REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MENTOURI- CONSTANTINE

FACULTE DES SCIENCES DE L'INGENIEUR

DEPARTEMENT D'ELECTRONIQUE

THESE

Présentée pour l'obtention du diplôme de

Doctorat en sciences en Electronique

Option communication

Par

MOUNIR BELATTAR

THEME

PROTOCOLES STANDARDS DE COMMUNICATION DES DONNEES SPATIALES DES SYSTEMES

Soutenu le : 20 Juin 2012

Devant le Jury:

Dr BENNIA Abdelhak	**Professeur**	**Univ.Constantine**	**Président**
Dr BENATIA Djamel	**Professeur**	**Univ.Batna**	**Rapporteur**
Dr SOLTANI Fouzi	**Professeur**	**Univ.Constantine**	**Examinateur**
Dr SAIDI Lamir	**MCA**	**Univ.Batna**	**Examinateur**
Dr KEMIH Karim	**MCA**	**Univ.Jijel**	**Examinateur**

Dédicaces

A Mes chers parents, trésor de générosité et d'amour en témoignage de ma profonde affection, mon admiration, et à qui je dois beaucoup.

A ma chère femme qui m'a toujours soutenu aux moments difficiles.

A mes frères et sœurs.

A mes nièces et neveux

A mes enfants Zinou, Amine et Syrine.

A toute ma famille, avec toute ma sympathie et mon amour.

A tous mes amis .

Remmerciements

Messieurs, les professeurs D.Benatia et M.Benslama,

Vous avez été l'inspirateurs de cette thèse, cela a été pour moi un bon choix. Vous avez spontanément accepté de diriger ce travail, j'ai beaucoup appris à vos côtés. Cette thèse est pour vous, un témoignage de ma reconnaissance et mon rofond respect.

Je voudrais également remercier le professeur A.Benia d'avoir accepté la présidence du jury de soutenance, je lui suis sincèrement reconnaissant pour sa bienveillance et je le prie d'accepter l'expression de ma sincère gratitude.

J'adresse ma profonde reconnaissance aux membres du jyry:Monsieur F.Soltani professeur à l'université de Constantine, Monsieur L.Saidi Maître de conférence à l'université de Batna et Monsieur K.Kemih Maître de conference à l'université de Jijel, d'avoir accepré de juger cette thèse, l'interêt qu'ils ont porté à mon travail et la rapidité avec laquelle ils ont lu mon manuscrit.

Je ne saurais oublier le Dr M.Ras Lain pour son assistance et ses encouragements.

Sans oublier les amis B.Bourbia, K.Hireche, A.Keghida et R.Alouatni qui m'ont apporté une assistance remarquable pour mener à bien ce travail.

Résumé

Notre étude accentuée sur les protocoles standards de communication par satellite, décrit le modèle ISO et ses protocoes multicouche dérivées, puis présente les performances du protocole Aloha discrétisé à accès aléatore bsé sur le modèle de Markov pour définir les distributions binomiales de probabilités optimles de retransmssion et celles des nouveaux paquets arrivés, en observant leurs évolutions en fonction du nombre de retransmissions en une unité de temps (slot), avec les probabilités de refoulement et de génération de paquets prises comme paramètres.

Cela nous a permis de définir le délai moyen de transmission d'un paquet, et voir l'effet de l'augmentation du nombre de sources. L'information transmise de la station terrestre vers le satellite est sujet de perte suite à l'effet des bruits ou des collisions, A cet effet un standard dit codage Reed-Solomon est traité pour reconstruire les paquets perdus et améliorer la performance du débit.

Abstract

Our study focused on the standard protocols of satelite comunication, describes the ISO Model and its derived protocols with several layers, then the performances of random multiple access protocol Aloha are analyzed, after the model of Markov chain is presented , to define optim al binomial distribution probabilities of retransm ission and arrival packets , in the case of Slotted Aloha protocol, to describe their evolution in function of the number of transmissions in a slot time, with parameters, probabilities of backlogged an d generation packets, this has permitted us to define the average packet delay and the throughput which is the main parameter of random multiple a ccess model, an d see the effect of the increase of the number of sources. The in formation transmitted from earth station to the satellite is subject of loss by noise effects or collisions, so a standard named Reed-Solomon coding is considered to find out and recover lost or collided packets, and to improve the throughput performance.

ملخص

هده الدراسة عنيت بوصف النمط ايزو و مشتقاته المتعدد الطبقات و كذلك مميزات بروتوكول الوها ذات الطبيعة العشوائية و المتعددة المستعملين. كما أن نمط سلسلة ماركوف قد نوقش بغرض تعيين دوال توزيع احتمالات إعادة إرسال أو وصول حزم معلومات جديدة و كشف تغيراتها بدلالة عدد الحزم المرسلة خلال وحدة زمنية باستعمال احتمالات تأخير أو إنتاج حزم جديدة مما مكننا من تقدير مقدار الزمن المتوسط لإرسال حزم المعلومات و الكمية المرسلة خلال الزمن.

إن الاتصالات ما بين محطة أرضية و قمر اصطناعي معرضة للتلف إما بسبب تشويش المؤثرات الخارجية أو تلاطم حزم المعلومات الواردة من مصادر متعددة. و لهدا فان تشفير ريد- سولومون قد تم عرضه بغرض استرجاع المعلومات الضائعة.

Symboles et abréviations

BCG : Bose et Ray-Chaudhuri

CRC : Cyclic Redondancy Code

FIFO : First In Fisrt Out

FDM : Frequency Division Multiplexing

GF : Galois Field

IP : Internet Protocol

ISO : International Stadards Organization

ITDM: Intelligent Time Division Multiplexing

MAC : Medium Access Control

OSI : Open Systyem Interconnexion

PPP : Point to Point Protocol

STDM : Sunchronous Time Division Multiplexing

TCP: Transmission Control Protocol

WAN : Wide Area Network

Liste des figures

TABLE DES MATIERES

II Techniques d'accès Multiples aux canaux satellitaires. Analyse du protocole Aloha

III Analyse du modele de Markov du Protocole Aloha discrétisé en communication par satellite

IV Correction d'erreurs et d'effacements

INTRODUCTION GENERALE

En 1978, l'ISO *(International Standards Organisation)* publia un standard universel pour l'échange d'informations entre réseaux au-delà des frontières géographiques.
Ce standard d' architecture de réseau était le modèle à sept couches pour l'interconnexion de systèmes ouverts (OSI) [1].

Le m odèle OSI a contribué à plus de conformité dans la conception des réseaux de communications et dans le contrôle des traitements répartis.

Les constructeurs réalisent des ordinateurs et des équipements dotés de plus d'intelligence. De nombreux réseaux de données pr ivés ou publics ont été créés pour connecter ces matériels. Mais les communications entre ces systèmes et ces réseaux répartis requièrent une approche standard dans la conception de réseau, qui définit les relations et les intersections entre les services et les fonctions de réseau, grâce à des interfaces et des protocoles communs [2]-[4].

Cette approche stratifiée de l'architecture de réseau ramène à la conception du système d'exploitation. Du fait de leur com plexité, la pl upart des systèmes d'exploitation des ordinateurs sont développés en sections dont chacune assure une fonction particulière. Cela permet de peaufiner chaque fonction pour qu' elle atteigne son but fonctionnel. Enfin, toutes les sections sont intégrées pour assurer les fonctionnalités et les services d' un système d' exploitation qui tourne au rond.

