Etude de performances des réseaux par simulations

Simulation des réseaux

Quentin Pointeau

Antoine Rey

Sciences du Numérique

Architecture, Systèmes et Réseaux

ENSEEIHT

30 mai 2025

Table des matières

Table des figures

Liste des tableaux

1 Introduction

Ce document fait l'objet d'un rapport sur l'ensemble des travaux pratiques réalisés en simulation de réseaux. Chaque partie correspond à un TP.

2 TP1

Dans ce premier TP on considère une source de trafic qui génère des paquets dont l'inter-arrivée est exponentiellement distribuée de moyenne $\frac{1}{\lambda}$. Aucun mécanisme de contrôle n'est mis en place : ni contrôle de flux, ni contrôle de congestion.

Nous nous intéressons au dimensionnement des files de sortie d'un nœud de commutation. Dans la première partie, nous supposons que ces nœuds de commutations n'ont aucune limitation de capacité. Les tailles des datagrammes et des trames sont considérées infinies.

2.1 File M/M/1

Lors de la première simulation avec un temps $T = 10$, on obtient $E[R] = 48 \pm 6$ ms et $E[L] = 0,8 \pm 0,7$. Cela nous fait donc un taux d'erreur $\epsilon_r = \frac{0,7}{0,8} \simeq 90\%$. Or, nous voulons atteindre une précision de 10% soit $\frac{90\%}{9}$. Donc en utilisant la méthode de Le Gall, on détermine T' : le temps nécessaire de simulation pour atteindre une précision de 10%. On obtient $T' = T \times \alpha = 810$ s où $\alpha = 9^2$.

On a choisi de garder μ constant donc on va faire varier λ pour faire varier la charge ρ.

Afin de comparer les performances des différentes files d'attente, nous avons le choix de comparer les temps de simulation nécessaire pour atteindre un intervalle de confiance de 10% ou se fixer un temps de simulation très grand pour être sûr d'avoir un intervalle de confiance de 10% et dans ce cas on compare les intervalles de confiance obtenus pour les différentes valeurs de charge (0,3, 0,6 et 0,9).
Ici nous avons choisi d'utiliser la deuxième méthode pour plus de simplicité. Nos simulations seront donc effectuées avec un temps de 1500 secondes.

Charge(ρ)	0.3	0.6	0.9
$\lambda = \mu \times \rho$	9.9	19.8	29.7
Résultats obtenus par simulation			
$E[L]$	$0.493 \pm 0.047 (T = 1500)$	$1.66 \pm 0.11 (T = 1500)$	$9.22 \pm 0.51 (T = 1500)$
$E[R]$	$0.0464 \pm 0.0007 (T = 1500)$	$0.0784 \pm 0.0008 (T = 1500)$	$0.289 \pm 0.002 (T = 1500)$
Résultats analytiques			
$E[L] = \frac{\rho}{1-\rho}$	0.434	1.5	9
$E[R] = \frac{E[L]}{\lambda}$	0.043	0.076	0.30

Question : Commenter les résultats sur les intervalles de confiance (à durée de simulation constante et à charge variable).

On remarque que logiquement, plus la charge ρ augmente, plus $E[L]$ et $E[R]$ augmentent, c'est-à-dire qu'il y a plus de paquets en attente dans la file et le temps moyen de réponse est aussi plus important. Cependant, tandis que les intervalles de confiance de $E[L]$ augmentent quand la charge augmente, ceux de $E[R]$ restent plus ou moins constants.

Si on compare maintenant les résultats obtenus par simulation avec ceux calculés analytiquement, on se rend compte que les résultats analytiques ne sont souvent pas inclus dans les intervalles de confiance donnés par la simulation. Cela s'explique par le fait que la simulation démarre en ayant un système avec des serveurs et des files d'attente vides. On commence donc avec un régime transitoire avant d'atteindre le régime stationnaire du système. Or les résultats analytiques sont calculés en prenant en compte un

régime stationnaire déjà établi dès le début et donc ne prennent pas en compte le régime transitoire des simulations, d'où l'écart entre les résultats obtenus par simulation et les résultats calculés analytiquement.

2.2 File M/D/1

Charge(ρ)	0.3	0.6	0.9
$\lambda = \mu \times \rho$	9.9	19.8	29.7
Résultats obtenus par simulation			
$E[L]$	$0.365 \pm 0.031 (T = 1500)$	$1.062 \pm 0.061 (T = 1500)$	$5.98 \pm 0.33 (T = 1500)$
$E[R]$	$0.0367 \pm 0.0002 (T = 1500)$	$0.0534 \pm 0.0004 (T = 1500)$	$0.200 \pm 0.002 (T = 1500)$
Résultats analytiques			
$E[L] = \frac{\rho(2-\rho)}{2(1-\rho)}$	0.364	1.05	4.95
$E[R] = \frac{E[L]}{\lambda}$	0.037	0.053	0.167

Question : Comparer les résultats en moyenne et en termes d'intervalles de confiance entre M/M/1 et M/D/1.

Avec la file M/D/1, on se rend compte que les valeurs calculées analytiquement sont comprises dans les intervalles de confiance obtenus par simulation, excepté pour ceux simulés avec une charge de 0,9, où le problème du régime transitoire reste visible.

2.3 Simulation de files de taille finie, file M/M/1/K

Dans le modèle précédent, les buffers sont considérés de très grande taille afin d'éviter les pertes. Utiliser les buffers de capacité limitée, peut être modélisé par une file M/M/1/K.

$$\text{Probabilité de rejet} = \Pi_K = \begin{cases} \frac{\rho^K(1-\rho)}{1-\rho^{K+1}} & \text{si } \rho < 1 \\ \frac{1}{K+1} & \text{si } \rho = 1 \end{cases}$$

Question : Déterminer par simulation la probabilité de rejet de paquets pour $K = 2$, $\lambda = 20$ et $\mu = 33$.

D'après les résultats de la simulation, on a $\Pi_k = 0.186$ or selon la forme au dessus on obtient 0.183.

Question : Tracer le taux de rejet en fonction de K.

Le tracé du taux de rejet en fonction de K est représenté sur la figure 1.

2.4 Simulation d'un système composé de deux nœuds de commutation.

Question : Trouver la loi d'arrivée des paquets dans la deuxième file lorsque la première suit une loi exponentielle de paramètre λ.

Si on suppose un régime stationnaire déjà établi, alors d'après le théorème de Burke, pour une première file d'attente M/M/1, la loi d'arrivée de la deuxième file d'attente est aussi une loi de Poisson de même paramètre λ.

Question : Observer par simulation le comportement de la deuxième file dans le cas d'une première file M/D/1. Expliquer ce comportement.

