N° d'ordre : 42/2012-M/ EL

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université des Sciences et de la Technologie Houari Boumediene
(U.ST.H.B)

Faculté d'Electronique et d'Informatique

MÉMOIRE

Présenté pour l'obtention du diplôme de **MAGISTER**
En : ELECTRONIQUE

Spécialité : **Systèmes Radio Fréquences et Micro Ondes**

Par: Rennane Ahmed

THÈME

Etude de l'algorithme d'Aloha anticollision pour un système RFID

Soutenu publiquement le 06/06/2012, devant le jury composé de :

CHITROUB Salim	Professeur,	à l'USTHB	Président
TOUHAMI Rachida	Professeur,	à l'USTHB	Directrice de mémoire
FERGANI Belkacem	Maitre de Conférences/A,	à l'USTHB	Examinateur
BOUKHENOUS Samir	Maitre de Conférences/A,	à l'USTHB	Examinateur
SAADI Hadjer	Maitre Assistant/A,	à l'INPTIC	Invitée

Remerciements

Je tiens tout d'abord à remercier vivement ma promotrice, Mme Touhami pour ses conseils précieux, pour sa disponibilité et son suivi permanent durant la période du projet.

Mes remerciements les plus vifs s'adressent aussi aux membres de jury d'avoir accepté d'examiner et d'évaluer ce travail.
Sans omettre bien sûr de remercier profondément tous ceux qui ont contribué de près ou de loin à la réalisation du présent travail.

Et enfin, que mes chers parents, et bien avant tout, trouvent ici l'expression de mes remerciements les plus sincères et les plus profonds en reconnaissance de leurs aides, soutien et encouragement.

TABLE DES MATIERES

Chapitre 02: Environnement de la simulation d'un système RFID

Chapitre 03: Simulation et résultats

❖

Table des figures

Liste des tableaux

Liste des abréviations:

ACPB	Anti Collision using Parity Bit
AFSA	Advanced Framed Slotted Aloha Algorithm
Aloha	Arial Location Of Hazardous Atmospheres
BFSA	Basic Framed Slotted Aloha Algorithm
BSA	Binary Search Algorithm
BS-CTTA	Bi-Slotted Collision Tracking Tree Algorithm
BS-QTA	Bi-Slotted Query Tree Algorithm
BTA	Binary Tree Algorithm
BTWA	Binary Tree Working Algorithm
CFMA	Conflict-Free Multiple Access
CSMA	Carrier Sense Multiple Access
CTTA	Collision Tracking Tree Algorithm
DBSA	Dynamic Binary Search Algorithm
DCMA	Dual Channel Multiple Access
DCS	Distributed Color Selection
DFSA	Dynamic Framed Slotted Aloha Algorithm
EAA	Enhanced Anti-collision Algorithm
EDFSA	Enhanced Dynamic Framed Slotted ALOHA Algorithm
FAP	Frequency Assignment Problem
FDMA	Frequency division Multiple Access
FPGA	Field-programmable gate array
FSA	Frame Slotted Aloha
HQT	Hybrid Query Tree
H2QT	Hybrid Hyper Query Tree
ISO	International Organization for Standardization
ICA	Independent component analysis
JDS	Anti-collision Based on Jumping and Dynamic Searching
PA	Pur Aloha
PS	Progressing Scanning
QTA	Query Tree Algorithm

QL	Querry learning
RFID	Radio Frequency Identification.
RTF	Reader Talk first
S-Aloha	Slotted Aloha
SDMA	Space Division Multiple Access
TDMA	Time Division Multiple Access
VLSI	Very-large-scale integration

Introduction générale

Introduction

Ces dernières décennies, les techniques de transmission de données ne cessent d'évoluer, récemment des systèmes complets et intelligents de communication sans fil ont été développés.

Actuellement, les étiquettes code-barres omniprésentes, peu couteuses, deviennent inadaptées pour un nombre croissant d'articles. La solution optimale serait le stockage des données dans une puce en silicium.

A cause des procédures utilisées pour le transfert de puissance et de données dans les systèmes d'identification sans contact, ces derniers sont appelés des systèmes RFID, pour Radio Frequency Identification.

La technologie RFID est l'une des technologies très répandues et faisant partie des technologies d'identification automatique, telles que la reconnaissance optique de caractères ou des codes barre classiques. Le principal objectif de la RFID est d'assurer l'identification d'objets ou d'individus, la traçabilité, la sécurisation dans des activités variées.

La technologie RFID utilisent des bandes de fréquences situées à : 125 kHz (bande BF, Basses Fréquences), 13,56 MHz (bande HF, Hautes Fréquences), 860-960 MHz (bande UHF, Ultra Hautes Fréquences), 2,45 GHz (bande micro-ondes) et 3,1-10,6 GHz (bande ULB, Ultra Large Bande, ou UWB, pour Ultra Wide Band). Un système RFID se compose en général d'un lecteur (interrogateur) et d'un identificateur (aussi appelé transpondeur ou tag) et d'un ordinateur. L'identificateur est fixé à l'article à identifier, et interrogé par un lecteur qui émet un signal radio et reçoit en retour les réponses des étiquettes qui se trouvent dans son champ d'action. L'alimentation électrique des étiquettes RFID se fait selon le principe d'induction électromagnétique. Afin de communiquer une information au lecteur, elles vont utiliser le principe de la modulation de charge en agissant sur les signaux incidents qu'elles reçoivent.

La technologie RFID a été considérée comme une technologie capable d'identifier simultanément de façon rapide et fiable un nombre très important d'objets. Les investigations doivent répondre aux exigences des industriels de systèmes RFID, pour une lecture à distance de produits dotés d'un tag, et qui consiste en la réalisation d'un système RFID capable d'identifier avec efficacité un nombre important de produits sans qu'il y ait d'interférences entre les signaux de réponses. Les travaux de recherche en cours sont confrontés, pour la phase de lecture du système RFID, à des problèmes de collision des signaux de communication en provenance des tags. A cet effet, de nombreux algorithmes d'anticollision d'étiquettes dans les deux familles stochastique et déterministe ont été ainsi développés et mis en service pour améliorer les performances du système RFID. L'objectif de notre étude consiste en l'étude de certains

algorithmes d'anticollision de nature aléatoire, telle que la famille des algorithmes d'Aloha, slotted Aloha, np-CSMA et la famille déterministe telle que BTA, QTA, CTTA, BS-QTA, BS-CTTA. Et pour cela, nous avons organisé notre mémoire en trois grands chapitres:

Dans le premier chapitre, nous avons dressé un état de l'art qui donne une idée très claire sur les différents algorithmes qui ont été établit pour résoudre les problèmes de collision dans les systèmes RFID. Nous allons voir comment ces algorithmes se divisent en trois grandes familles à savoir : les algorithmes déterministes, les algorithmes stochastiques et les algorithmes hybrides. Aussi, nous allons exposer quelques algorithmes d'anticollision, quelques méthodes de planification utilisant des algorithmes d'optimisations et certains algorithmes implémentés sur circuits FPGA ainsi que leurs techniques de simulation.

Le deuxième chapitre est consacré à la modélisation de l'environnement de la simulation sur ordinateur d'un système RFID, ainsi que les paramètres d'évaluations des protocoles sont discutés. Nous allons présenter la configuration fondamentale du programme de simulation des protocoles de la méthode de résolution stochastique et nous montrons aussi l'influence de certains paramètres sur les résultats de la simulation.

Les résultats obtenus et leurs interprétations, ainsi que l'explication des sous-programmes des protocoles feront l'objet du chapitre trois.

Enfin, la conclusion résumera les travaux exposés dans ce mémoire et présentera les principales perspectives de recherches sur les protocoles anticollision du système RFID et au niveau des protocoles MAC en général envisagées.

Chapitre 01

Etat de l'art des Algorithmes anticollision RFID

1.1. Introduction

Plusieurs applications, dans les systèmes RFID, exigent aux lecteurs RFID de fonctionner dans une proximité étroite (lorsque deux ou plusieurs lecteurs sont trop rapprochés). Les lecteurs placés physiquement les uns à côté des autres peuvent s'interférer entre eux. Une telle situation va engendrer la collision de lecteurs [1]. Cette contrainte est l'une des problématiques les plus rencontrées dans les réseaux de lecteurs RFID d'où l'obligation de la résoudre afin d'assurer d'une part le bon fonctionnement du système RFID et soutenir d'autre part l'efficacité du système du réseau entier [2].

Un autre type de collision peut se produire lorsque deux ou plusieurs étiquettes transmettent en même temps et au même lecteur (tag-tag collisions). Ce problème devient important, lorsque la densité des puces RFID qui doivent être identifiées est élevée. Par exemple, on rencontre ce genre de collision lors d'un inventaire dans un grand magasin où on trouve une forte densité de tags RFID placés sur les articles, engendrant ainsi l'augmentation du temps nécessaire pour établir le processus d'inventaire [3].

Pour remédier aux problèmes cités ci-dessus, de nombreux travaux ont été établit afin de développer des algorithmes anticollision parmi lesquels deux familles seulement ont été implémenté à savoir: les algorithmes stochastiques et les algorithmes déterministes [3]. Nous présentons dans ce qui suit les différentes solutions qui ont été proposé pour l'implémentation de ces algorithme afin d'achever les processus d'identification.

1.2. Méthodes de résolution déterministes

1.2.1. Algorithme (*Tree-Based)* ou *Recherche en arbre*

Les algorithmes Tree-Based, ou algorithme basés sur l'arbre, appartiennent à la classe des protocoles de résolution de collision. Le processus d'identification des tags dans ce cas est déterministe, il est basé sur l'interrogation itérative d'un sous-ensemble de tags, correspondant à la même propriété, jusqu'à ce que tous les tags soient identifiés. L'avantage dans cette méthode est que le lecteur crée des adresses *ad hoc* pour les tags et ce après la résolution de la collision, pour les utiliser lors de la commande lecture/écriture [4].

1.2.2. Algorithme BSA (*Binary Search Algorithm*)

Dans cet algorithme, un codage Manchester est employé afin d'identifier le bit où il y a une collision. Le lecteur peut savoir qu'une collision est produite dans la séquence reçue si les états du code ne changent pas dans certain bit comme le montre la Fig. 1.1 [5].

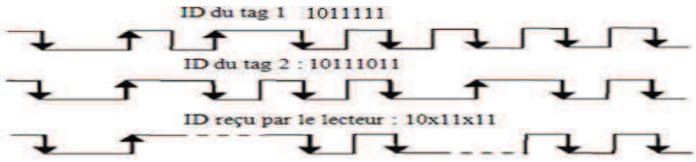


Fig. 1.1. Collisions dans l'identification reçue [5]

Dans cet algorithme le lecteur transmet un numéro de série aux tags de sa zone. Les tags comparent ce numéro de série avec leurs identifiants respectives. Les tags qui présentent un préfixe égal ou inférieur à ce numéro de série répondent au lecteur. Ensuite le lecteur gère les réponses des tags bit par bit en utilisant le codage de Manchester. Une fois la collision apparait, le lecteur divise les tags en deux sous-ensembles en se basant sur les bits endommagés [6].

Une autre performance a été ajoutée au protocole de l'algorithme BS d'où en résulte un nouvel algorithme : DBSA (*Dynamic BS Algorithm*). Le processus d'identification, dans cet algorithme, n'utilise pas la totalité des bits du numéro de série envoyé par le lecteur ainsi que les bits des identifiants des tags. Par exemple si le lecteur reçoit une séquence telle que 01X, les tags n'ont besoin que de transmettre les bits restants de leurs ID puisque le lecteur a identifié le préfixe 01[6].

1.2.3. Algorithme BTWA (*Binary Tree Working*)

Dans ce processus l'algorithme donne la priorité de choisir soit 0 ou 1 lorsque l'arbre se divise en nœuds. Cette priorité doit être gardée durant le procédé d'arbitrage. On peut résumer le déroulement de cet algorithme comme suit [7]:

Initialement le lecteur transmet une requête qui contient un préfixe de taille K bits, puis les étiquettes présentant les même K premiers bits envoient le $(K+1)^{ièmes}$ bit de leurs IDs au lecteur. En cas de collision dans les bits reçus, le préfixe se prolonge et le lecteur attache '0' ou '1' à l'ancien préfixe, et le nouveau préfixe est retransmis par le lecteur. D'autre part, si les bits reçus n'ont pas subi de collision, le bit reçu est attaché au préfixe pour le prochain préfixe envoyé. S'il n'y a pas de réponse la branche est ignorée et le lecteur passera à une autre branche et ce processus se répète jusqu'à l'identification de toutes les étiquettes.

1.2.4. Algorithme QTA (*Query Tree Algorithm*)

L'algorithme d'arbre de question (QTA) a le même principe que l'algorithme BTWA [8].

1.2.5. Algorithme CTTA (*Collision Tracking Tree Algorithm*)

L'algorithme d'arbre de cheminement de collision (CTTA) a le même principe que l'algorithme QTA sauf que cet arrangement utilise le cheminement de collision [8].

1.2.6. Algorithme à deux slot : BS-QTA & BS-CTTA (*Bi-Slotted*)

Ces deux algorithmes dérivent respectivement des deux algorithmes QTA et CTTA, seulement dans ce cas deux slot sont pris en considération pour chaque requête [9].

1.2.7. L'Algorithme Anticollision basé sur l'Arbitrage bit: (Bit Arbitration Based Anti-collision Algorithm)

Dans cet algorithme, toutes les étiquettes non identifiées dans la zone d'interrogation du lecteur sont activées par celui-ci. Toutes les étiquettes actives participeront au processus d'arbitrage et celles qui rentrent dans la zone en cour du déroulement du processus seront ignorées et ne participeront pas au processus d'arbitrage.

L'idée principale de cet algorithme se base sur la décomposition des étiquettes actives sollicitées par le lecteur, selon le bit envoyé par les tags, en deux groupes. Le lecteur choisira un groupe pour continuer une autre étape d'arbitrage de bit et puis pour s'enquérir du prochain bit jusqu'à ce que le code d'identification d'une étiquette soit connu.

La période d'identification d'une étiquette présente une fonction linéaire de la longueur du code d'identification. Ainsi s'il y a n étiquettes dans la zone d'interrogation du lecteur et la longueur du code de l'identification de chaque étiquette est m, le nombre total d'itérations est $m*n$ [10].

1.2.8. Algorithme JDS (*Anti-collision Based on Jumping and Dynamic Searching*)

L'algorithme JDS (*Anti-collision Based on Jumping and Dynamic Searching*) est une autre variante de l'algorithme BS proposée par [11]. La différence entre les deux est que dans le JDS le lecteur ne reprend pas le processus une fois les tags sont identifiés. Ainsi, à l'état initial, le lecteur transmet un ' 1 ' au lieu d'envoyer un numéro de série dont la totalité de ses bits sont égaux à '1' [11].

1.2.9. L'algorithme anticollision amélioré

L'algorithme d'anticollision proposé dans [12] est amélioré, comparativement à l'algorithme BSA et à l'algorithme JDS. Son but est d'augmenter la vitesse d'identification des étiquettes quand les collisions se produisent pendant la transmission [12].

Le point clé de cet algorithme est que toutes les étiquettes doivent répondre au lecteur en même temps, de sorte que les étiquettes puissent envoyer leurs codes d'identification au lecteur synchroniquement. C'est le seul cas où l'algorithme fonctionnera correctement.
Cet algorithme se différencie des deux algorithmes BSA et JDS comme suit:

- S'il y a seulement un seul bit en collision à n'importe quelle position dans l'identifiant, le lecteur peut automatiquement identifier deux étiquettes à la fois sans avoir besoin de renvoyer une nouvelle commande Request.
- L'algorithme sera employé si les collisions se produisent en succession. Après détection de collisions successives, chaque bit correspondant dans la commande Request sera placé à zéro. Différemment aux algorithmes existants mentionnés précédemment, la prochaine commande Request est directement égale à l'ancienne plus une. Le processus d'identification continuera jusqu'à ce que toutes les étiquettes dans la zone d'interrogation soient identifiées [12].

1.2.10. Algorithme ACPB (*Anti Collision using Parity Bit*)

L'algorithme d'anticollision basé sur arbre ACPB (Anti Collision algorithm using Parity Bit) utilise le bit de parité pour arbitrer la collision sans vérifier la totalité des bits. Il détermine l'identifiant d'étiquette en s'appuyant sur trois paramètres qui sont : le bit de parité, le nombre de bits en collision et le nombre de '1' dans l'identifiant d'étiquette (Fig.1.2). L'algorithme ACPB peut réduire le nombre de requêtes du lecteur, ainsi il diminue le temps d'identification des étiquettes dans la zone d'interrogation et fournit aux lecteurs une vitesse d'identification plus rapide [13].

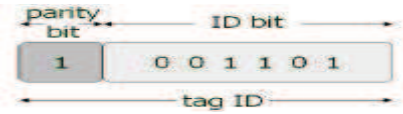


Fig.1.2. Structure de l'identifiant d'étiquette avec le bit de parité [13]

1.2.11. Algorithme EAA (*Enhanced Anti-collision Algorithm*)

L'algorithme EAA est une version améliorée des algorithmes basés sur l'arbre binaire dans lesquels une pile est employée pour maintenir la partie non traversée de l'arbre d'identification. Les compteurs sont initialisés à 0 adoptés dans les étiquettes indiquant le temps approprié de réponse [14].

Si une collision se produit, tous les compteurs des transpondeurs s'incrémentent, sauf ceux dans la catégorie choisie; sinon, si un succès survient, tous les compteurs se décrémente. Lorsqu'un compteur est égal à 0 il peut répondre dans ce cas à l'interrogateur.

Un tel algorithme résout les contraintes présentes dans les tags qui pourrait être causés par le nombre aléatoire des algorithmes basés sur ALOHA et ce en réduisant le nombre total de bits transmis. Cependant, cet algorithme présente certains inconvénients à savoir : le long délai d'identification, la dépendance de la performance de l'identifiant du transpondeur.

1.3. Méthodes de résolution stochastique (aléatoire)

Comme le réseau RF à diffusion implique un support unique pour n émetteurs/récepteurs (tags). L'une des solutions pour réaliser ces accès multiples des tags est l'accès aléatoire. Nous allons présenter ci-après les algorithmes appliqués dans ce cadre.

1.3.1. ALOHA pure

Les systèmes basés sur Aloha pure (PA) ont plusieurs variantes, on peut citer les plus importantes. La durée d'émission de la trame (paquet de données) est fonction inverse du débit sur le réseau. Le " débit utile " est égal au nombre de bits que le réseau peut transmettre par seconde sans collision : c'est une fraction du débit total. Pour calculer la durée d'un transfert de fichier il faut soustraire à ce débit utile l'" overhead " provoqué par le découpage du fichier en trames (paquets) ainsi que par l'adjonction au contenu utile des adresses et contrôles. De tout cela découle que le débit utilisable pour transporter du contenu n'est qu'une fraction du débit physique offert par le réseau.

1.3.1.1. PA avec silence

Lorsque le mode silencieux est utilisé, le nombre d'étiquettes dans la zone d'interrogation est réduit après chaque réponse réussie. Par conséquent, le silence a pour effet de réduire la charge délivrée au lecteur après chaque identification.

La Fig1.3 montre le processus de l'algorithme Aloha pure. Initialement, les étiquettes 1 et 3 rentrent en collision, elles vont attendre alors un temps aléatoire avant de retransmettre à nouveau. Après identification, le lecteur met en silence les étiquettes identifiées en utilisant la commande "mute" [6].

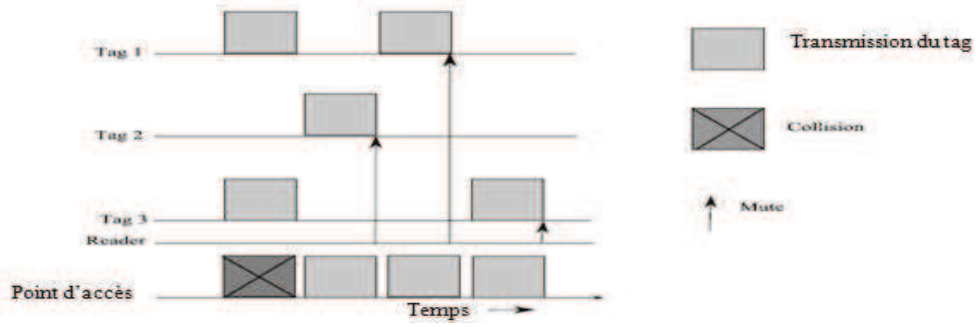


Fig.1. 3. Aloha avec silence [6].

1.3.1.2. PA avec le ralentissement

Dans ce cas, au lieu d'être mis en silence, l'étiquette peut être sollicitée en utilisant une commande "slow down" (ralentissement) afin de réduire son taux de transmissions, diminuant ainsi la probabilité de collision. La Fig.1.4 montre comment le lecteur ralentit l'étiquette 1 après identification, ayant pour résultat l'étiquette 1 adaptant son back-off aléatoire de façon à réduire son débit de transmission [6].

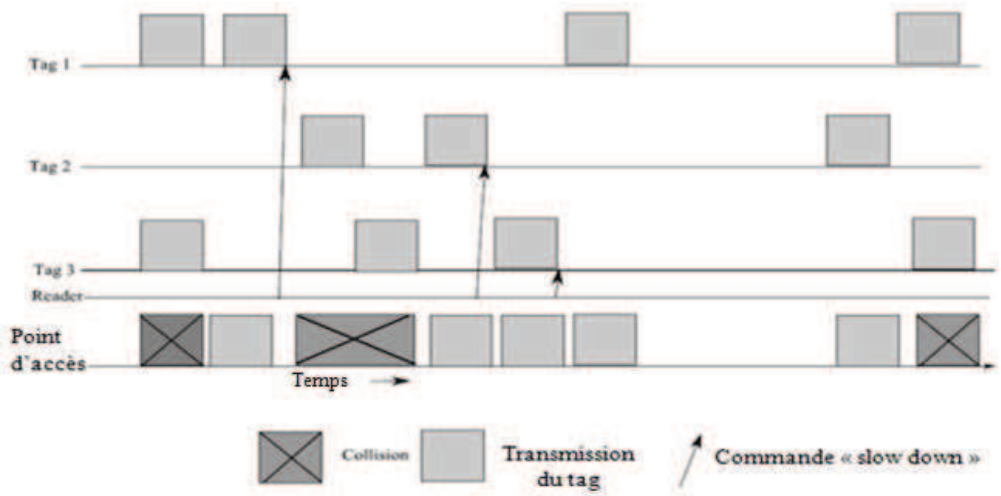


Fig. 1.4. Aloha avec le ralentissement [6].

1.3.1.3. PA avec le mode rapide

Cet algorithme est présenté à la Fig. 1.5 où une commande " silence " est envoyée par le lecteur une fois qu'il a détecté le début d'une transmission d'étiquette. Cette commande a pour effet d'arrêter la transmission d'autres étiquettes. Les étiquettes sont autorisées à transmettre à nouveau après que le lecteur envoi une commande ACK ou bien jusqu'à ce que leur temps d'attente expire [6].

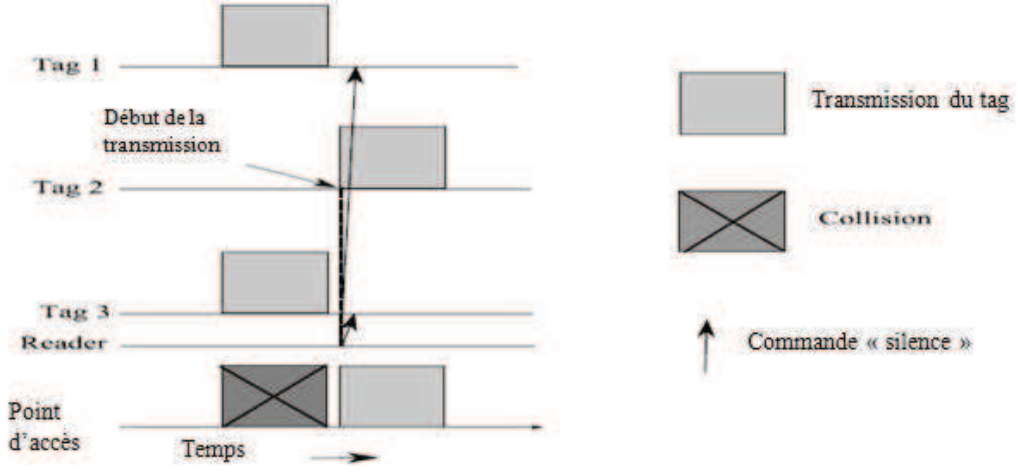


Fig. 1.5. Aloha avec le mode rapide [6].

Quand le nombre d'étiquettes dans la zone d'interrogation du lecteur augmente, la situation devient plus compliquée. Ainsi les chercheurs commencent à trouver de nouveaux moyens pour résoudre les performances instables de l'algorithme ALOHA et une variété d'arrangements ont été proposées.

1.3.2. Algorithme S-Aloha (*Slotted Aloha*)

Algorithme Slotted Aloha ou Aloha discrétisé est un processus d'anticollision stochastique basé également sur le TDMA. Dans ce cas, la trame est divisée en plusieurs slots et les transpondeurs doivent commencer à transmettre leurs paquets de données au début d'un slot synchrone [15]. En raison de la limitation du nombre des slots, cet algorithme est utilisé dans le cas où il y a une faible densité d'étiquettes dans la zone d'interrogation du lecteur [16].

1.3.3. Algorithme FSA (*Frame Slotted Aloha*)

Dans l'algorithme FSA, la trame est définie comme un intervalle de temps entre les demandes du lecteur et se compose d'un certain nombre de slots. Les étiquettes choisissent de façon aléatoire le slot de temps dans lequel ils vont répondre [17]. Avec les trames multiples le lecteur identifiera toutes les étiquettes qui se trouvent dans sa portée. Il existe trois types d'algorithmes anticollision FSA [16].

a) **Basic Framed Slotted Aloha Algorithm (BFSA):** Dans cet algorithme, le processus d'identification est géré par le lecteur, ce dernier a comme rôle d'offrir aux tags des informations sur la taille de la trame et le nombre aléatoire qui est utilisé pour sélectionner un slot dans une trame. Chaque tag sélectionne un numéro d'emplacement dans la trame pour y accéder en utilisant le nombre aléatoire et répond au numéro d'emplacement dans la trame.

   Comme la taille de la trame de l'algorithme BFSA est fixe, sa mise en œuvre est simple, cependant, son efficacité d'identification d'étiquette est faible.

b) **Dynamic Framed Slotted Aloha Algorithm (DFSA) :** Dans cet Algorithme la taille de la trame peut être changée par le lecteur. Par exemple, quand le nombre des slots produisant des collisions est plus grand que la limite supérieure, le lecteur augmentera la taille de la trame; quand le nombre de slots vides est plus grand que la limite supérieure, le lecteur diminuera la taille de la trame [17].

En raison des différentes méthodes utilisées pour modifier la taille de la trame (Annexe 1), l'algorithme DFSA a plusieurs versions. L'algorithme DFSA a plus d'avantages que les autres versions de l'algorithme ALOHA, ainsi il est largement plus répandu.

Lorsque le nombre de slots dans une trame est identique au nombre d'étiquettes dans la zone d'interrogation du lecteur, le système montre une meilleure performance en terme de temps et de sortie. Dans cette méthode la sortie de système; quand la taille de la trame est constante; peut être obtenu comme le montre la Fig. 1.6 **[**16**]**.

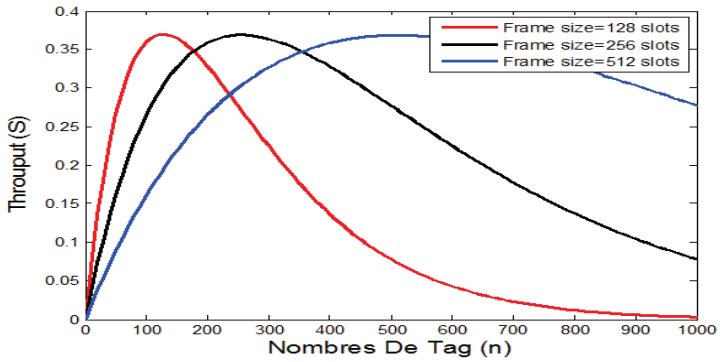


Fig. 1.6. Sortie S en fonction de nombre de tag(n) à plusieurs tailles de la trame (N) [16]

c) **Advanced Framed Slotted Aloha Algorithm (AFSA):** L'algorithme AFSA emploie une fonction d'évaluation du nombre d'étiquettes comme équation (1.1). Le nombre d'étiquettes est estimé en utilisant la taille de la trame (N) utilisée dans le cycle de lecture et les résultats du cycle de lecture précédent comme les nombres (c_0, c_1, c_k) qui mesurent respectivement les slots vides, les slots remplies d'une étiquette, et les slots en collision. Dans l'équation (1.1), ($a_0^{N,n}$,$a_1^{N,n}$, $a^{N,n} \geq 2$) sont respectivement le nombre prévu de slots vides, des slots remplies d'une étiquette, et des slots en collision où N et n indiquent respectivement la taille de la trame et le nombre d'étiquettes. La fonction d'évaluation du nombre d'étiquettes mesure la différence entre les vrais résultats et les valeurs prévues pour estimer le nombre d'étiquettes dont lequel la différence devient minimale [18-19].

$$\varepsilon_{vd}(N, c_0, c_1, c_k) = \min \left| \begin{pmatrix} a_0^{N,n} \\ a_1^{N,n} \\ a^{N,n} \geq 2 \end{pmatrix} - \begin{pmatrix} c_0 \\ c_1 \\ c_k \end{pmatrix} \right| \qquad (1.1)$$

Cette fonction d'estimation permet de réduire au minimum la probabilité de collision et rendre cet algorithme plus efficace par rapport aux autres. Cependant, l'algorithme AFSA a le même problème que l'algorithme Aloha, il ne peut pas augmenter la taille de la trame indéfiniment lorsque le nombre d'étiquettes augmente. Ainsi, cet algorithme fonctionne bien si le nombre d'étiquettes est relativement petit, cependant, si le nombre devient grand il commence à montrer de faible performance [18-19].

1.3.4. Enhanced Dynamic Framed Slotted ALOHA Algorithm (EDFSA)

Si la densité de tags dans la zone d'interrogation d'un lecteur augmente, la probabilité de collision d'étiquettes augmente rapidement. Ce problème peut être résolu lorsqu'on limite le nombre d'étiquettes répondant approximativement les mêmes que la taille de la trame. L'algorithme EDFSA est présenté pour résoudre ce problème [17].