Il en est de même pour la conception de systèmes de réseau. Une archit ecture de réseau spécifie une hiérarchie entre le s couches indépendantes dont les modules réalisent des fonctions déterminées. Cela se tra duit en un jeu de règles qui définit la manière des nœuds de réseau participants à inte ragir pour communiquer et échanger des infor mations. Le m odèle OSI définit les relations standards entre le logiciel e t le matériel des systèmes informatiques complexes d'aujourd'hui [1]-[4].

Dans les réseaux de télécommunications, un protocole de communication est une spécification de plusieurs règles pour un type de commmnication particulier. Initialement, on nommait protocole ce qui est utilisé pour communi quer sur une même couche d'abstraction entre

deux machines différentes. Par extension de langage, on utilise parfois ce m ot aussi aujourd'hui pour désigner les règles de communication [5], [6].

En effet, le modèle OSI est constitué de sept couches dont ch acune correspond à une onctionnalité particulière d'un réseau. Les quatre prem ières couches dites basses, assurent l'acheminement des informations entre les extrémités concernées et dépendent du support physique. Les trois autres couches, dites hautes, sont responsables du traitem ent de l'information relative à la gestion des échanges entre systèmes informatiques.

Le m odèle TCP/IP, inspiré du m odèle OSI, reprend le systèm e de couches m ais n'en contient que cinq. Le m odèle TCP/IP l'emporte sur le m odèle OSI du fait que ce dernier est trop complexe pour être efficacem ent implém enté à l'inverse du TCP/IP qui est beaucoup plus optimisé et efficace [3].

.

Lors d'une transm ission, les données traversent chacune des couches au niveau de la machine émettrice. A chaque couche, une information est ajoutée au paquet de données, il s'agit d'un en-tête, ensemble d'informations qui garantit la transmission. Au niveau de la machine réceptrice, lors du passage dans chaque couche , l'en-tête est lu, puis supprimé. Ainsi, à la réception, le message est dans son état originel.

Dans le cas d' un réseau avec plusieurs utilisateurs partageant une ressource c'est une partie spécif ique de la couche lia ison f aisant p artie de s couches basses, est appelée Medium Access Con trol (MAC) qui org anise l'accès à la ressource. Cette sous couche MAC est d' une importance capitale dans les systèmes de communication [7].

Les performances des systèm es de communications son t fortem ent liées aux choix techniques qui pe rmettent à des u tilisateurs multip les d'accéder à un canal de transmission [7], [8].

Le protocole ALOHA, inventé à l' Université d'Hawaï en 1970 par Norm an Abra mson. est un protocole de communication d'accès multiple toutes les stations émettent et reçoivent sur la même bande de fréquence [1], [9], [10].

Les messages sont découpés en trames, identifiées par un numéro d'ordre et l'adresse de la station destinataire et les stations reçoivent toutes les trames, identifient celles qui leur sont destinées et jettent les autres.

ALOHA doit prévoir le cas où deux stations (ou plus) voudraient commencer à émettre en même temps. Cette circonstance est nommée *collision*, et résulte en trames brouillées, incompréhensibles, en un mot perdues. Il va falloir réémettre ces trames perdues [11]-[16].

S.Ghez et S.Schartz ont abordé les propriétés de stabilité du protocole Aloha discrétisé à accès multiple [17], et ont fait l'objet de notre analyse dans le cas de deux utilisateurs en définissant les régions de stabilité, et ce en utilisant la théorie des files d'attente [18] [19], En s'inspirant des travaux antérieurs dans la littérature [20]-[27].

Ainsi, en se basant sur la théorie des chaînes de Markov employée pour l'analyse du protocole Aloha discrétisé à accès multiple [29] -[32], nous avons procédé à la définition des distributions optimales des probabilités de retransmission et d'arrivée de paquets d'informations et décrire leurs évolutions en fonction des probabilités de refoulement et de génération de paquets dans un intervalle de temps dit « slot » et d'arriver enfin à définir les deux paramètres caractérisant une transmission à accès multiple qui sont le débit et le délai moyen d'un paquet [33]. L'optimisation de ses deux grandeurs a constitué déjà un souci et un objet d'étude de nombreuses recherches [34]-[36].

En effet, en communication spatiale, les données transmises par satellite sont sujet d'erreurs et des effacements, ce qui mène employer le code de Reed-Solomon [37]-[55].

Dans notre approche, nous appliquons ce code pour améliorer les performances du protocole Aloha discrétisé afin de retrouver les paquets perdus décrits par les travaux de K.S.Chan , L.K.Yeung et W.Shao [55], .et nous avons défini ainsi le comportement du débit avec correction d'effacement en utilisant une interpolation polynomiale du débit en fonction du trafic offert.

Pour cela, nous avons structuré notre thèse en quatre chapitres dont le premier est consacré à la description du modèle ISO et ses dérivés ainsi qu'aux structurex multi-couches des protocoles standards de communication, les protocoles à accès multiple tel que le protocole Aloha.

Le deuxième chapitre présente le protocole Aloha discrétisé comme un protocole d'accès aléatoire, il traite le débit d'une communication satellitaire, son amélioration, le problème de perte des information par collisions des trames, ainsi que la stabilité du protocloe dans le cas de deux utilisateurs.

Le troisième chapitre offre une analyse du modèle de Markov appliqué au protocole Aloha discrétisé et les performence de ce dernier comme débit et délais de transmissions des paquets d'information.

Le quatrième chapitre présente les techniques de corrections des erreurs et des effacements des informations lors d'une communication par satellite en utilisant le codage de Reed-Solomon et les algorithme de décodage de Forney, Berlekamp Massey et Forney.

Ensuite, une conclusion générale donnera une synthèse du travail présenté avec un apperçu sur les travaux futurs et perspectives.

CHAPITRE I

Modélisation des réseaux- le Modèle ISO et ses dérivées

I.1 Introduction

Le problème de base à résoudre pour concevoir et réaliser un réseau d'ordinateurs consiste à établir un échange de données entre deux ordinateurs distants. Ce problème se divise en deux parties : pour que les données circulent correctement elles doivent être représentées selon un codage approprié commun aux deux extrémités, et il y faut un support physique également approprié.

La position de ce problème remonte au moins à Aristote, qui a envisagé la communication d'information entre deux personnes en termes de message et de code. Incidemment, ce modèle est beaucoup mieux adapté à la communication entre ordinateurs qu'à la communication entre êtres humains.

L'invention du téléphone a conduit à le formaliser sous le nom de « communication sur un canal bruité ». En effet il y a du bruit, c'est-à-dire qu'aucun canal de communication n'est parfait, certains éléments du message sont altérés ou perdus. Henrik Nyquist, dès les années 1920, et Claude Shannon en 1948 ont posé les bases théoriques précises constituant la théorie dite de l'information (Annexe A).

Dans ce cas précis le support matériel de la communication était une ligne de téléphone en fils de cuivre, mais les réseaux informatiques peuvent utiliser toutes sortes de supports matériel sans que cela modifie la nature du problème à résoudre : fibre optique, faisceau hertzien, canal satellite, câble coaxial, rayons infrarouge, signal laser etc. Il suffit que les équipements à chaque extrémité soient configurés correctement [1].