Dans le cas d'une première file M/D/1, on se rend compte par simulation que le temps de réponse de la deuxième file est constant. En effet, les arrivées dans la deuxième file sont périodiques donc les paquets

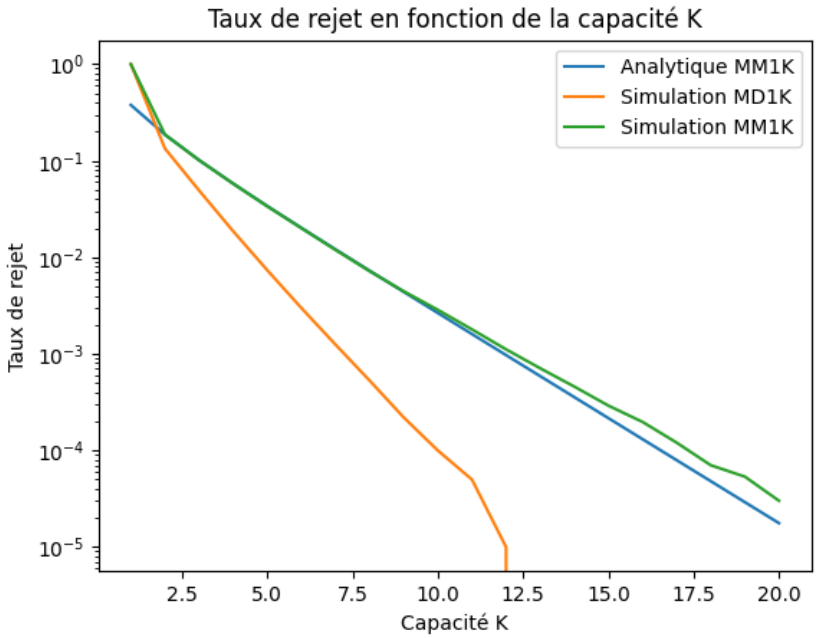


FIGURE 1 – Taux de rejet en fonction de la capacité K

sont servis immédiatement.

Le temps de service dans la première file étant constant et égal à $\frac{1}{\mu}$, on se retrouve avec une loi d'arrivée dans la deuxième file quelconque, représenté sur la figure ci-dessous. On a donc une deuxième file G/M/1.

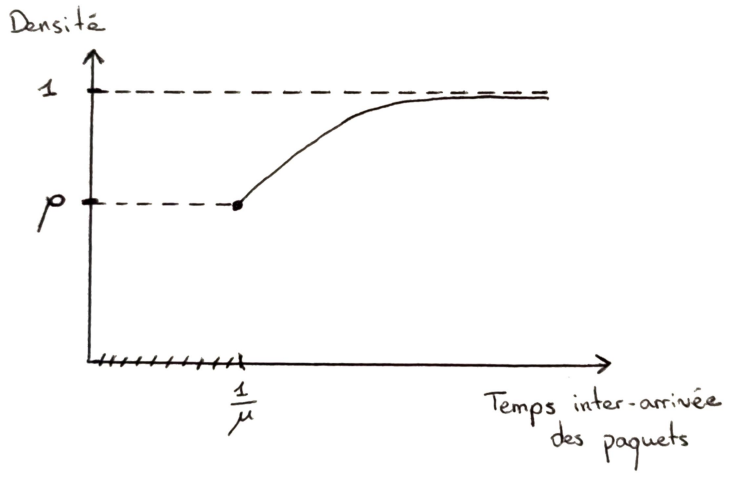


FIGURE 2 – Loi d'arrivée des paquets dans la deuxième file

3 TP2

Dans les réseaux informatiques, pour exploiter efficacement les supports de communication partagés, on a depuis toujours préférés mettre en place des méthodes d'accès dynamiques permettant de tenir compte des besoins des utilisateurs. Parmi ces méthodes, on distingue essentiellement les solutions qui évitent toute collision et celles qui les supportent et entraînent des retransmissions ultérieures.

3.1 Aloha pur

On appelle Aloha pur le cas où l'on a une infinité d'utilisateurs partageant un support qui génèrent des trames de taille constante T (que l'on confond avec le temps d'émission) avec un débit global poissonien de taux λ. Le canal est sans erreur : quand une trame est émise et qu'elle ne rentre en conflit avec aucune autre trame, elle est reçue correctement. Quand deux trames (ou plus) ont des périodes d'émission qui se chevauchent, aucune n'est reçue correctement. À la fin de chaque émission, on suppose que l'utilisateur sait s'il y a eu collision ou non. La population étant infinie, on peut considérer que chaque trame appartient à un utilisateur différent. On ne gère pas alors d'utilisateurs (avec des files d'attente) mais des trames à envoyer ou à renvoyer en cas de collision.

On note alors γ le débit des trames émises (ou réémises sur le support) : $\gamma > \lambda$ et Λ le débit de trames transmises correctement. On appelle capacité du lien $\rho_{\max}$, la charge maximale de sortie soit encore le débit maximal de sortie multipliée par le temps moyen d'émission $\rho_{\max} = \Lambda_{\max} T$.

Il n'y a pas de modèle exact d'un tel système, les approximations classiques consistent à dire que les émissions de paquets sur le lien constituent un processus de Poisson de paramètre γ (la caractérisation de ce processus est extrêmement compliquée mais les études ont montré que si la durée avant retransmission était choisie selon une loi uniforme sur une grande plage de valeurs, cette approximation était raisonnable). Sous cette approximation, on obtient que la probabilité qu'une transmission soit fructueuse est égale à la probabilité qu'il n'y ait aucun débit de transmission pendant l'intervalle $]-T, T[$ appelé période de vulnérabilité. Il vient :

$$P_{\text{succ}} = e^{-2\gamma T}$$

Soit une proportion de temps pendant laquelle le support est efficacement utilisé

$$\rho = \gamma T e^{-2\gamma T} = G e^{-2G}$$

On trouve alors que $\rho_{\max}$ est atteint pour $G = \frac{1}{2}$ et vaut :

$$\rho_{\max} = \frac{1}{2e} \approx 0.18$$

Question : Utiliser le programme de simulation reproduisant ce premier système (on supposera un grand nombre d'utilisateurs au moins 100). Les temps de retransmissions seront tirés selon une loi uniforme sur $[0, T_{\max}]$, $T_{\max}$ étant un paramètre à faire varier. Observer en fonction du débit d'arrivée, le débit soumis au support γ (débit offert), le temps moyen pour transmettre correctement une trame ainsi que la charge efficace du support ρ.