- **Description de l'algorithme EDFSA :** En général, nous ne pouvons pas augmenter la taille de la trame indéfiniment. Donc pour réaliser une efficacité élevée du système quand le nombre d'étiquettes non lues est trop grand, nous devrons limiter le nombre d'étiquettes non lues en divisant ce nombre en groupes de sorte que le nombre optimal d'étiquettes dans ce groupe répond à la taille donnée de la trame.

Dans chaque cycle de lecture, le lecteur estime le nombre d'étiquettes non lues et calcule le nombre de groupes qui donnent la sortie maximale pendant le prochain cycle de lecture. Après estimation du nombre d'étiquettes, si le lecteur trouve que le nombre d'étiquettes, comparé avec la taille maximale donnée de la trame est beaucoup plus grand que celui qui donne l'efficacité optimale du système, il divise les étiquettes non lues en un certain nombre de groupes et permet à seulement un groupe de répondre.

Quand le nombre d'étiquettes estimées non lues est au-dessous d'un certain seuil, le lecteur ajuste la taille de la trame sans grouper les étiquettes non lues, et dans ce cas la fonction d'estimation de l'équation (1.1) est utilisée.

Après chaque cycle de lecture, le lecteur estime le nombre d'étiquettes non lues et ajuste la taille de la trame et le processus se répète jusqu'à ce que toutes les étiquettes soient identifiées [17].

1.3.5. Algorithme Q protocol

Cet algorithme, basé sur l'algorithme DFSA, est appliqué dans le standard EPC class 1 Gen 2; la fonction d'estimation d'étiquettes proposée exige au lecteur d'incrémenter et de décrémenter avec une constante la taille de la trame. Un lecteur diffuse au début une commande contenant un compteur de slot Q la taille de la trame dans ce cas est égale à 2^Q où Q est un nombre entier entre zéro et huit. Les étiquettes choisissent un slot aléatoirement de 0 à 2^Q-1. Pour chaque collision ou slot vide, le lecteur incrémente ou décrémente Q (selon l'évènement) par une constante c où, $0.1 \leq c \leq 0.5$. Les slots avec une seule réponse ne changent pas. La valeur résultante de Q est alors employée pour déterminer la taille de la trame du prochain cycle [20].

1.4. Algorithmes de résolution hybrides

1.4.1. Algorithme HQT (*Hybrid Query Tree*)

Contrairement à l'algorithme QT qui présente plusieurs bits et plusieurs collisions lors du processus d'interrogation, l'algorithme HQT emploi la technique de recherche d'arbre 4-aires qui permet d'augmenter le préfixe par deux bits à la fois dans la durée d'un bit engendrant ainsi la diminution des cycles de collision entre les étiquettes mais la contrainte dans ce cas est l'augmentation des cycles vide. Pour pallier à cette dernière, l'algorithme HQT adopte une autre technique qui consiste à retarder les slots contenant les étiquettes transférant leurs données après un certain temps.

Toutefois, il est toujours impossible de détecter certains slots qui n'ont pas de réponses transmises, car la méthode proposée peut vérifier seulement l'état occupé [21].

1.4.2. Algorithme H2QT (*Hybrid Hyper Query Tree*)

Cet algorithme est illustré à la Fig.1.7 où le lecteur diffuse un préfixe k, avec lequel il exige aux étiquettes de répondre selon le nombre de '1' contenus dans leurs identifiants après le $k^{iéme}$ bit, donc si ces bits ne contiennent que des '0' le tag correspondant répond au slot '0' et si ces bits contiennent un seule '1' le tag correspondant répond au slot '1' et ainsi de suite. Contrairement aux algorithmes 4-aire et QT, le H2QTne génère pas le cycle d'inactivité, et réduit le temps de reconnaissance en réduisant le nombre de collisions et de requêtes.

Aussi, l'évaluation des performances par la simulation a démontré que l'algorithme H2QT présente une meilleure performance en termes du nombre de requêtes puisqu'il réduit le nombre de requêtes de 3 à 4 fois davantage par rapport aux autres algorithmes basé sur l'arbre [21].

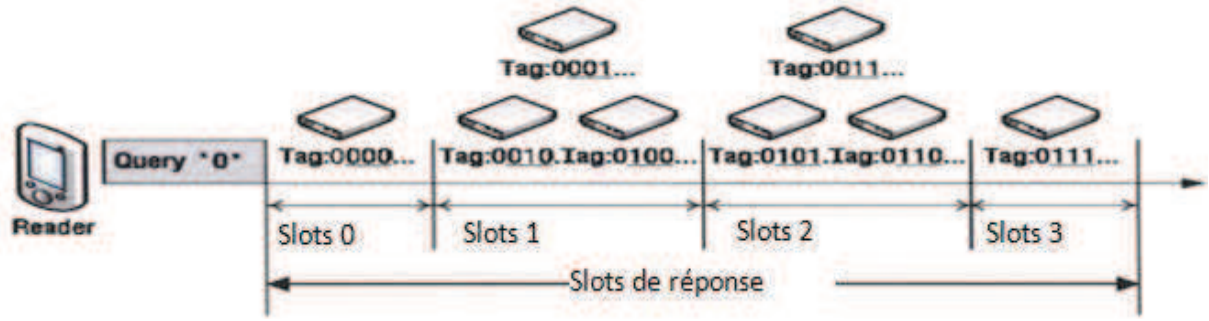


Fig. 1.7 Réponse d'algorithme H2QT [21]

1.4.3. L'algorithme de balayage progressé PS (*Progressing Scanning*)

C'est une variation hybride basée sur l'algorithme FSA utilisant une technique de balayage progressée qui permet d'améliorer les performances du FSA quand le nombre d'étiquettes dans le secteur est trop élevé (>1000) en divisant les étiquettes en groupes et en

donnant chaque fois la possibilité au lecteur de communiquer avec un groupe en les traitant individuellement. Il peut ainsi fournir un degré élevé d'intégrité de données dans le système RFID, même avec l'utilisation de petites tailles de trame [3].

Les paramètres qui commandent les performances de l'algorithme PS sont le niveau minimum de puissance transmise par le lecteur, la taille de la trame et la taille du pas d'augmentation de la puissance dans chaque cycle [22].

Pour l'analyse de performances, l'algorithme PS a été évalué avec les paramètres d'un système RFID passif typique à 2,45 GHz. Les résultats de simulation montrent que l'algorithme PS améliore l'efficacité du système RFID et fourni une solution fiable pour des cas à forte densité d'étiquettes (plus de 800 étiquettes), Ainsi l'algorithme PS présente de meilleurs résultats quand un plus grand pas est utilisé comme le montre les Fig.1.8 et Fig.1.9 [22].

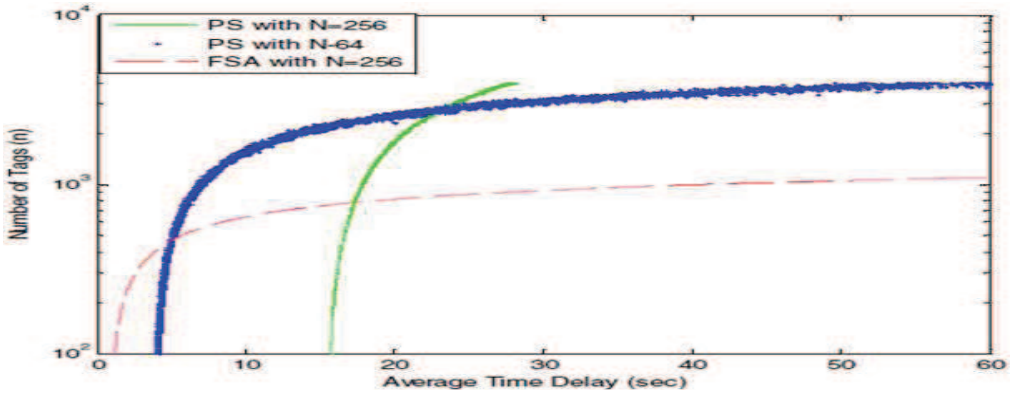


Fig. 1.8. Comparaison de délai entre PS et le FSA [22]

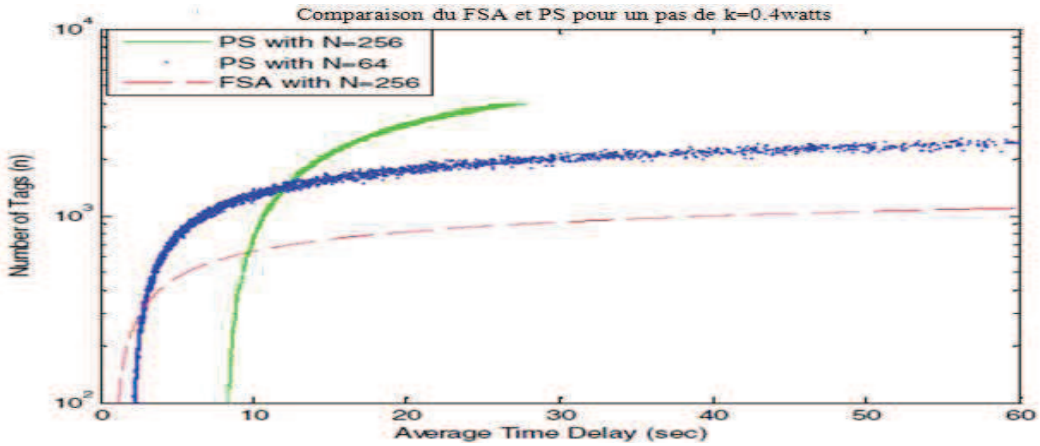


Fig. 1.9. Comparaison de délai entre le PS et le FSA avec un pas de valeur plus grande [22]

1.4.4. Algorithme d'anticollision basé sur des antennes intelligentes dans le système RFID :

L'Antenne intelligente est une nouvelle conception d'antenne radio logicielle, il combine les technologies de Formation de faisceau Numérique (Digital Beamforming) (DBF) et la radio logicielle. La technologie DBF peut rapporter des faisceaux multiples

simultanément disponibles qui peuvent être réalisés pour avoir un gain élevé avec faibles lobes latéraux [23].

Cet algorithme est basé sur le principe de la technologie SDMA, la zone d'interrogation d'un lecteur peut être divisée en plusieurs sous-espaces. En utilisant la technologie de transmission numérique de faisceaux, le lecteur peut contrôler chaque direction du maximum du faisceau de l'antenne et activer plusieurs étiquettes dans chaque direction. Ensuite, le lecteur reçoit les signaux de retours et les traites afin de séparer et lire les identifiants des étiquettes [23].

Comme les étiquettes sont séparées dans l'espace, nous pouvons distribuer le même slot et la même fréquence aux tags dans différents sous-espaces sans produire de collision entre eux. Par conséquent, l'efficacité et le temps d'identification du système seront améliorés [23].

1.5. Techniques d'accès

1.5.1. L'algorithme CSMA (*Carrier Sense Multiple Access*)

L'accès multiple avec détection de porteuse (CSMA) est une méthode d'anticollision qui permet de réduire le nombre de conflits par rapport au protocole Aloha et ses dérivés. Selon le type de décision prise lorsque le canal est détecté occupé on distingue trois types de protocole CSMA : CSMA non-persistant, CSMA persistant, CSMA p-persistant [24].

1.5.2. L'algorithme DCMA (Dual Channel Multiple Access)

Cet algorithme est utilisé pour des systèmes sans fonction de contrôle physique caractérisés par une transmission de paquets longs. Le premier avantage du DCMA est que les clients sont en mode power-down pendant la période Backoff, ce qui réduit la consommation de puissance. Le second avantage est que le nouveau tag ajouté au système ne subit pas de collision avec le paquet de donnée [25].

Ce protocole dispose deux canaux de fréquence qui sont identifiés comme canal de données et canal de commande. Ceci offre une possibilité de transmettre séparément les signaux de surveillance virtuels tels que RTS/CTS et les signaux de données tels que les données et le signal ACK.

Dans certains cas, il y a des tags qui envoient une commande RTS sans être sollicité mais heureusement ce RTS ne causera pas de collision avec les données transmises, car ces

dernières sont transmises dans le canal 2 contrairement au RTS qui est supporté par le canal 1 [25].

1.6. Implémentation matérielle

1.6.1. Algorithmes implémentés sur FPGA

L'inconvénient majeur lors de l'implémentation des algorithmes RFID, reste la complexité du calcul qui est insupportable par l'étiquette et le lecteur RFID vue la faible capacité du calcul dans ces derniers et ceci devient important quand plusieurs étiquettes arrivent en même temps. Motivé par ceci, on propose de combiner des techniques logicielles et une implémentation matérielle pour réaliser un algorithme anticollision plus robuste [26].

a) Le protocole anticollision *binary tree-like* dans le cas d'un simulateur FPGA

Le travail proposé dans [26] contribue principalement à l'évaluation des performances de l'algorithme anticollision en cas de l'utilisation d'un simulateur FPGA [26].

Les performances de l'algorithme anticollision sont évaluées par rapport au nombre maximum d'étiquettes identifiées par cycle de lecture, la vitesse d'identification et l'efficacité d'algorithme. La méthodologie ainsi que le résultat sont tous les deux utiles pour concevoir, montrer et implémenter un système RFID pratique et donc réel.

Le résultat indique que cette stratégie de combinaison des techniques logicielles et leur implémentation matérielle présente de meilleures performances et réalise un calcul plus rapide avec un faible coût [26].

b) Algorithme de communication Q d'un lecteur RFID basé sur le processeur NIOSII

Dans le but de concevoir et implémenter un algorithme de communication du protocole Q, une conception de la bande de base du lecteur RFID basée sur une nouvelle structure est proposée [27].

Cette nouvelle structure est mise en œuvre dans le processeur NIOSII au niveau de la bande de base du lecteur FPGA. Le système conçu dont le lecteur peut communiquer avec plusieurs étiquettes est compatible avec le Standard EPC Classe 1, Génération2, fonctionnant en bande 915MHz. Dans la structure NIOSII, la plupart des fonctionnalités du lecteur RFID sont mises en œuvre dans la puce NIOSII à l'exception du codage/décodage et de la modulation / démodulation.

Cette simulation montre que cette nouvelle structure proposée peut communiquer avec plusieurs étiquettes tout en assurant correctement un fonctionnement du codage, décodage ainsi que le déroulement de l'algorithme de communication [27].

c) Algorithme anticollision pour un système RFID actif pour la collection rapide d'étiquettes

Cette conception se concentre sur la collection d'étiquette dans les systèmes RFID actifs respectant la norme ISO/IEC 18000-7. En modifiant l'algorithme de cette norme un lecteur actif et des étiquettes RFID ont été développés [28].

L'algorithme modifié admet que le lecteur peut choisir la taille optimale du slot pour recevoir une réponse d'étiquette selon les capacités de son processeur.

Comme le montre la Fig.1.10, le lecteur comprend un module soft et un module RF et leurs processus s'exécutent en parallèle. En utilisant l'algorithme modifié le lecteur peut réduire au maximum la taille du slot [28].

Fig.1.10. Architecture proposée du lecteur [28].

Les résultats expérimentaux, avec un seul lecteur et 30 étiquettes dans un environnement réel, ont prouvé que ce système RFID actif utilisant l'algorithme de collection d'étiquette de la norme ISO/IEC 18000-7 est performant, aussi l'algorithme modifié a montré de meilleures performances en collection d'étiquettes que l'algorithme standard tout en respectant la norme ISO/IEC 18000-7 [28].

1.6.2. Conception VLSI d'un protocole d'Anticollision pour des étiquettes RFID

L'amélioration des performances de l'anticollision et la réduction du coût de production du circuit sont des considérations importantes dans beaucoup d'applications. En raison de soutenir ces deux considérations, un algorithme d'anticollision de tag basé sur le principe de l'arbre a été proposé dans [29] pour un système RFID. L'implémentation de cet algorithme sur FPGA est ainsi réalisée en utilisant un logiciel de simulation, puis une analyse pour la communication à plusieurs accès dans le système RFID a été faite [29].

La méthodologie de conception et les résultats ont aidés à concevoir et mettre en œuvre un système RFID pratique. Les circuits logiques sont développés et finalisés, puis

implémentés par la suite dans des circuits intégrés à très grande échelle (VLSI) d'où le nom de cette technologie.

La conception du système est réalisée en utilisant le Complementary Metal Oxide Semiconductor (CMOS) et l'extraction de l'arrangement final du protocole d'anticollision utilise la technologie VLSI et est développée en exigeant une réponse plus rapide entre le lecteur et l'étiquette, en améliorant ainsi les fonctionnalités et le coût du circuit d'anticollision [29].

1.7. Autres solutions proposées pour l'anticollision

1.7.1. Système d'anticollision RFID utilisant la technique d'étalement du spectre

Les données à transmettre sont multipliées par une séquence pseudo-aléatoire de débit beaucoup plus élevé que le débit des données. Chaque étiquette contient une séquence unique. La sortie haute fréquence de cette multiplication est utilisée pour basculer une diode afin de renvoyer à l'interrogateur une forme d'onde de haut débit [30].

L'interrogateur reçoit une combinaison de plusieurs ondes haute fréquence qui arrivent en même temps. La clé du processus de récupération se trouve dans la séquence pseudo-aléatoire unique qui a été combiné avec les données de l'étiquette désirée. Si l'interrogateur connaît la séquence pseudo-aléatoire associé à l'information d'étiquette désirée, il peut récupérer les données. Pour ce faire, théoriquement, tout ce qui doit être fait est de multiplier le signal entrant par la même séquence pseudo-aléatoire qui est combinée avec les données souhaitées. Lorsque cette séquence pseudo-aléatoire unique est multipliée avec les autres ondes comportant la forme d'onde entrante reçue par l'interrogateur, d'autres ondes hautes fréquences sont créées. Par conséquent la seule onde produite de basse fréquence est celle qui porte les données originales désirées [30].

Une fois la multiplication est terminée, les données originales peuvent être récupérées en faisant passer l'onde résultante à travers un filtre passe-bas. Toutes les ondes à haute fréquence générés par autres étiquettes seront supprimées [30].

1.7.2. Analyse par composante indépendante (ICA) combinée avec la technologie FDMA

L'analyse par composante indépendante (ICA) combinée avec la technologie FDMA est proposée pour améliorer les capacités d'identification d'un système RFID. Cette méthode a été vérifiée par simulation sur ordinateur qui prouve qu'elle est performante comparée à l'algorithme traditionnel d'Aloha, notamment lors de l'augmentation considérables du nombre d'utilisateurs [31].

Dans cette technique les signaux reçus avec le bruit sont séparés correctement, et les étiquettes peuvent être également ajustées et détectées d'une manière adaptative. En raison de sa généralité le modèle d'ICA possède différentes applications dans différents secteurs.

L'algorithme ICA. D'après la Fig. 1.11, on pourrait voir que la performance du système RFID basée sur la technologie ICA est meilleure par rapport à celle de la technologie d'Aloha, particulièrement quand le nombre d'utilisateurs augmente, la sortie du système RFID basé sur l'algorithme d'Aloha sera près de zéro, toutefois la sortie du système basé sur la technologie ICA présente toujours une bonne performance [31].

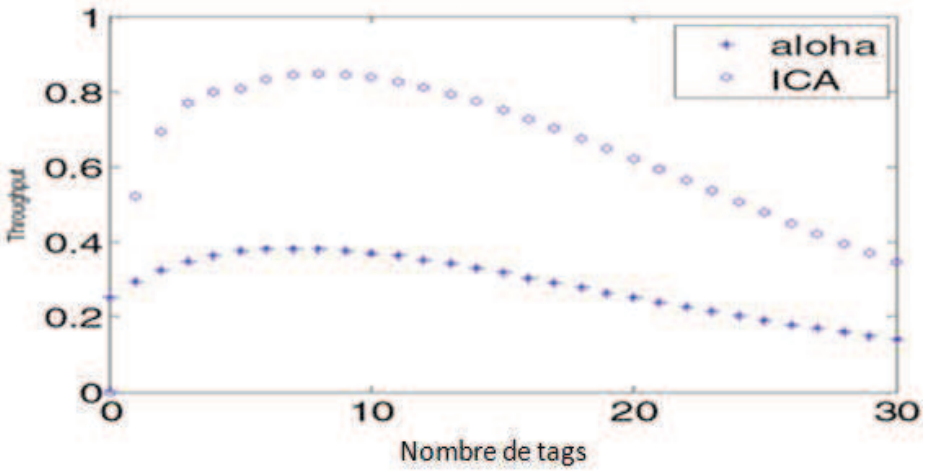


Fig. 1.11. La sortie de système RFID basée sur l'algorithme d'Aloha et l'ICA en fonction du nombre de transpondeurs [31]

1.8. Méthode de planification dans les systèmes RFID

L'identification par fréquence radio (RFID) possède plusieurs applications et la planification de réseau RFID sans fil est l'une des questions les plus importantes dans ce sujet. C'est pour cela que les systèmes RFID ont pour vocation de gérer la planification des réseaux sans-fil et ce tout en prenant en considération les aspects suivants : le choix adéquat de la position d'antenne du lecteur, assurer des communications performantes et efficaces et contrôler les coûts d'équipement dans une marge acceptable [32].

Par conséquent, de raisonnables réseaux doivent être implémentés afin de satisfaire les exigences citées ci-dessus. Les méthodes traditionnelles d'optimisation ne permettent pas d'implémenter efficacement de tels réseaux d'où la nécessité de nouvelles méthodes de planification de réseau RFID basées sur de puissants algorithmes dont nous allons citer deux parmi elles [32].

1.8.1. Planification basée sur l'algorithme génétique

La planification des réseaux RFID consiste à assurer une communication fiable entre les tags dans un espace tridimensionnel qui est généralement simplifié en deux plans égaux. L'un est celui contenant le lecteur et le second est celui des tags [32].

Dans la référence [32] une planification basée sur un algorithme génétique de gestion de réseau RFID a été proposée, c'est une solution d'optimisation multi-objective. Cette solution trace les issues du système RFID dans les algorithmes génétiques, produit de la présentation du gène et du chromosome, et met en application le mécanisme du choix individuel et de l'opération génétique. Cette méthode élimine non seulement le défaut de recherche de la méthode d'optimisation multi objective traditionnelle, mais fournit également une solution faisable pour la planification sans fil dans la gestion du réseau RFID [32].

1.8.2. Planification de réseaux RFID en utilisant des algorithmes évolutionnaires et intelligence d'essaim

Afin d'obtenir une planification du réseau précise et fiable dans le système de communications RFID, les endroits des lecteurs et les valeurs associées à leurs paramètres doivent être déterminés.

Dans la référence [33], le problème de planification de réseau RFID a été examiné comme problème d'optimisation multi-objectif, où la portée, l'interférence, l'efficacité économique et l'équilibre de charge dans le réseau sont en particulier considérés comme exigences fondamentales du système RFID. Ainsi, un développement d'un modèle mathématique pour la planification des réseaux RFID basés sur l'application de deux techniques d'optimisation puissantes à savoir les algorithmes évolutionnaires (EAs) et l'intelligence d'essaim (SI) a été élaboré. En outre, une étude comparative entre les trois algorithmes a été présentée, à savoir l'algorithme génétique (GA), stratégies d'évolution (ES) et l'optimisation d'essaim de particules (PSO) [33].

Les résultats de simulation prouvent que l'algorithme PSO obtient la meilleure solution pour le problème par rapport aux deux autres en termes d'exactitude d'optimisation et robustesse de calcul [33].

1.9. Algorithmes d'anticollision des lecteurs

1.9.1. Algorithme basé sur QL (*querry learning*)

L'algorithme proposé pour résoudre le problème de collision du lecteur est basé sur un algorithme *Q-learning* qui a pour but d'éviter les problèmes relatifs à l'assignement de fréquence [34].

Les problèmes d'assignement de fréquence (FAP) est la tâche qui désigne d'une façon optimale l'allocation des fréquences radio aux liaisons dans un réseau.

Eviter la collision de lecteur consiste à assigner correctement des fréquences radio aux étiquettes RFID de telle sorte que l'interférence entre les lecteurs est fortement réduite. Pour réaliser la performance optimale du lecteur, un algorithme doit assigner les fréquences limitées avec le temps [35].

L'algorithme proposé est capable d'assigner efficacement des ressources de temps et de fréquence aux lecteurs sans connaissance globale de leurs contraintes en mesurant les performances d'assignement passé. Une implémentation réussie et une analyse expérimentale de performance de l'algorithme ont été fournies [35].

1.9.2. Algorithme d'Anticollision *Colorwave*

L'algorithme *Colorwave*, est conçu pour un réseau dense des dispositifs de nœud tels que les lecteurs d'étiquette RFID, il offre une solution au problème de collision de lecteur qui est près de la solution statique optimale. La nature dynamique de l'algorithme basée sur l'information locale permet au système RFID de s'adapter automatiquement aux changements dans le système et dans son environnement de fonctionnement [35].

Le partage global de communication et d'informations n'est pas exigé. Ainsi, le *Colorwave* permet au système RFID de s'adapter facilement aux perturbations locales, telles que l'installation d'un nouveau lecteur RFID ou la présence d'un lecteur RFID mobile. Le *Colorwave* montre une meilleure performance en termes de densité de nœud et de fréquence de communications par rapport aux algorithmes d'anticollision de backoff [35].

1.9.3. Algorithme d'anticollision utilisant un serveur et des lecteurs mobiles basés sur l'accès multiple sans conflits

Les algorithmes existants d'anticollision RFID de *DCS* et de *colorwave* peuvent ne pas montrer la performance souhaitable dans des environnements avec un nombre dynamiquement variable des lecteurs voisins. Un algorithme d'anticollision de lecteur simple RFID proposé *CFMA* (*conflict-free multiple access*) emploie l'information d'endroit reçue à

partir des lecteurs. Le serveur exploite cette information et décide rapidement si un lecteur peut être utilisé sans collisions avec des lecteurs voisins ou non et informe les lecteurs de sa décision. Ces mécanismes d'aides permettent aux lecteurs d'éviter des collisions de lecteur efficacement [36].

1.9.4. Nouveau type d'algorithme anticollision de lecteur RFID

Sur la base de recherche des algorithmes d'anticollision de lecteur existants, comme l'algorithme *Color-wave*, l'algorithme *HIQ* et l'algorithme d'impulsion, etc.., un nouveau type d'algorithme d'anticollision a été proposé [37].

Principe de l'algorithme proposé : l'algorithme proposé possède les exigences suivantes :

- Le canal est divisé en deux parties: canal de contrôle et canal de données. Le canal de données est dédié pour des communications entre les étiquettes et les lecteurs et le canal de commande est consacré pour les communications entre les lecteurs; il n'y a aucune interférence entre le canal de données et le canal de commande.
- Les lecteurs peuvent recevoir les données du canal de données et du canal de commande en même temps et ils peuvent également envoyer les données au canal de commande et au canal de données.
- Le mouvement des lecteurs est sans délimitation, ce qui signifie que les lecteurs peuvent se déplacer librement dans le réseau entier.

Cet algorithme possède des caractéristiques telles que la commande simple de mis en application, qui le rendre très adapté pour le réseau de lecteur mobile [37].

Conclusion

Dans ce chapitre, un état de l'art a été dressé et a donné une idée très claire sur les différents algorithmes qui ont été établit pour résoudre les problèmes de collision dans les systèmes RFID, des implémentations matérielles d'algorithmes et planification des réseaux RFID.

Nous avons vue comment ces algorithmes se sont classés en trois grandes familles à savoir : les algorithmes déterministes, les algorithmes stochastiques et les algorithmes hybrides. La première catégorie renferme une diversité d'algorithme présentant chacun des avantages et des inconvénients. La seconde est présentée principalement par l'algorithme ALOHA ainsi que ses dérivées. Quant à la troisième, elle a démontré de performants algorithmes comparée aux deux premières.

Nous avons aussi exposés quelque algorithmes d'évitement de collision et quelque méthodes de planification utilisant des algorithmes d'optimisations et nous avons aussi exposés quelque algorithmes implémentés sur circuits FPGA ainsi que leurs techniques de simulation afin d'évaluer leurs performances pour pouvoir enfin les réaliser.

Dans le chapitre qui suit nous allons aborder l'environnement de la simulation du système RFID et discuter du principe général de chaque algorithme anticollision étudié.

Chapitre 02

Environnement de la simulation d'un système RFID

2.1. Introduction

En vue d'établir une estimation réelle des aspects temporels et performances des protocoles d'anticollisions déterministes et aléatoires d'un système RFID en utilisant la simulation sur ordinateur, il est nécessaire de définir un modèle de référence qui spécifie les exigences du temps de communication lecteur-tag et tag-lecteur, définir les exigences physiques et logiques pour une rétrodiffusion passive. Dans ce qui suit, la description des différents blocs d'un système RFID est présentée.

2.2. Présentation du modèle général de temps de transmission pour les protocoles d'anticollision des réseaux RFID

Le modèle de transmission de référence, a été calculé par la norme EPCglobal classe 1, Gen2 qui définit les exigences physiques et logiques pour une rétrodiffusion passive, un système d'identification RFID qui utilise le protocole RTF (interrogateur parle-premier), basée sur la trame ALOHA.

Nous pouvons tirer depuis cette norme, deux aspects importants qui doivent être considérés lors de l'évaluation du temps de transmission [3].

- D'abords, les performances physiques doivent être considérée, tel que la transmission Reader-tag (R => T) et Tag-Reader(T => R) lesquelles commencent par un préambule (champ dans la trame). Le préambule (R => T) est transmis seulement dans la première requête émise au début d'une trame, son effet est négligeable. cependant, le préambule (T => R), envoyé au début de chaque transmission du tag, a un impact important sur les performances du protocole. La taille de ce préambule dépend de l'encodage des données.
- Le deuxième aspect à prendre en compte est la synchronisation de lien:

Il s'agit d'estimer la durée du temps de propagation, le temps de transmission et le temps de réaction des tags. Le délai de propagation dans les deux directions est un temps estimé à 1 /30µs, pour une distance de référence de 10 m, mais il peut varier en fonction du débit de transmission de données et de la quantité des bits à transmettre. Selon la spécification EPC, le temps de réaction dépend des caractéristiques de l'appareil. Du côté étiquette, il est estimé que [R1 = 10/ (débit de donnée)], tandis que du côté lecteur, il est estimé que [R2 = 1/ (débit de donnée)] [3].

