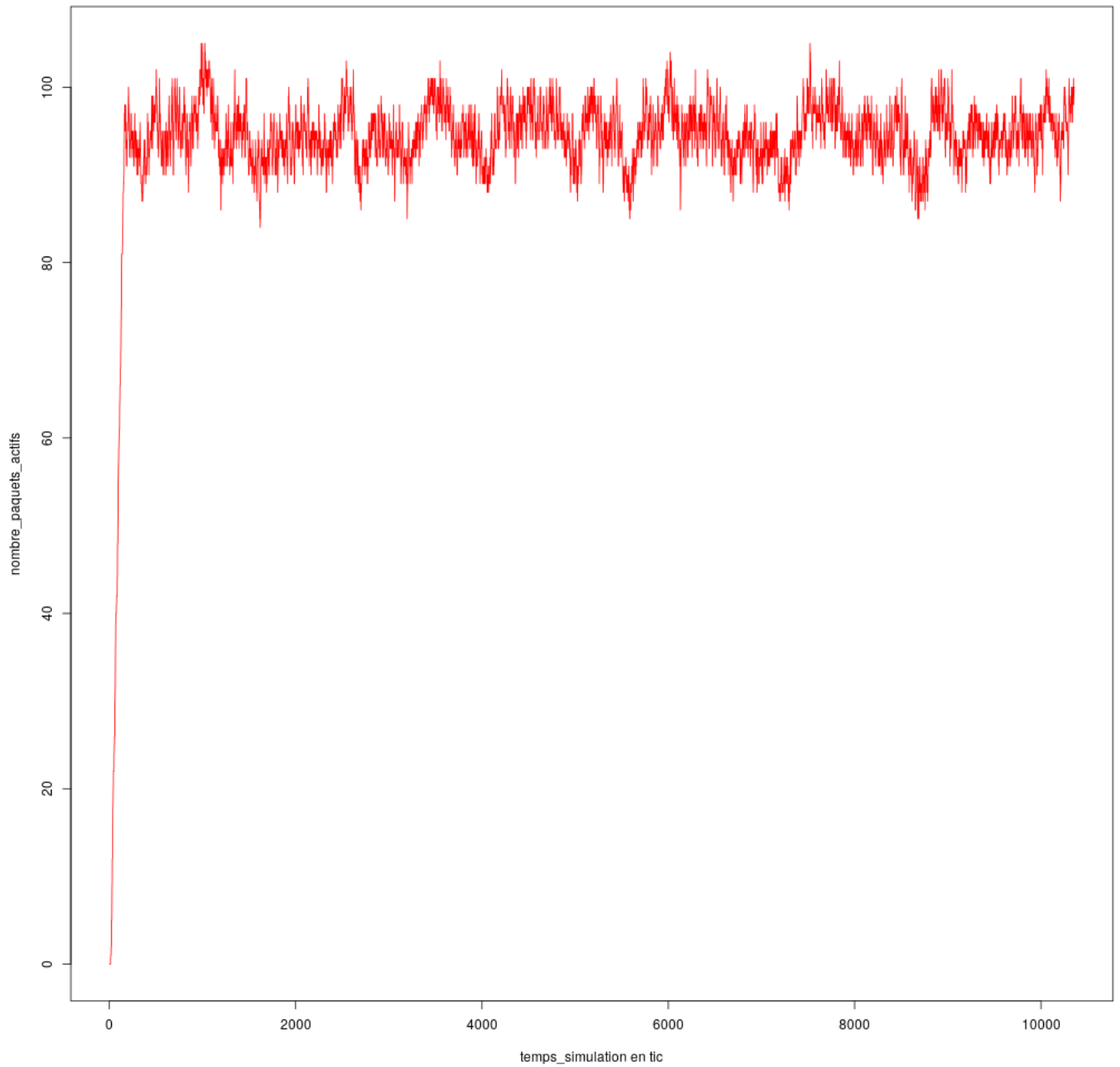


Cette courbe représente le temps d'attente en pourcentage du temps de simulation pour la station 10 et avec $K = 22$ stations. On remarque que la simulation se stabilise à 9% de temps d'attente pour environ 120000 itérations.