Le réseaux de files d'attente peut être modélisé comme sur la figure 3.
On a $G = \gamma T$ la charge utile et $\rho = G e^{-2G}$ la charge du système. On a aussi

$$\Lambda = \gamma P_{succ} = \gamma e^{-2\gamma T}$$

où $e^{-2\gamma T}$ signifie qu'aucun paquet n'arrive sur une période de $2T$. De plus, on ne sait pas quelle est la loi de γ mais expérimentalement on pense que c'est une loi de Poisson. On a donc ici une hypothèse de plus.

En traçant la charge du système ρ en fonction de la charge utile G (*c.f.* figure 4), on se rend compte que les résultats sont très différents. Cela s'explique par le fait que dans le modèle théorique, on considère un temps de propagation nul et une infinité d'utilisateurs. Cependant ces hypothèses ne sont pas recevables pour la simulation car on a fixé un maximum de 100 utilisateurs et on simule aussi le temps de propagation.

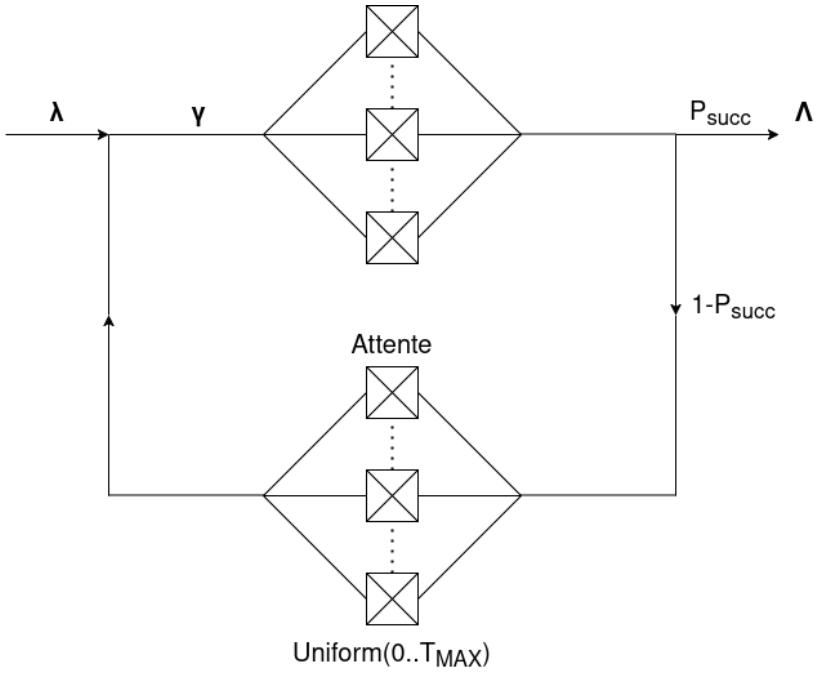


FIGURE 3 – Modélisation du réseaux de files d'attente

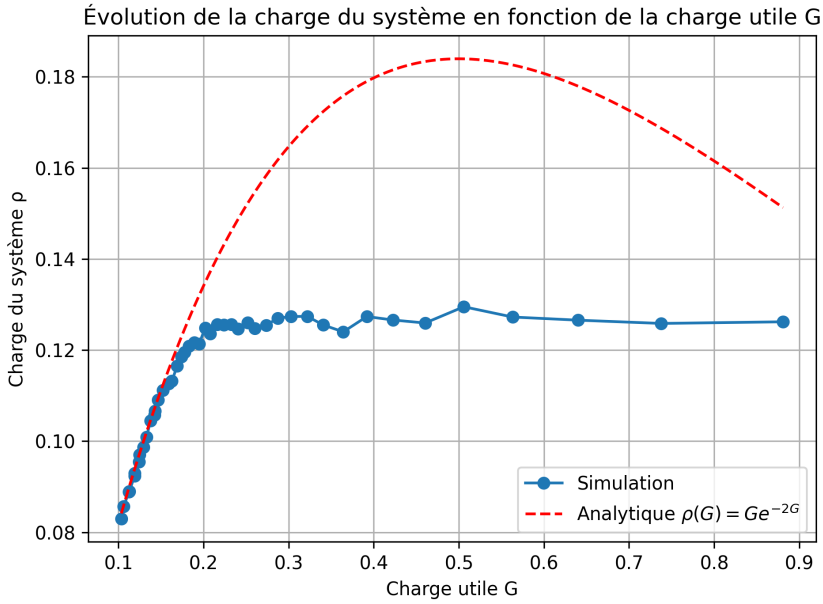


FIGURE 4 – Évolution de la charge du système en fonction de la charge utile G

3.2 Détermination du idle time optimal

Afin de déterminer le `idle_time` optimal, nous avons besoin de connaître la valeur de la charge utile G pour laquelle la charge ρ du système est maximale. En effet, $G = \gamma T$ et γ et lié à λ et

$$\lambda = \sum_{i=1}^{N=100} \lambda_i = \sum_{i=1}^{N} \frac{1}{\text{idle\_time}} = \frac{100}{\text{idle\_time}}$$

donc G est bien lié à `idle_time`.

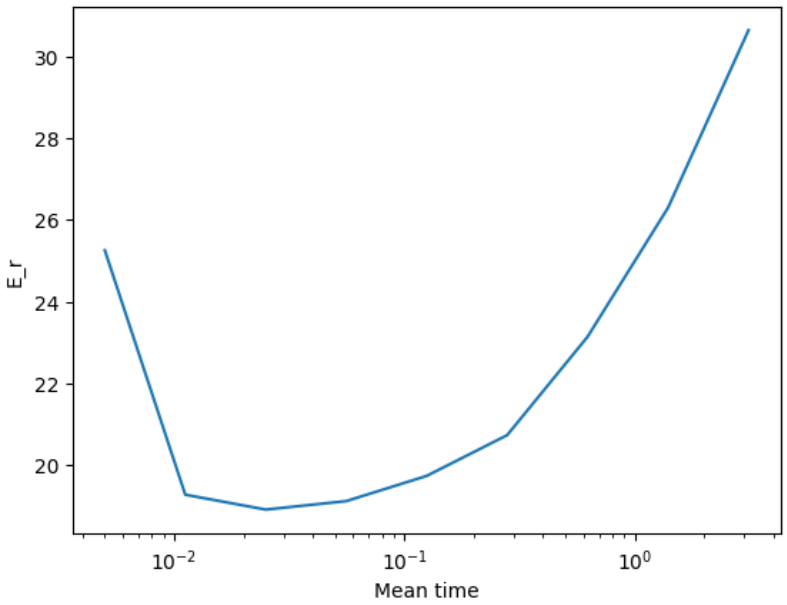


FIGURE 5 – Temps de réponse du système en fonction du temps de backoff

Comme on peut le voir sur la figure 5, si le backoff est trop petit on a plus de chance de rencontrer une collision et donc on va mettre plus de temps a réussir à envoyer le message. Aussi, si le message est trop grand, on va mettre trop de temps avant d'essayer de transmettre à nouveau ce qui va aussi augmenter le temps de réponse du système.