Les constructeurs informatiques ont proposé des architectures réseaux propres à leurs équipements. Par exemple, IBM a proposé SNA, DEC a proposé DNA... Ces architectures ont toutes le même défaut : du fait de leur caractère propriétaire, il n'est pas facile de les interconnecter, à moins d'un accord entre constructeurs. Aussi, pour éviter la multiplication des solutions d'interconnexion d'architectures hétérogènes, l'ISO (International Standards Organization), organisme dépendant de l'ONU et composé de 140 organismes nationaux de normalisation, a développé un modèle de référence appelé modèle OSI (Open Systems Interconnection). Ce modèle décrit les concepts utilisés et la démarche suivie pour normaliser l'interconnexion des systèmes ouverts (un réseau est composé de systèmes ouverts lorsque la modification, l'adjonction ou la suppression d'un de ces systèmes ne modifie pas le comportement global du réseau) [1], [2].

Au moment de la conception de ce modèle, la prise en compte de l'hétérogénéité des équipements était fondamentale. En effet, ce modèle devait permettre l'interconnexion avec des systèmes hétérogènes pour des raisons historiques et économiques. Il ne devait en outre pas favoriser un fournisseur particulier. Enfin, il devait permettre de s'adapter à l'évolution des flux d'informations à traiter sans remettre en cause les investissements antérieurs. Cette prise en compte de l'hétérogénéité nécessite donc l'adoption de règles communes de communication et de coopération entre les équipements, c'est à dire que ce modèle devait logiquement mener à une normalisation internationale des protocoles.

Le modèle OSI n'est pas une véritable architecture de réseau, car il ne précise pas réellement les services et les protocoles à utiliser pour chaque couche. Il décrit plutôt ce que doivent faire les couches. Néanmoins, l'ISO a écrit ses propres normes pour chaque couche, et ceci de manière indépendante au modèle, i.e. comme le fait tout constructeur.

Les premiers travaux portant sur le modèle OSI datent de 1977. Ils ont été basés sur l'expérience acquise en matière de grands réseaux et de réseaux privés plus petits ; le modèle devait en effet être valable pour tous types de réseaux. En 1978, l'ISO propose ce modèle sous la norme ISO IS7498. En 1984, Douze constructeurs européens, rejoints en 1985 par les grands constructeurs américains, adoptent le standard [1]- [4].

I.2 Définition du modèle standard OSI de l'ISO

Le modèle d'Interconnexion des Systèmes Ouverts (*Open Systems Interconnection*) a été proposé par l'ISO (*International Standards Organization*) en 1977. Il s'agit d'une norme internationale pour une architecture multicouche qui permet l'interconnexion de matériels hétérogènes, ceci pour avoir :

1° Facilité le développement et la modification : une couche (un protocole) peut être modifiée de façon indépendante tant que l'interface avec les deux couches adjacentes reste inchangées.

2° Inter opérabilité : une même couche de niveau *n+1* peut utiliser les services de couches de niveau *n* très différentes à condition que l'interface *n/n+1* soit la même à un modèle commun pour pouvoir communiquer c'est-à-dire :

- une couche doit être créée lorsqu'un nouveau niveau d'abstraction est nécessaire.

- chaque couche a des fonctions bien définies,

- les fonctions de chaque couche doivent être choisies dans l'objectif de la normalisation internationale des protocoles,

- les frontières entre couches doivent être choisies de manière à minimiser le flux d'information aux interfaces,

- le nombre de couches doit être tel qu'il n'y ait pas de cohabitation de fonctions très différentes au sein d'une même couche et que l'architecture ne soit pas trop difficile à maîtriser.

I.3 Notion de protocole

Pour acheminer un flux de données sur un canal de transmission, les systèmes matériels et logiciels qui composent le réseau utilisent un ensemble de règles, de conventions et de mises en forme des données qui constituent un protocole de communication. Une des fonctions des protocoles de communication est la détection et la correction des erreurs de transmission, qui constituent un des problèmes classiques des réseaux informatiques.

Les canaux de communication sont soumis à des erreurs aléatoires, qui constituent le bruit. Il faut donc instaurer des moyens de vérifier l'intégrité des données et, quand c'est possible, de la restaurer quand elle a été altérée. Ceci devra être fait pour chaque couche du protocole de transmission, mais c'est spécialement important pour la couche 2, dont le rôle est de garantir aux couches supérieures un canal de transmission sans erreur d'un flux de bits. Le traitement de ce problème peut différer selon qu'il s'agit de liaisons point à point ou de diffusion [1].

Le protocole définit également, pour chaque couche, d'autres caractéristiques de la transmission : les messages sont généralement découpés en unités de taille homogène (appelés trames pour la couche 2 et paquets pour la couche 3), les nœuds reçoivent lorsque c'est nécessaire des adresses qui permettent, comme celles des personnes, de les trouver et de les identifier dans le réseau.

Dans un modèle en couches les protocoles définissent les règles du dialogue entre couches de même niveau sur des systèmes différents, cependant que les interfaces spécifient les services qu'une couche inférieure fournit à la couche qui lui est immédiatement supérieure au sein du même système.

I.4 Architecture Multicouches pour protocoles standards de communication

Le modèle OSI comporte 7 couches, et sont décrites comme suit :

Les couches basses (1, 2, 3 et 4) sont nécessaires à l'acheminement des informations entre les extrémités concernées et dépendent du support physique. Les couches hautes (5, 6 et 7) sont responsables du traitement de l'information relative à la gestion des échanges entre systèmes informatiques. Par ailleurs, les couches 1 à 3 interviennent entre machines voisines, et non entre les machines d'extrémité qui peuvent être séparées par plusieurs routeurs. Les couches 4 à 7 sont au contraire des couches qui n'interviennent qu'entre hôtes distants (Voir Fig.(I.1)) [1]-[4].