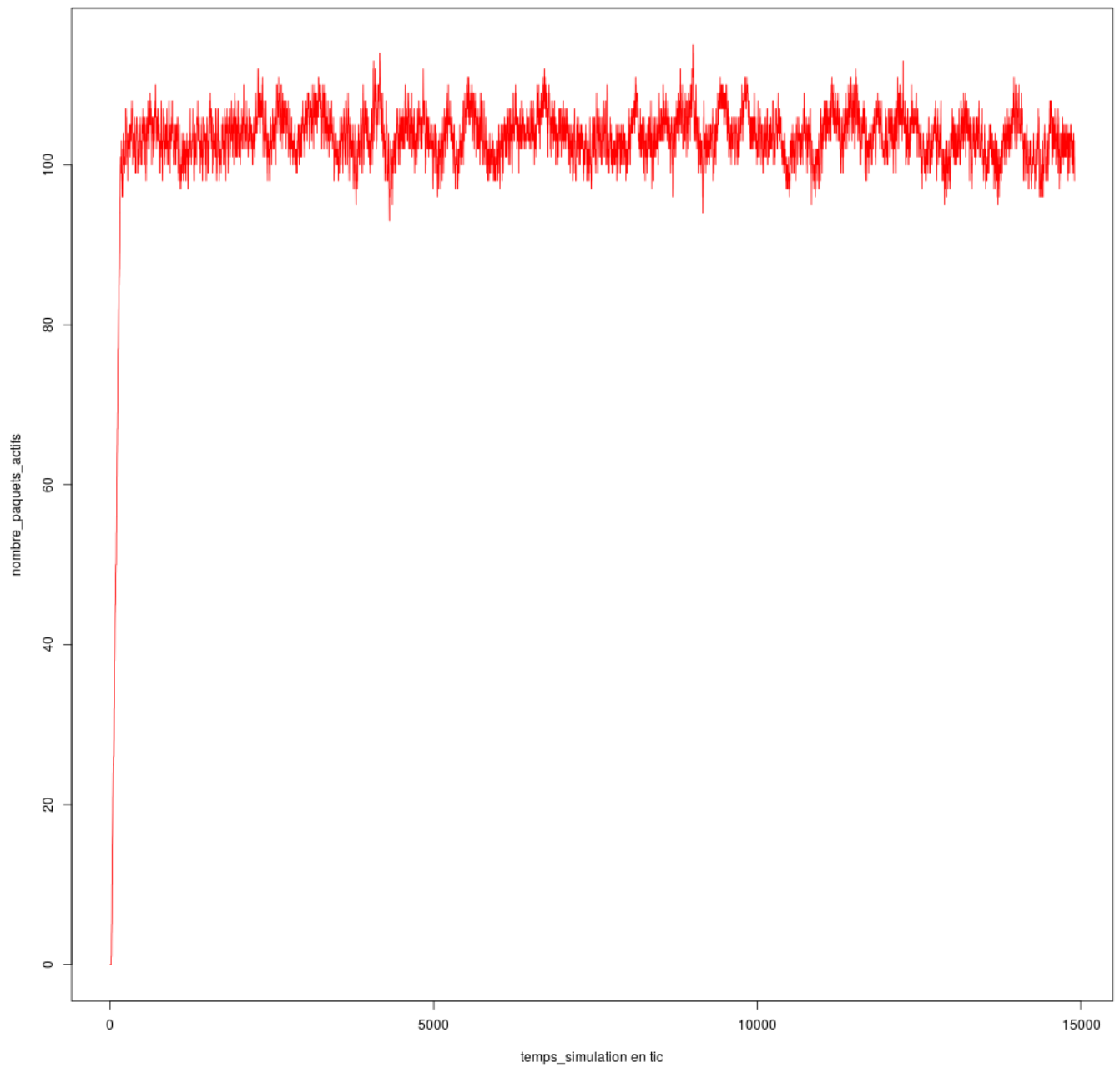


Cette courbe représente le temps d'attente en pourcentage du temps de simulation pour la station 10 et avec $K = 24$ stations. On remarque que la simulation se stabilise à 12% de temps d'attente pour environ 150000 itérations.