3.3 Détermination du temps de backoff optimal

On cherche maintenant à déterminer le temps de backoff optimal afin de maximiser la charge ρ du système. On obtient par simulation que le temps de backoff moyen optimal est de 0,025 seconde (*c.f.* figure 6). Or

$$\text{mean\_backoff} = \frac{T_{MAX}}{2}$$

donc

$$T_{MAX} = 0,05 \text{ s}$$

3.4 Détermination du nombre de nœuds optimal

On cherche pour cette dernière partie à déterminer le nombre de nœuds c'est-à-dire le nombre moyen d'utilisateurs optimal, toujours pour maximiser la charge ρ du système. On trace donc par simulation sur la figure 7 ρ en fonction du nombre de nœuds N.
On remarque qu'un minimum d'utilisateurs est requis pour maximiser la charge du système mais dépassé un certain seuil, il y a trop d'utilisateurs donc les collisions deviennent courantes et ainsi la charge du système diminue.

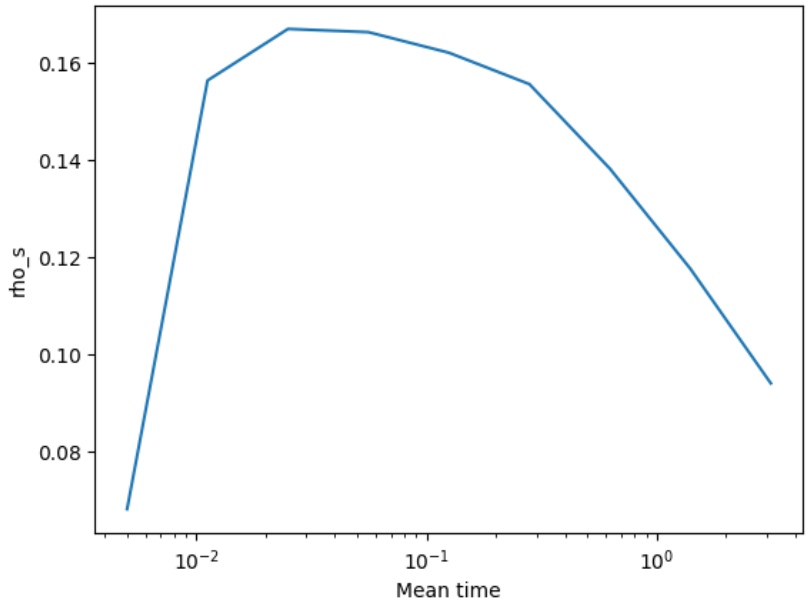


FIGURE 6 – Charge du système en fonction du temps moyen de backoff

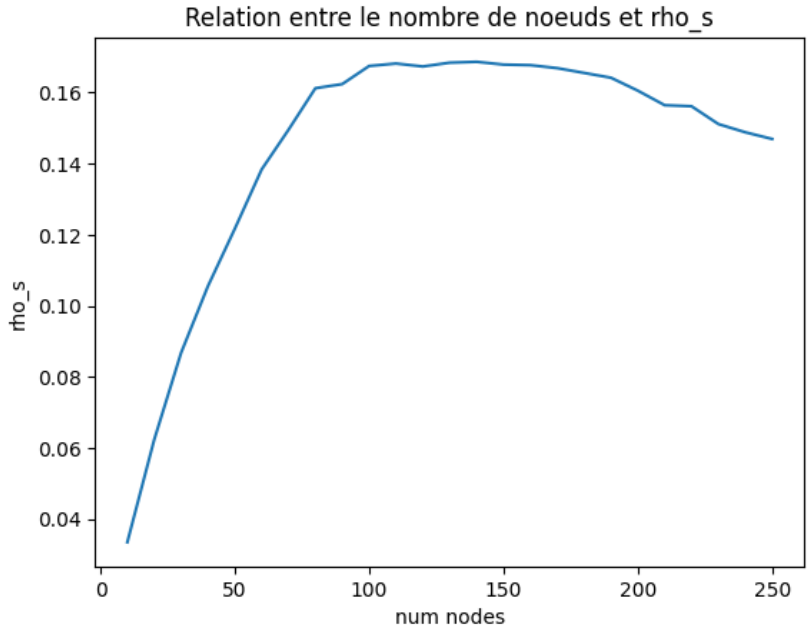


FIGURE 7 – Charge du système en fonction du nombre de nœuds N

Ainsi, le "plateau" observé sur la figure 4 était dû à une simulation effectuée avec un nombre trop petit d'utilisateurs ce qui ne nous permettait pas d'observer une diminution de la charge du système quand la charge utile G augmentait.

4 TP3

4.1 Introduction

Ce dernier TP a été créé par Hugo Chelle, ancien doctorant de l'ENSEEIHT. Toutes les schémas utilisés dans cette partie et l'énoncé du TP font partis de sa création.

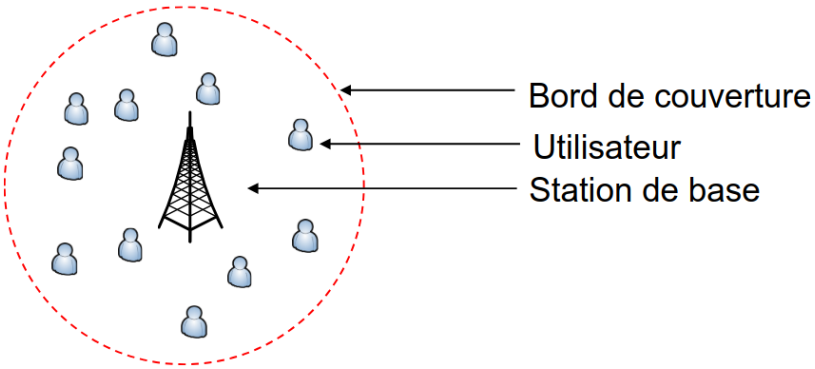


FIGURE 8 – Schéma réseaux d'accès sans fil (Hugo Chelle)

Lorsqu'un utilisateur veut transmettre des données, il effectue les étapes suivantes :

1. L'utilisateur a besoin de ressources pour transmettre, il envoie une requête de ressources à la station de base via un canal d'accès aléatoire partagé par tous les utilisateurs. Pour transmettre sur ce canal, l'utilisateur utilise une méthode d'accès.
2. L'utilisateur attend ensuite la réponse de la station de base, si il ne reçoit pas de réponse il retransmet la demande de ressources.
3. Transmission de la donnée utile via les ressources attribuées par la station de base.