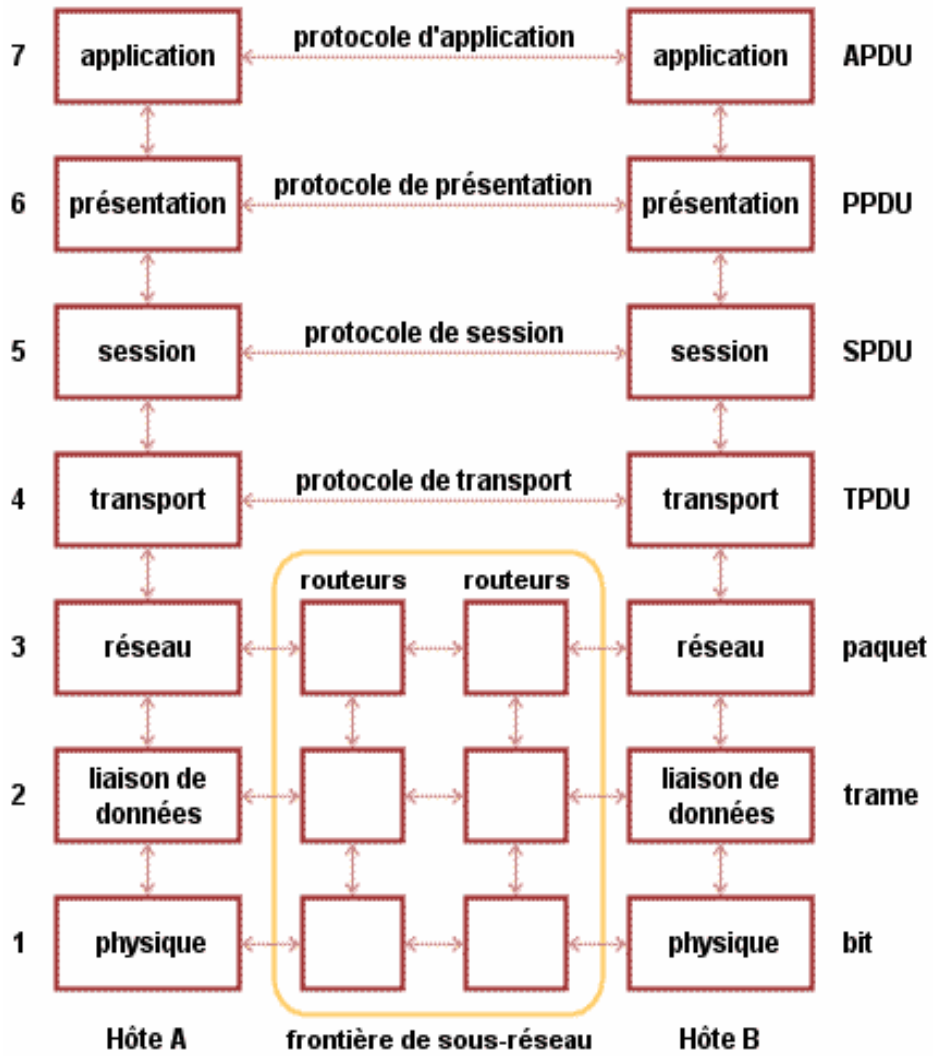

Figure (I.1) : Architecture du modèle standard ISO

I.4.1 Couche Physique ou niveau 1

I.4.1.1 Définition

Elle décrit les interfaces mécaniques, électriques, fonctionnels et procédurales nécessaires à l'activation, au maintien et à la désactivation des connexions physiques destinées à la transmission de bits entre deux entités de liaison de données. Ce niveau est chargé de piloter le matériel de transmission. C'est dans cette couche que sont définis le support physique ou médium, les signaux, les voies de transmission ou canaux, le raccordement des communicants. Les entités principales, à ce niveau, sont le signal analogique et le bit avec comme caractéristiques le débit binaire et le taux d'erreur bits.

I.4.1.2. Multiplexage

On appelle multiplexage, la capacité à transmettre sur un seul support physique des données provenant de plusieurs paires d'équipements (émetteurs et récepteurs).

a- Le multiplexage fréquentiel

Le multiplexage fréquentiel (FDM, Frequency Division Multiplexing) est utilisé lorsque la bande passante du support de transmission est supérieure à celle nécessaire au signal. Le principe de cette technique, consiste à partager la bande passante de la voie de transmission en plusieurs bandes de plus faible largeur, chacune de ces sous bandes passante est affectées à un émetteur qui devra donc émettre dans cette bande. Les signaux de chacune de ces bandes sont transmis par un canal unique, ce qui est réalisé par un multiplexeur, à la réception un démultiplexeur qui sépare les différents signaux par une série de filtre passe bande, les dirige vers leurs destinataires respectifs, aucun adressage explicite n'est nécessaire puisque chaque émetteur est identifié par la bande de fréquences utilisées.

Pour assurer une bonne transmission, on laisse une bande de fréquence inutilisée entre chaque sous bande (appelée bande de garde), il s'agit essentiellement d'empêcher des chevauchements entre signaux appartenant à des bandes voisines.

Les signaux transmis avec cette technique sont du type analogique, les signaux numériques doivent donc être codés à l'aide de MODEM pour être transmis, chaque MODEM doit assurer la translation du signal dans la gamme des fréquences du sous canal utilisé. La figure (I.2), illustre le principe du multiplexage fréquentiel avec une bande passante du canal 300-1400 Hz, divisée en trois sous bandes.

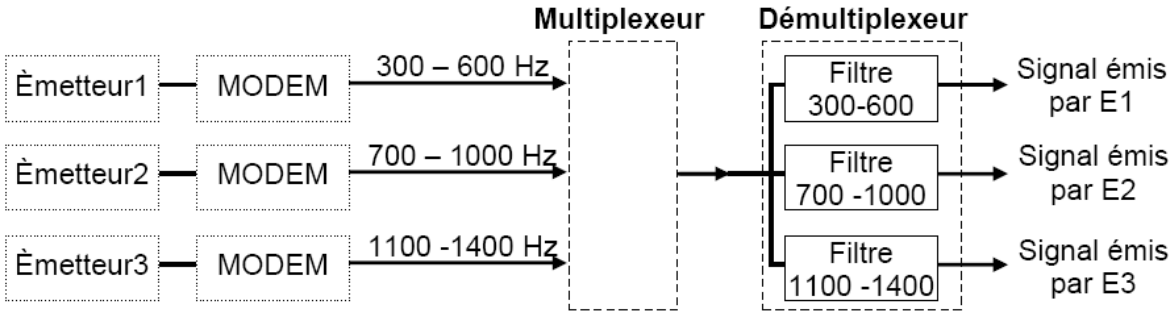

Figure (I.2): Exemple de multiplexage fréquentiel.

b- Le multiplexage temporel

Le multiplexage temporel (TDM, Time Division Multiplexing) suit le même mécanisme que la FDM, mais au lieu de découper la bande passante du support en plusieurs bandes, il découpe le temps en tranches qui sont affectées régulièrement à chaque émetteur.

Le multiplexage temporel est plus efficace que le précèdent, puisqu'il fait une meilleure utilisation de la bande passante. Avec cette technique, les différentes sources émettrices utilisent différents intervalles de temps, c'est à dire qu'elles utilisent le support de transmission à tour de rôle. Selon la méthode d'allocation du temps, on distingue deux types de multiplexage temporel : le multiplexage temporel synchrone (STDM, Synchronous Time Division Multiplexing) et le multiplexage statistique.

c- Le multiplexage temporel synchrone

Dans le multiplexage temporel synchrone, toutes les sources émettrices ont accès au support de transmission durant un intervalle de temps égal même si la source n'est autorisée à émettre. Imaginons par exemple qu'il y'ait trois appareils devant transmettre des données : l'ordinateur A, l'ordinateur B et l'ordinateur C. Le premier à avoir accès au support de transmission est l'ordinateur A, une fois l'intervalle de temps écoulé, il passe le relais à l'ordinateur B, qui peut transmettre ses données pendant une durée identique avant de laisser l'ordinateur C émettre à son tour. Les données émises durant un intervalle de temps peuvent être divisées en bits, en octet ou autre regroupement.

Le multiplexage temporel synchrone fonctionne comme un commutateur.