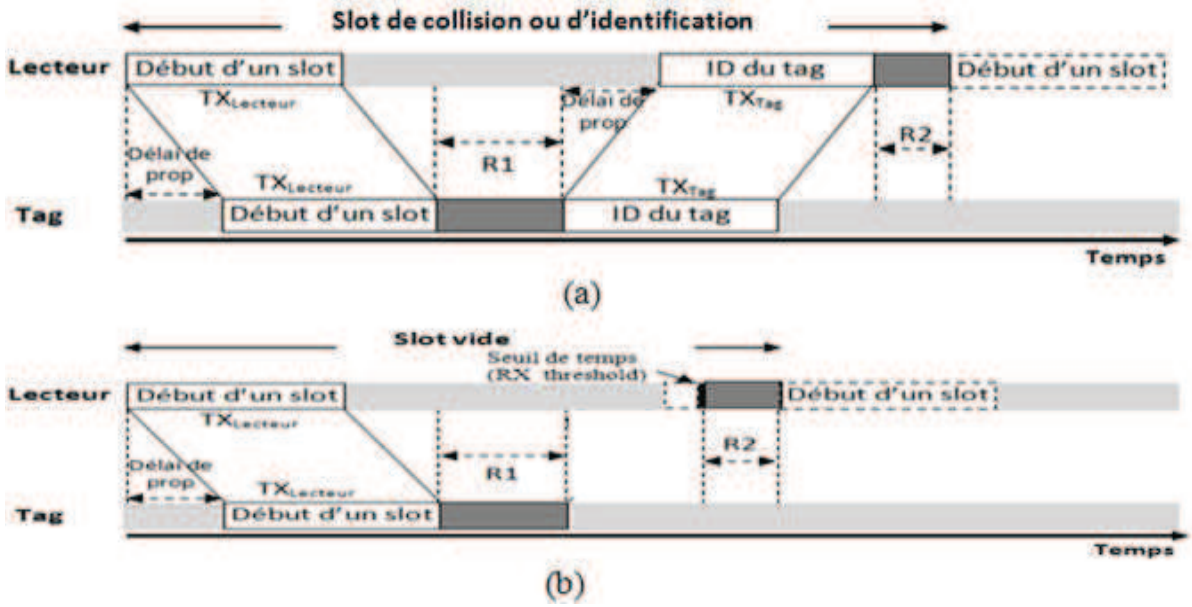


Fig.2.1. Synchronisation de lien pour la communication lecteur-étiquette et étiquette-lecteur pendant :(a) slot de collision ou d'identification, (b) slot vide [3]

La Fig.2.1 montre la synchronisation de lien pour un échange de messages entre un lecteur et une étiquette. Dans chaque slot, le lecteur commence en transmettant un début de slot pour les tags. Une fois le temps de transmission, *TX_lecteur*, et le délai de propagation s'écoulent le message est reçu par les tags. Après réception du message, le tag prend un temps R1 pour réagir et répondre. Avant d'être en mesure de renvoyer un nouveau slot, le lecteur patiente un temps R2. Si aucun tag ne répond au lecteur, cela veut dire qu'un résultat slot vide apparait, comme le montre la Fig.2.1(b).Après un seuil de temps RX (*RX_threshold*) le lecteur comprend qu'aucune transmission n'est de retour. Dans ce cas, le slot se termine lorsque le seuil RX s'écoule, et le lecteur émet un nouveau début de slot après un temps de réaction R2.
Un modèle similaire pour les protocoles basés sur l'arbre (*Tree-based*), dans lequel le message envoyé par le lecteur est une requête, et les étiquettes répliquent avec leurs identifiants [3].

Les points forts de ce modèle est que la durée d'un slot dépend strictement de la quantité de bits transmis par le lecteur (dans le slot de signalisation ou le message de requête) et par le tag (dans le message de réponse).

2.3. Modélisation d'un système RFID de communication par paquet pour la simulation des protocoles d'anticollision

Nous nous basons sur le modèle présenté précédemment, nous avons pris deux modèles similaires pour la simulation des protocoles dans les deux familles (aléatoires et déterministes) comme nous allons voir dans ce qui suit.

2.3.1. Méthodes stochastiques

2.3.1.1. Configuration entière d'un système RFID

Quant à la configuration entière de la simulation du système RFID de communication par paquet, chaque tag accède à un point d'accès (Lecteur) en utilisant un protocole d'accès, comme représenté sur la Fig.2.2.

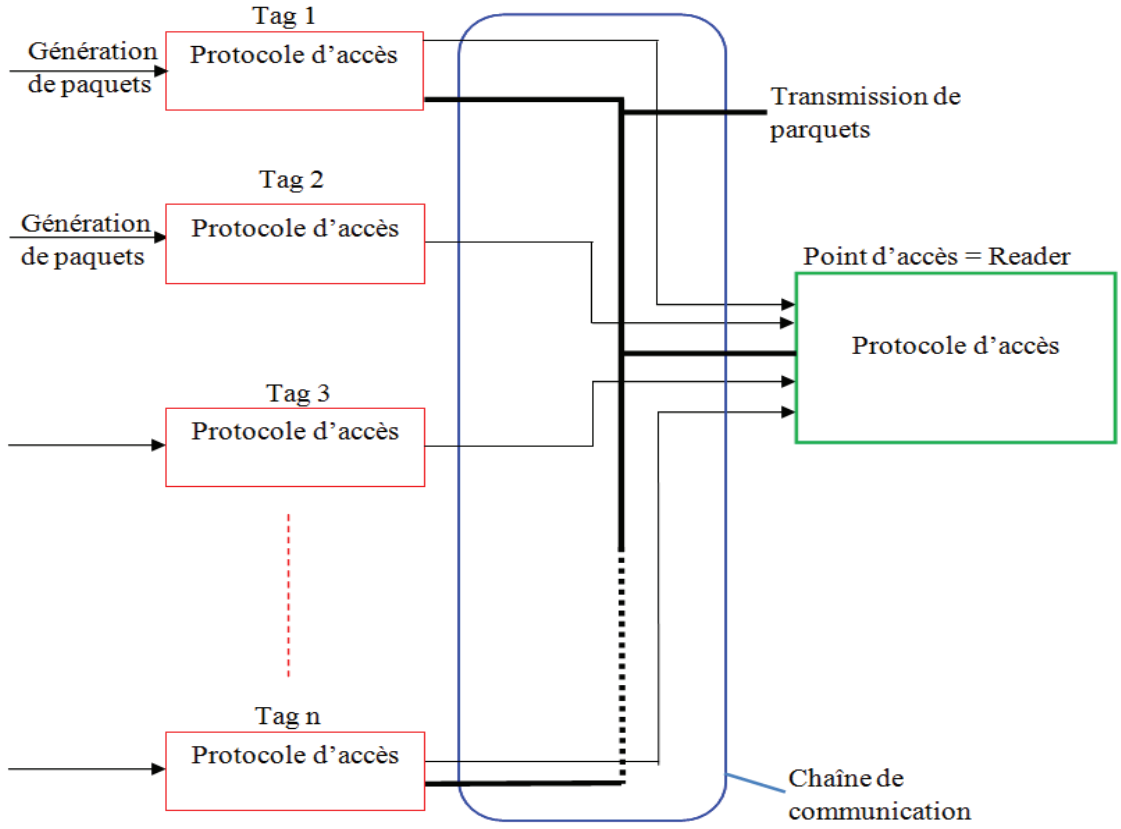


Fig.2.2. Configuration d'un système RFID de communication par paquet

a) Canal de transmission

La différence entre les systèmes de communication filaire et sans fil est le modèle utilisé dans le canal de la transmission; dans notre simulation nous avons adopté les modèles suivants :

- Dans le système de communication filaire, on suppose l'absence des erreurs de transmission, et les puissances reçues par le lecteur au point d'accès des différents tags sont identiques.
- Dans le système de communication sans fil, la qualité de la voie de transmission dépend de deux paramètres :

✓ **Pertes de propagation:** La puissance reçue diminue d'une façon monotone avec la distance lecteur-tag, avec une proportion $\gamma^{-\alpha}$, où γ est la valeur moyenne du signal reçu et α est la constante d'atténuation. Dans le cas normal, la valeur *de* α est de 2 à 5 [38].

✓ **(Shadowing*)*:** Il est provoquée par les phénomènes de réflexion et diffraction subit par les ondes radio sur les obstacles lors des trajets multiples entre les tags et le lecteur. A cet effet la puissance reçue de l'onde radio multiple, fluctue intensément. La fluctuation s'appelle (fading). La valeur moyenne du fading pour une dizaine de longueurs d'onde s'appelle la valeur moyenne dans une seule petite échelle. La valeur moyenne de cette dernières à travers dix petites échelles est appelée *Shadowing* [38].

b) Génération de paquet

Chaque tag produit des paquets d'une manière aléatoires et indépendante. Une telle génération de paquet s'appelle la distribution de Poisson.

c)Collision

Dans les systèmes de communication filaire et sans fil, les paquets qui se sont chevauchés (collision complète ou partielle) sont manipulés comme suit :

- Système de communication filaire: Tous les paquets chevauchés sont détruits, et la transmission d'un paquet est échouée, car tous les niveaux des signaux relatifs aux paquets sont les mêmes. Dans ce cas, on utilise le modèle de collision pour un canal d'accès multiple, qui signifie que si deux étiquettes ou plus qui sont dans la gamme du lecteur transmettent simultanément, alors le lecteur ne reçoit aucun paquet transmis [4].
- Système de communication sans fil: La puissance reçue de chaque paquet dépend de la position du tag et de l'état de la voie de transmission. Dans certains cas même si plusieurs paquets sont recouverts (collision), un paquet peut poursuivre sa transmission si sa puissance est suffisamment grande. Ce phénomène s'appelle l'effet de capture.

2.3.1.2. Paramètres d'évaluations du système dans la famille stochastique

Les éléments les plus utilisés pour évaluer le protocole d'accès sont le trafic offert G, la sortie (*throughput*) S, et le délai moyen de transmission.

a) Le Trafic Offert: La charge offerte G correspond au nombre de transpondeurs transmettant simultanément à un certain point dans le temps t 0 (c.-à-d. 0, 1, 2, 3,…). La moyenne de la charge offerte G est la moyenne sur une période d'observation T et est extrêmement simple à calculer à partir d'une durée **τ** de transmission d'un paquet de données [15] :

$$G = \sum_1^n \frac{\tau_n}{T} * r_n \quad (2.1)$$

Où:

n = 1, 2, 3... : est le nombre de transpondeurs dans le système

r_n = 0, 1, 2… : est le nombre de paquets de données qui sont transmis par le transpondeur numéro n pendant la période d'observation.

Plusieurs transpondeurs envoient leurs paquets de données aléatoirement dans le temps. Ceci peut causer des collisions de données, dont en raison la sortie S (de données) chute à zéro pour les paquets de données qui sont interférées (voir Fig.2.3).

Le nombre de paquets qui inclue les nouveaux paquets produits ainsi que les paquets de retransmission dans un intervalle de temps s'appelle le trafic offert. Le trafic offert normalisé par un débit de transmission est nommé G_n et s'exprime comme suit :

$$G_n = \frac{Ts}{R} \quad (2.2)$$

Où *R* (bit/s) représente le débit de transmission et *Ts* (bit) le nombre de bits transmis dans cet intervalle.

b) Sortie (*Throughput*): Si la transmission des paquets s'effectue sans collision et dans le délai exigé, la sortie S sera égale à 1. Dans tout autres cas, elle est égale à 0. La sortie moyenne S, d'un canal de transmission, en fonction de la charge offerte G représentée sur la Fig.2.9 est donnée par: [15]

$$S = G * e^{-2*G} \quad (2.3)$$

Le nombre de paquets qui ont été transmis avec succès au point d'accès dans l'intervalle de temps désiré s'appelle sortie, et la sortie normalisée par rapport au débit de transmission de données est montrée comme S. On peut exprimer (2.3) autrement par la formule (2.4), où *n* représente le nombre de paquets transmis avec succès dans une unité de temps et *T* (bits) la quantité d'information dans un paquet.

$$S = \frac{T*n}{R} \quad (2.4)$$

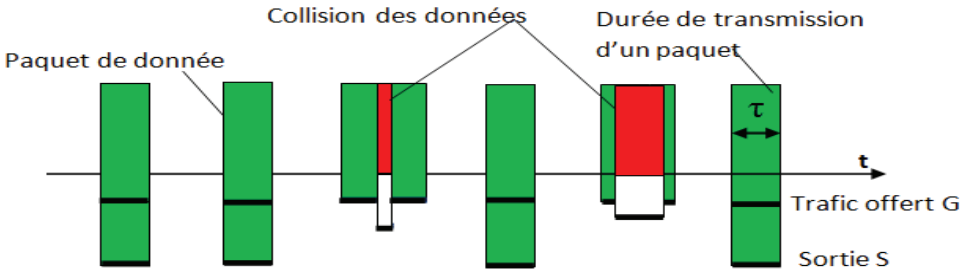


Fig.2.3. Définition de la charge offerte G et de la sortie S du protocole ALOHA [15]

c) **Le délai de transmission moyen :** il représente la période nécessaire pour la production et la transmission d'un paquet. Il dépend de la longueur du paquet, de la durée de création et la transmission du paquet ainsi que du temps de traitement des signaux au niveau du lecteur.

2.3.1.3. Configuration fondamentale du programme de simulation

Nous décrivons dans cette section le procédé d'évaluation de la sortie (*Throuput*) et le délai de transmission et le nombre de tags identifiés des protocoles d'accès lors de leur simulation, et nous allons voir l'influence de certains paramètres comme l'effet de capture et la valeur d'espérance de l'intervalle de génération des paquets sur la performance du protocole. La Fig.2.4 représente l'organigramme de la simulation des protocoles stochastiques.

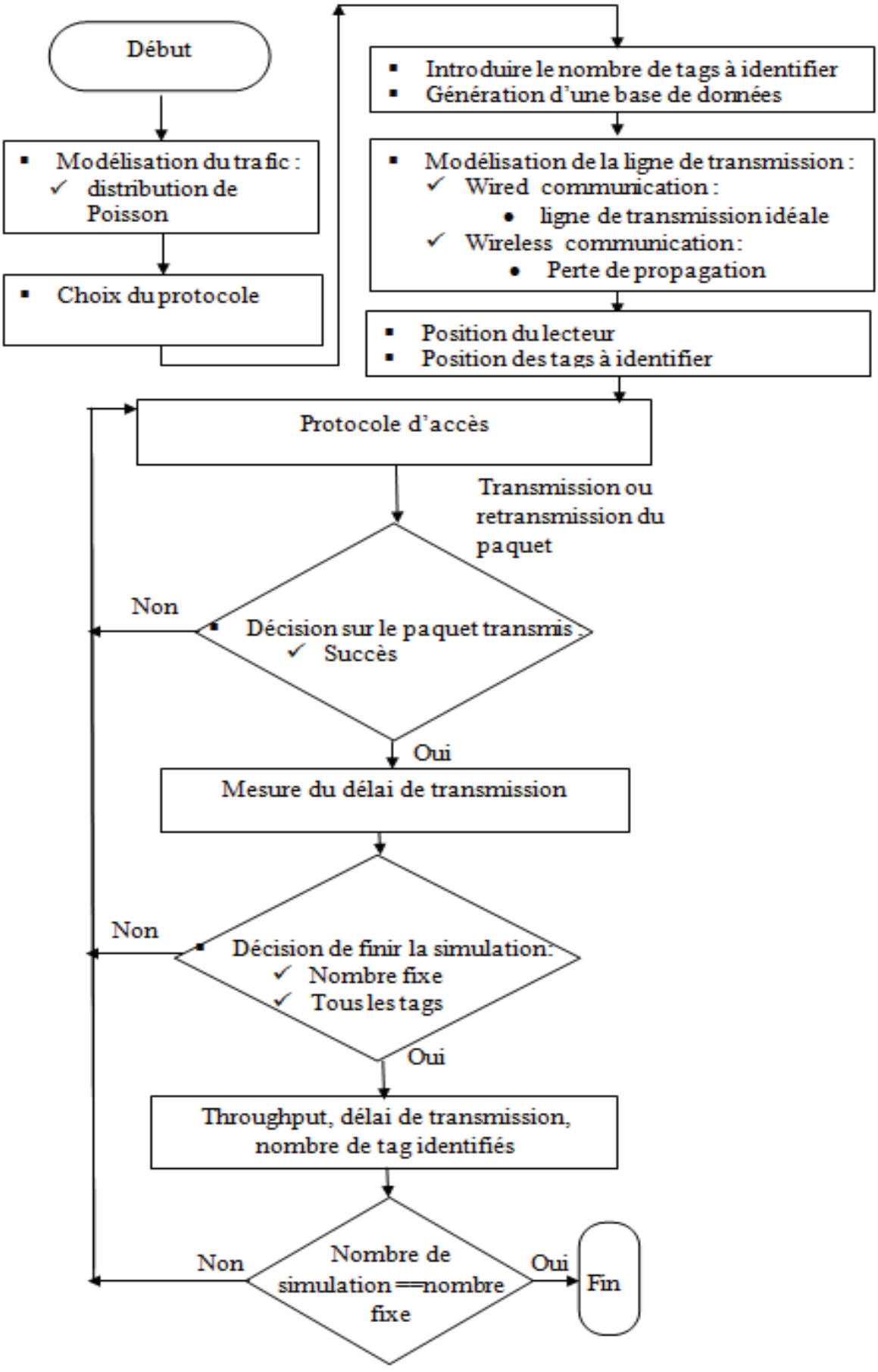


Fig.2.4.Organigramme de la simulation des protocoles stochastiques

❖ **Description des différents blocs de l'organigramme**

La simulation sur ordinateur est montrée sur la Fig.2.4. Dans cette simulation, chaque valeur d'espérance de l'intervalle de génération de paquet continuera jusqu'à ce que la condition configurée sur l'interface graphique soit réalisée.

a) **Vue générale de l'interface graphique Matlab**

La Fig.2.5 représente une vue générale de l'interface Matlab pour la simulation des protocoles de la famille aléatoire.

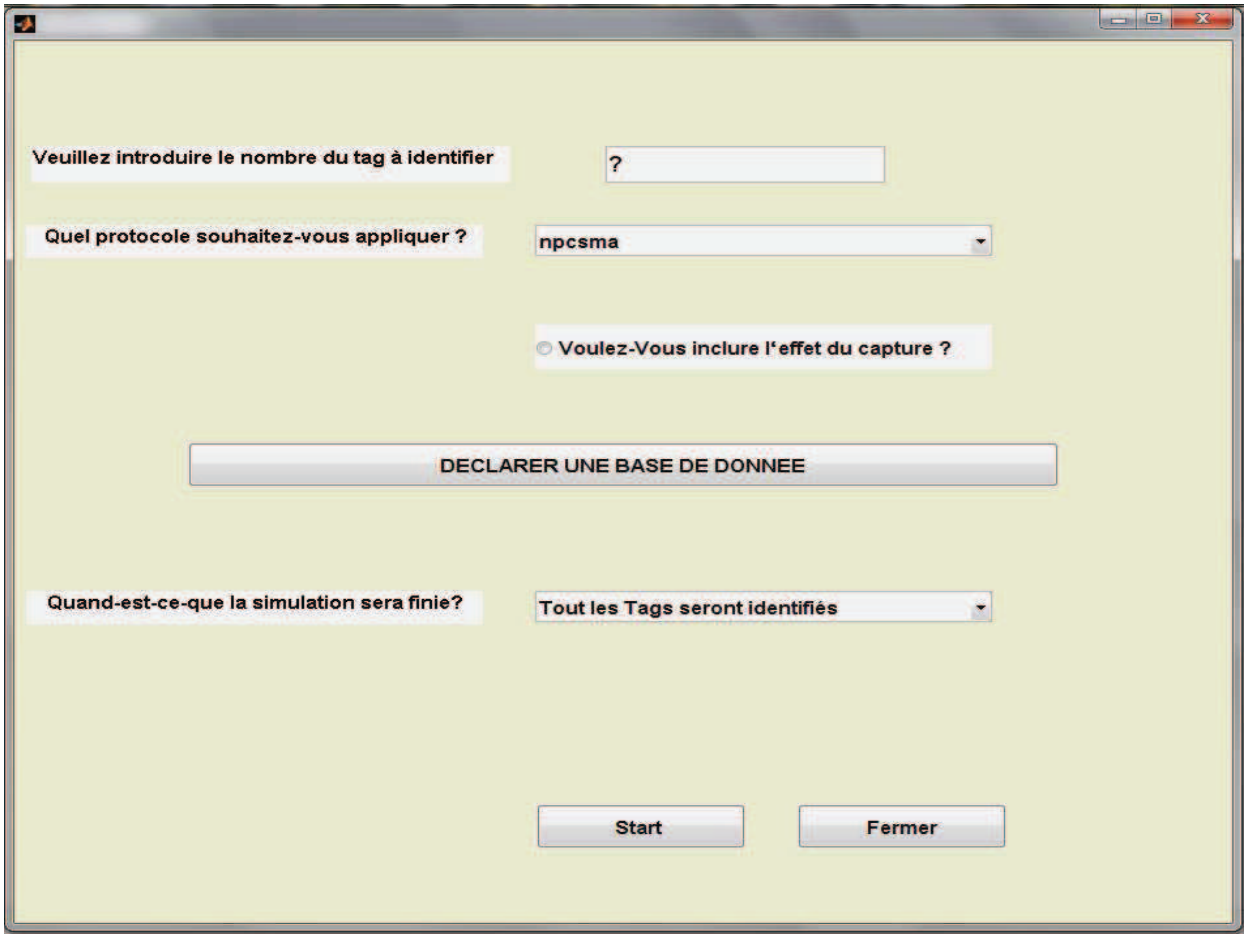


Fig.2.5. Interface graphique Matlab

b) **Modélisation du trafic :** Nous avons adopté comme modèle du trafic celui de Poisson, la probabilité dans laquelle une étiquette produit n paquets au temps *t* est montrée dans (2.5).

$$P_n(t) = \frac{e^{-\lambda * t}(\lambda t)^n}{n!} \tag{2.5}$$

La probabilité qu'aucun paquet ne se produit pendant la période du temps 0 *à t* est donné comme suit:

$$P_0(t) = e^{-\lambda * t} \tag{2.6}$$

Par conséquent, la probabilité qu'au moins un paquet sera produit est donnée comme suit:

$$P_1(t) = 1 - e^{-\lambda * t} \quad (2.7)$$

Dans la simulation, la période de génération du paquet est exigée. Le temps est calculé à partir de (2.8) sur une période *t.*

$$\mathrm{x} = 1 - \mathrm{e}^{-\frac{\mathrm{t}}{\mathrm{Tint}}} \quad (2.8)$$

En conclusion, nous pouvons obtenir la période de génération du paquet par :

$$t = -\mathrm{Tint} * \log(1 - x) \quad (2.9)$$

Où *x* est le nombre aléatoire uniforme.

- **Influence de la valeur Tint sur les résultats de performance du système**

D'après l'équation (2.9) nous remarquons que le temps de transmission du paquet suivant est égal au temps de génération du paquet *t* qui est proportionnel à la valeur d'espérance de l'intervalle de génération des paquets. Donc nous pouvons dire que le choix de la valeur d'espérance de l'intervalle de génération des paquets a une certaine influence sur les performances du système. Pour cette raison, nous avons simulé les protocoles de la famille stochastique en prenant plusieurs valeurs de T_{int}. Donc la valeur optimale de T_{int} c'est la valeur pour laquelle la ligne de transmission sera bien exploitée ce qui donne de remarquables résultats de performances pour le système.

c) **Choix du protocole** : Le choix du protocole est assuré par la configuration de la fenêtre « popupmenu » sur l'interface graphique (Matlab) comme le montre la Fig.2.6.

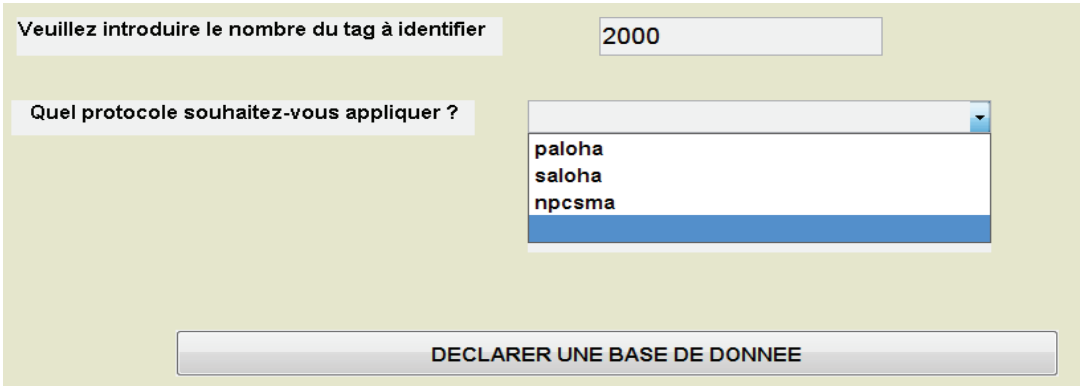


Fig.2.6.Choix du : protocole, nombre de tag à identifier et bouton de génération des IDs

d) Le nombre de tag à identifier : aussi, il est assuré par une fenêtre qui nous offre la possibilité de saisir un nombre voulu de tags à identifier comme montré sur la Fig.2.6.

e) La génération de la base de données : est assuré par le bouton « DECLARER UNE BASE DE DONNEE ». L'appui sur ce bouton va appeler un sous programme « base.m »

qui génère aléatoirement les codes d'identifications de tous les tags à identifier comme montré sur la Fig.2.6.

f) **Modélisation de la ligne de transmission :** le choix du modèle de la ligne de transmission est assuré par la disponibilité d'une fenêtre sur l'interface graphique comme montré sur la Fig.2.7.

Si nous choisissons de ne pas inclure l'effet de capture dans la simulation, le programme est simulé sur un canal de transmission idéal. Si nous voulons inclure l'effet de capture dans la simulation, le programme est simulé par une transmission sans fil.

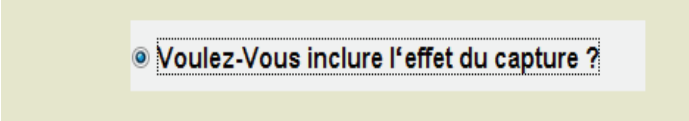


Fig.2.7.Cas où l'effet de capture est inclu dans la simulation du protocole

g) Positionnement des tags à identifier dans la zone d'interrogation du lecteur

Nous avons supposé que la zone d'interrogation du lecteur est une cellule de rayon r. Les n tags sont placés aléatoirement dans cette zone, comme représenté sur la Fig.2.8. Le centre de la cellule est l'origine (0.0). La position du lecteur est fixé dans la cellule et possède les coordonnées suivantes : $[X_L, Y_L, Z_L] = [0, 0, 6]$ en mètre.

Les positions des tags doivent être différentes. La hauteur de chaque tag prend une valeur aléatoire de Z_T=1 à 3 m qui limite la distance maximale d'un tag à$dmax = \sqrt{(5^2 + 3^2)} \approx$ 5.853m.

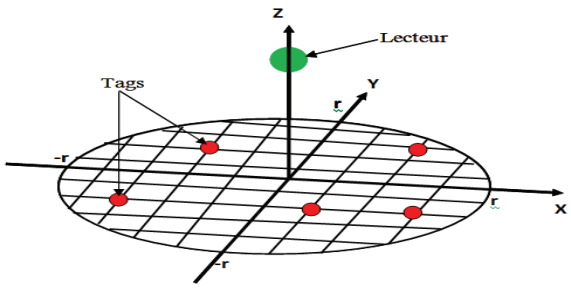


Fig.2.8.Positionnement des tags dans la zone de couverture du lecteur

Après le positionnement aléatoirement des tags, les pertes de propagations (shadowing) entre les tags et le lecteur sont données par la génération des valeurs aléatoires avec une distribution normale assurée par la fonction « randn» du Matlab.

h) **Protocole d'accès :** Le protocole d'accès, est un sous-programme qui assure la communication entre le lecteur et les tags présents dans sa zone d'interrogation.

Nous allons présenter dans ce qui suit les principes des protocoles aléatoires simulés à savoir les algorithmes de résolution de la collision (Aloha, Slotted Aloha), ainsi que celui d'évitement de la collision (CSMA).

- **ALOHA**

Cet algorithme est basé sur le principe TDMA, il est appliqué dans le cas des tags à lecture seulement, qui doivent généralement transférer peu de données (numéros de série); ces données sont envoyées au lecteur dans un ordre cyclique. Quand l'étiquette rentre dans la zone d'interrogation d'un lecteur, elle transmettra son identifiant immédiatement vers ce dernier [15]. L'étiquette attend que le lecteur répond par un accusé de réception positif *ACK,* indiquant que son identifiant a été reçu correctement, ou par un accusé de réception négatif *NACK*, signifiant qu'une collision s'est produite. Si deux ou plusieurs étiquettes transmettent en même temps, une collision complète ou partielle peut se produire et les étiquettes alors choisissent un temps aléatoire avant de retransmettre leurs identifiants [6].

On note *S,* la probabilité de succès dans les algorithmes d'ALOHA, elle est exprimée en fonction de la charge G qui représente le nombre moyen de tentatives de transmission de paquets pendant une durée, s'écrit donc [15]:

$$\mathbf{S = G\, e^{-2G}} \tag{2.10}$$

- **Algorithme S-Aloha(*Slotted Aloha*)**

Algorithme Slotted Aloha ou Aloha discrétisé est un processus d'anticollision stochastique basé également sur la TDMA. Dans ce cas, la trame est divisée en plusieurs slots et les transpondeurs doivent commencer à transmettre leurs paquets de données au début d'un slot synchrone [15]. La synchronisation de tous les transpondeurs doit être contrôlée par le lecteur. Dans le cas d'une collision, l'étiquette retransmet son paquet de donnée au début des intervalles de temps suivants après une durée d'attente aléatoire fournie par le lecteur. Les paquets dans ce cas se chevauchent complètement et non partiellement mais lorsqu'une seule étiquette est présente dans un slot, le lecteur peut l'interroger en recevant l'information correctement [39].

On peut remarquer à partir de la Fig. 2.9 que la probabilité de succès *S* en fonction de G est améliorée comparativement à celle d'Aloha pure et elle est donnée dans ce cas par [12]:

$$\mathbf{S = G\, e^{-G}} \tag{2.11}$$

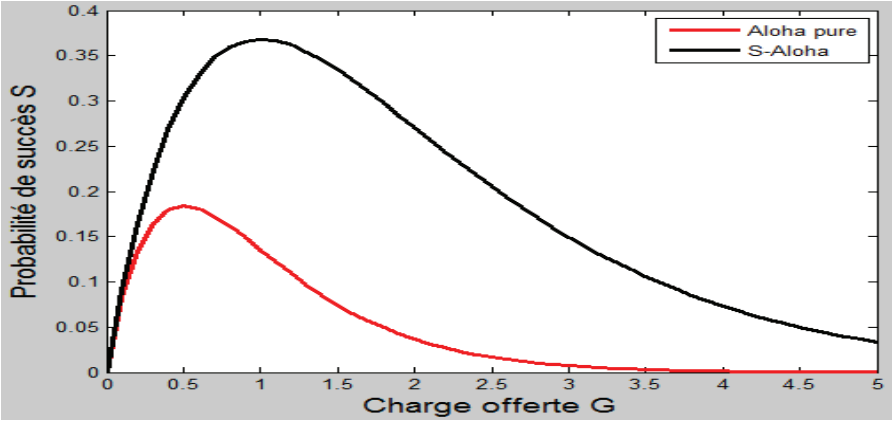


Fig. 2.9. Comparaison des courbes de sortie d'ALOHA et de S-Aloha.