Question : Quel est le nom de la méthode d'accès la plus basique ?

La méthode d'accès la plus basique est Aloha.

Question : Deux versions de cette méthode d'accès sont possibles. Donnez pour les deux versions la formule mathématique qui exprime le débit en fonction de la charge.

Aloha pur : On suppose que le nombre de tentatives de transmission par intervalle de temps T suit une loi de Poisson de paramètre G. Ainsi, la probabilité que k tentatives de transmission aient lieues durant un intervalle de temps T est de $\frac{G^k e^{-G}}{k!}$. De plus, la probabilité de n'avoir aucune collision durant la période de vulnérabilité $2T$ est de e^{-2G}. Ainsi, le débit est le nombre G de tentatives qui ne font pas l'objet d'une collision pendant un intervalle de temps T. Donc le débit D est donné comme suit :

$$D = Ge^{-2G}$$

Aloha slotté : Dans cette version, tous les émetteurs sont synchronisés temporellement sur des slots de temps de durée T. Une transmission ne peut donc commencer qu'au début d'un slot. Ainsi, on réduit la période de vulnérabilité à T. On obtient donc un débit de

$$D = Ge^{-G}$$

Question : Tracez le débit en fonction de la charge pour les deux versions, qu'en pensez vous ?

Question : Citez des méthodes d'accès qui améliorent le débit, dans quel(s) contexte(s) ces nouvelles méthodes sont utilisées et pourquoi ?

Diversity Slotted Aloha (DSA) [1] est un protocole d'accès aléatoire basé sur Slotted Aloha qui vise à créer des copies du même paquet afin de les envoyer sur des time slots différents choisis aléatoirement ou alors sur des fréquences différentes. Il a été montré que le fait d'envoyer plusieurs copies lors d'un trafic faible de données, comme les communications IoT par exemple, réduisait en moyenne le délai et un meilleur débit dû aux probabilités plus faibles d'avoir à retransmettre un paquet. Cependant, avec un haut trafic, l'envoi de plusieurs copies du même paquet participe activement à la surcharge du réseau.

Contention Resolution Diversity Slotted ALOHA (CRDSA) [2] est un protocole qui tout comme DSA, CRDSA envoi des salves de copies du même paquet tout en utilisant des techniques d'annulation d'interférences afin d'améliorer le débit. Cette technique assure des communications de faible latence pour des envois de petits paquets, ce qui encore une fois est particulièrement adapté à l'IoT par satellites.

Reservation-Contention Resolution Diversity Slotted ALOHA (R-CRDSA) [3] est un protocole utilisé pour les communications par satellites qui se base sur deux constats : plus la charge augmente, plus le débit diminue et le fait que CRDSA a besoin de plus de bande passante et consomme plus d'énergie que Slotted Aloha. R-CRDSA apporte donc la solution à ces deux problèmes en contrôlant le nombre de copies envoyées afin de réduire la probabilité d'interférences et donc de permettre un meilleur taux de réussite d'accès au réseau.

Multi-Slots Coded ALOHA (MuSCA) [4] est un protocole qui peut être considéré comme une généralisation de CRDSA donc adapté pour la communication avec des satellites ou pour de l'IoT. La principale différence avec CRDSA est qu'ici, ce ne sont pas des copies des paquets qui sont envoyées mais plusieurs parties d'un seul mot d'un code de correction d'erreur. Les messages envoyés (bursts) sont constitués d'une partie contenant des informations de signalisation et d'une autre partie contenant des données à proprement parler. Plusieurs processus de réduction d'interférences sont appliqués (Interference Cancellation et Successive Interference Cancellation) afin de décoder les informations et d'assurer une transmission correcte.

Multi-slot Coded ALOHA with Irregular Degree Distribution [5] est une amélioration de MuSCA proposé par les mêmes auteurs qui consiste à choisir les taux de codage en fonction de distribution probabilistes de degrés irréguliers.

Irregular Repetition Slotted ALOHA (IRSA) [6] est un protocole utilisant aussi l'envoi de copies d'un même paquet afin d'augmenter la probabilité qu'au moins une copie soit transmise sans collision. De plus, en ayant plusieurs copies du paquet, on augmente les chances de décoder l'information même si toutes les copies ont subit une collision. Ce protocole peut supporter des charges de trafic plus importantes que Slotted Aloha.

Enhanced Dynamic Framed Slotted ALOHA (EDFSA) [7] est un algorithme anti-collision pour l'identification de puces RFID basé sur Framed Aloha. L'algorithme estime le nombre de tags non lus et ajuste le nombre de tags répondants ou la taille de la frame en conséquence. Cet algorithme assure que le nombre de slots nécessaire pour lire les tags augmente linéairement en fonction du nombre de tags présents.

An Adaptive RFID Anti-Collision Algorithm Based on Dynamic Framed ALOHA [8] présente un algorithme adaptatif permettant de réduire le nombre de time slots et de rounds nécessaires à l'identification de puces RFID. Cet algorithme est plus performant que les autres algorithmes fondés sur Framed Aloha.

4.2 Modélisation simple d'un réseau d'accès 4G

4.2.1 Abstraction couche physique

La méthode d'accès utilisée par la 4G est l'Aloha slotté en temps, tslot représente la durée du slot temporel (t_{slot} = 10 ms). Des codes orthogonaux sont utilisés pour améliorer les performances du canal d'accès aléatoire, le nombre de codes orthogonaux est donné par N_{codes}.

Question : Selon vous pourquoi le 3GPP a fait ce choix de méthode d'accès ?

Parce que c'est la méthode d'accès la plus simple, elle est peu coûteuse et tout le monde peu facilement l'implanter. Elle ne nécessite pas non plus de se synchroniser temporellement avec d'autres équipement, ce qui rend cette méthode encore moins contraignante à implanter sur les terminaux utilisateurs et moins contraignante pour les eNodeB.
On pourrait alors se demander pourquoi le 3GGP n'a pas décidé d'utiliser des méthodes d'accès de type CSMA. Plusieurs arguments s'opposent à cela. D'abord, des questions énergétiques se posent. L'écoute active du support de communication demande beaucoup d'énergie et l'utilisation de la batterie des terminaux est un enjeu important dans les réseaux mobiles. De plus, les terminaux utilisateurs sont bien plus loin des eNodeB que dans les réseaux locaux avec par exemple le WiFi où les utilisateurs ne sont qu'à quelques dizaines de mètres maximum du point d'accès. Ainsi, dans les réseaux mobiles l'écoute du support serait réalisée avec un délai bien plus important ce qui rendrait donc cette méthode bien moins efficace et donc très peu intéressante.