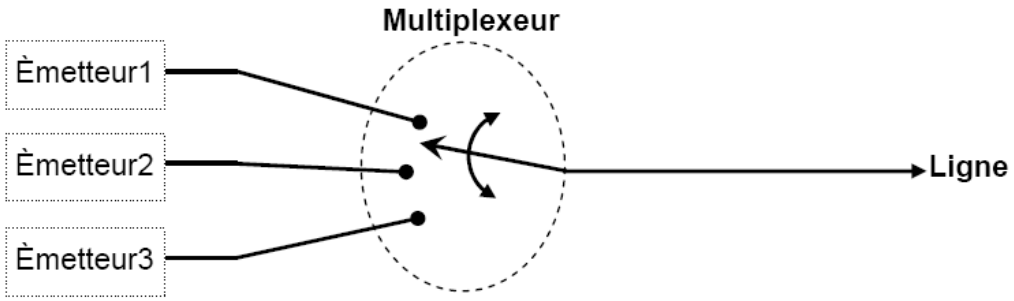

Figure (I.3): Multiplexage temporel

De même que dans le cas du multiplexeur fréquentiel, aucun adressage n'est nécessaire, la position de l'information dans le flot permet d'identifier l'émetteur.

Le multiplexage temporel synchrone présente l'inconvénient que, même si l'ordinateur A et B n'ont aucune données à transmettre, l'ordinateur C doit quand même attendre que leur temps d'émission soit écoulé avant de pouvoir émettre à son tour. Le multiplexage statistique permet d'utiliser plus efficacement le support de transmission lorsqu'une ou plusieurs sources n'ont aucune donnée à émettre.

d- Le multiplexage statistique

Le multiplexeur statistique, appelé aussi multiplexage intelligent (ITDM, Intelligent Time Division Multiplexing) attribue également un intervalle de temps d'émission à chaque appareil, qui doit attendre son tour pour transmettre, mais lorsqu'un appareil n'a aucune donnée à envoyer, le multiplexeur saute son tour et passe directement à l'appareil suivant. Ces multiplexeurs intelligents sont aussi appelés concentrateur. La plupart des réseaux informatiques utilisent une forme de multiplexage statistique, car les différents ordinateurs reliés n'ont pas besoin d'envoyer des données en permanence. Pour être sûr qu'un ordinateur transmettant des données laissera les autres émettre à leur tour, on fixe une limite supérieure pour la quantité de données pouvant être envoyés en une fois : on appelle cette quantité maximale un paquet.

I.4.2 Couche liaison de données ou niveau 2

La couche liaison utilise la couche physique pour offrir à la couche réseau une liaison sans erreurs. C'est elle qui fractionne les données en trames et en paquets, elle gère les trames d'acquittement et les réémissions ainsi que les collisions à partir de mesures effectuées par la couche physique.

Cette couche donc assure la transmission fiable d'un flux de bits entre deux nœuds adjacents du réseau sur un support physique procuré par la couche 1, dont les deux équipements terminaux sont connectés par un canal de transmission, c'est à dire que les bits émis à une extrémité sont délivrés exactement dans le même ordre à l'autre extrémité. Le travail de la couche 2 consiste à faire en sorte qu'il y ait une transmission sans omission ni déformation et remis à la couche 3.

Elle doit donc fournir également les moyens fonctionnels et procéduraux nécessaires à l'établissement, au maintien et à la libération des connections de liaison de données point à point entre 2 entités du réseau.

La liaison point à point gérée par le protocole PPP (*Point to Point Protocol* est généralement le procédé utilisé pour les liaisons à longue distance qui constituent un réseau étendu, ou WAN (Wide *Area Network*).

La figure (I.4) représente cette coopération des fonctions fournies par chaque couche.

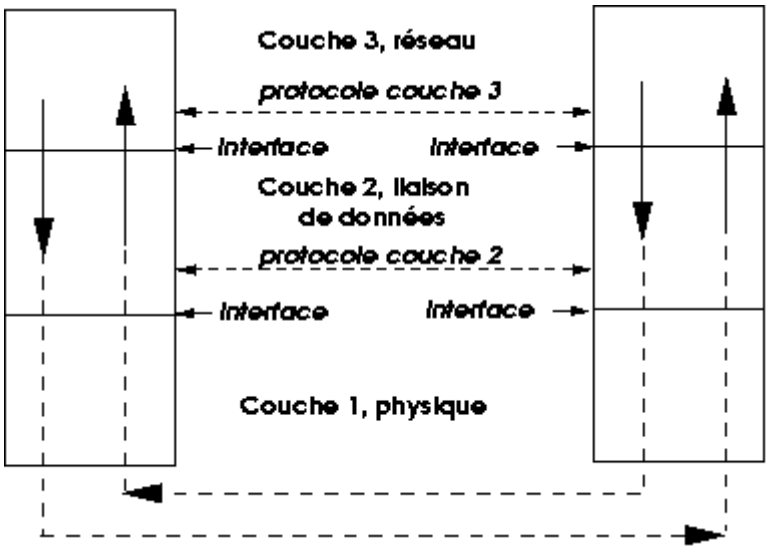

Figure (I.4): Les trois couches basses

Dans le cas d'un réseau avec plusieurs utilisateurs partageant une ressource c'est une partie spécifique de cette couche appelée Medium Access Control (MAC) qui organise l'accès à la ressource. Cette sous couche MAC est d'une importance capitale dans les systèmes d'accès radio.

Les performances des systèmes de communications sont fortement liées aux choix techniques qui permettent à des utilisateurs multiples d'accéder à un canal de transmission.

Cette section aborde les mécanismes d'allocations des ressources physiques que ces dernières soient des fréquences, des 'times slots' ou des codes d'étalement. Cet ensemble de mécanismes qui constitue la sous couche (MAC), de la couche "Liaison de Données" définie par l'ISO. On rappellera donc brièvement le modèle en couche puis on s'intéressera plus particulièrement aux techniques d'accès aléatoires.

I.4.2.1 Protocoles d'Accès et fonctions de la couche MAC

On rappelle qu'il s'agit en fait d'une sous couche de la couche liaison dont le but principal est d'attribuer un canal à accès multiples à plusieurs utilisateurs. Ce type d'allocation revêt une importance capitale dans les systèmes radio dit point multipoints pour lesquels une station centrale, station de base d'un système cellulaire terrestre ou station terrienne d'un système satellitaire, est reliée à un grand nombre d'utilisateurs équipés de terminaux.

L'allocation des ressources peut être statique. C'est le cas dans les systèmes fonctionnant selon un mode dit "circuit". Un circuit de capacité fixée est établi à l'initialisation de la communication et il est libéré à la fin de la communication. Ce «circuit" peut revêtir différentes formes. Ainsi dans le GSM il s'agira d'un time slot revenant régulièrement en voie descendante et en voie montante. Ce time slot régulier permet au terminal de recevoir et d'écouler un trafic régulier [1]-[4].

L'occurrence du time slot est identique à la disponibilité continue d'une paire de cuivre. On parle donc de circuit. Dans d'autres systèmes, le circuit peut être la disponibilité régulière d'un code d'étalement sur un time slot et sur une fréquence porteuse. La régularité de la disponibilité de la ressource fait que l'on parle toujours de circuit.

Appliquée à un trafic de données, de nature très sporadique, cette approche statique est fortement sous optimale.

Considérons un canal radio ayant une capacité égale à C bits/sec. Considérons un trafic de données mises sous forme de paquets avec un rythme d'arrivée des paquets égal à λ paquets par seconde. La taille des paquets suit une loi exponentielle avec en moyenne $1/\mu$
bits par paquet.