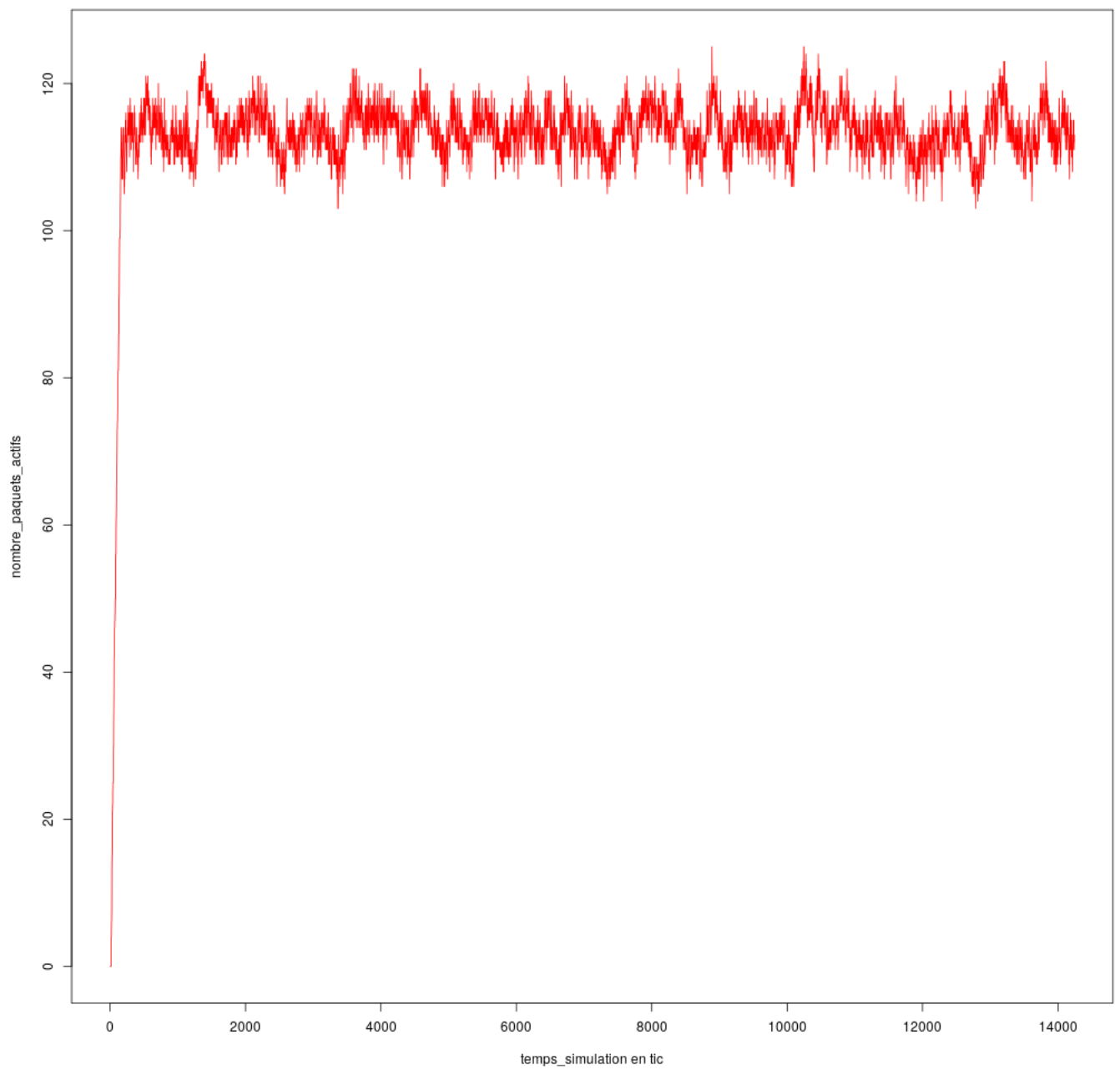
Pour les courbes suivantes, nous avons changé la valeur de l'EPSILON pour $1e-3$, avec $1e-5$ le nombre trop élevé d'itérations rendait la courbe peu pertinente (courbe extrêmement dense sans aucune variation lisible).