Une nouvelle formule de la probabilité de succès est définie dans (2.12) pour cet algorithme qui considère un autre paramètre qui est l'effet de capture:

$$S = G \cdot e^{-\left(\frac{b \cdot G}{1+b}\right)} \tag{2.12}$$

Où : b représente un seuil de supériorité d'un paquet de données par rapport aux autres pour qu'il puisse être détecté par le lecteur [15].

L'algorithme S-Aloha emploie trois commandes pour trier les étiquettes: REQUEST, SELECT, et READ [40].

Les transitions d'état du lecteur et des étiquettes pendant le protocole de conversation est montré en annexe 2sous forme de schémas dans Fig.1 [40].

- **Principe général du CSMA (*Carrier Sense Multiple Access*)**

En employant le processus de détection de la porteuse, il est possible de juger si d'autres tags transmettent leurs paquets parce que chaque tag ne transmet aucune onde porteuse excepté sa transmission de paquet, comme représenté sur la Fig. 2.10. Si une onde porteuse est détectée sur la voie de transmission, le canal est dit: "occupé"; autrement, il est dit: "libre". Dans le CSMA non-persistent, quand le paquet est produit dans un tag, le tag commence le processus de détection de la porteuse.

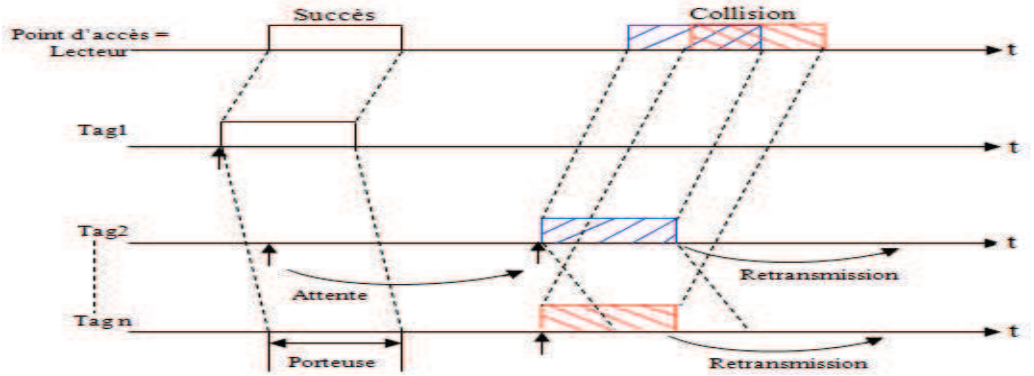


Fig. 2.10. Principe de CSMA non-persistant

Si le résultat d'une détection de la porteuse est "libre", le paquet produit est transmis au lecteur immédiatement. Cependant, si le résultat de détection de la porteuse est "occupé", le tag arrête processus de détection, attend un moment, et commence alors à nouveau le processus de détection [41].

Malgré que chaque tag détecte la porteuse avant de transmettre son paquet, mais la collision des paquets se produit toujours. Une des raisons est le délai de propagation. Dans un véritable système de communication RFID, quand un tag transmet ses paquets, les autres tags détectent la transmission après un délai de propagation. Si d'autres tags transmettent leurs paquets pendant le délai de propagation, les collisions se produisent au lecteur. À cause des obstacles qui existent entre les tags dans un système de communication RFID sans fil, la porteuse parfois ne peut pas être détectée sur quelques tags même si un tag transmet un paquet [41].

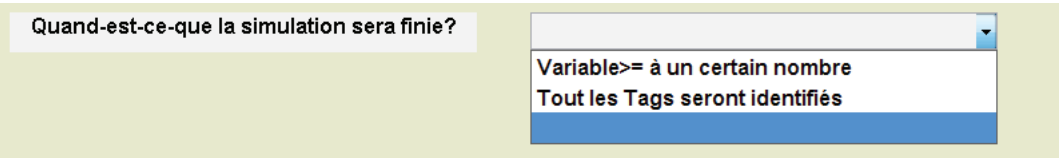
La sortie S du CSMA non-persistent en fonction du trafic offert G est montrée comme suit [42]:

$$S = \frac{Ge^{-aG}}{G(1+2a)+e^{-aG}} \tag{2.13}$$

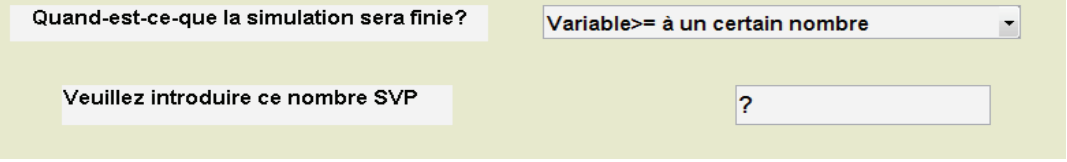
Où : a est le délai ou temps de propagation normalisé.

Les opérations fondamentales dans la simulation sont montrées comme suit :

Si le nombre total de tags identifiés par le lecteur ayant transmets leurs paquets au point d'accès avec succès est plus grand qu'une variable ayant la possibilité d'être configurée en deux cas dans l'interface graphique comme le montre la Fig. 2.11(a), la simulation pour la valeur courante de G sera terminée.



(a)



(b)

Fig.2.11. Configuration de la condition de fin de simulation pour une valeur de G courante

Si nous choisissons la première case, un nouveau champ va apparaître dans la fenêtre de l'interface qui nous permettra de saisir ce nombre comme montré dans la Fig.2.11(b).

Les tags qui transmettent leurs paquets à la station de base (lecteur) sont comptés.

Nous considérons deux procédures pour déterminer si les paquets transmis par les tags sont reçus avec succès ou non au point d'accès :

- Quand l'effet de capture n'est pas considéré dans la simulation du protocole : s'il y a plus de deux tags qui transmettent simultanément, il y a de forte chance qu'il y ait collision, l'état de chaque tag sera COLLISION et il sera stocké dans la variable Tag_etat. Dans la simulation, s'il y a un seul tag qui transmet son paquet, la puissance calculée de ce paquet est toujours plus grande que le seuil minimal (tcn) qui est défini par le rapport de capture C/N en dB. Les données instantanées dans le paquet sont transmises au point d'accès avec succès, et le résultat est stocké dans le variable Tag_etat.

Avec : (tcn= C/N en dB) est la valeur seuil de la puissance reçue pour la quelle aucune erreur de paquet ne se produit au point d'accès (lecteur), elle est égale à 3 dB. Elle dépend de la sensibilité du lecteur.

- Quand nous considérons l'effet de capture dans la simulation du protocole : le paquet dont la puissance reçue est maximale peut échapper à la collision. La distance entre le lecteur et les tags transmettant les paquets est calculée dans le sous programme distance.m.

Puis, la puissance au point d'accès est calculée pour tous les paquets qui sont transmis à partir des tags, et le paquet qui a la puissance maximale est sélectionné. En suite, la puissance instantanée du paquet sélectionné est calculée en considérant les autres paquets comme bruit. Si cette dernière est plus grande que le seuil minimal de puissance (tcn), les données instantanées dans le paquet sont transmises au point d'accès avec succès, et le résultat est stocké dans la variable Tag_etat. D'autre part, si la puissance calculée est inférieure au seuil minimal de puissance (tcn), tous les paquets des tags ne seront pas transmis au lecteur, et le résultat « COLLISION » est stocké dans la variable Tag-état.

Initialement, l'état de chaque transpondeur est placé à «STANDBY» et lorsque un transpondeur commence à générer un paquet de données sa variable d'état va prendre l'état «TRANSMISSION », quand ce transpondeur reste dans cet état pendant la durée de transmission d'un paquet, la transmission de paquet est terminée avec succès même si la collision se produit. Quand un transpondeur transmet son paquet avec succès au lecteur, il sera identifié et sa variable d'état passe à l'état « CLOSE » jusqu'à la nouvelle valeur de G.

- Le compteur de temps s'incrémente à chaque changement d'état du tag (par exemple, un nouveau paquet est produit).
- Ici, le trafic moyen et la sortie sont définis comme suit :

  Trafic = temps nécessaire pour transmettre tout les paquets / période de la simulation;

S= temps de transmission des paquets qui sont seulement transmis avec succès / période de la simulation

Quand toutes les simulations seront terminée en réglant le paramètre G dans une boucle for, les fichiers qui stockent les résultats de la simulation sont fermés, et les résultats seront montrées dans des figures par l'exécution des sous programmes : graph.m, graphA.m, graphnpcsma.m.

2.3.2. Méthodes de résolution déterministes

Dans les protocoles anticollision basés sur les algorithmes en arbre au niveau radio, nous supposons que les étiquettes qui se trouvent dans la zone d'interrogation du lecteur pourront toujours recevoir des impulsions de ce lecteur sans erreurs. De même, si une seule étiquette dans la zone d'interrogation du lecteur transmet, alors son paquet est reçu avec succès par le lecteur. Nous utilisons le modèle de collision pour un canal d'accès multiple, qui signifie que si deux étiquettes ou plus qui sont dans la gamme du lecteur transmettent simultanément, alors le lecteur ne reçoit aucun paquet transmis. Dans les protocoles anticollision basés sur les arbres pour des systèmes RFID passifs, plusieurs facteurs peuvent causer des erreurs de transmission, tels que l'évanouissement, l'orientation d'étiquette, les obstacles entre le lecteur et l'étiquette, etc. Néanmoins, le modèle adopté par la couche MAC idéalisé est suffisant pour exprimer les concepts pertinents aux protocoles basés sur les arbres. Une étiquette envoie une réponse uniquement quand elle est interrogée par une impulsion envoyée par le lecteur. Ceci est illustré par la Fig.2.12, où les étiquettes envoient des réponses à l'impulsion.

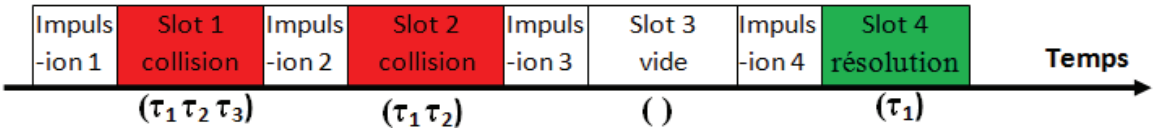


Fig.2.12. Diagramme de temps du canal lors de l'arbitrage et la communication étiquette-lecteur [4].

En autorisant les réponses de certains groupes d'étiquettes, le lecteur contrôle les collisions sur le canal. Par exemple, à la réception de l'impulsion 1, les étiquettes $\tau 1$, $\tau 2$, $\tau 3$ envoient leurs réponses. Nous supposons que la durée du paquet envoyé par chaque étiquette est constante et égales à un slot, tel que si deux étiquettes transmettent simultanément leurs paquets sont complètement recouverts. Le lecteur retourne un paquet d'information dans la prochaine impulsion. Par exemple, lorsque k étiquettes transmettent dans un slot, alors l'interrogateur perçoit le canal dans ce slot comme :

• *Repos ou libre(I) si k = 0 pas de réponses des étiquettes.*

• *Réception correcte (S) ou résolution d'étiquette si k = 1.*

• *Collision (C)* si $k \geq 2$.

Dans l'exemple de la Fig. 2.12, l'impulsion 2 envoyée par le lecteur porte une instruction informant les étiquettes d'une collision au niveau du slot 1.

De façon à introduire les protocoles d'arbitrage basés sur les arbres, nous supposons que chaque étiquette peut générer un vecteur binaire aléatoirement décrivant son identifiant (ID) qui est utilisé pour l'arbitrage.

2.3.2.a. Paramètres d'évaluation du système

- En absence d'erreurs, l'efficacité des protocoles d'arbitrage est mesurée en termes de temps et du nombre de messages. S'il y a (n) étiquettes, alors nous nous intéresserons au temps moyen *Tn* que le protocole prend pour identifier toutes les étiquettes. A partir de la Fig2.12 nous pouvons remarquer que *Tn* possède deux composantes : (1) le temps utilisé pour envoyer les impulsions ; et (2) les slots de temps utilisés pour la transmission des réponses à partir des étiquettes. Nous supposerons que la durée de l'impulsion envoyée par le lecteur est nulle, et nous nous baserons uniquement sur les slots de temps consommés par transmission des réponses. Avec une impulsion de longueur zéro, l'efficacité du temps d'un protocole d'arbitrage donnée dans [4] est défini par :

$$\eta_n = \frac{n}{T_n} \tag{2.14}$$

- Notez qu'une autre mesure qui est traditionnellement utilisée pour estimer la performance des protocoles d'arbre est le nombre moyen des messages Mn envoyés durant le processus d'arbitrage. Ce serait intéressant si les étiquettes actives seront considérées, puisque dans ce cas chaque message transmis consomme de l'énergie de l'étiquette. D'autre part, avec les étiquettes passives, Mn est plutôt inutile, puisque l'énergie de l'étiquette est fournie par le lecteur.

En plus de ces paramètres d'évaluations discutées ci-dessus nous nous sommes intéressés par l'observation des autres paramètres qui sont résumés dans les trois points importants suivants :

- Le premier point d'observation est la moyenne des bits exigés pour l'identification d'une étiquette avec une longueur d'identification de l'étiquette de 12 bits au lieu de 96-bits déterminée dans EPC Class 1 Gen2 à cause de la limitation de la capacité mémoire de nos micro-ordinateurs disponibles.
- Le deuxième point est les itérations moyennes exigées pour l'identification d'une étiquette avec une longueur d'identification de l'étiquette de 12 bits au lieu de 96-bits. Une itération

est déterminée par une seule demande REQUEST et un des états de réponses possibles: collision, résolution, et aucune réponse.

- Le troisième point est la tendance des performances de préfixe et de la réponse; qui sont liées à l'aspect de consommation d'énergie au niveau du lecteur et au niveau des étiquettes.

2.3.2.b. Les algorithmes de la méthode de résolution déterministes simulés

Dans cette famille nous avons simulé cinq protocoles d'anticollision. Nous avons adopté pour leur simulation le modèle discuté précédemment. Nous allons en discuter, ainsi leur principe et leurs organigrammes seront présentés. Pour monter l'aspect général de l'algorithme binary tree, l'arbre binaire et le diagramme de temps pour un exemple de 5 tags seront présentés.

- **Description de l'algorithme *Tree Based***

Nous pouvons voir le processus clairement dans l'exemple suivant :

Supposons qu'il y a quatre tags dont les codes d'identification sont respectivement :

Tag1=«100», tag2=«101», tag3=« 001», tag4=« 011».

Le lecteur commence par l'envoie d'une demande, par exemple, toutes les étiquettes dont le premier chiffre est "0" doivent répondre par leur ID unique.

Dans notre exemple, seulement les tag3 et tag4 vérifiant cette condition qui vont répondre au lecteur. Les deux tags vont répondre et le lecteur conclut qu'une collision s'est produite entre deux tags ou plus.

Le lecteur envoie la deuxième requête. Par exemple : Si votre premier chiffre = " 0 " et votre 2^{em} chiffre = " 1 ", répondez par « présent ».

Le tag4, étant le seul dans cet exemple dont le 1^{er} chiffre de son identifiant est un "0" et le 2^{em} chiffre est un "1", donc il doit répondre.

Puisque il y a seulement une réponse, le lecteur entend la réponse et identifie le tag.

En examinant toutes les valeurs possibles des chiffres, le lecteur peut par la suite trouver et identifier chaque tag dans sa zone de lecture [4].

- **Description de l'Algorithme QTA (*Query Tree Algorithm*)**

  L'algorithme d'arbre de question (QTA) a le même principe que l'algorithme BTWA [8].

  Le lecteur transmet toujours un préfixe de longueur K et les étiquettes possédant les k premiers bits de leurs IDs identiques au préfixe envoient de $(K + 1)^{ém}$ *bit* jusqu'à l'extrémité de leurs IDs. Si une collision se produit dans les séquences reçus, le préfixe se prolonge et le lecteur attache '0' ou '1' à l'ancien préfixe, et le nouveau préfixe est

retransmis. Par contre, si une étiquette est la seule à répondre à la requête, l'étiquette est identifiée par le lecteur [8].

- **Algorithme CTTA (*Collision Tracking Tree Algorithm*)**

  L'algorithme d'arbre de cheminement de collision (CTTA) a le même principe que l'algorithme QTA sauf que cet arrangement utilise le cheminement de collision [8].

  Dans le CTTA, les étiquettes possédant les K premiers bits de leurs IDs identiques au préfixe envoient toujours leur IDs de $(K + 1)^{ième}$ *bit* jusqu'à l'extrémité. En cas de collision, le lecteur transmet un signal d'ACK pour cesser la transmission des bits. L'algorithme de cheminement d'arbre de collision réduit la perte de temps provoquée par des collisions. Pour cela, le prochain préfixe est constitué par la concaténation de l'ancien préfixe de longueur K bits avec les (n-1) bits plus0 ou 1 quand la collision se produit au $n^{ième}$ bit dans les séquences reçus du lecteur [8].

- **Description des protocoles (bi-slotted tree)**

L'organigramme de l'algorithme BSQTA (Bi-Slotted Query Tree) est montrés en annexe 01dans la Fig. 3 et celui de l'algorithme BSCTTA est montré dans la Fig. 4 [9].

La procédure des deux algorithmes a quatre points principaux:

1) REQUEST: Le lecteur commence par transmettre une requête aux étiquettes, contenant un préfixe de taille (n-1) bits.

2) GROUPEMENT: Les étiquettes qui se trouvent dans la portée du lecteur et possédant les (n-1) premiers bits de leurs IDs identiques au préfixe vont répondre au lecteur. Elles choisissent un des deux slots de temps selon si leur nième bit est ' 0 ' (premier slot) ou ' 1 ' (deuxième slot). Ainsi, le slot de temps indique la valeur du $n^{ième}$ bit dans l'identifiant du tag.

- BSQTA: les étiquettes envoient leur IDs de $(n + 1)^{ième}$ bit jusqu'à l'extrémité
- BSCTTA: les étiquettes envoient leur IDs à partir du $(n + 1)^{ième}$ bit jusqu'au temps que le signal d'ACK, qui est envoyé par le lecteur quand une collision se produise, est reçu.

3) DÉCISION: Le lecteur déclenche une procédure de marche selon les cas suivants:

- S'il y a une collision, le lecteur sauvegarde un nouveau préfixe dans une pile (LIFO). Dans l'algorithme BSQTA, le nouveau préfixe est constitué par la concaténation des bits du préfixe de longueur (n-1) et l'indication du slot choisi. Mais dans le BSCTTA, il est constitué par la concaténation des bits du préfixe de longueur (n-1), l'indication du slot choisi, et les bits reçus avant que des collisions se produisent.

- Si une collision se produit au dernier bit dans l'IDs d'étiquette, le lecteur suppose qu'il y a deux étiquettes en raison de l'unicité de l'IDs d'étiquette.
- S'il y a seulement une réponse d'étiquette (pas de collision), le lecteur entend la réponse et identifie l'étiquette.

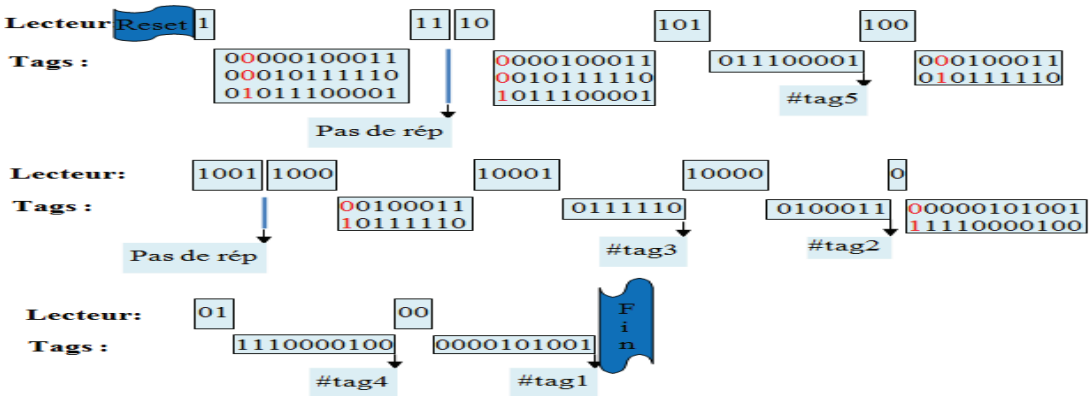
4) Ce procédé se répète jusqu'à ce que la pile LIFO soit « *nul* » [9].

2.3.2.c. Un exemple

Pour une comparaison objective entre les algorithmes et entre les résultats de simulation et ceux de la théorie, nous avons pris un exemple de cinq étiquettes dans la zone d'interrogation du lecteur RFID, leurs IDs sont : Tag1= « 000000101001 », Tag 2= «100000100011», Tag 3= « 100010111110 », Tag4= « 011110000100 », et Tag 5= « 101011100001 ». Alors nous évaluons les bits totaux nécessaires pour l'identification les cinq étiquettes afin de comparer le QTA au BSQTA et le CTTA au BSCTTA.

L'utilisation d'une pile LIFO permet d'améliorer la performance du protocole puisque dans chaque requête nous ne recherchons pas à partir du niveau de la racine, mais nous recherchons à partir du nœud le plus étroit de l'arbre (quand on dépile le préfixe empilé). Dans les Fig.2.13 et Fig.2.14, l'axe de temps est de la gauche vers la droite et les bits du côté du lecteur sont les bits du préfixe, et les bits du côté d'étiquette sont les bits de la réponse correspondante. Donc, la longueur de gauche à droite montre le délai pour l'identification des cinq étiquettes. Ce délai dépend strictement de la quantité de bits transmis par le lecteur dans le message de requête et par le tag dans le message de réponse.

Query tree algorithm (QTA)



Bi-Slotted Query Tree Algorithm (BSQTA)

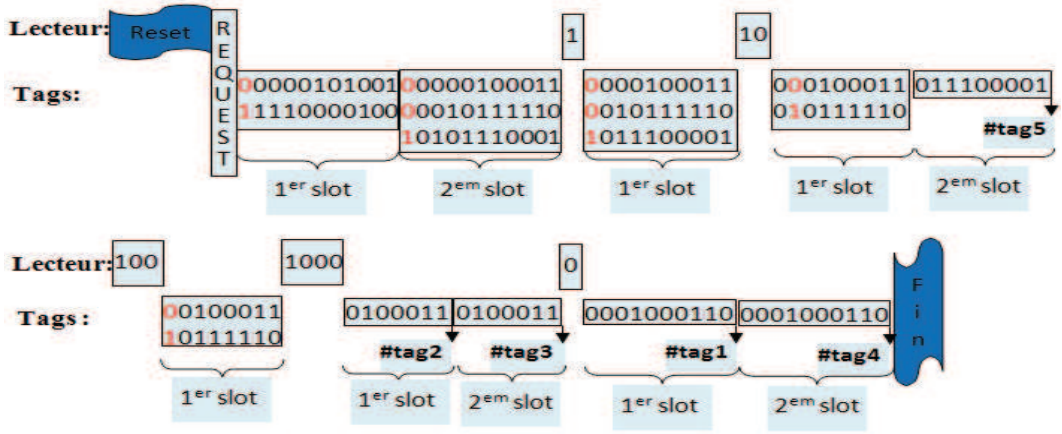
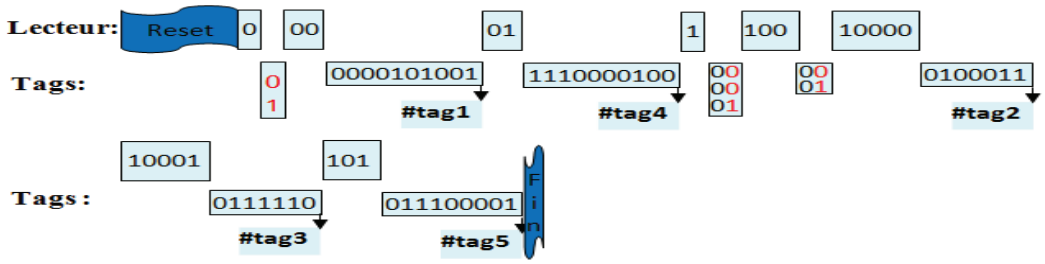


Fig.2.13. Une comparaison entre QTA et BSQTA

Collision Tracking Tree Algorithm (CTTA)



Bi-Slotted Collision Tracking Tree Algorithm (BSCTTA)

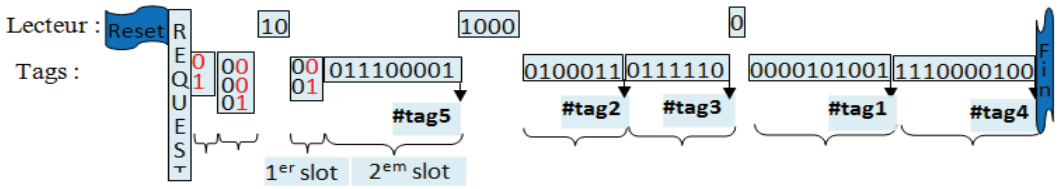


Fig.2.14. Une comparaison entre CTTA et BSCTTA

Pour le protocole binary tree, nous avons pris le même exemple de cinq tags. Nous avons construit l'arbre binaire ainsi le diagramme de temps correspond à l'arbre pour montrer le déroulement du processus et pour qu'on puisse comparer ces résultats théoriques à ceux de la simulation. L'arbre binaire ainsi que le diagramme de temps correspond sont montrés respectivement dans la Fig.2.15 et Fig.2.16. Dans la Fig.2.15 le nœud noté S_j est relatif à la sortie dans le $j^{ième}$ slot qui est représentée sur le diagramme du temps dans la Fig.2.16. Par exemple, la collision dans le premier slot S1 a une multiplicité de cinq.

D'après les Fig. 2.13 et Fig. 2. 14, pour identifier les cinq étiquettes, QTA exige 34 bits pour les préfixes et 92 bits pour les réponses, ainsi 126 bits au total sont nécessaires pour l'identification de toutes les étiquettes. D'autre part, le BS-QTA exige 11 bits pour les préfixes et 92 bits pour les réponses, ainsi 103 bits au total sont nécessaires pour l'identification de toutes les étiquettes. Le CTTA exige 22 bits pour les préfixes et 48 bits pour les réponses, ainsi 70 bits au total sont nécessaires pour l'identification de toutes les étiquettes. D'autre part, le BS-CTTA exige 7 bits pour les préfixes et 48 bits pour les réponses, ainsi 55 bits au total sont nécessaires pour l'identification de toutes les étiquettes.

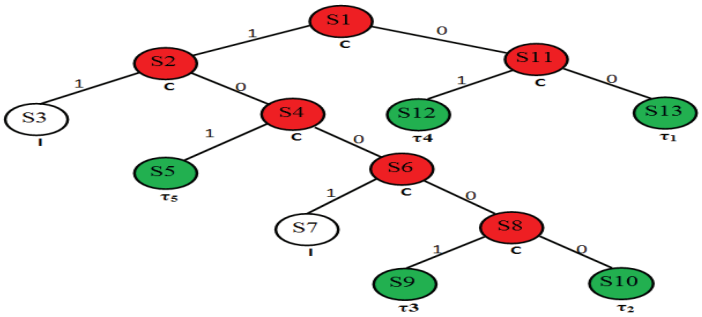


Fig.2.15. Arbre binaire pour une multiplicité de collision de N = 5.

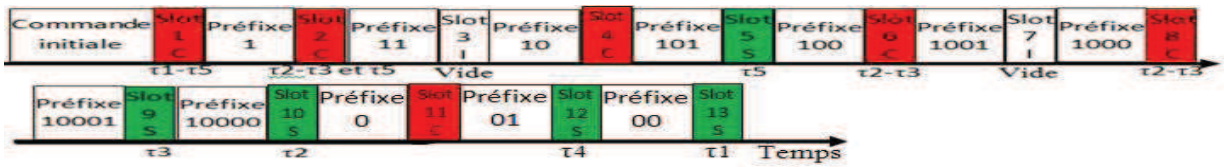


Fig.2.16. Chronogramme complet de l'exécution du protocole avec l'exemple de 5 tags

Conclusion

Dans ce chapitre, nous avons présenté la modélisation d'un système RFID de communication par paquet pour la simulation des protocoles d'anticollision sur ordinateur.

Au premier lieu, nous avons commencé par la définition d'un modèle de référence qui spécifie les exigences physiques et logiques pour une rétrodiffusion passive d'un système RFID, ensuite nous avons abordé la modélisation d'un système RFID pour la simulation des

protocoles de la méthode de résolution stochastique en décrivant la configuration entière d'un système RFID et les différents paramètres d'évaluation du système pour la simulation de ces protocoles, enfin la configuration fondamentale du programme de simulation des protocoles de la méthode de résolution stochastique est présenté. Ainsi, une description des différents blocs de l'organigramme est abordée et les principes des protocoles aléatoires simulés sont présentés.

En second lieu, nous avons discuté de la modélisation d'un système RFID pour la simulation des protocoles d'anticollision de la méthode de résolution déterministe en décrivant les différents paramètres d'évaluation du système pour la simulation de ces protocoles, ensuite, nous avons présenté les protocoles simulés dans cette famille.

Enfin pour des raisons de comparaison entre les algorithmes déterministes et entre les résultats de simulation et ceux de la théorie, nous avons pris un exemple de cinq étiquettes dans la zone d'interrogation du lecteur RFID.

Chapitre 03

Simulations et résultats

3.1. Introduction

Avant de présenter les différents résultats de simulation obtenus des protocoles d'anticollision déterministes et aléatoires du système RFID, nous allons définir tout d'abord l'environnement de notre étude :

Nous avons développé les algorithmes d'anticollision de la famille déterministes en utilisant le ***C++ Dev V 4.9.9.2*** sous Windows 7, et pour les protocoles de la famille aléatoire, nous les avons développés en utilisant ***Matlab V 7.8.0 (R2009a)*** sous l'environnement Windows.

Nous avons utilisé comme support de traitement un PC Portable de processeur Intel Core 2 Duo 2.00 GHz, possédant une mémoire installée (RAM) de 3GO et un système d'exploitation de 32 bits. Ces performances n'étaient pas suffisantes pour simuler les protocoles d'anticollision de la famille déterministes dans la norme EPC Class 1 Gen 2 proposée par EPC global à ISO/ICE C 18000-6 qui exige deux conditions : la première porte sur la longueur de l'identifiant qui doit être de 96 bits, la seconde porte sur l'unicité d'identifiant car chaque tag doit avoir son propre identifiant.