Considérons un mécanisme d'allocation sachant gérer au mieux la totalité de la capacité du canal. Le délai moyen d'attente T est alors donné par l'expression suivante :

$$T = \frac{1}{\mu C - \lambda} \qquad (I.1)$$

Si on considère maintenant que le mécanisme d'allocation effectue une "division" du canal en N sous-canaux indépendants, chaque canal a une capacité de C/N bits/sec. Le débit moyen sur chaque canal est égal à λ/N.

. En recalculant le délai moyen d'attente T', on obtient :

$$T' = \frac{1}{\mu C/N - \lambda/N} = NT \qquad (I.2)'$$

Ethernet est un exemple de liaison de données, dont le nom Ethernet est un hommage à un ancêtre de ce protocole, ALOHA, inventé à l'Université d'Hawaï en 1970 par Norman Abramson.

Comme Abramson voulait relier les campus de l'université, situés sur des îles différentes, ALOHA était un protocole de réseau par radio, les communications circulaient dans l'éther.

Selon ALOHA, qui est un protocole d'accès multiple toutes les stations émettent et reçoivent sur la même bande de fréquence. Les messages sont découpés en trames, identifiées par un numéro d'ordre et l'adresse de la station destinataire. C'est une conversation à plusieurs : toutes les stations reçoivent toutes les trames, identifient celles qui leur sont destinées, jettent les autres. La communication par radio entre sites distants interdit toute idée de contrôle centralisé : ALOHA doit prévoir le cas où deux stations (ou plus) voudraient commencer à émettre en même temps. Cette circonstance est nommée *collision*, et résulte en trames brouillées, incompréhensibles, en un mot perdues. Il va falloir réémettre ces trames perdues, et si possible en évitant de provoquer une nouvelle collision, sinon le protocole n'aboutira jamais.

Pour diminuer le risque de nouvelles collisions, ALOHA utilise un algorithme probabiliste : chacune des stations qui a émis une trame perdue à cause de la collision calcule un nombre aléatoire, en déduit un intervalle de temps et réémet. La probabilité que deux stations aient calculé le même délai est très faible ; si néanmoins c'est le cas, une nouvelle collision a lieu, et il faut réitérer le calcul. La probabilité que deux stations calculent trois fois de suite le même délai est tellement faible que cet algorithme, en pratique, fonctionne très bien. Il peut être encore amélioré au moyen d'une horloge centrale qui émet un signal dont la période est égale au délai de transmission d'une trame à travers le réseau : les trames ne peuvent être émises qu'au « top » d'horloge donné par ce signal. Cette discrétisation des émissions améliore l'efficacité du réseau en diminuant l'intervalle de temps minimum avant réémission en cas de collision.

Ethernet utilise le même principe qu'ALOHA : le support physique du réseau est accessible par toutes les stations simultanément.

I.4.2.2 Détection de trames endommagées

Le flot de bits, que la couche 1 est prête à fournir à la demande, est découpé par les protocoles de couche 2 en entités discrètes de longueur limitée appelées *trames* (*frame*).

Le découpage du flot de données en trames nécessite un moyen de reconnaître le début et la fin d'une trame. Les solutions raisonnables sont :

- séparer les trames par des intervalles de « silence » ; cette solution est employée pour les réseaux locaux parce que le débit est élevé et la bande passante disponible à profusion ;
- utiliser des trames de longueur soit fixe, soit variable mais connue en cours de transmission, et compter les bits ; cette solution et la suivante sont employées notamment pour les liaisons à longue distance ;
- utiliser des configurations de bits particulières et inutilisées par le codage des données pour marquer le début et la fin de trames, et guetter leur occurrence.

Les trames seront les entités dont les procédures de détection d'erreur vérifieront l'intégrité. Nous considérons ici, dans un premier temps, le traitement des trames qui arrivent à destination mais dont le contenu a été altéré par un parasite quelconque.

Le principe de base de la détection de ce type d'erreur est la redondance : avant d'émettre une trame, la station émettrice ajoute au message à transmettre (ici le contenu de la trame) une information supplémentaire calculée à partir des bits du message selon un algorithme dit de hachage *(hash)*. À la réception, la station réceptrice effectue le même calcul ; si elle ne trouve pas le même résultat c'est qu'il y a eu une erreur. Cette information supplémentaire calculée à partir de l'information utile s'appelle *somme de contrôle* (checksum). L'algorithme de calcul de la somme de contrôle doit bien sûr être le même aux deux extrémités : cette convention fait partie du protocole. Une méthode très répandue est le code de redondance cyclique (CRC).

Si le calcul prévu par la procédure donne le résultat attendu, il n'y a pas d'erreur et alors, dans le cas des réseaux longue distance (WAN), le protocole de couche 2 (côté récepteur) envoie un acquittement (conventionnellement ACK) à l'émetteur ; sinon il envoie un acquittement négatif (NAK) qui demande à l'émetteur de retransmettre la trame considérée. Pour savoir quelle trame est acquittée, le protocole prévoit aussi que chaque trame comporte un numéro de séquence permettant de la distinguer des précédentes et des suivantes. Ethernet procède sans échange d'acquittements : les détections d'erreur sont signalées par un signal de couche 1.

Il existe aussi des codes auto-correcteurs, dont le plus célèbre est le code de Hamming : ces codes permettent de connaître précisément les positions des bits erronés, s'il n'y en a pas trop, et procèdent à les corriger.

I.4.2.3 Contrôle de flux

Les contrôles d'erreurs faisaient l'hypothèse que les délais de transmission d'une trame et de l'acquittement en sens inverse étaient négligeables, ce qui est vrai pour un réseau local mais beaucoup moins pour une liaison à longue distance.

Dans le cas où un émetteur a une interface réseau beaucoup plus rapide que celle du récepteur, et de ce fait, le récepteur ne pourra pas absorber toutes les trames qui lui sont envoyées et des données vont se perdre. Il faut donc un algorithme pour réguler les transmissions, et notamment s'assurer que les trames envoyées sont bien reçues.

La solution qui consiste, avant d'envoyer une nouvelle trame, à attendre d'avoir reçu un acquittement positif pour la précédente, impose un ralentissement et une utilisation inefficace de la bande passante. Cette inefficacité est particulièrement grande pour les liaisons par satellite géostationnaire, qui peuvent avoir un débit élevé mais un temps de transit incompressible de l'ordre d'un quart de seconde : s'il fallait attendre l'acquittement, on aurait un débit de deux trames par seconde, ce qui ne serait évidemment pas acceptable.

Pour améliorer cette situation, les protocoles adaptés aux liaisons à délai de transit important utilisent un algorithme basé sur le principe suivant : l'émetteur n'attend l'acquittement de la trame numéro *n* qu'après l'émission de la trame *n*+*p*, avec *p*>1. Le délai nécessaire à la transmission de *p* trames est appelé *délai de garde (timeout interval).* Cette méthode est appelée *pipeline*, parce que l'on enfonce les trames dans le « tuyau » sans attendre que les précédentes soient sorties. Comme à la section précédente, chaque trame est dotée d'un numéro de séquence qui permet de savoir, notamment, quelle trame acquitte tel ACK (dans le cas des protocoles WAN).