Question : Donnez le packet loss ratio (PLR) en fonction du nombre de trames transmises. En déduire le débit du canal d'accès aléatoire.

Le packet loss ratio (PLR) est le nombre de paquets perdus par rapport au nombre de paquets transmis. On sait que la probabilité de rejet d'un paquet est de $1 - e^{-G}$ avec G la charge du système. Soit $N_{paquets}$ le nombre de paquets envoyés et N_{codes} le nombre total de codes orthogonaux pour la transmission. Alors on peut écrire la charge du système comme

$$G = \frac{N_{paquets}}{N_{codes}}$$

Donc

$$PLR = 1 - e^{-\frac{N_{paquets}}{N_{codes}}}$$

Ainsi, on peut calculer le débit D.

$$\begin{aligned} D &= G(1 - PLR) \\ &= G(1 - (1 - e^{-G})) \\ D &= Ge^{-G} \end{aligned}$$

Question : Combien de trames la station de base peut recevoir correctement au maximum par time slot ?

Dans Aloha slotté, si deux stations émettent sur le même time slot, alors il y aura nécessairement un collision et les stations devront tenter à nouveau d'émettre après une attente aléatoire (le backoff). Cependant, nous pouvons émettre ici sur N_{codes} codes différents en même temps grâce à leur orthogonalité deux à deux. Ainsi, une collision aura lieu uniquement si deux stations émettent en même temps et sur le même code. Donc dans le meilleur des cas, la station de base reçoit un message par code. On en conclut donc que la station de base peut recevoir N_{codes} trames correctement par time slot.

4.2.2 Abstraction couche MAC

Nous supposons que toutes les requêtes transmises durant un time slot sont traitées par la station de base pendant $d_{traitement}$ time slots. Ensuite, des acquittements sont envoyés aux utilisateurs. Pour des raisons de simplification nous supposons que les ressources attribués à l'utilisateur sont envoyés avec l'acquittement.

- Lorsqu'un utilisateur reçoit un acquittement, il arrête la procédure de contention pour transmettre sa donnée utile.
- L'utilisateur ne reçoit pas d'acquittement, il continue la procédure de contention. Son time slot de retransmission est déterminé par d_{rand}. Nous considérons que le nombre de transmissions de requêtes de ressources est borné par $N_{MaxTransmission}$, lorsque l'utilisateur dépasse cette borne il abandonne la procédure de contention.

La figure 9 illustre le fonctionnement de l'abstraction de couche MAC utilisée.

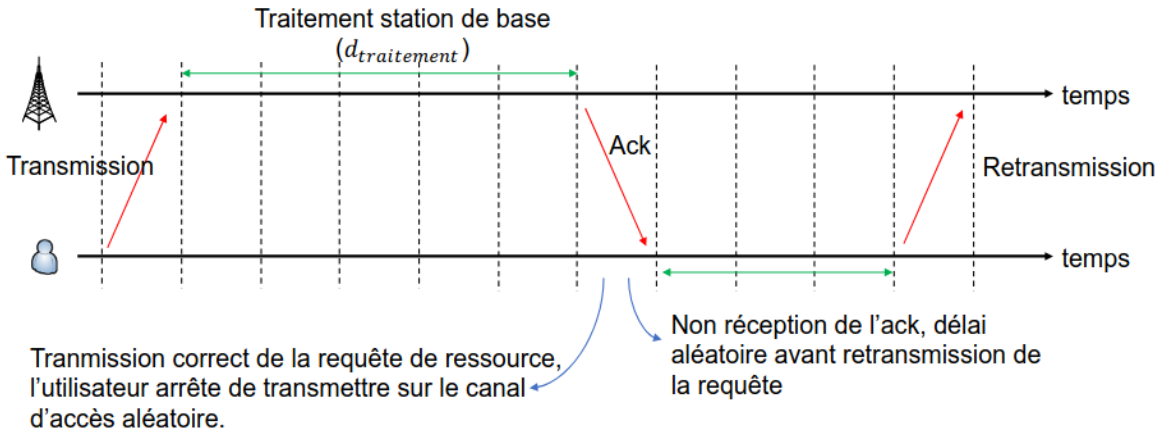


FIGURE 9 – Schéma explicatif de l'abstraction de la couche MAC (Hugo Chelle)

$d_{traitement}$	5 t_{slot}
d_{rand}	$rand(\llbracket 1;3 \rrbracket)$
$N_{MaxTransmission}$	10

TABLE 1 – Paramètres couche MAC considérés pour notre étude (Hugo Chelle)

Question : Quelles sont les hypothèses importantes faites par cette abstraction ?

Plusieurs hypothèses ont été faites pour abstraire la couche MAC. On peut noter la plus importante d'entre elles : nous supposons que le temps de traitement d'une demande de ressource est constant (égal à $d_{traitement}$). Nous faisons aussi une autre grosse hypothèse : si un paquets d'acquittement (ACK) est envoyé par la station de base, alors on suppose qu'il est forcément reçu par le terminal utilisateur. En d'autres termes, on suppose qu'aucune collision n'a lieu pour les paquets envoyés sur le lien descendant. Enfin, une autre hypothèse a été faite mais d'un peu moindre importance cette fois-ci : nous supposons que chaque paquet est transmis sur un unique time slot.

Question : Quel est l'intérêt de d_{rand} ?

Pour comprendre l'intérêt de d_{rand}, raisonnons sur un exemple. Supposons que deux terminaux utilisateur envoient une demande de ressource durant le même time slot et sur le même code. Dans ce cas, il y a évidemment collision et les deux terminaux utilisateurs vont devoir émettre leur demande à nouveau. Logiquement, ils essaient de retransmettre sur le prochain time slot. Ils seront donc à nouveau deux à transmettre une demande de ressources sur le même time slot et le processus se répètera à l'infini. Il est donc nécessaire de différer leur demande d'un temps aléatoire afin de les *désynchroniser*. Ainsi, on tire un nombre aléatoire d_{rand} pour chaque terminal utilisateur afin de déterminer le prochain time slot sur lequel il émettra. On résout donc le problème de synchronisation des retransmissions.

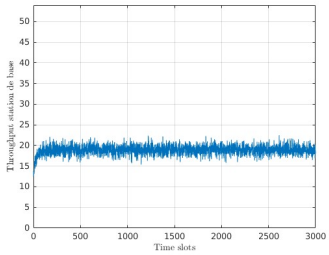
Question : Quel est l'intérêt de $N_{MaxTransmission}$?