En vue de satisfaire ces exigences, nous avons adopté le modèle idéalisé discuté en chapitre 2 qui utilise une longueur d'identifiant de 12 bit (à cause de la limitation de la capacité mémoire de nos micro-ordinateurs disponibles) au lieu de 96 bits. Néanmoins, ce modèle à couche MAC idéalisé, reste suffisant pour exprimer les concepts pertinents des protocoles basés sur les arbres.

Pour la simulation et l'analyse des performances des protocoles d'anticollision de la famille stochastiques, nous avons travaillé sur la norme 18000-4 d'un système RFID passif à 2,45 GHz, avec densité élevée des étiquettes dans le secteur (plus de 1500 étiquettes).

3.2. Discussion des sous-programmes aléatoires

3.2.1. Aloha

Le sous-programme Paloha.m peut être divisé en six fonctionnalités.

1. Le premier bloc, renferme les définitions des variables globales et les variables locales.
2. Le deuxième bloc initialise les états des tags. C'est dans ce bloc où le moment initial où le premier paquet est produit est définit ainsi que la longueur du paquet.
3. Le troisième bloc réalise un comptage lors d'un succès de transmission du nombre de paquets transmis, et calcule le délai de la transmission ainsi que le moment de production d'un nouveau paquet pour fournir cette information à tous ces tags.
4. Le quatrième bloc fournit l'ordre de retransmission d'un paquet lors d'un éventuel échec.

5. Le cinquième bloc cherche les tags qui commencent la transmission de leur paquet et qui n'ont pas encore été identifiés. Puis, l'état bascule vers le mode « TRANSMISSION », la durée nécessaire pour terminer une transmission de paquet est calculée, et le nombre de paquets transmis est compté.
6. Le sixième bloc fournit l'instant du prochain état aux tags.

3.2.2. Slotted ALOHA : Par une simple modification d'ALOHA, le taux de collisions peut être réduit par la moitié, à savoir, que les messages sont exigés pour être transmis dans des slots de temps entre deux impulsions de synchronisation, et peut être commencé seulement au début d'un slot de temps. Pour transmettre le paquet au lecteur avec succès, le nombre de paquets qui est produit dans un slot de temps doit être 1. Si plus de deux paquets sont produits dans un slot de temps, une collision se produit.

3.2.3. np-CSMA : Le sous-programme npcsma.m est presque identique à paloha.m. En plus des cinq blocs discutés dans aloha, le npcsma permet de détecter la porteuse avant d'envoyer un paquet pour chaque tag en employant une fonction carrier sense.m. Si le résultat est « vide ou libre», un paquet est transmis au lecteur. Si le résultat est « occupé », l'instant de la retransmission de paquet est fourni. Toute information sur les instants d'envoi des paquets pour tous les tags est stockée dans la variable globale Mstime.
Dans le sous-programme carrier sense.m, la variable Mstime est ajoutée au délai de propagation qui dépend de la distance entre les tags. Puis, la porteuse est captée pour tous les tags. Si la porteuse est détectée, alors la condition se met à « occupée » (result=l), sinon la condition est placée « vide ou libre» (result=0).

3.3. Discussion des sous-programmes déterministes

La simulation de chaque protocole RFID d'anticollision d'étiquettes basé sur l'arbre porte sur la présentation du déroulement du processus par la génération des numéros de série unique correspondant au nombre introduit de tags à identifier en se basant sur l'évaluation des paramètres correspondant à la famille des algorithmes déterministes discutés en chapitre 2.
Pour cela nous avons pris le même exemple de cinq étiquettes dans la zone d'interrogation du lecteur avec les mêmes identifiants.
Nous évaluons les bits totaux (bits de préfixe et bits des réponses), ainsi le nombre d'itérations nécessaires pour l'identification des cinq étiquettes pour chaque protocole en utilisant le modèle de collision pour un canal d'accès multiple, qui signifie que si deux étiquettes ou plus qui sont dans la gamme du lecteur transmettent simultanément, alors le lecteur ne reçoit aucun paquet transmis.

4. Simulations et résultats

4.1. Méthodes aléatoires

4.1.1. Environnement de simulation des protocoles d'anticollision aléatoires

Les paramètres de simulation sont :

- ✓ Il y a seulement un lecteur. Dans la zone du lecteur, le nombre d'étiquettes peut augmenter jusqu'à 3000 tags.
- ✓ Le débit binaire tag-lecteur et le débit binaire lecteur-tag sont fixés à 40K bit/s, car c'est la vitesse moyenne dans EPC Class 1 Gen. 2 proposée par EPC global à ISO/ICE C 18000-4.
- ✓ La taille de paquet égale à 96 bits;
- ✓ Une portée de lecture égale à 5m;
- ✓ Un nombre de tag à identifier égale à 1500 tags ont été aléatoirement produites et uniformément placées autour du lecteur avec des distances de 0-5 m. Les tags utilisent le signal du lecteur pour la synchronisation.
- ✓ Un environnement intérieur avec l'exposant du chemin de perte $\alpha = 3$. Pour les applications, telles que la numérisation des produits dans un magasin, $\alpha = 1,6$ est généralement le cas: ainsi, une distance de lecture allant jusqu'à 5,5 m peut être atteint.
- ✓ Afin d'identifier l'étiquette; le lecteur exige une sensibilité de 3 dB de différence entre le signal de l'étiquette considérée et le bruit du canal (l'interférence d'autres étiquettes).

Nous évaluons les principaux protocoles d'anticollision, discutés précédemment, nous caractérisons leur comportement. Cela donne un aperçu des avantages et inconvénient de chaque protocole. Les performances de la sortie et du délai moyen de transmission et l'évaluation en termes de temps sont déterminées; les résultats de simulation sont montrés dans les figures qui suivent.

4.1.2. Aloha

Les résultats de simulation de la sortie, du délai moyen de transmission et l'évaluation en termes de temps pour le protocole d'anticollision d'ALOHA sont montrés dans les Fig. 3.1 jusqu'à Fig. 3.6.

a) Résultats de la simulation avec un seuil de capture Tcn=3dB

- **Simulation sous une voie de communication filaire**

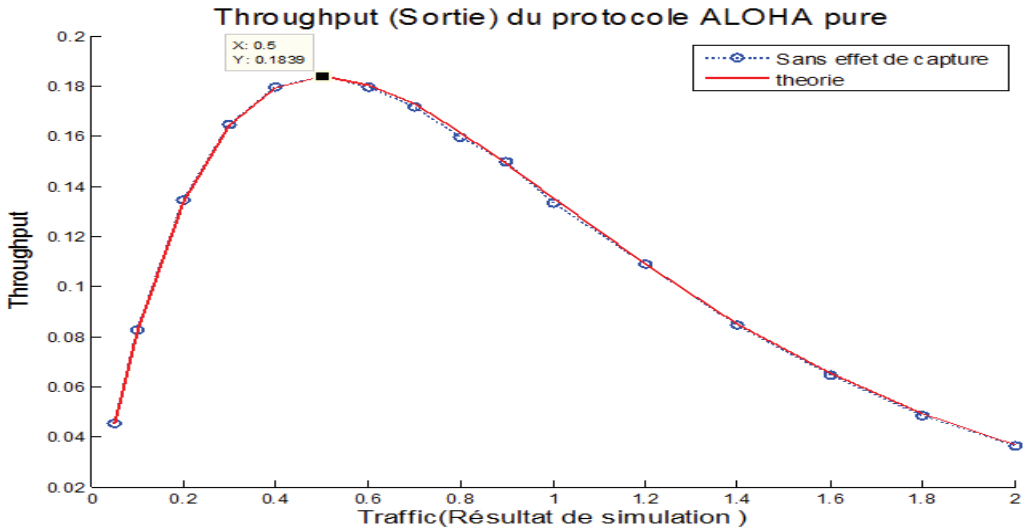


Fig.3.1. Résultat d'évaluation de la sortie (théorie et réelle) lors de la simulation du protocole Aloha sous une voie de communication filaire

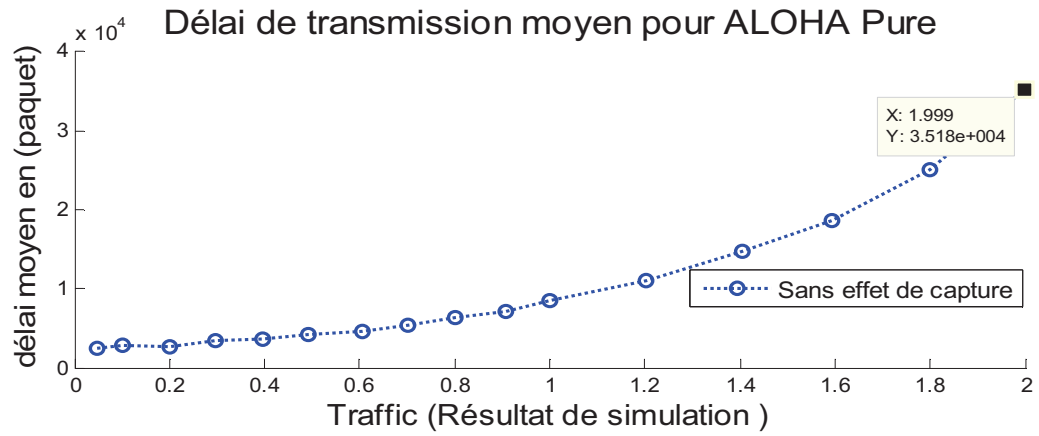


Fig.3.2. Résultat d'évaluation de délai moyen de transmission lors de la simulation du protocole Aloha sous une voie de communication filaire

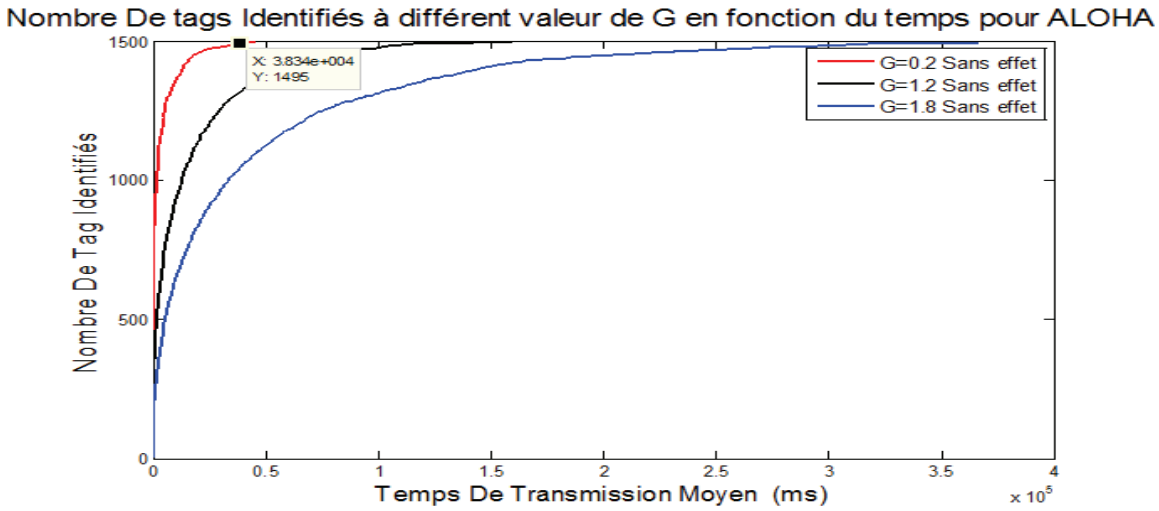


Fig.3.3. Nombre de tags identifiés en fonction du temps à différentes valeurs d'espérance de l'intervalle de génération de paquet pour le protocole Aloha.

❖ **Discussion des résultats**

Quand l'effet de capture n'est pas considéré, c-à-d lors de la simulation du protocole Aloha sous un canal de communication filaire (idéal) :

- ✓ la sortie (Throuput) est près de la valeur théorique même si le nombre de tags est 1500 ou autre nombre. Elle présente un maximum de 18.39% pour un trafic de 0.5 (Fig. 3.1).
- ✓ le délai moyen de transmission donné en paquet est représenté dans la Fig. 3.2, il atteint la valeur 34740 pour une valeur du trafic de 2.
- ✓ Le nombre de tags identifiés en fonction du temps à différentes valeurs d'espérance de l'intervalle de génération de paquets pour le protocole Aloha est présenté dans la Fig. 3.3. Nous remarquons que le temps nécessaire pour identifier le même nombre de tags augmente lorsque la valeur d'espérance de l'intervalle de génération de paquet augmente aussi, et ce temps est proportionnel à la valeur d'espérance de l'intervalle de génération des paquets. Pour la valeur de l'intervalle de génération des paquets, correspond à G=0.2, nous arrivons à identifier 1495 tags dans un temps de 38.34 s.

❖ **Simulation sous une voie de communication sans fil**

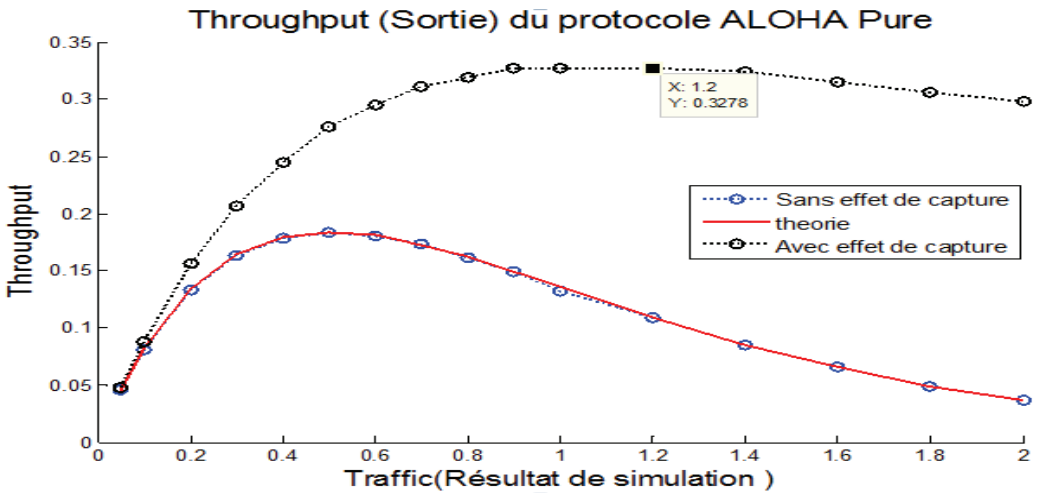


Fig.3.4. Résultat d'évaluation de la sortie (théorie, réelle) lors de la simulation du protocole Aloha sous une voie de communication filaire et sans fil.

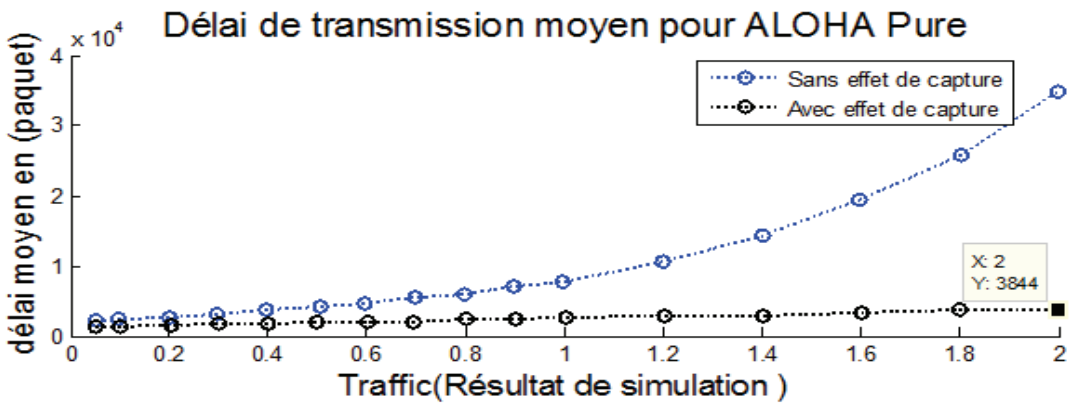


Fig.3.5.Comparaison des résultats d'évaluations de délai moyen de transmission lors de la simulation du protocole Aloha pour les deux voies de communication.

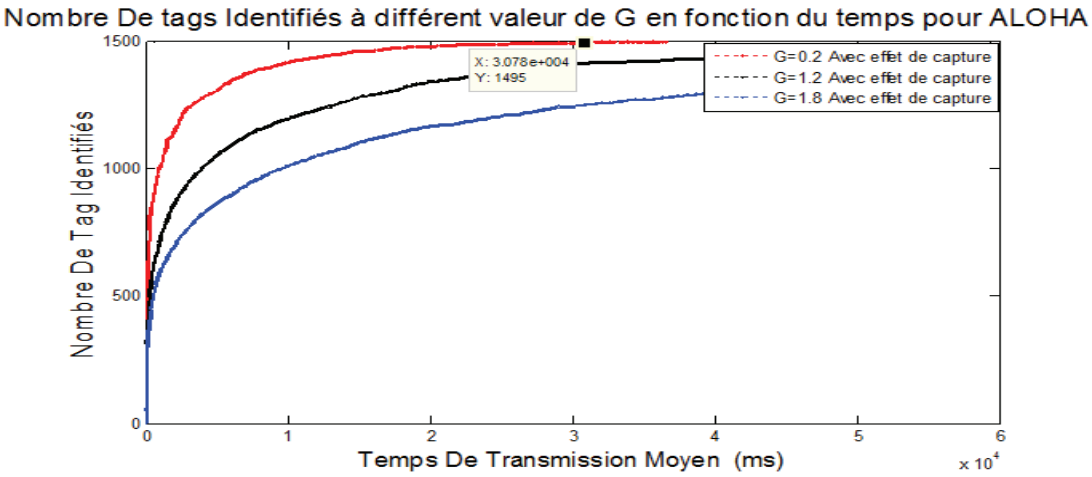


Fig.3.6. Nombre de tags identifiés en fonction du temps à différentes valeurs d'espérance de l'intervalle de génération du paquet pour le protocole Aloha.

❖ Discussion des résultats

- Quand l'effet de capture est considéré, c-à-d lors de la simulation du protocole Aloha sous une voie de communication sans fil (modèle adopté) :

✓ la sortie est plus grande que celle lorsque l'effet de capture n'est pas considéré. Elle présente un maximum de 32.78% pour un trafic de 1.2. Donc la sortie est améliorée voir (Fig. 3.4).

✓ le délai moyen de transmission donné en paquet est représenté dans la Fig. 3.5, il est réduit à 3844 pour une valeur du trafic de 2.

✓ Le nombre de tags identifiés en fonction du temps pour la même valeur d'espérance de l'intervalle de génération du paquet, utilisée dans le cas de l'effet de capture pour le protocole Aloha, est présenté dans la Fig. 3.6. Nous remarquons que le temps nécessaire pour identifier le même nombre de tags (1495 tags) est réduit à 30.78 s.

Nous pouvons dire que le modèle théorique; c-à-d le choix de la valeur d'espérance de l'intervalle de génération des paquets; a une certaine influence sur les résultats de performances du système. Donc la valeur optimale de Tint c'est la valeur pour laquelle la ligne de transmission sera bien exploitée et donne un meilleur résultat de performances pour le système.

Dans le cas d'ALOHA, les collisions se produisent souvent; donc, l'effet de capture est une raison pour améliorer les performances du protocole.

4.1.3. Slotted aloha

Les résultats de simulation de la sortie, de délai moyen de transmission et l'évaluation en termes de temps pour le protocole d'anticollision de Slotted ALOHA sont montrés dans les figures Fig. 3.7 jusqu'à Fig. 3.12.

a) Résultats de la simulation avec un seuil de capture Tcn=3dB

❖ **Simulation sous une voie de communication filaire**

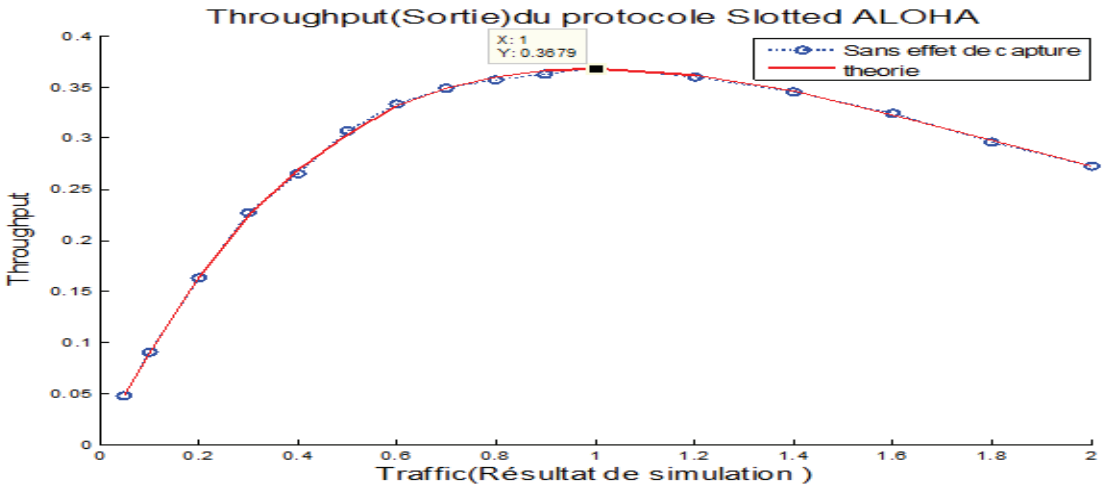


Fig.3.7. Résultat d'évaluation de la sortie (théorie et réelle) lors de la simulation du protocole S-Aloha sous une voie de communication filaire

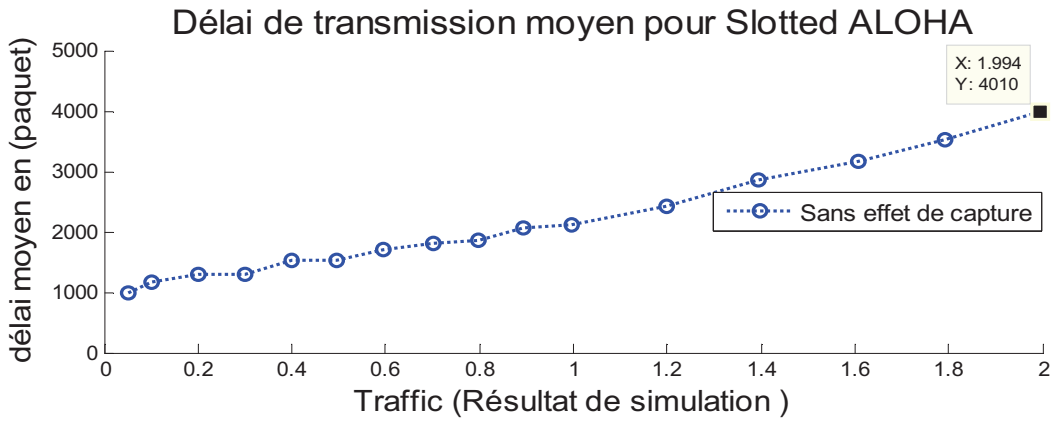


Fig.3.8. Résultat d'évaluation de délai moyen de transmission lors de la simulation du protocole S-Aloha sous une voie de communication filaire

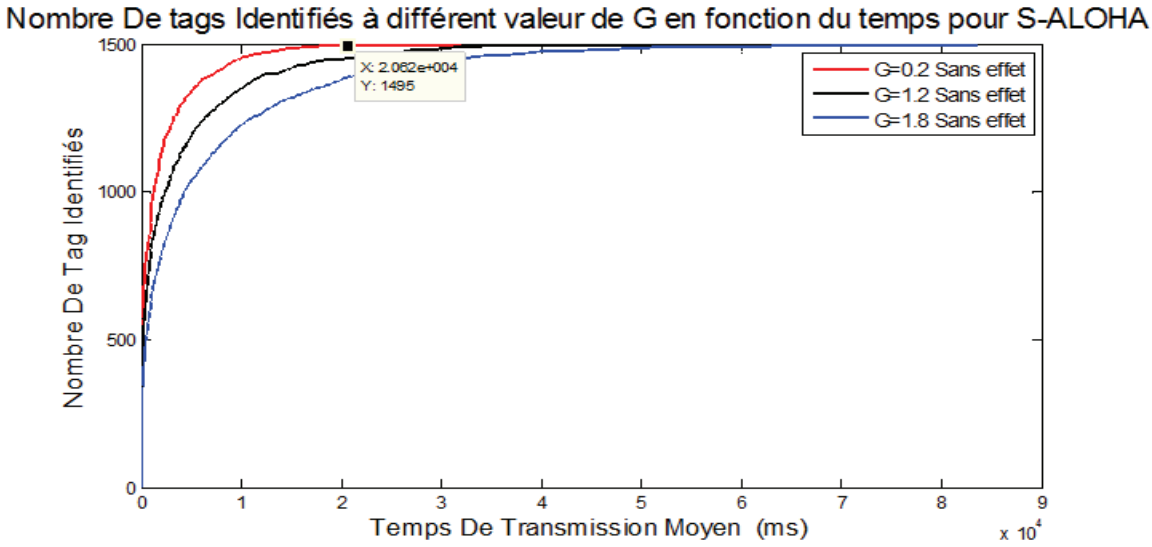


Fig.3.9. Nombre de tags identifiés en fonction du temps à différentes valeurs d'espérance de l'intervalle de génération du paquet pour le protocole S-Aloha.

❖ **Discussion des résultats**

- Quand l'effet de capture n'est pas considéré, c-à-d lors de la simulation du protocole Slotted Aloha sous une voie de communication filaire (idéale) :

✓ La sortie (Throuput) est près de la valeur théorique même si le nombre de tags est 1500 ou autre nombre. Elle présente un maximum de 36.79% pour un trafic de 1 (Fig. 3.7). Elle est améliorée par rapport à ALOHA.

✓ Le délai moyen de transmission donné en paquet représenté dans la Fig. 3.8 est réduit par rapport à ALOHA, il atteint la valeur 4010 pour une valeur du trafic de 2.

✓ Le nombre de tags identifiés en fonction du temps pour la même valeur d'espérance de l'intervalle de génération du paquet, utilisée dans le cas de l'effet de capture pour le protocole Aloha, est présenté dans la Fig. 3.9. Nous remarquons que le temps nécessaire pour identifier le même nombre de tags (1495 tags) est réduit par rapport au cas d'ALOHA à 20.62 s.

❖ **Simulation sous une voie de communication sans fil**

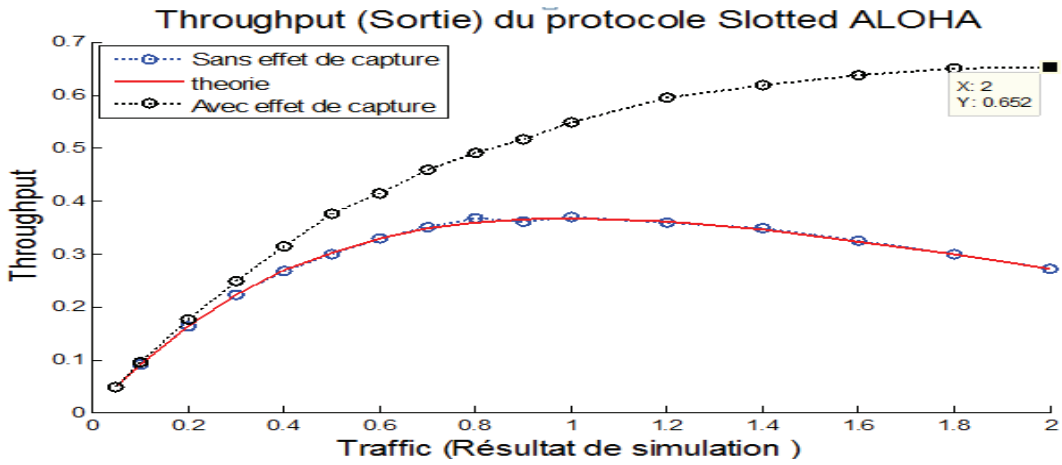


Fig.3.10. Résultat d'évaluation de la sortie (théorie, réelle) lors de la simulation du protocole S-Aloha sous une voie de communication filaire et sans fil.

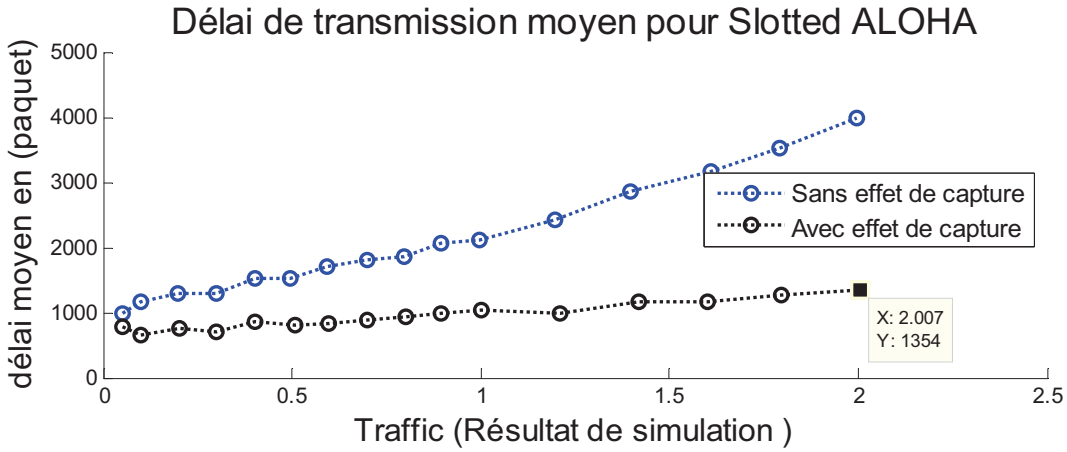


Fig.3.11.Comparaison des résultats d'évaluations de délai moyen de transmission lors de la simulation du protocole S-Aloha pour les deux voies de communication.