Cette courbe représente le nombre de packets actifs avec $K = 20$ stations. La quantité de packets dans l'anneau varie entre 90 et 100 pour 10000 itérations.



Cette courbe représente le nombre de packets actifs avec $K = 22$ stations. La quantité de packets dans l'anneau varie entre 100 et 105 pour 15000 itérations.



Cette courbe représente le nombre de packets actifs avec $K = 24$ stations. La quantité de paquets dans l'anneau varie entre 110 et 120 pour 14000 itérations.

3.2 Questions

Nous n'avons pas observé de valeur K pour laquelle la courbe ne se stabilise définitivement pas, cependant nous avons pu observer qu'à partir de $K = 28$ le réseau se retrouve fortement saturé, et pour $K \geq 30$, le réseau se retrouve dès les premières itérations saturé à 90% et reste ainsi pendant toute la durée de la simulation, le système se stabilisant alors extrêmement vite mais à une très forte saturation.

Pour corriger ce problème, la méthode la plus évidente semble d'agrandir l'anneau, cependant après avoir testé l'hypothèse, celle-ci ne fonctionne pas, car en réalité la suppression des packets dépend également de la taille de l'anneau. Avec un anneau plus grand, la suppression se faisant moins fréquemment, la croissance du taux d'occupation est retardée mais finit tout de même par tendre et se stabiliser vers 85% (avec $K = 32$ et un anneau de taille 5000). Cela semble cohérent, vu que la vitesse d'arrivée des packets ne change pas.

Une solution pourrait donc être d'inclure dans les packets un marqueur indiquant le nombre de stations présentes sur l'anneau. Ce nombre se réduirait à chaque visite, et une fois à zéro, le packet se fait supprimer du réseau par la dernière station visitée. Ainsi, les packets ne feraient plus tout le tour de l'anneau et agrandir le nombre de slots pourrait réellement apporter une réduction de la saturation, mais celle-ci ne semble pas significative car chaque packet ne traverserait en moins qu'un nombre de slot égal à l'écart entre deux stations. Il faudrait adapter le simulateur pour vérifier cette idée.

Une autre possibilité serait de superposer plusieurs anneaux, l'un tournant dans le sens horaire et l'autre dans le sens anti-horaire, ainsi chaque packet n'aurait qu'à traverser une moitié de chemin avant d'être supprimé. On peut ensuite rajouter de plus en plus d'anneaux et insérer les packets dans les anneaux les moins utilisés.

3.3 Analyse

Il ressort de la simulation que le réseau proposé fonctionne correctement jusqu'à environ 28 stations, nombre à partir duquel le réseau sature à 75% et au-delà duquel celui-ci sature immédiatement à 90% de temps d'attente, ce qui rend le modèle non-viable. Le but étant de maximiser l'utilisation du réseau, un nombre de 26 stations semble convenir, valeur pour laquelle 135 slots de l'anneau sont occupés en moyenne pour un temps d'attente de 15%.

La simulation reste cependant incomplète dans le cadre du projet NGREEN, la notion de deadline, le temps de copie des packets, le taux de remplissage des packets ne sont pas pris en compte. Elle pourrait donc être utilisée comme base à compléter afin de réaliser une étude plus précise du conteneur optique.