L'intérêt de $N_{MaxTransmission}$ est de signaler au terminal utilisateur que sa demande ne peut être satisfaite (station de base non joignable, réseau surchargé, etc.). C'est aussi très utile pour ne pas continuer à surcharger un réseau déjà surchargé avec de vieilles requêtes. Plus simplement, un terminal utilisateur peut être *"impatient"* et arrêter sa demande de ressource.

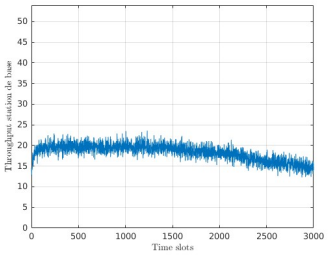
4.2.3 Analyse des résultats

Question : A partir de combien de nouveaux utilisateurs par time slot le système commence-t-il à être instable (fournissez des courbes) ? Est-ce que c'était prévisible mathématiquement ?

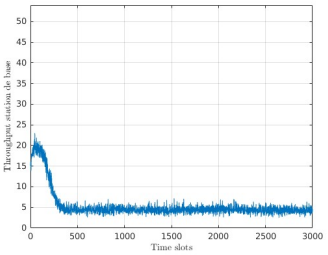
Après plusieurs simulations, nous trouvons que le système commence à être instable à partir de 20 nouveaux utilisateurs par time slot. Les simulations de la figure 10 ont été effectuées avec 54 codes et un temps de simulation de 3000 time slots.



(a) 19 nouveaux utilisateurs par time slot



(b) 20 nouveaux utilisateurs par time slot



(c) 23 nouveaux utilisateurs par time slot

FIGURE 10 – Graphes du débit de la station de base en fonction du nombre de nouveaux utilisateurs par time slot.

Ce résultat était prévisible mathématiquement. En effet, le débit maximal avec Aloha slotté sur un code donné est de

$$D = \frac{1}{e}$$

($G = 1$) donc sur N_{codes} codes, ce débit maximal est de

$$\frac{N_{codes}}{e}$$

Donc pour $N_{codes} = 54$, on obtient

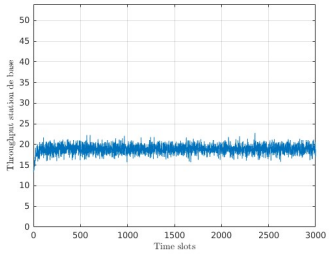
$$D = \frac{54}{e} \simeq 19,87$$

Question : L'impatience des utilisateurs n'est pas simulé dans le simulateur, rajoutez la. Quel est l'impact sur les simulations ?

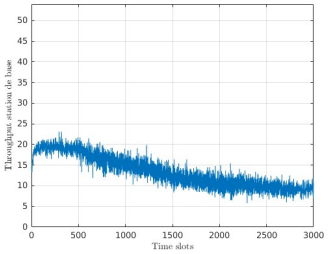
Nous avons simulé l'impatience des utilisateurs en mettant la valeur de d_{rand} à 0. En effet, dès que la demande de ressource a échoué, on émet à nouveau immédiatement. Les simulations décrites par la figure 11 montrent que en-dessous de 20 nouveaux utilisateurs par time slot, le système reste stable. Cependant, à partir de 20 nouveaux utilisateurs par time slot, le système devient plus instable que dans les précédentes simulations. Le système devient *"inutilisable"* plus rapidement.

Question : Une station de base peut couvrir une zone assez vaste, en vous basant sur toutes vos réponses précédentes, expliquez pourquoi dans certaines situations il est compliqué d'accéder "au réseau".

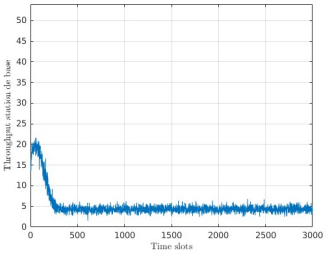
Au vu des réponses précédentes, nous pouvons imaginer une situation où beaucoup d'utilisateurs essaient de se connecter au réseau en même temps (lors d'un événement sportif de grosse ampleur par exemple). Dans ce cas, la charge du système va devenir trop élevée et si les utilisateurs sont impatients, alors ce sera encore pire. Dans un cas comme celui-ci, il peut en effet être compliqué d'accéder au réseau.



(a) 19 nouveaux utilisateurs par time slot



(b) 20 nouveaux utilisateurs par time slot



(c) 23 nouveaux utilisateurs par time slot

FIGURE 11 – Graphes du débit de la station de base en fonction du nombre de nouveaux utilisateurs par time slot en prenant en compte l'impatience des utilisateurs.

4.3 Introduction au contrôle de charge

4.3.1 Présentation du mécanisme back-off

Supposons lorsque la station de base est surchargée de requêtes, elle utilise un mécanisme de type back-off afin de limiter le nombre de requête transmises par les utilisateurs. Le mécanisme est composé de deux paramètres :

- Une probabilité d'accès p_{access}.
- Un nombre de slots maximal de blocage $N_{\text{Slots Barring}}$

Avant de transmettre un utilisateur va tirer un nombre aléatoire et le comparer à p_{access} pour savoir s'il est autorisé à transmettre. Si l'utilisateur échoue ce test, il essayera de transmettre de nouveau $rand(\llbracket 1; N_{\text{Slots Barring}} \rrbracket)$ time slots plus tard.

4.3.2 Analyse des résultats

Question : Selon vous quelle(s) métrique(s) peux t-on utiliser pour évaluer les performances du contrôle de charge ?

Afin d'évaluer les performances du contrôle de charge, nous pouvons étudier :

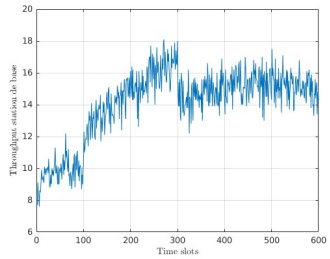
- Le temps de réponse moyen des requêtes.
- Le débit de sortie du système.
- Le nombre de personnes qui n'ont pas réussi à transmettre leur requête car certains utilisateurs vont arriver à leur nombre maximum de retransmissions ($N_{MaxTransmission}$).

Question : En faisant varier les paramètres du contrôle de charge (p_{access} et $N_{\text{Slots Barring}}$), expliquez l'influence de chacun des paramètres.