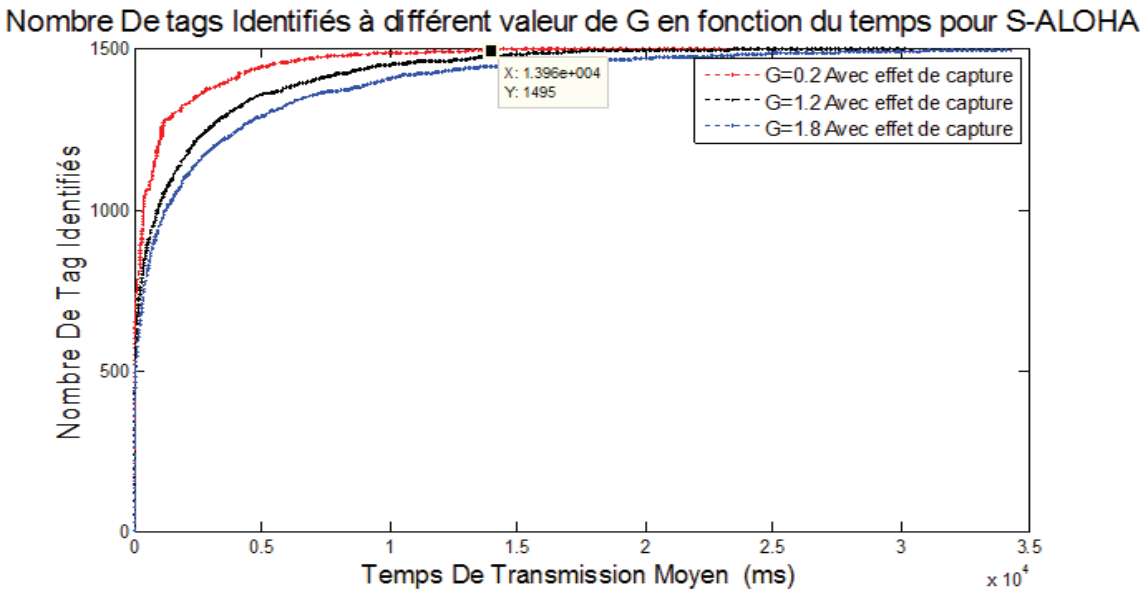


Fig.3.12. Nombre de tags identifiés en fonction du temps à différentes valeurs d'espérance de l'intervalle de génération du paquet pour le protocole S-Aloha.

❖ Discussion des résultats

Quand l'effet de capture est considéré, c-à-d lors de la simulation du protocole slotted aloha sous une voie de communication sans fil (modèle adopté) :

- ✓ La sortie est plus grande que celle lorsque l'effet de capture n'est pas considéré. Elle présente un maximum de 65.2% pour un trafic de 2. Donc la sortie est améliorée par rapport à celle d'Aloha pure (voir Fig. 3.10).
- ✓ Le délai moyen de transmission donné en paquet est représenté dans la Fig. 3.11, il est réduit à 1354 pour une valeur du trafic de 2.
- ✓ Le nombre de tags identifiés en fonction du temps pour la même valeur d'espérance de l'intervalle de génération du paquet, utilisée dans le cas de l'effet de capture pour le protocole Slotted Aloha, est présenté dans la Fig. 3.12. Nous remarquons que le temps nécessaire pour identifier le même nombre de tags (1495 tags) est réduit à 13.96 s.

Nous pouvons dire que le modèle théorique c-à-d le choix de la valeur d'espérance de l'intervalle de génération des paquets a une certaine influence sur les résultats de performances du système. Donc il faut toujours choisir la valeur optimale de Tint qui est la valeur pour la quelle la ligne de transmission sera bien exploitée et donne un meilleur résultat de performances pour le système.

Donc nous pouvons dire que dans le protocole Slotted Aloha l'influence de l'effet de capture est remarquable et le protocole S-aloha améliore les performances par rapport à celles d'Aloha.

4.1.4. np-CSMA

Les résultats de simulation de la sortie, du délai moyen de transmission et l'évaluation en termes de temps pour le protocole d'anticollision de np-CSMA pour un nombre de tags de 1000 sont montrés dans les figures Fig. 3.13 jusqu'à Fig. 3.16.Ainsi que deux autres courbes du protocole ALOHA en raison de comparaison sont aussi simulées sous même nombre de tags (1000) comme le montrent les figures Fig. 3.17 et Fig. 3.18.

- **Résultats de la simulation du protocole npcsma avec un seuil de capture T=3dB pour 1000 tags**

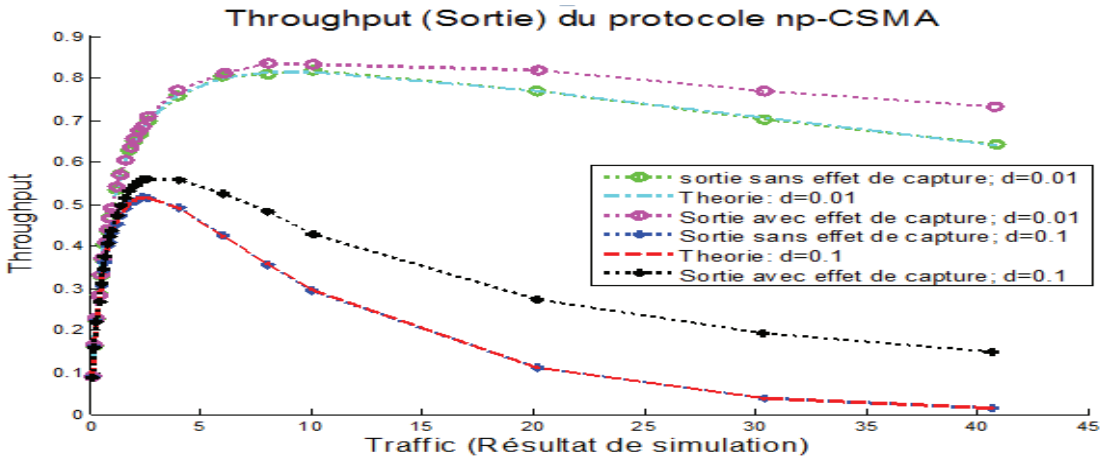


Fig.3.13. Résultat d'évaluation de la sortie (théorie, réelle) lors de la simulation du protocole np-CSMA avec différentes valeurs du délai de propagation normalisé d.

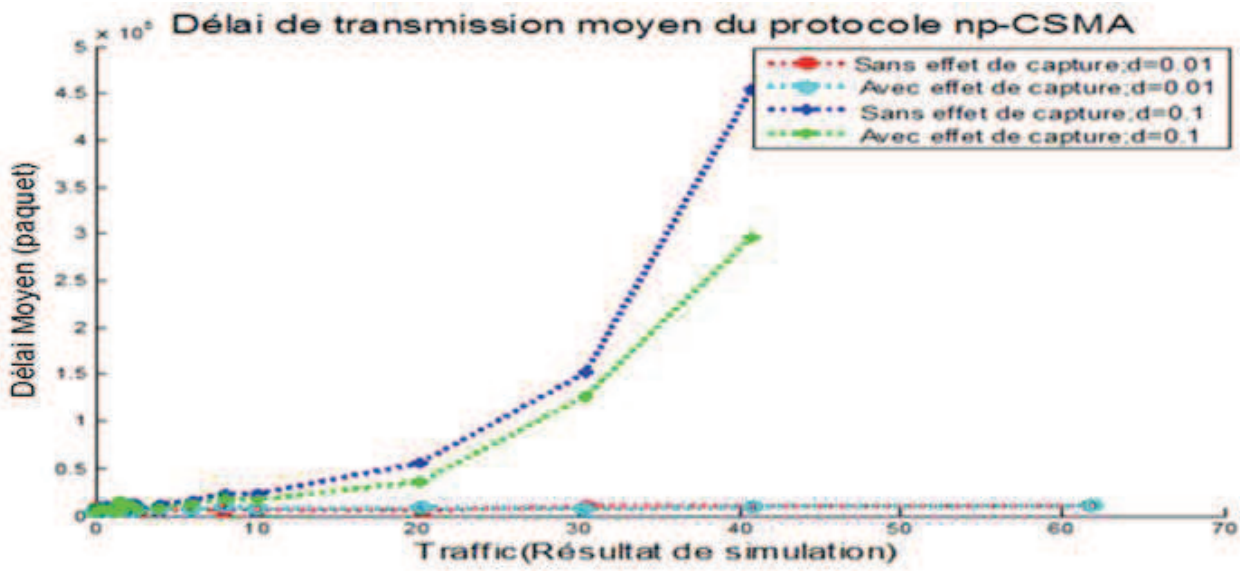


Fig.3.14.Résultats d'évaluations de délai moyen de transmission lors de la simulation du protocole np-CSMA avec différentes valeurs du délai de propagation normalisé d.

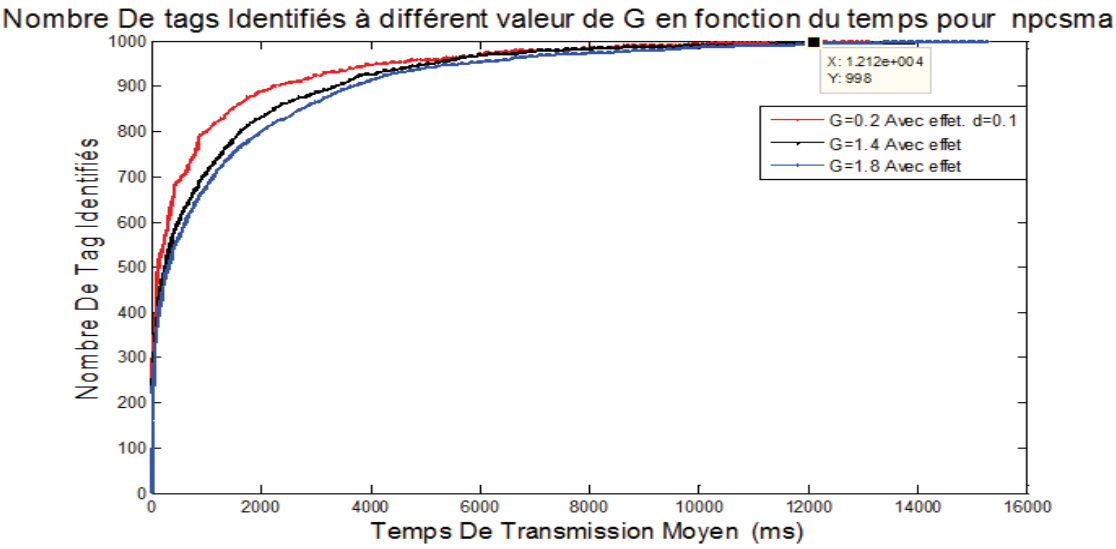


Fig.3.15. Nombre de tags identifiés en fonction du temps à différentes valeurs d'espérance de l'intervalle de génération du paquet pour le protocole np-CSMA avec d=0.1.

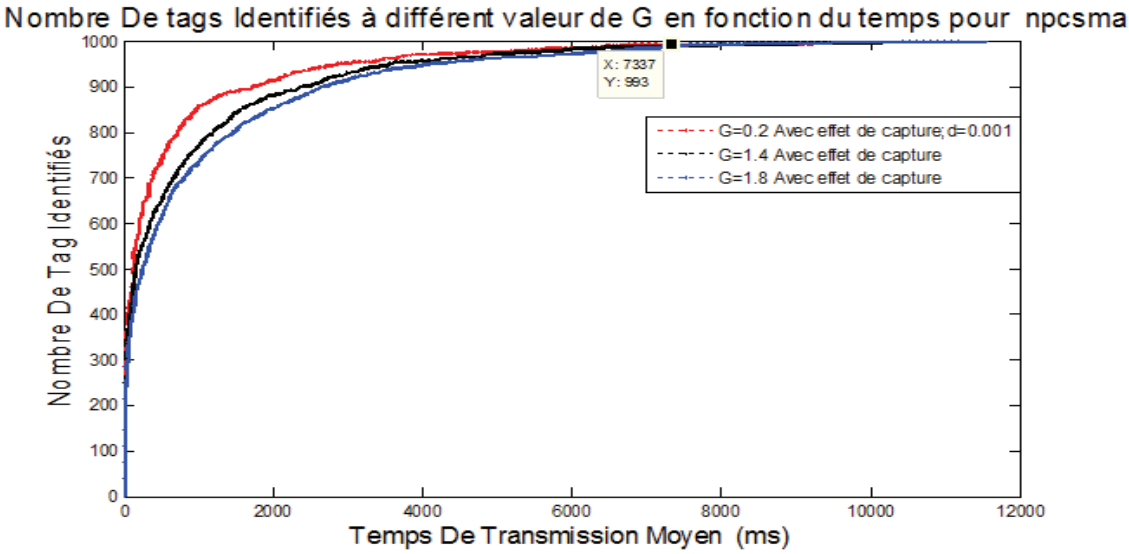


Fig.3.16. Nombre de tags identifiés en fonction du temps à différentes valeurs d'espérance de l'intervalle de génération du paquet pour le protocole np-CSMA avec d=0.001.

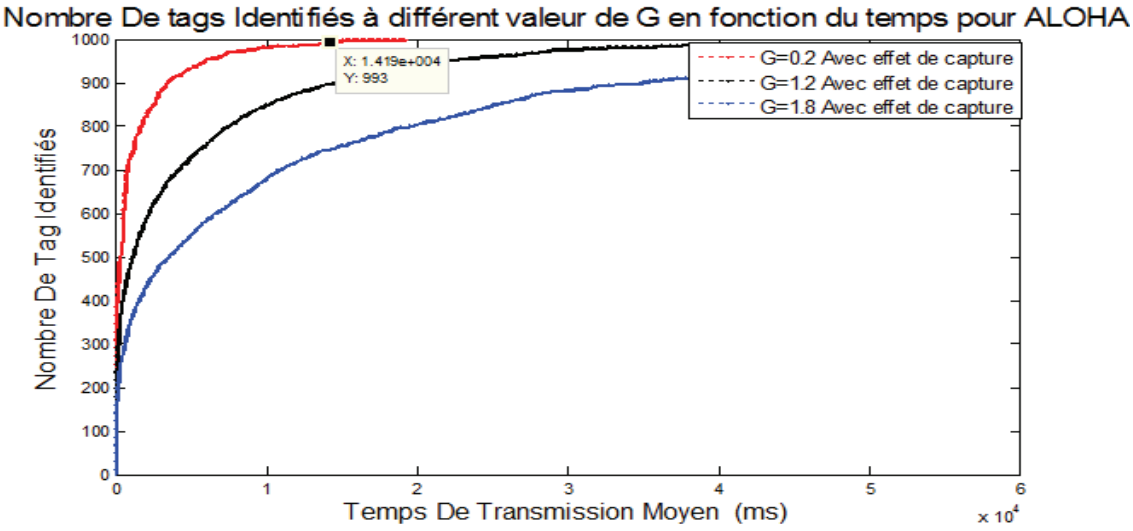


Fig.3.17. Nombre de tags identifiés en fonction du temps à différentes valeurs d'espérance de l'intervalle de génération du paquet pour le protocole ALOHA avec 1000 tags.

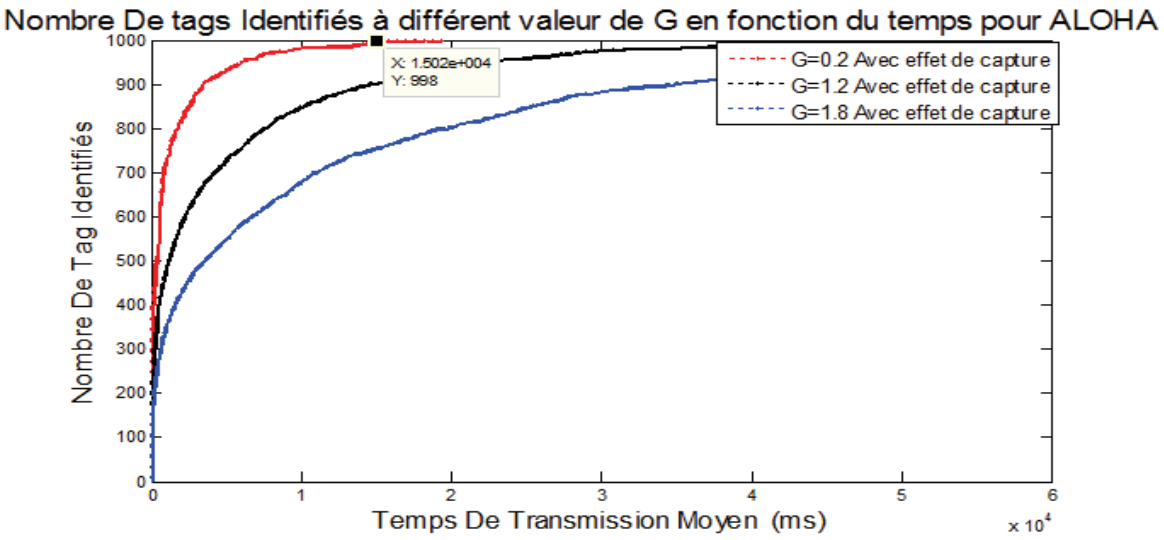


Fig.3.18. Nombre de tags identifiés en fonction du temps à différentes valeurs d'espérance de l'intervalle de génération du paquet pour le protocole ALOHA avec 1000 tags.

❖ **Discussion des résultats**

Dans la simulation, le délai de propagation normalisé (d) était 0,01 ou 0,1 et aussi 0.001. Nous remarquons d'après les courbes que la sortie maximale de np-CSMA dépend du délai de propagation normalisé (d). Si *d* est petit, la sortie maximale de np-CSMA est plus grande que celle d'ALOHA. Cependant, si *d est* grand, la performance est près d'ALOHA. C'est parce que d'autres tags peuvent transmettre leurs paquets pendant ce grand temps de propagation "d" même si un tag transmet son paquet après la sensation de la porteuse. En conséquence, la collision se produit. D'ailleurs, si l'effet de capture est considéré, le paquet transmis survit parfois en raison de la différence de la puissance reçue de chaque tag. Par conséquent, une amélioration des performances du protocole est obtenue.

4.1.5. Influence de la sensibilité du lecteur sur les performances des protocoles

Pour étudier l'influence de la sensibilité du lecteur ou bien le rapport de capture sur les performances des protocoles, nous avons simulé les protocoles sous une nouvelle valeur de l'effet de capture qui est de T=6dB en gardant les conditions de simulation précédentes telles qu'elles sont. Les figures de Fig. 3.19 jusqu'à Fig. 3.23 montrent les résultats de simulation auxquels nous avons abouti.

❖ **Résultats de la simulation avec un seuil de capture T=6dB**

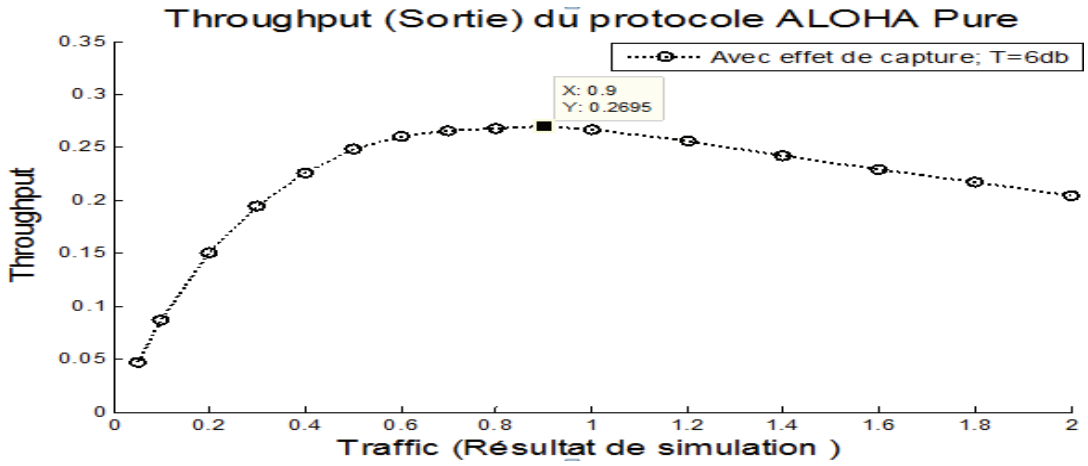


Fig. 3.19. Résultat d'évaluation de la sortie (réelle) lors de la simulation du protocole Aloha sous une voie de communication sans fil avec T=6dB.

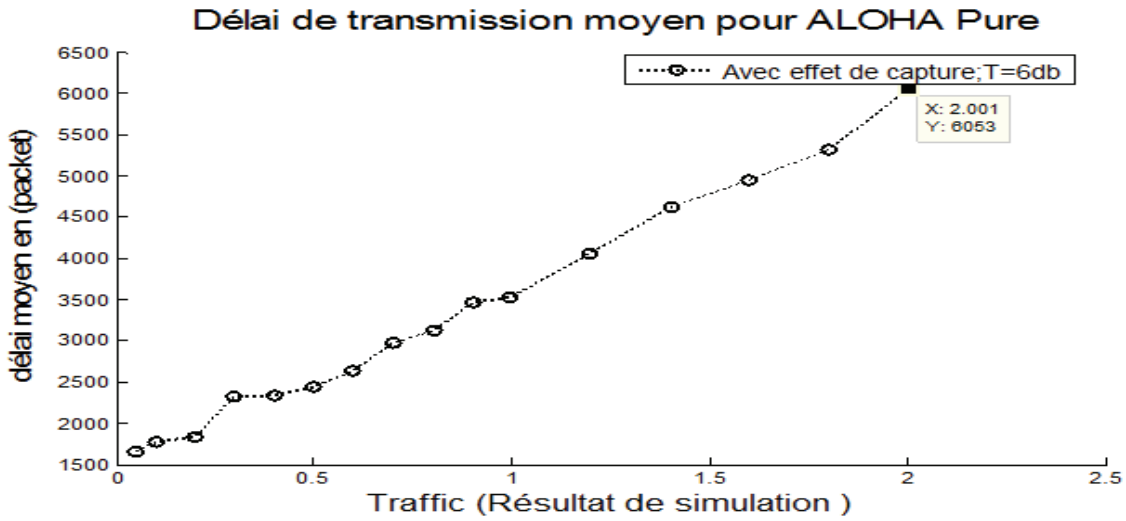


Fig. 3.20. Résultat d'évaluation de délai moyen de transmission lors de la simulation du protocole Aloha sous une voie de communication sans fil avec T=6dB.

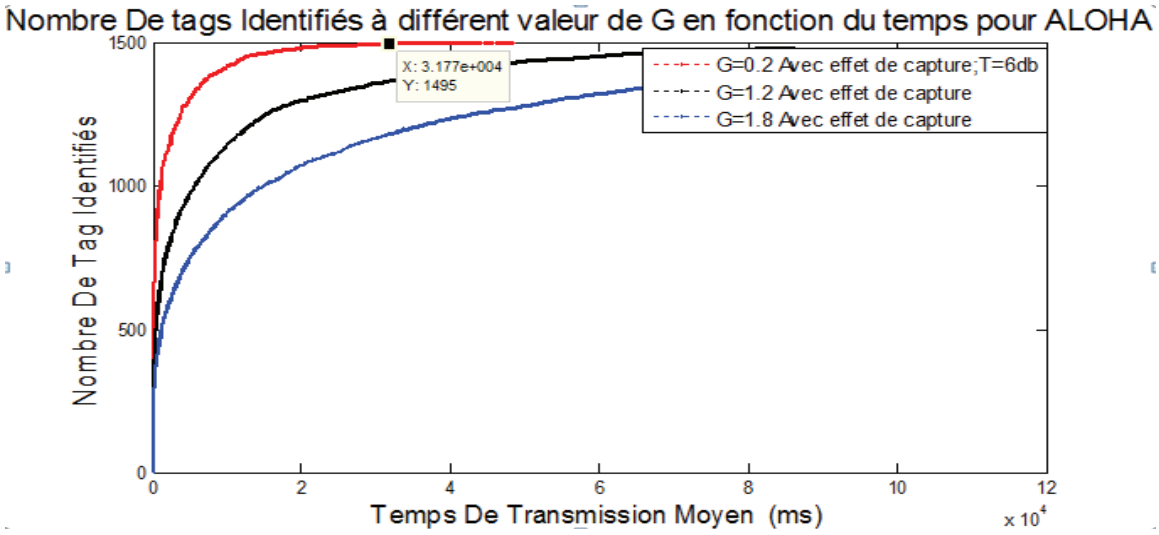


Fig. 3.21. Nbre de tags identifiés en fonction du temps à différentes valeurs d'espérance de l'intervalle de génération du paquet pour le protocole Aloha avec T=6dB.

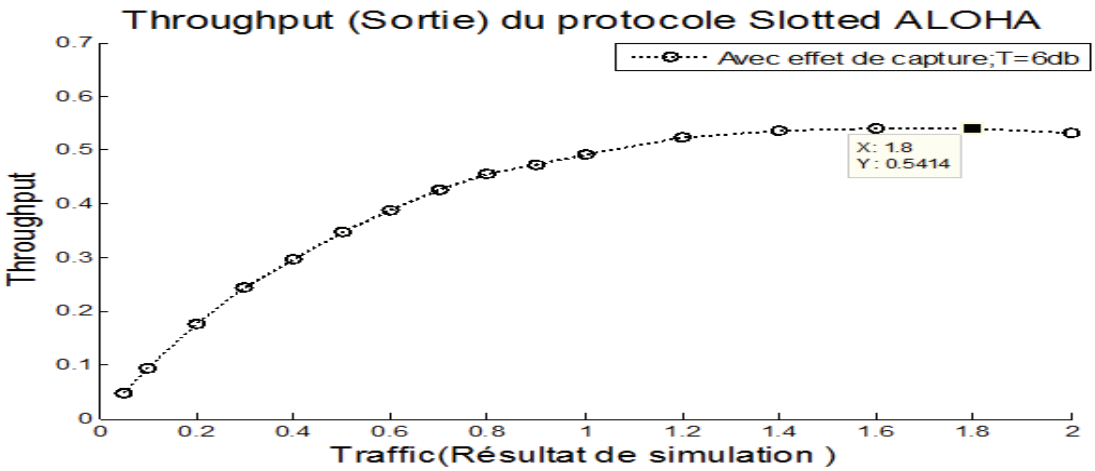


Fig. 3.22. Résultat d'évaluation de la sortie (réelle) lors de la simulation du protocole S-Aloha sous une voie de communication sans fil avec T=6dB.

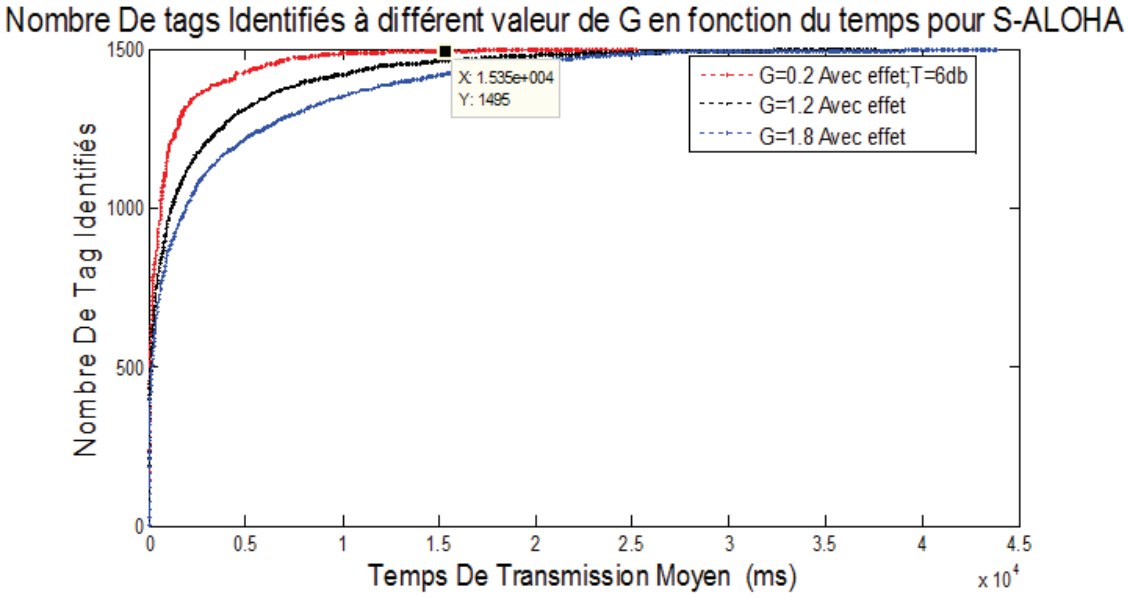


Fig. 3.23. Nombre de tags identifiés en fonction du temps à différentes valeurs d'espérance de l'intervalle de génération du paquet pour le protocole S-Aloha avec T=6dB.

❖ Discussion des résultats

- ✓ La sortie (Throuput) du protocole Aloha présente un maximum de 26.95% pour un trafic de 0.9 (Fig.3.19). Elle est diminuée par rapport au cas de seuil T=3dB qui était 32.78%(voir Fig.3.4).
- ✓ Le délai moyen de transmission donné en paquet du protocole Aloha est représenté dans la Fig.3.20, il augmente à 6053 pour une valeur du trafic de 2 par rapport au cas de seuil T=3dB qui était 3844 (voir Fig.3.5).
- ✓ Le temps nécessaire pour identifier le même nombre de tags (1495 tags) dans le cas d'Aloha augmente à 31.77 s (voir Fig.3.21) par rapport au cas de seuil T=3dB qui était 30.78 (voir Fig.3.6).

- ✓ La sortie (Throuput) du protocole S-Aloha présente un maximum de 54.14% pour un trafic de 1.8 (Fig.3.22). Elle est diminuée par rapport au cas de seuil T=3dB qui était 65.2% pour un trafic de 2(voir Fig.3.10).
- ✓ Le temps nécessaire pour identifier le même nombre de tags (1495 tags) dans le cas de S-Aloha augmente à 15.35 s (voir Fig.3.23) par rapport au cas de seuil T=3dB qui était 13.96 s (voir Fig.3.12).

La valeur T indique un seuil, choisi au niveau du lecteur, appelé le rapport de capture. Il correspond à la valeur minimale du rapport de la plus grande puissance reçue d'une étiquette spécifique dans la zone du lecteur, sur la somme des puissances reçues des étiquettes restantes qui occupent le même slot, afin d'être identifiées par le lecteur [43].

- ✓ La sortie atteint sa valeur maximale de $(1 + T)/e^{T}$ quand le trafic G est égal à $1 + 1/T$.

D'après les résultats obtenus nous pouvons conclure que pour un seuil T petit, la performance du système RFID augmente.

Le choix d'un seuil bas dans le lecteur permet d'éliminer le problème de collision, or ce cas ne peut pas se produire dans la pratique, car T est déterminé par la sensibilité du lecteur.

4.2. Méthodes déterministes

4.2.1. Environnement de simulation des protocoles d'anticollision déterministes

Nous avons simulé les protocoles d'anticollision de la famille déterministes dans la norme EPC Class 1 Gen 2 proposée par EPC global à ISO/ICE C 18000-6 qui exige deux conditions : la première sur la longueur de l'identifiant qui doit être de 96 bits, la seconde porte sur l'unicité d'identifiant car chaque tag doit avoir son propre identifiant.