Lorsqu'une trame, au milieu d'un long message, est corrompue ou n'arrive pas, comme c'est illustré par la figure (I.5) où nous supposerons *p*=3. La trame 2 est émise, mais perdue ou détériorée en route. Le récepteur détecte le problème : si la trame est détériorée, par une des méthodes de détection d'erreur indiquées à la section précédente; si elle est perdue en route, le contrôle des numéros de séquence montre que la trame 1 devrait être suivie de la trame 2, or il reçoit à la place la trame 3 (ou une autre...), qui ne satisfait pas aux règles du protocole.

Dès que le récepteur constate la défaillance de la trame 2, il rejette toutes les trames que l'émetteur continue à lui envoyer, et après *p* émissions, soit après la trame *n*+*p*=2+3=5, il s'attend à recevoir l'acquittement de la trame 2. Il va réémettre la trame 2, puis toutes ses suivantes. Le récepteur va recevoir la trame 2 attendue, il va l'accepter et l'acquitter, ainsi que les suivantes.

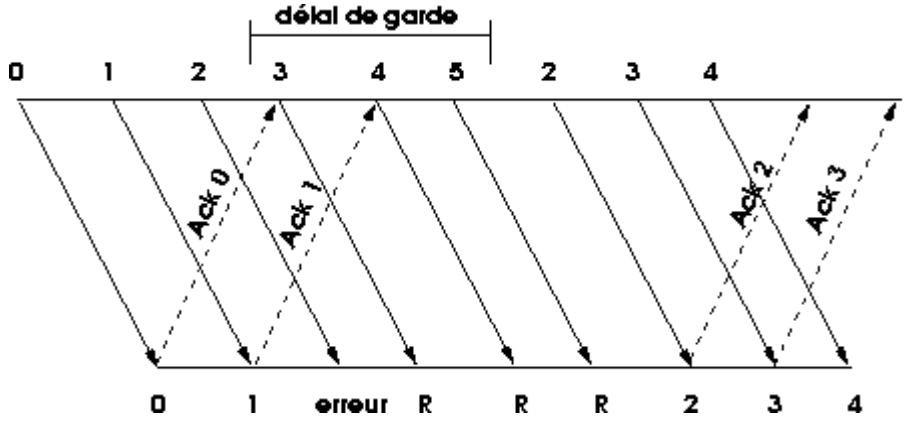

Figure (I.5) : Fenêtre glissante

I.4.3 Couche réseau ou niveau 3

Ce niveau est chargé de l'acheminement et de la communication en paquets d'information ainsi que de la gestion des connexions réseaux. Ces principales fonctionnalités sont donc : le routage, la recherche du chemin, le tri des unités de données, la gestion des adresses.

De façon générale, un certain nombre de points reliés par des lignes constituent un graphe. Les points reliés sont appelés sommets (*vertex*) du graphe, et la ligne qui relie deux points est appelée arc (*edge*). Si l'arc qui relie un sommet **A** à un autre, **B** par exemple, relie également **B** à **A**, on dit que le graphe n'est pas orienté ; dans les graphes orientés les arcs ont un sens ; les graphes dont nous parlerons sont non orientés. La figure (I.6) représente un graphe non orienté *G* à sept sommets.

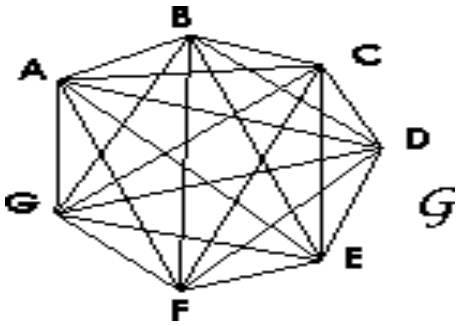

Figure (I.6) : Graphe connexe complet

Le graphe *G*, ou **ABCDEFG**, est tel qu'entre deux quelconques de ses sommets il existe au moins un chemin constitué d'arcs du graphe : un tel graphe est dit *connexe*. De surcroît, chaque sommet est relié par un arc à chacun des autres : c'est un graphe connexe *complet*. Chaque sommet est relié par un arc à chacun des *n*-1 autres sommets, et chaque arc joue le même rôle pour les deux sommets qu'il relie, *G* possède donc $n \times (n-1)/2$ arcs.

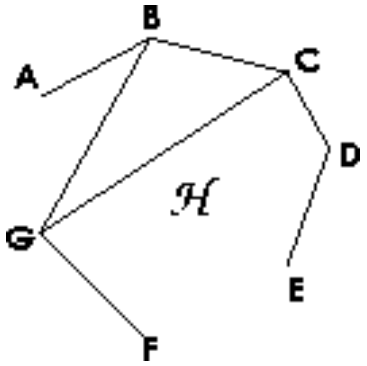

Figure (I.7) : Graphe simplement connexe

La figure (I.7) représente un graphe *H* à sept sommets simplement connexe. Lorsque l'on parle de réseaux informatiques, il peut être commode de les représenter par des graphes. Les sommets sont alors généralement appelés nœuds du réseau, et les arcs, lignes ou liaisons.

Dès que notre réseau comportera un nombre *n* d'équipements terminaux supérieur à 2 ou 3, il ne sera plus raisonnable de les relier deux à deux par des supports physiques dont le nombre serait égal à $n \times (n\text{-}1)/2$, même s'il s'agit de faisceaux hertziens. Il faut donc acheminer les messages par un trajet complexe qui passe par plusieurs segments de liaison de données.

I.4.4 Couche transport ou niveau 4

Cette couche est responsable du bon acheminement des messages complets au destinataire. Le rôle principal de la couche transport est de prendre les messages de la couche session, de les découper s'il le faut en unités plus petites et de les passer à la couche réseau, tout en s'assurant que les morceaux arrivent correctement de l'autre côté.

Cette couche effectue donc aussi le réassemblage du message à la réception des morceaux.

Cette couche est également responsable de l'optimisation des ressources du réseau : en toute rigueur, la couche transport crée une connexion réseau par connexion de transport requise par la couche session, mais cette couche est capable de créer plusieurs connexions réseau par processus de la couche session pour répartir les données, par exemple pour améliorer le débit. A l'inverse, cette couche est capable d'utiliser une seule connexion réseau pour transporter plusieurs messages à la fois grâce au multiplexage. Dans tous les cas, tout ceci doit être transparent pour la couche session.

Cette couche est également responsable du type de service à fournir à la couche session, et finalement aux utilisateurs du réseau : service en mode connecté ou non, avec ou sans garantie d'ordre de délivrance, diffusion du message à plusieurs destinataires à la fois... Cette couche est donc également responsable de l'établissement et du relâchement des connexions sur le réseau.

Un des tous derniers rôles à évoquer est le contrôle de flux. C'est l'une des couches les plus importantes, car c'est elle qui fournit le service de base à l'utilisateur, et par ailleurs c'est elle qui gère l'ensemble du processus de connexion, avec toutes les contraintes qui y sont liées.

I.4.5 Couche session ou niveau 5

Elle fournit des outils de synchronisation et de gestion du dialogue entre entités communicantes.

Ces fonctionnalités sont principalement des services nécessaires à l'établissement d'une connexion de session entre deux entités de présentation (niveau 6), ce qui induit entre autres, la gestion des interruptions, des reprises, checkpoints.