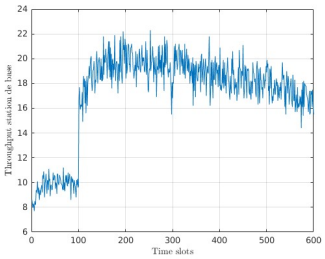
Grâce aux simulations dont les résultats sont récensés dans la figure 12, on se rend compte que pour une valeur de $N_{\text{Slots Barring}}$ fixée (figures 12a, 12b, 12c), si p_{access} est trop petit, alors on limite trop le débit et on ne profite pas assez de toutes les ressources disponibles du système. Au contraire, lorsque que p_{access} est trop grand, trop de transmissions sont retardées donc le débit s'écroule.

Pour une valeur de p_{access} fixée, si $N_{\text{Slots Barring}}$ (figures 12d, 12e, 12f) est trop petit alors trop d'utilisateurs vont essayer de retransmettre en même temps et donc on aura beaucoup de collisions. En résulte un débit qui s'écroule. Au contraire, si $N_{\text{Slots Barring}}$ est trop grand, alors on retarde tellement les retransmissions qu'on n'utilise plus toute la capacité disponible du système.
Question : Selon vous, quel couple de paramètres p_{access}, $N_{\text{Slots Barring}}$ permet d'obtenir les meilleurs performances pour un scénario de trafic donné ?

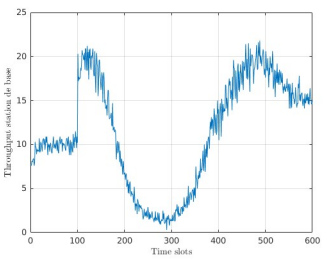
Si on expérimente un pic de trafic à l'instant t, alors on souhaite retarder $N_{utilisateurs}(t) - N_{optimal}$ utilisateurs, c'est-à-dire le nombre d'utilisateurs *en trop* par rapport au nombre d'utilisateurs pour un



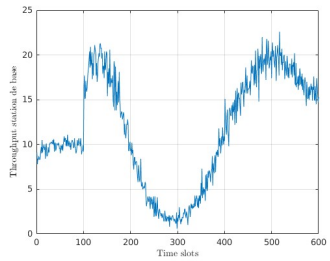
(a) $p_{access} = 0,5$
$N_{\text{Slots Barring}} = 250$



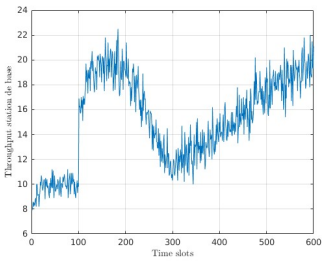
(b) $p_{access} = 0,8$
$N_{\text{Slots Barring}} = 250$



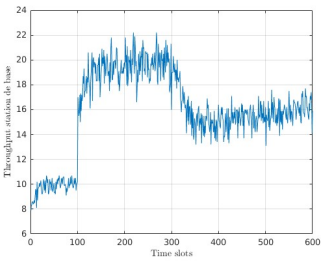
(c) $p_{access} = 1,0$
$N_{\text{Slots Barring}} = 250$



(d) $p_{access} = 0,8$
$N_{\text{Slots Barring}} = 10$



(e) $p_{access} = 0,8$
$N_{\text{Slots Barring}} = 100$



(f) $p_{access} = 0,8$
$N_{\text{Slots Barring}} = 500$

FIGURE 12 – Graphes du débit de la station de base en fonction de p_{access} et de $N_{\text{Slots Barring}}$

fonctionnement optimal du système. Une solution serait de modifier p_{access} de telle sorte que

$$p_{access} = \frac{N_{utilisateurs}(t) - N_{optimal}}{N_{optimal}(t)}$$

ce qui permettrait de retarder la proportion d'utilisateur qui est *en trop* au temps t.
Cependant si avant le pic on avait un taux d'utilisation inférieur au taux optimal et que le pic est important, on va avoir une grande probabilité de retarder les paquets et le système va donc mettre beaucoup de temps avant d'atteindre le régime optimal.

On souhaite aussi modifier le $N_{\text{Slots Barring}}$ de manière à implanter un back-off exponentiel, c'est-à-dire multiplier par deux $N_{\text{Slots Barring}}$ à chaque tentative de retransmission si le réseau est toujours surchargé. Cependant, si le réseau subit une longue surcharge, le débit du système va trop retarder les requêtes et donc le problème sera juste remis à plus tard (comme vu précédemment sur la figure 12f).

Références

[1] G. Choudhury et S. Rappaport, "Diversity ALOHA - A Random Access Scheme for Satellite Communications," *IEEE Transactions on Communications*, t. 31, n° 3, p. 450-457, 1983. doi : 10.1109/TCOM.1983.1095828.

[2] E. Casini, R. De Gaudenzi et O. Del Rio Herrero, "Contention Resolution Diversity Slotted ALOHA (CRDSA) : An Enhanced Random Access Schemefor Satellite Access Packet Networks," *IEEE Transactions on Wireless Communications*, t. 6, n° 4, p. 1408-1419, 2007. doi : 10.1109/TWC.2007.348337.

[3] M. Lee, J.-K. Lee, J.-J. Lee et J. Lim, "R-CRDSA : Reservation-Contention Resolution Diversity Slotted ALOHA for Satellite Networks," *IEEE Communications Letters*, t. 16, n° 10, p. 1576-1579, 2012. doi : 10.1109/LCOMM.2012.082012.120573.

[4] I. S. C. Author, "An enhanced multiple random access scheme for satellite communications," in *2012 Wireless Telecommunications Symposium*, sér. 2012 Wireless Telecommunications Symposium (WTS 2012), [Place of publication not identified] : IEEE, 2012, p. 1-6, isbn : 1457705796.

[5] H.-C. Bui, J. Lacan et M.-L. Boucheret, "Multi-slot coded ALOHA with irregular degree distribution," in *2012 IEEE First AESS European Conference on Satellite Telecommunications (ESTEL)*, 2012, p. 1-6. doi : 10.1109/ESTEL.2012.6400144.

[6] M. Ghanbarinejad et C. Schlegel, "Irregular repetition slotted ALOHA with multiuser detection," in *2013 10th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, 2013, p. 201-205. doi : 10.1109/WONS.2013.6578348.

[7] S.-R. Lee, S.-D. Joo et C.-W. Lee, "An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification," in *The Second Annual International Conference on Mobile and Ubiquitous Systems : Networking and Services*, 2005, p. 166-172. doi : 10.1109/MOBIQUITOUS.2005.13.

[8] C. W. LEE, H. CHO et S. W. KIM, "An Adaptive RFID Anti-Collision Algorithm Based on Dynamic Framed ALOHA," *IEICE Transactions on Communications*, t. E91.B, n° 2, p. 641-645, 2008. doi : 10.1093/ietcom/e91-b.2.641.