En vue de satisfaire ces exigences, nous avons adopté le modèle à couche MAC idéalisé qui reste suffisant pour exprimer les concepts pertinents des protocoles basés sur les arbres.

4.2.2. Simulation des protocoles d'anticollision déterministes

Cette simulation porte sur la comparaison des performances entre différents protocoles RFID d'anticollision d'étiquettes basés sur l'arbre et aussi sur la présentation du déroulement du processus d'identification de chaque protocole.

Ainsi, pour des raisons de comparaison entre les résultats de la simulation et ceux de la théorie, l'arbre binaire, ainsi que le déroulement du processus correspondant à l'arbre sont montrés pour chaque protocole. Ainsi l'efficacité de l'algorithme Binary tree est présentée.

Pour cela nous avons pris le même exemple de cinq étiquettes dans la zone d'interrogation du lecteur avec les mêmes identifiants.

Nous évaluons les bits totaux (bits de préfixe et bits des réponses), ainsi le nombre d'itérations nécessaires pour l'identification des cinq étiquettes pour chaque protocole.

Les résultats de simulation sont montrés dans les schémas suivants :

a) Le protocole binary tree (BTA):

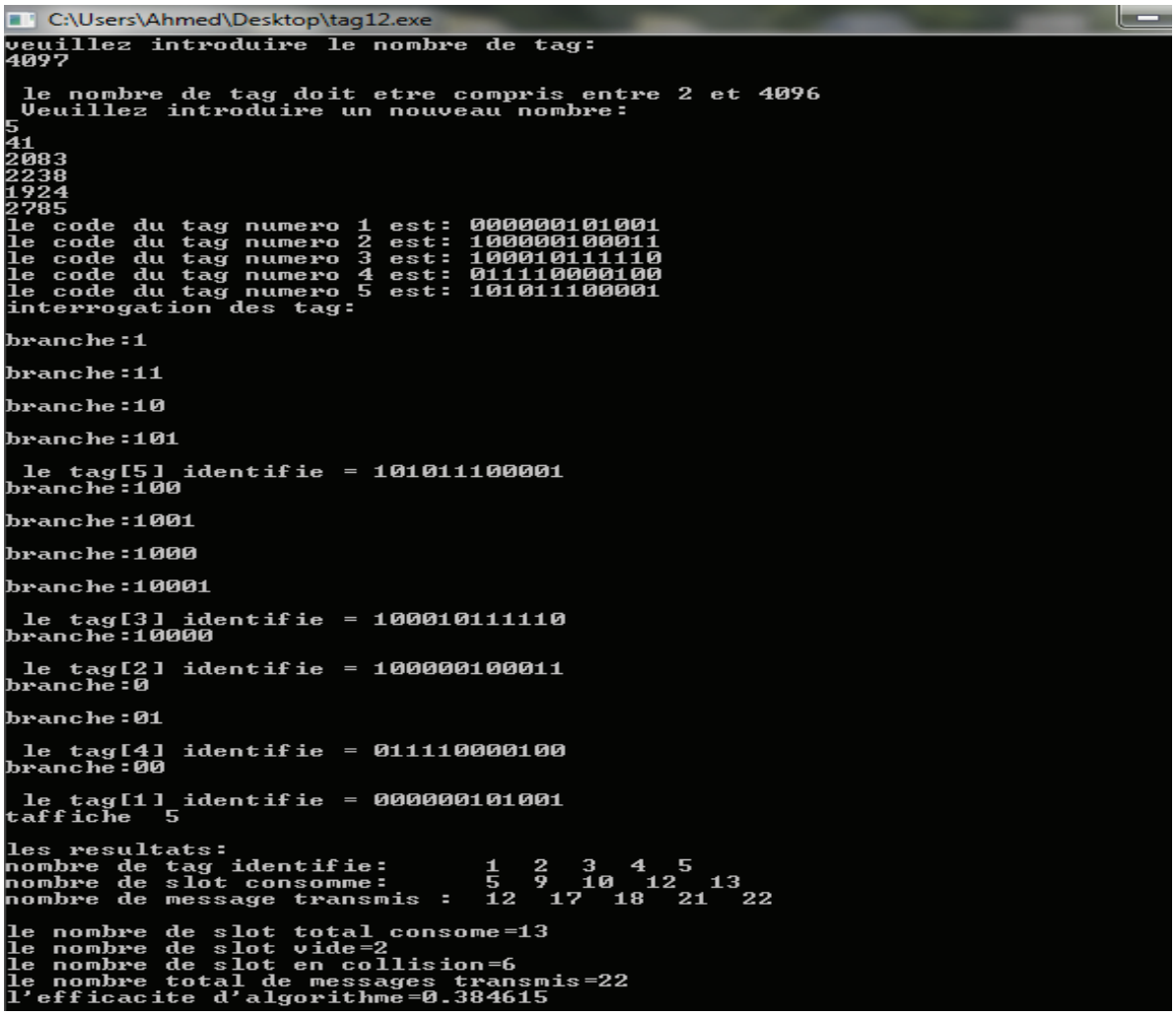


Fig.3.24.Simulation du protocole binary tree avec l'exemple de 5 tags.

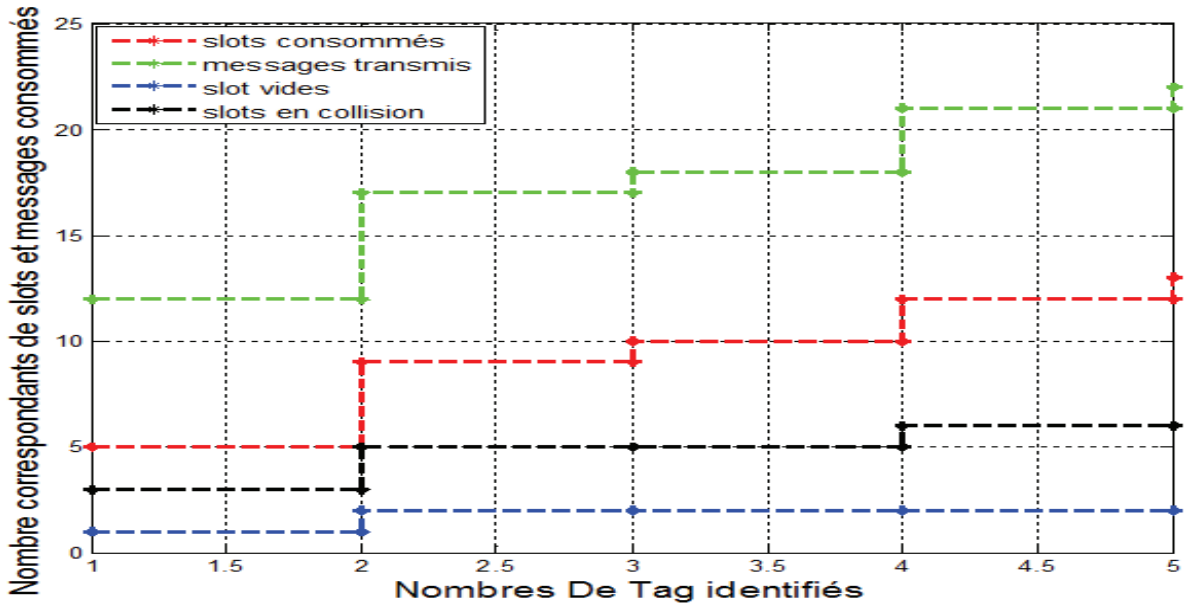


Fig. 3.25. Chronogramme complet de l'exécution du protocole avec l'exemple de 5 tags.

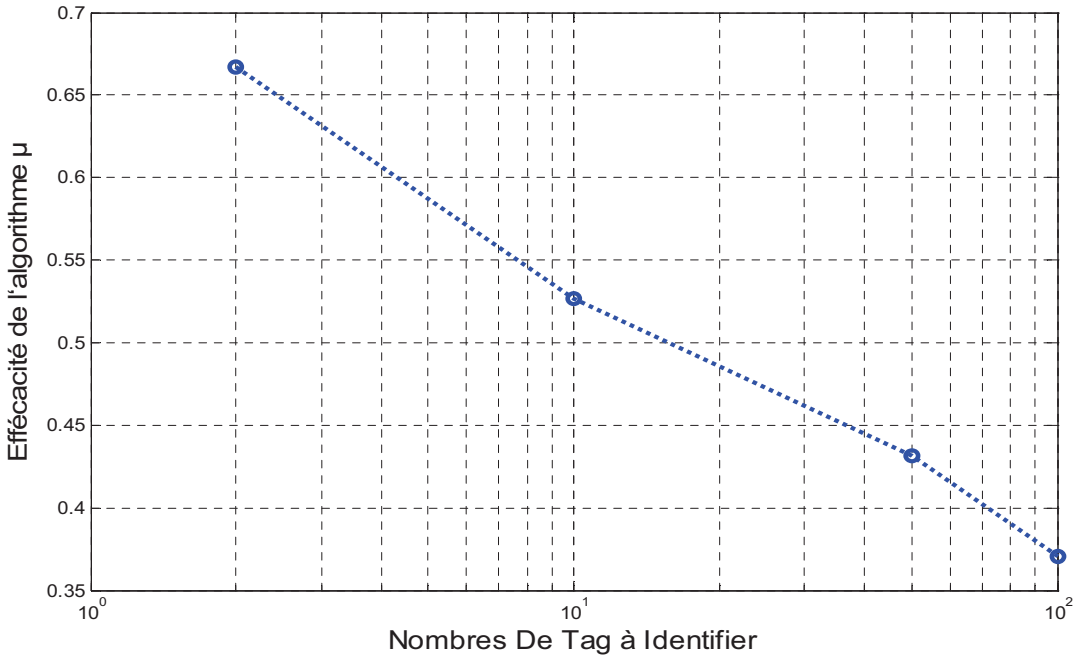


Fig.3.26. Efficacité de l'algorithme *binary tree* en fonction du nombre de tags à identifier.

❖ Discussion des résultats

La figure 3.25 représente un chronogramme complet de l'exécution du protocole pour un cas de 5 tags. Nous remarquons que l'arbre binaire, ainsi que son processus d'identification sont identiques à ceux discutés dans la partie théorique du chapitre 2 où ils sont représentés par les Fig. 2.15 et Fig. 2.16. Comme il est illustré à la Fig. 3.24, le premier tag identifié est le tag numéro 5. Le nombre de messages transmis, le nombre de slots vides, le nombre de slots en collision corresponds à cet instant sont respectivement 12 messages, 1 slot, 3slots (2 plus le slot

de la requête initiale) et l'instant correspondant à l'identification du dernier tag (tag 1), le nombre de messages transmis, le nombre de slots vides, le nombre de slots en collision seront respectivement 22 messages, 2 slots, 6slots.

Dans ce cas, le nombre total des slots utilisés pour identifier les cinq étiquettes est de 13 slots.

L'efficacité du protocole d'arbitrage binay tree est évaluée en fonction du temps et du nombre de messages donnés dans le chapitre 2 par l'équation 2.14. Nous avons simulé le protocole pour plusieurs valeurs de n (nombre d'étiquette dans la zone d'interrogation du lecteur). La Fig. 3.26 présente le tracé de l'efficacité de l'algorithme en fonction du nombre de tags à identifier (n).

Nous avons constaté que l'efficacité de l'algorithme diminue lorsque la densité des tags augmente dans la zone d'interrogation du lecteur, et ceci est dû à l'augmentation du nombre de slots en collision.

b) Le protocole query tree (QTA)

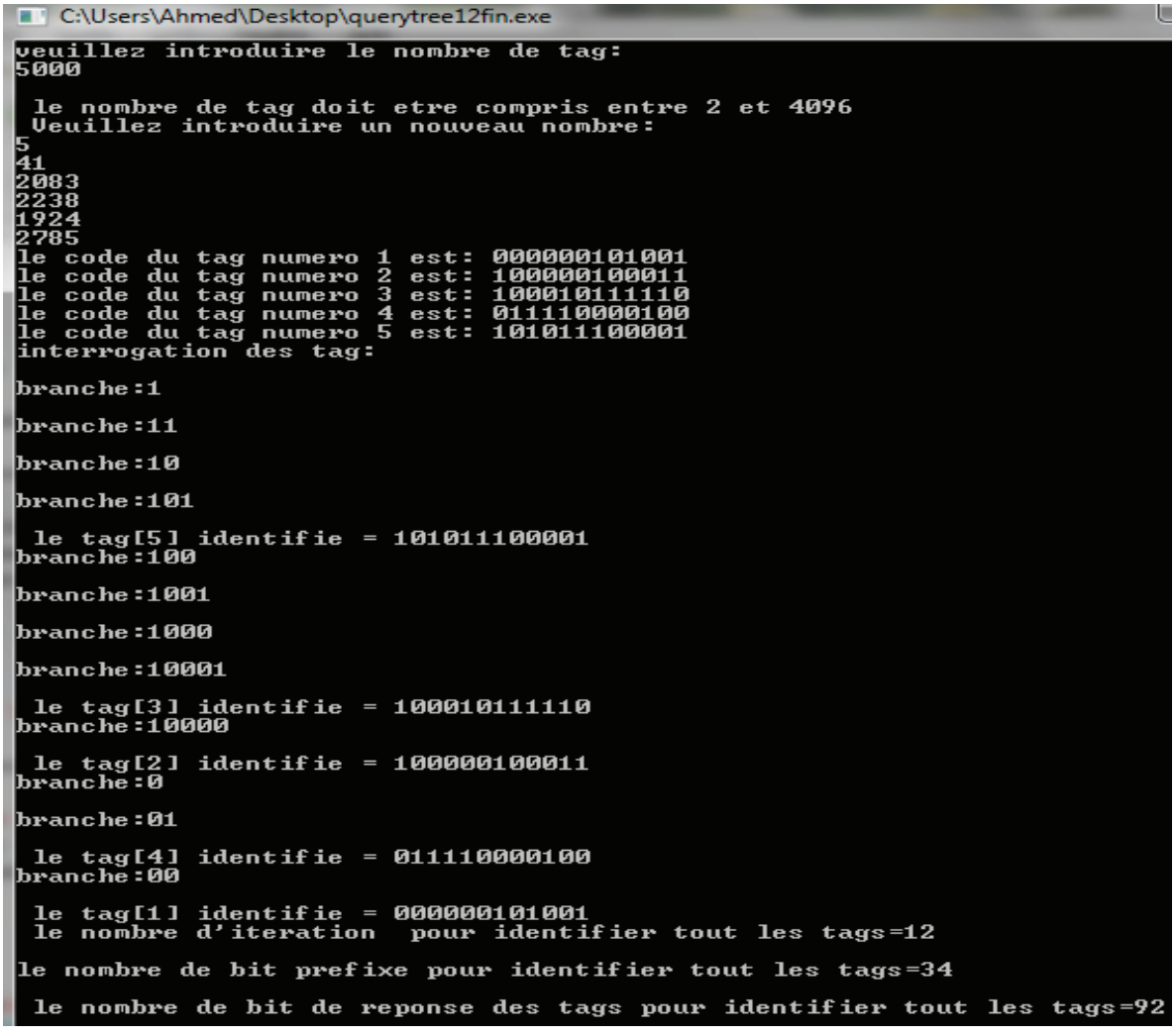


Fig.3.27. Simulation du protocole Querytree avec un l'exemple de 5 tags.

c) Le protocole collision tracking tree (CTTA)

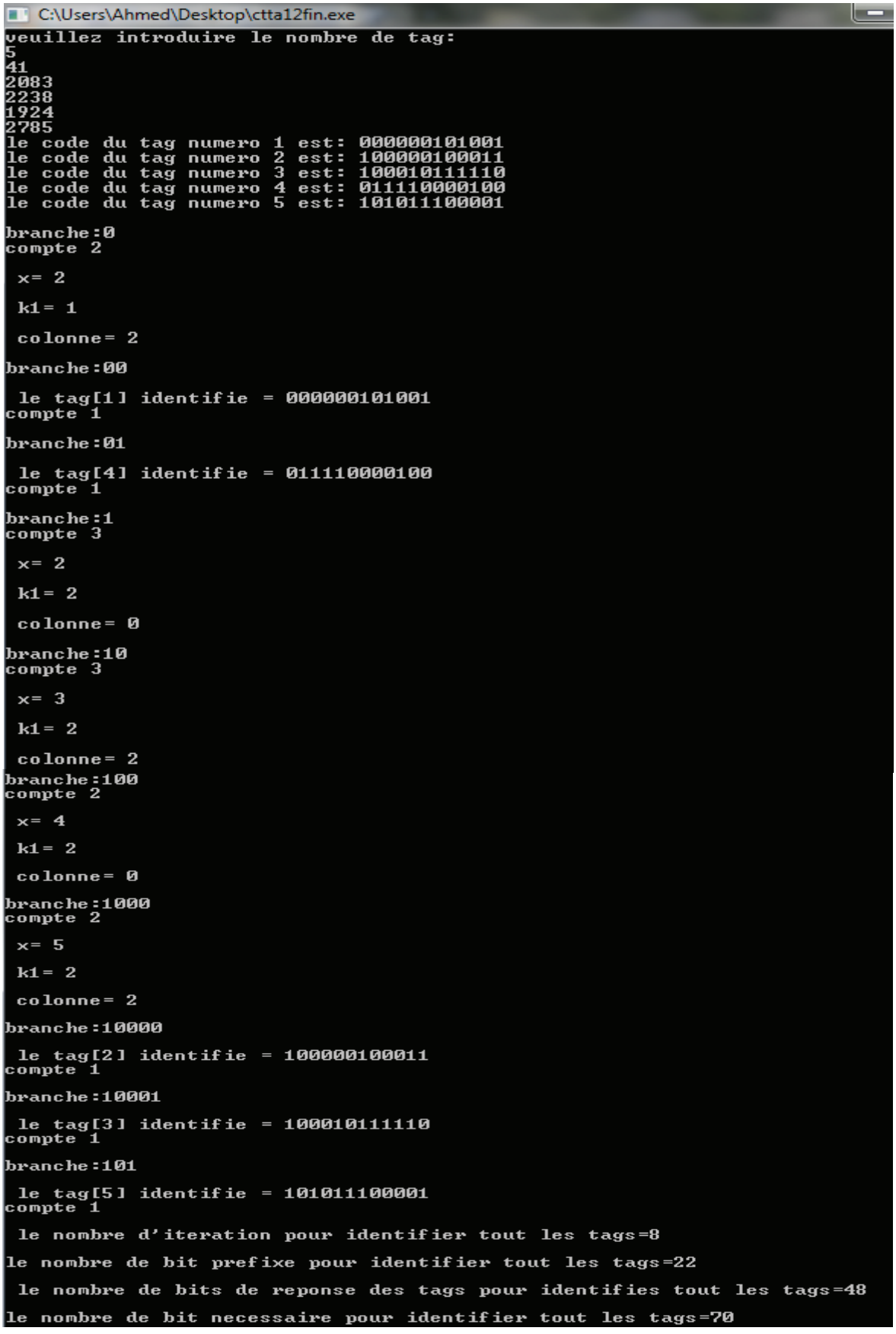


Fig.3.28.Simulation du protocole CTTA avec l'exemple de 5 tags.

d) Le protocole bi-slotted query tree (BSQTA)

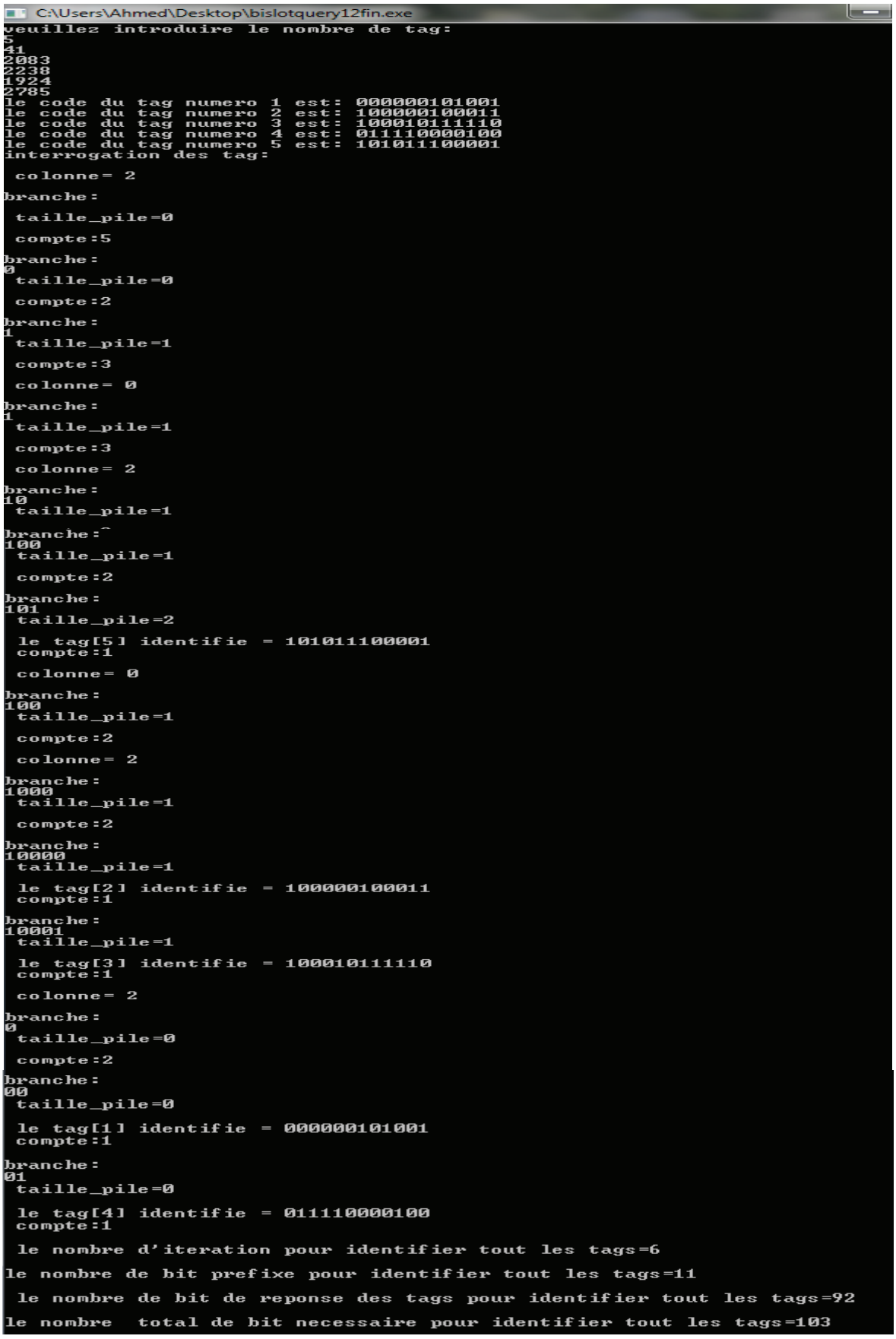


Fig.3.29.Simulation du protocole BSQTA avec l'exemple de 5 tags.

e) Le protocole bi-slotted collision tracking tree (BSCTTA):

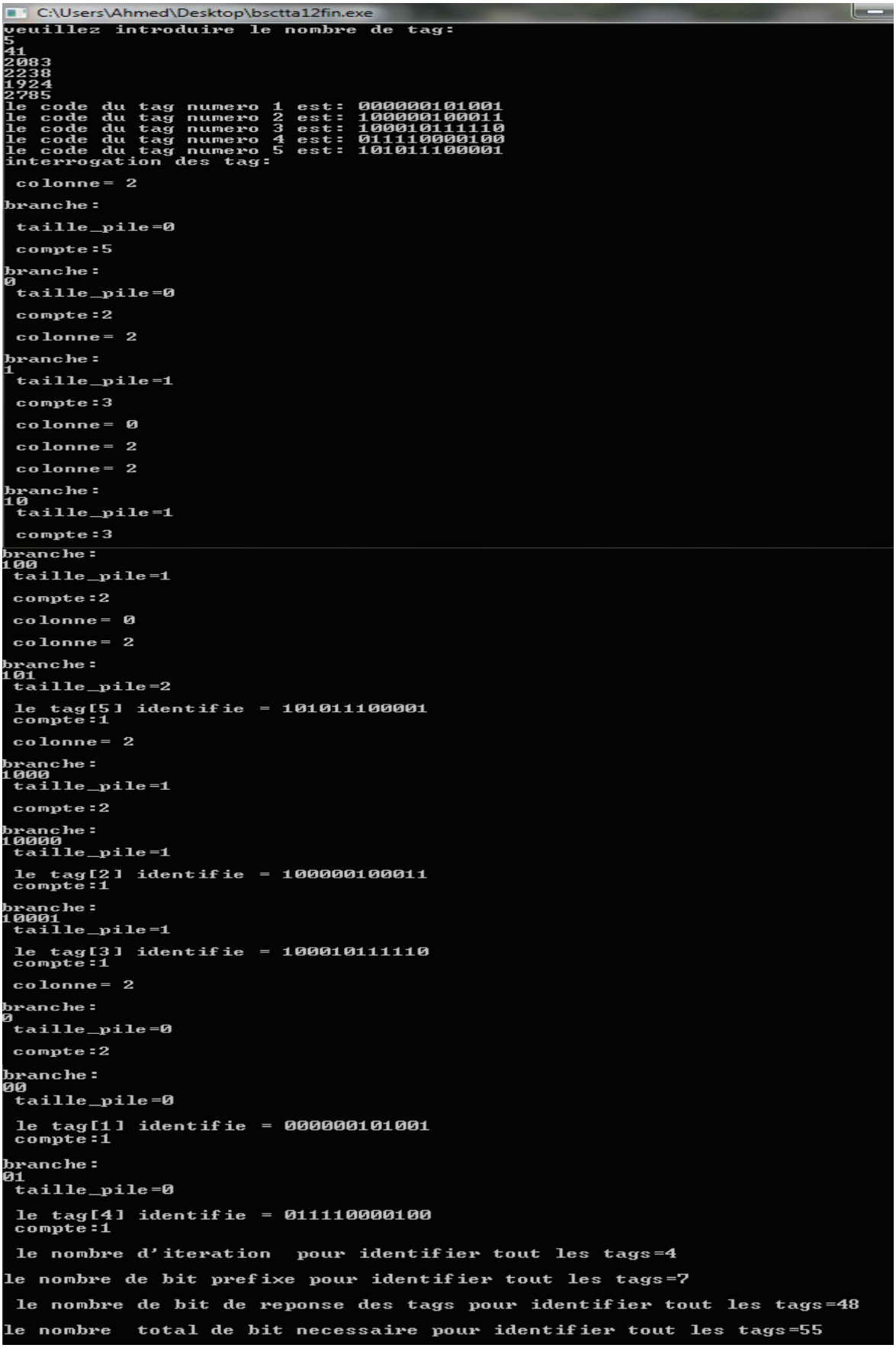


Fig.3.30. Simulation du protocole BSCTTA avec l'exemple de 5 tags.

❖ Discussion des résultats des protocoles QTA, BS-QTA, CTTA, BS-CTTA

D'après la simulation des protocoles, nous avons remarqué que les résultats obtenus concernant les bits consommés, ainsi que le nombre d'itérations sont identiques avec ceux de la théorie.

Les résultats de la simulation obtenus pour un exemple de cinq étiquettes sont récapitulés dans le tableau suivant :

Tab.3.1 : paramètres d'évaluation des protocoles pour un exemple de 5 étiquettes

Paramètres / Protocoles	Nombre d'itération	Nombre de bit préfix	Nombre de bit de réponse
QTA	12	34	92
BS-QTA	6	11	92
CTTA	8	22	48
BS-CTTA	4	7	48

Selon les figures Fig.3.27 et Fig.3.*29,* le protocole QTA exige 34 bits de préfixes et 92 bits de réponses pour l'identification des cinq étiquettes. D'autre part, le protocole BSQTA exige 11 bits de préfixes et 92 bits de réponses. Par conséquent, en comparaison avec QTA, le protocole BSQTA a une meilleure réduction des préfixes le que QTA.

Les figures Fig.3.28 et Fig.3.30 montrent une comparaison entre le protocole CTTA et BSCTTA. Pour identifier les cinq étiquettes, le protocole CTTA exige 22 bits de préfixes et 48bits pour les réponses, ainsi 70 bits au total sont nécessaires. Quant au protocole BSCTTA exige 7 bits pour les préfixes et 48bits pour les réponses, ainsi 55 bits au total sont nécessaires. D'où, l'algorithme BSCTTA présente un nombre de bits de préfixe plus petit.

Le nombre d'itérations exigées pour l'identification de toutes les étiquettes est un autre facteur important pour évaluer la performance de système RFID, car les itérations affectées par le processeur dans le lecteur et dans l'étiquette, nécessitent un délai pour l'identification d'étiquette. Les Fig.3.27 et Fig.3.29 indiquent que BSQTA exécute la moitié des itérations qu'en QTA. Par conséquence, BSQTA réalise une performance légèrement meilleure que QTA dans la moyenne des bits exigés et l'itération moyenne exigée pour l'identification d'une étiquette. A partir des Fig. 3.28 et Fig. 3.30 nous remarquons que les itérations nécessaires pour l'identification de toutes les étiquettes dans BSCTTA sont la moitié de celles dans CTTA.

Par conséquent, les systèmes RFID appliquant BSQTA ou BSCTTA présentent une identification d'étiquette plus rapide que des systèmes RFID qui utilisent QTA ou CTTA.

5. Comparaisons entre les deux familles des algorithmes simulés:

Etant donné que chaque protocole a été défini pour satisfaire et prendre en compte une architecture système RFID bien précise, donc chaque famille présente des avantages et des inconvénients, ce qui rendre la comparaison de leurs performances difficile.

Dans les protocoles déterministes:

- On peut prévoir la durée maximale du processus d'identification, et comme l'erreur de collision peut être détectée pendant la phase d'anticollision, la vitesse peut aussi être optimisée.
- La méthode bit à bit est facile à implémenter en logique câblée.

Dans les protocoles probabilistes:

- Les performances des protocoles de cette famille sont fortement affectées par la taille utilisée de la trame. Parce que la totalité des tags à identifier n'est pas connue, donc le dimensionnement des trames est un gros problème pour les protocoles probabilistes.
- Le nombre d'échanges est minimalisé, par conséquent les perturbations radioélectriques sont réduites, Cependant, les algorithmes déterministes restent plus rapides que les algorithmes probabilistes.

Conclusion

Dans ce chapitre nous avons présenté les résultats auxquels nous avons abouti. Nous avons présenté en premier lieu les résultats obtenus par la simulation des protocoles d'anticollision stochastiques.

En seconde lieu, Nous avons présenté les résultats obtenus par la simulation des protocoles d'anticollision déterministes.

En dernier point, nous pouvons dire que pour une identification rapide d'étiquettes, les protocoles d'anticollisions qui réduisent le temps d'identification et améliorant la robustesse du système sous la contrainte du coût, sont meilleurs d'êtres implémentés.

Conclusion générale

Conclusion et perspectives

Le marché de la RFID se développe très rapidement en UHF et microondes. Ces bandes de fréquences autorisent des distances de communication importantes de plusieurs mètres et la bande UHF semble s'imposer comme une référence pour l'amélioration de la portée de lecture. Les systèmes RFID UHF sont considérés parmi des technologies les plus prometteuses utilisées pour l'identification d'objet sans contact. Pour une identification rapide d'étiquettes et sans interférences, les protocoles anticollisions sont exigés.

Ces travaux du mémoire s'inscrivent dans le contexte de l'identification radiofréquence (RFID), dans des gammes avoisinant le Gigahertz. Nous avons adressé des protocoles de résolution de collision pour des systèmes RFID et aussi le protocole d'évitement de collision np-CSMA est étudié. Nous avons défini une nouvelle métrique (le nombre de tags identifiés en fonction du temps) pour évaluer de tels protocoles. Cette nouvelle métrique fournit une mesure directe de la latence encourue par les systèmes d'anticollision RFID pour lire un groupe d'étiquettes. Basé sur cette métrique, nous avons évalué la performance des protocoles proposés existantes et nous avons également effectué une comparaison entre eux.

Nous avons consacré une autre partie qui porte sur l'évaluation des performances des protocoles d'anticollision à base d'arbre binaire à savoir le protocole BTA, QTA, CTTA, BS-QTA, et BS-CTTA et nous avons également effectué une comparaison entre eux.

Le but n'est donc pas de concevoir un nouveau type de lecteur RFID précis mais bien de maitriser les algorithmes anticollisions et fournir aux concepteurs de systèmes RFID des outils d'aide à la conception sur un point clé qu'est les protocoles d'anticollision pour améliorer les paramètres pertinents des protocoles et obtenir un système RFID fiable et efficace, qui peut identifier rapidement un nombre très important d'objets, minimisant les frais généraux consommés lors du processus d'identification d'étiquettes.

Les résultats de simulation basés sur les paramètres d'évaluation discutés dans le deuxième chapitre montrent que l'application de np-CSMA améliore les performances des procédures d'anticollision des tags par rapport aux protocoles slotted aloha et au protocole aloha qui a de plus faibles performances. Ainsi l'application de l'effet de capture dans cette famille d'algorithme anticollision améliore les performances de chaque protocole.

D'autres part, les résultats de simulation des protocoles d'anticollision à base d'arbre binaire montrent que l'application des protocoles à deux slot (*Bi-Slotted*) améliore les performances des procédures d'anticollision par rapport aux cas des protocoles QTA et CTTA.

Perspectives

Basé sur les perspicacités gagnées de l'évaluation et de la comparaison des protocoles d'anticollision étudiés dans les deux familles: aléatoire basée sur Aloha et déterministe basée sur l'arbre binaire, nos travaux ouvrent de nombreuses perspectives au niveau des protocoles MAC.

- Ces perspicacités donnent un aperçu de la façon dont ces protocoles existants peuvent être optimisés et comment un nouveau protocole peut être conçu pour réduire les inefficacités en termes de temps ou bien de puissance consommée pour résoudre les problèmes de requêtes de groupe. Pour cela nous proposons dans un premier temps de concevoir de nouveau algorithme d'anticollision utilisant la combinaison de plusieurs actuels protocoles anticollision basée sur les deux familles des protocoles (déterministe et aléatoire) afin de réaliser un système robuste plus efficace par rapport aux autres existants, et aussi robuste aux réseaux RFID s'étendant de petites tailles aux tailles très grandes sans connaissance a priori de la population d'étiquettes.
- L'étape suivante pourrait consister de rendre le processus d'identification de cet algorithme encore plus performant en incluant des modèles rapides ou bien utilisant autre techniques et pouvant aider à la prise de décision. Nous pensons ici aux modèles neuronaux ou à base de logique floue ayant largement démontré leur efficacité dans des problèmes similaires.
- L'influence de l'environnement immédiat (structure et localisation des lecteurs et/ou d'autres tags, problèmes de réflexions multiples, présence d'autres tags et/ou plusieurs lecteurs impliquant des problèmes d'anticollision, …) sur les performances du système RFID du point de vu aspect anticollision toujours, mais cette fois-ci dans la couche physique, est un point important. Il serait intéressant de rechercher s'il existe des solutions adéquates minimisant ces influences et de les inclure dans le comportement du circuit.

D'une façon plus générale, des axes de recherche très importants et prometteurs se développent actuellement et ouvrent la voie à des travaux de recherche plus vastes dans la technologie RFID.

Bibliographie

[1] Waldrop, J.; Engels, D.W.; Sarma, S.E.; " Colorwave: An Anticollision Algorithm for the Reader Collision Problem", IEEE Communications, vol. 2, pp. 1206 – 1210, 2003.

[2] Leian Liu; Dashun Yan; Xiaozheng Lai; Shengli Lai, "A New Kind of RFID Reader Anti-collision Algorithm", IEEE Circuits and Systems for Communications, pp. 559 - 563, 2008.

[3] La Porta, T.F.; Maselli, G.; Petrioli, C., "Anticollision Protocols for Single-Reader RFID Systems:Temporal Analysis and Optimization", IEEE Mobile Computing , Vol. 10, pp. 267 - 279, 2011.

[4] Miodrag Bolic, David Simplot-Ryl, Ivan Stojmenovic: *RFID SYSTEMS RESEARCH TRENDS AND CHALLENGES*, John Wiley & Sons 2010.

[5] Leian Liu; Zhenhua Xie; Jingtian Xi; Shengli Lai, "An Improved Anti-collision Algorithm in RFID System", IEEE Mobile Technology, Applications and Systems, pp.-5, 2005.

[6] Klair, D.K.; Kwan-Wu Chin; Raad, R., "A Survey and Tutorial of RFID Anti-Collision Protocols", IEEE Communications Surveys & Tutorials, Vol. 11, PP. 400 - 421, 2010.

[7] J. I. Capetanakis, "Tree algorithms for packet broadcast channels," *IEEE Trans. Information Theory*, vol. 25, pp. 505-515. 1979.

[8] F. Zhou, D. Jin, C. Huang, and M. Hao, "Optimize the Power Consumption of Passive Electronic Tags for Anti-collision Schemes," *Proc. The 5th Inter. conf. on ASIC*, Vol. 2, pp.1213-1217, Oct. 21-24, 2003.

[9] Ji Hwan Choi; Dongwook Lee; Hyoungsuk Jeon; Jongsub Cha; Hyuckjae Lee, "Enhanced Binary Search with Time-Divided Responses for Efficient RFID Tag Anti-Collision", IEEE Communications, PP. 3853–3858, 2007.

[10] M. Jacomet, A. Ehrsam, and U. Gehrig, "Contactless Identification Device With Anticollision Algorithm," IEEE Computer Society CSCC99, Conference on Circuits, Systems, Computers and Communications, 1999.

[11] S. Yu, Y. Zhan, Z. Wang, and Z. Tang, "Anti-collision algorithm based on jumping and dynamic searching and its analysis," *Computer Engineering*, vol. 31, pp. 19–20, 2005.

[12] Leian Liu; Zhenhua Xie; Jingtian Xi; Shengli Lai, "An Improved Anti-collision Algorithm in RFID System", IEEE Mobile Technology, Applications and Systems, PP. – 5, 2005.

[13] SungSoo Kim; YongHwan Kim; SeongJoon Lee; KwangSeon Ahn, "An Improved Anti Collision Algorithm using Parity Bit in RFID System", IEEE Network Computing and Applications, PP. 224 – 227, 2008.

[14] Majun Zheng; Jing Xie; Zhigang Mao; Yongxin Zhu, "A Hybrid Anti-Collision Algorithm for RFID with Enhanced Throughput and Reduced Memory Consumption", IEEE Embedded and Ubiquitous Computing, Vol. 1, PP. 259 – 265, 2008.

[15] *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification; Second Edition*; Klaus Finkenzeller.

[16] Leian Liu; Shengli Lai, "ALOHA-based Anti-collision Algorithms Used in RFID System",IEEE Wireless Communications, Networking and Mobile Computing, PP. 1 - 4, 2006.

[17] Su-Ryun Lee; Sung-Don Joo; Chae-Woo Lee, "An Enhanced Dynamic Framed Slotted ALOHA Algorithm for RFID Tag Identification", IEEE Mobile and Ubiquitous Systems , PP. 166 - 172, 2005.

[18] H. Vogt. "Multiple Object Identification with Passive RFID Tags", *IEEE International Conference on Systems*, 2002.

[19] H. Vogt. "Efficient Object Identification with Passive RFID Tags", *Proc. Pervasive 2002*. PP. 98–113, 2002.

[20] Tao Cheng; Li Jin, "Analysis and Simulation of RFID Anti-collision Algorithms", IEEE Advanced Communication Technology, Vol.1, PP. 697 – 701, 2007.

[21] Tae Hee Kim; Seong Joon Lee, "A Hybrid Hyper Tag Anti-Collision Algorithm in RFID System", IEEE Advanced Communication Technology, Vol. 2, PP. 1276 - 1281, 2009.

[22] Su, Weilian; Alchazidis, Nikolaos; Ha, Tri T., "DATA INTEGRITY IN RFID SYSTEMS", IEEE Military Communications Conference, PP. 1 – 7, 2007.

[23] Jiexiao Yu; Kai Hua Liu; Xiangdong Huang; Ge Yan, "An Anti-collision Algorithm Based on Smart Antenna in RFID System", IEEE Microwave and Millimeter Wave Technology, Vol. 3, PP. 1149 - 1152, 2008.

[24] E. Egea-Lopez, J. Vales-Alonso, A. S. Martinez-Sala, M. V. Bueno-Delgado, and J. Garcia-Haro, Performance evaluation of non-persistent CSMA as anti-collision protocol for active RFID tags, Department of Information Technologies and Communications, Polytechnic University of Cartagena, Spain.

[25] Nan Li; Xiaohui Duan; Yakun Wu; Shi Hua; Bingli Jiao, " An Anti-Collision Algorithm for Active RFID" IEEE Wireless Communications, Networking and Mobile Computing, PP.1-4, 2006.

[26] Huansheng Ning; Yu Cong; Xu, Z.-Q.; Hong, T.; Zhao, J.-C.; Yan Zhang, "Performance Evaluation of RFID Anti-Collision Algorithm with FPGA Implementation", IEEE Advanced Information Networking and Applications Workshops, Vol. 2, PP. 153 – 158, 2007.

[27] Chen Ying, "Design and Implementation of Communication Algorithm in NIOSII-Based RFID Reader", IEEE Communications and Mobile Computing, Vol. 2, PP. 286 - 289, 2009.

[28] Won-Ju Yoon; Sang-Hwa Chung; Seong-Joon Lee; Young-Sik Moon, "Design and Implementation of an Active RFID System for Fast Tag Collection", IEEE Computer and Information Technology, PP. 961 - 966, 2007.

[29] S. M. A. Motakabber; Mohd Alauddin Mohd Ali; Nowshad Amin, " VLSI Design of an Anti-Collision Protocol for RFID Tags", European Journal of Scientific Research, Vol.28 No.4, PP.559-565, 2009.

[30] The Propagation Group, "RFID Anti-Collision System Using the Spread Spectrum Technique", 2005.

[31] Lifen Yuan; Yigang He; ZhouGuo Hou; Bing Li, "A New Method of Signal Processing in RFID system" IEEE Computer Science and Software Engineering, Vol. 5, PP. 1040 – 1043, 2008.

[32] Yahui Yang ; Yujie Wu; Min Xia; Zhijing Qin, "A RFID Network Planning Method Based on Genetic Algorithm", IEEE Networks Security, Wireless Communications and Trusted Computing, PP.534-537, 2009.

[33] Hanning Chen; Yunlong Zhu, "RFID Networks Planning Using Evolutionary Algorithms and Swarm Intelligence", IEEE, PP.1-4,2008.

[34] Hu Shengbo, "An algorithm based on querry learning for solving Frequency assignment in RFID system", IEEE Control and Decision Conference, PP. 225 – 229, 2008.

[35] Waldrop, J.; Engels, D.W.; Sarma, S.E., " Colorwave: An Anticollision Algorithm for the Reader Collision Problem", IEEE Communications, Vol.2, PP. 1206 – 1210, 2003.

[36] Jun-Bong Eom; Tae-Jin Lee, "RFID Reader Anti-collision Algorithm Using a Server and Mobile Readers Based on Conflict-Free Multiple Access", IEEE Performance, Computing and Communications Conference, PP. 395 – 399, 2008.

[37] Leian Liu; Dashun Yan; Xiaozheng Lai; Shengli Lai, "A New Kind of RFID Reader Anti-collision Algorithm", IEEE Circuits and Systems for Communications, PP. 559 – 563, 2008.

[38] Kasahara, H, S. Hara, and N. Morinaga, Modeling and Simulation Analysis of Indoor Packet Radio Communication Systems, *IEICE Trans. Fundamentals,* Vol. 78-A, No. 8, August 1995.

[39] Klair, Dheeraj K.; Chin, Kwan-Wu; Raad, Raad, "An investigation into the energy efficiency of pure and slotted Aloha based RFID anti-collision protocols," IEEE World of Wireless, Mobile and Multimedia Networks, PP. 1 - 4, 2007.

[40] *Himanshu Bhatt, Bill Glover, RFID Essentials, O'Reilly, January 2006.*

[41] Adlen ksentini, Qualité de service (QoS) dans les réseaux locaux sans fil basés sur la technologie IEEE802.11, école doctorale science et ingénierie de l'université cergy-Pontoise, 2005.

[42] Zhanyou Ma; Wuyi Yue; Naishuo Tian, "Performance Analysis of Multi-Channel and Multi Traffic on Wireless Communication Networks", IEEE Wireless Communications and Networking Conference, PP. 1-9, 2009.

Annexes

ANNEXE I

- **Méthode d'estimation d'étiquette I :** Schoute a présenté une méthode pour estimer le nombre de tag qui n'a pas encore été identifié dans les systèmes de communication à plusieurs accès. On suppose que le nombre de transpondeurs qui choisissent le slot i d'une trame pour transmettre leurs paquets de données a la distribution de Poisson avec un moyen de 1. L'estimation du nombre d'étiquettes (n ') qui n'ont pas été identifiées par le lecteur après la trame courante est donnée par [17]:

$$n' = 2.39 * C \quad (1.1)$$

- **Méthode d'estimation d'étiquette II:** On note par : (N) le nombre de slots dans une trame et (n) le nombre d'étiquettes, (t) le nombre d'étiquettes dans un des slots est une distribution binomiale s'écrit comme suit [14]:

$$P(X = t) = \binom{n}{t} (\frac{1}{N})^t (1 - \frac{1}{N})^{n-t} \quad (1.2)$$

Le nombre prévu de slots (m) dont chacun va avoir t étiquette arrangée à chacun d'eux peut être reçu par:

$$m = N \binom{n}{t} (\frac{1}{N})^t (1 - \frac{1}{N})^{n-t} \quad (1.3)$$

À partir de l'équation (1.3), nous pouvons obtenir:

$$E = N(1 - \frac{1}{N})^n \quad (1.4)$$

$$S = n(1 - \frac{1}{N})^{n-1} \quad (1.5)$$

$$C = N - E - S \quad (1.6)$$

Nous définissons la sortie T comme suit:

$$T = \frac{S}{N} = \frac{n}{N}(1 - \frac{1}{N})^{n-1} \quad (1.7)$$

La dérivée de l'équation (1.7) donne le résultat suivant:

$$\frac{dT}{dN} = -\frac{n}{N^2}(1 - \frac{1}{N})^{n-1} + \frac{n}{N}(n-1)(1 - \frac{1}{N})^{n-2} \frac{n}{N^2} = 0 \quad (1.8)$$

À partir de l'équation (1.8), on obtient le résultat suivant:

$$N = n \quad (1.9)$$

Ceci signifie lorsque le nombre de slots dans une trame est identique au nombre d'étiquettes dans la zone d'interrogation du lecteur, la sortie maximale sera obtenu et ainsi le temps total pour identifier toutes les étiquettes sera minimale. Selon l'équation (1.7), nous pouvons

obtenir la sortie de système quand la taille de la trame est constante comme le montre la Fig 1 **[**16**]**.

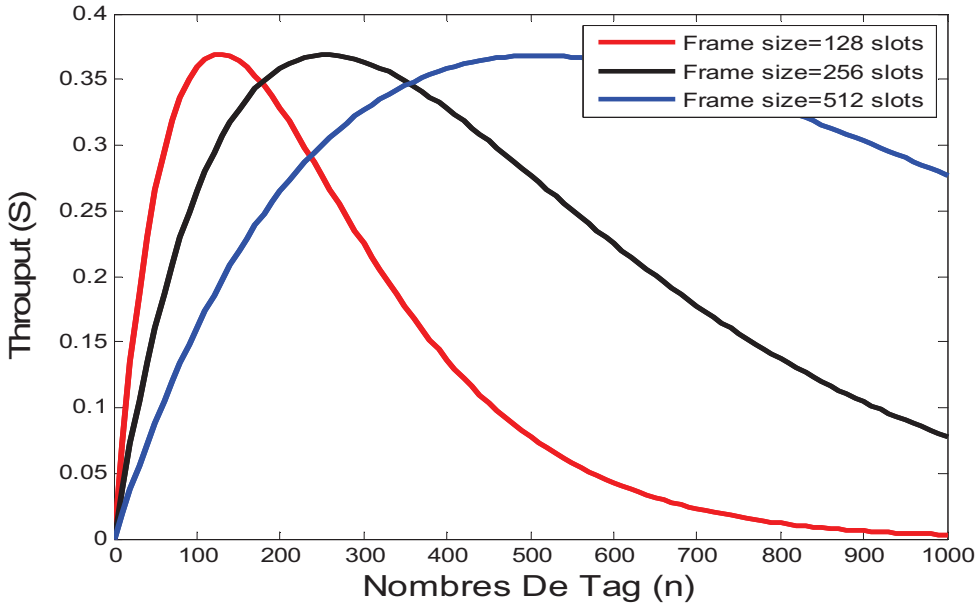


Fig 1. Sortie S en fonction de nombre de tag(n) à plusieurs tailles de la trame (N)

A chaque demande d'interrogation d'étiquettes, les nombres E, S et C sont reçus et N dans la trame courante est déjà connu, ainsi selon l'équation (1.4), (1.5) et (1.6), nous pouvons obtenir le nombre d'étiquettes (n ") qui n'ont pas été identifiées par le lecteur après la trame courante comme suit:

$$n" = n - S \qquad (1.10)$$

ANNEXE II

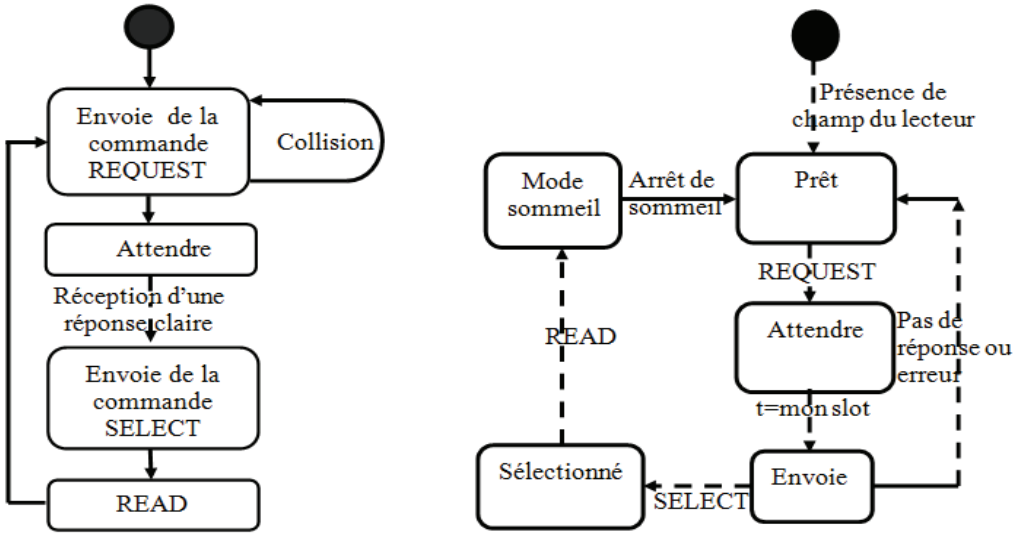


Fig. 2. (A) les transitions d'état du lecteur. (B) transitions d'état de l'étiquette. [40]

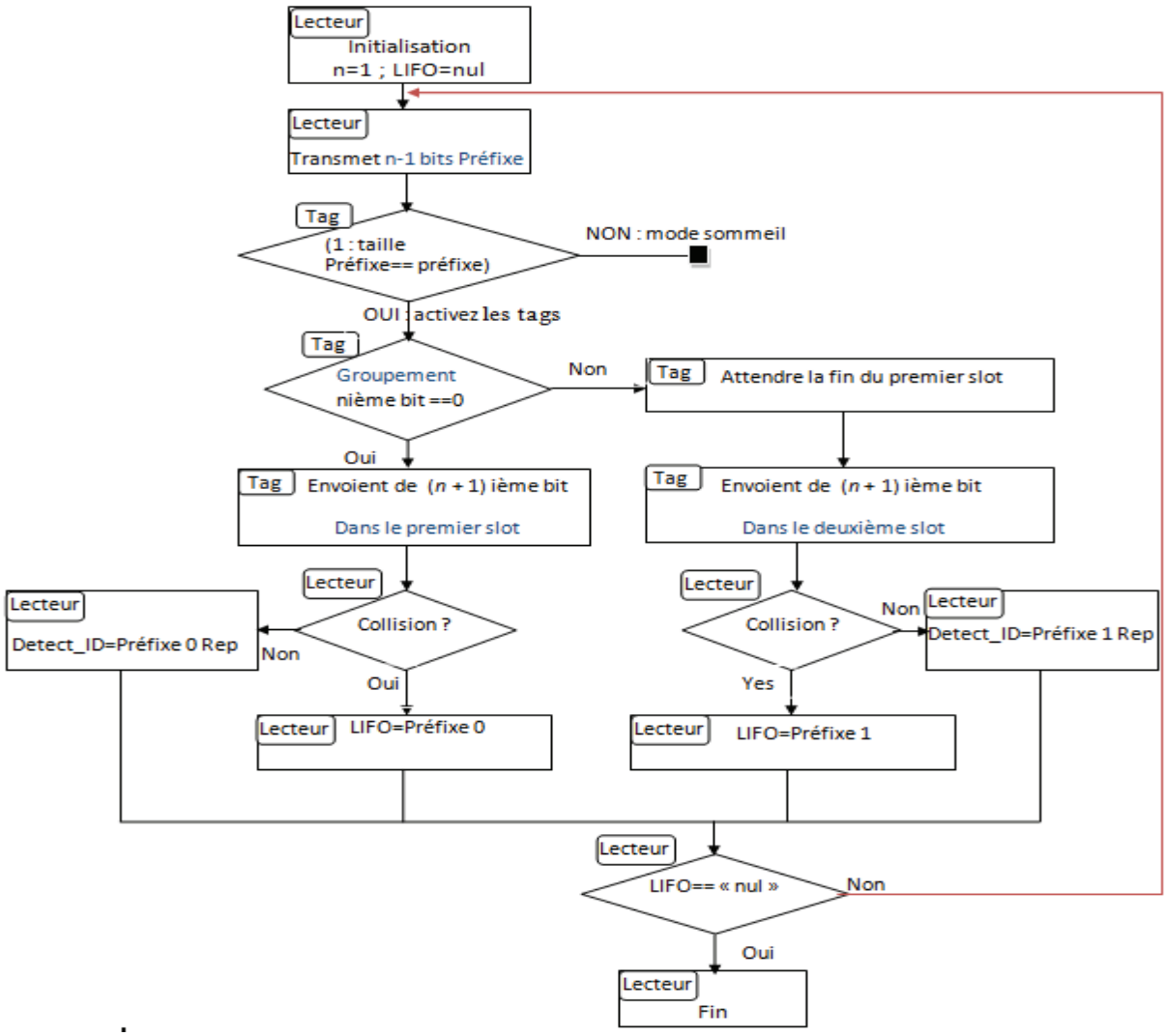


Fig. 3. Organigramme de l'algorithme BSQTA [9]

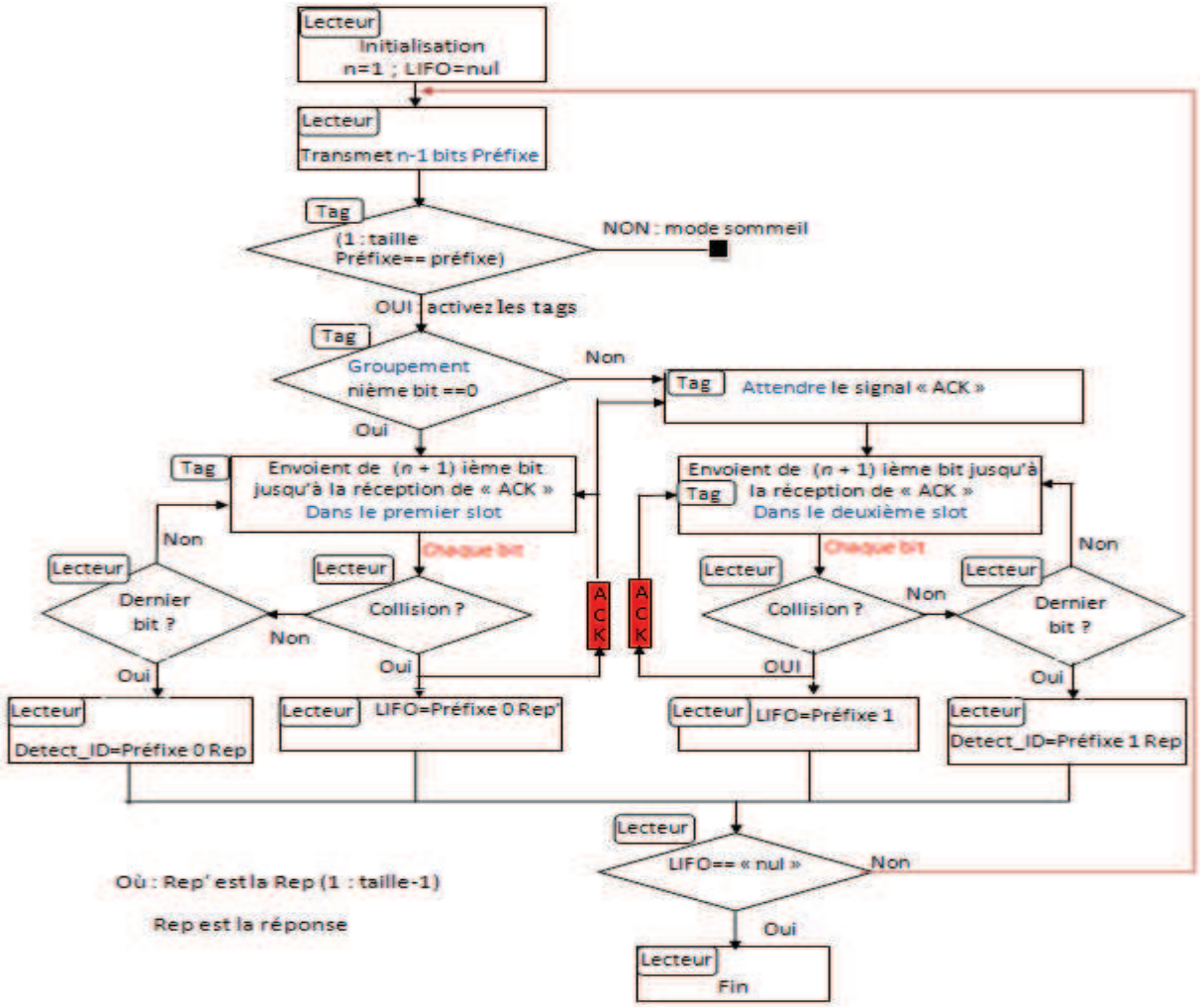


Fig. 4. Organigramme de l'algorithme BSCTTA [9]

Résumé :

La RFID est une technologie de transmission sans fil qui permet l'indentification automatiques des objets à distance. Les principaux composants d'un système RFID sont les tags qui sont déposés sur les cibles et les lecteurs qui communiquent à distance avec ces tags. La présence de plusieurs tags dans le champ d'un seul lecteur, ou d'un seul tag dans le champ croisé de plusieurs lecteurs produit des signaux tag-tags ou lecteurs-lecteurs qui s'interfèrent pour produire des mélanges au niveau des lecteurs. Ce travail s'inscrit dans le cadre du développement d'algorithmes d'anticollisions pour les systèmes RFID passifs. Le principal objectif était l'étude et la simulation d'un algorithme anticollision basé sur le protocole d'ALOHA, ainsi de proposer une étude comparative d'algorithmes d'anticollisions de signaux RFID passifs.

Mots clés : RFID, Tag, Lecteur, Algorithme Anticollision, ALOHA.

Abstract :

The RFID is a wireless transmission technology which allows the automatic identification of objects at a distance. The main components of an RFID system are tags that are deposited on the targets and readers that communicate remotely with these tags. The presence of multiple tags in the field of a single drive, or a single tag in a crossed field of multiple drives produces tags-tag or readers-readers signals which interfere. This work is in development of anti collision algorithm for passive RFID systems. The main objective was to study and simulation of an anti collision algorithm based on ALOHA protocol, and to propose a comparative study for these algorithms.

Keywords : RFID, Tag, Reader, Anti-collision Algorithm, ALOHA.

ملخص:

أنظمة التعرف اللاسلكي هي طريقة تبادل المعلومات لاسلكية مستعملة في التعرف الأتوماتيكي للأجسام البعيدة. المكونات الرئيسية لنظام التعرف اللاسلكي هما الشريحة المثبتة على الأجسام و القارئ الذي يتصل بهذه الأخيرة. المشكل المصادف في مثل هذه الأنظمة هو لما يكون عدة شرائح موزعة في مجال قارئ واحد أو لما يتداخل أكثر من قارئ من أجل التعرف على شريحة واحدة في مثل هذه الحالتان ينتج لدينا تداخل الإشارات. في هذا الصدد قدمنا هذا العمل المتضمن مبادرة تطوير أنظمة التعرف اللاسلكي و المتركزة على خوارزميات منع التداخل و بالأخص تلك المعتمدة على خوارزمية آلوها. كذلك قدمت دراسة مقارنة لخوارزميات أخرى مستعملة في مثل هذه الأنظمة.

كلمات البحث :شريحة, قارئ, خوارزميات منع التداخل, آلوها.