I.4.6 Couche présentation ou niveau 6

La couche présentation se charge de la représentation des informations lorsque des entités d'application (niveau 7) se communiquent, ou auxquelles elles se réfèrent au cours de leur dialogue. Cette couche est surtout utile en environnement hétérogène car c'est elle qui est chargée de décrire de manière cohérente les données et de les coder sous une forme universelle dans le réseau. C'est aussi cette couche qui gère une partie des problèmes de sécurité, en particulier ceux relatifs à la sûreté du contenu des messages. Le cryptage/décryptage est donc un des services présents dans cette couche comme la compression, typage, …

I.4.7 Couche application ou niveau 7

C'est la couche chargée de la communication entre les processus application et le modèle OSI. Elle définit les formats de données spécifiques à une application (mail, ftp, web, …) ; C'est la seule couche ouverte vers l'extérieur. Toute normalisation est donc très difficile.

I.5 Transmission des données à travers le modèle OSI

Le processus émetteur remet les données à envoyer au processus récepteur à la couche application qui leur ajoute un en-tête application AH (éventuellement nul).

Le résultat est alors transmis à la couche présentation. La couche présentation transforme alors ce message et lui ajoute (ou non) un nouvel en-tête (éventuellement nul). La couche présentation ne connaît et ne doit pas connaître l'existence éventuelle de AH ; pour la couche présentation, AH fait partie des données utilisateur. Une fois le traitement terminé, la couche présentation envoie le nouveau "message" à la couche session et le même processus recommence de nouveau. Les données atteignent alors la couche physique qui va effectivement transmettre les données au destinataire. A la réception, le message va remonter les couches et les en-têtes sont progressivement retirés jusqu'à atteindre le processus récepteur. (voir Fig.(I.8)) [2].

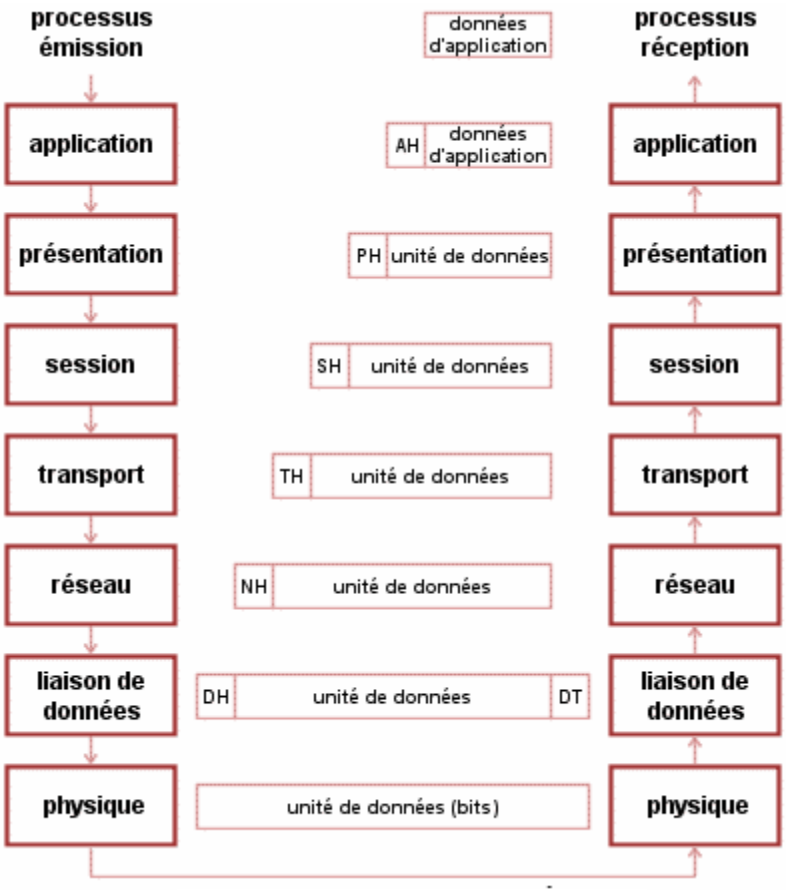

Figure (I.8): Canal de transmission de données

Le concept important est le suivant : il faut considérer que chaque couche est programmée comme si elle était vraiment horizontale, c'est à dire qu'elle dialoguait directement

avec sa couche paire réceptrice. Au moment de dialoguer avec sa couche paire, chaque couche rajoute un en-tête et l'envoie (virtuellement, grâce à la couche sous-jacente) à sa couche paire.

I. 6 Critique du modèle OSI

La chose la plus frappante à propos du modèle OSI est que c'est peut-être la structure réseau la plus étudiée et la plus unanimement reconnue et pourtant, ce n'est pas le modèle qui a su s'imposer. Les spécialistes qui ont analysé cet échec en ont déterminé quatre raisons principales.

David Clark a publié une théorie quant à l'art et la manière de publier une norme au bon moment. Pour lui, dans le cycle de vie d'une norme, il y a deux pics principaux d'activité : la recherche effectuée dans le domaine couvert par la norme, et les investissements des industriels pour l'implémentation et la mise en place de la norme. Ces deux pics sont séparés par un creux d'activité qui est en fait le moment idéal pour la publication de la norme : il n'est ni trop tôt par rapport à la recherche et on peut donc assurer une certaine maîtrise, et il n'est ni trop tard pour les investissements et les industriels qui sont prêts à utiliser des capitaux pour l'implémenter. Le modèle OSI était idéalement placé par rapport à la recherche, mais hélas, le modèle TCP/IP était déjà en phase d'investissement prononcé (lorsque le modèle OSI est sorti, les universités américaines utilisaient déjà largement TCP/IP avec un certain succès) et les industriels n'ont pas ressenti le besoin d'investir dessus.

Le modèle OSI est peut-être trop complet et trop complexe. La distance entre l'utilisation concrète (l'implémentation) et le modèle est parfois importante. En effet, peu de programmes peuvent utiliser ou utilisent mal l'ensemble des sept couches du modèle : les couches session et présentation sont fort peu utilisées et à l'inverse les couches liaison de données et réseau sont très souvent découpées en sous-couches tant elles sont complexes.

OSI est en fait trop complexe pour pouvoir être proprement et efficacement implémenté. Le comité rédacteur de la norme a même dû laisser de côté certains points techniques, comme le la sécurité et le codage, tant il était délicat de conserver un rôle bien déterminé à chaque couche ainsi complétée. Ce modèle est également redondant (le contrôle de flux et le contrôle d'erreur apparaissent pratiquement dans chaque couche). Au niveau de l'implémentation, TCP/IP est beaucoup plus optimisé et efficace [3], [4].

La plus grosse critique que l'on peut faire au modèle est qu'il n'est pas du tout adapté aux applications de télécommunication sur ordinateur ! Certains choix effectués sont en désaccord avec la façon dont les ordinateurs et les logiciels communiquent. La norme a fait le choix d'un "système d'interruptions" pour signaler les événements, et sur des langages de programmation de haut niveau, cela est peu réalisable.

Cela tient tout simplement du fait que le modèle est relativement complexe, et que du coup les premières implémentations furent relativement lourdes et lentes. A l'inverse, la première implémentation de TCP/IP dans l'Unix de l'université de Berkeley (BSD) était gratuite et relativement efficace. Historiquement, les gens ont donc eu une tendance naturelle à utiliser TCP/IP.

Le modèle OSI a en fait souffert de sa trop forte normalisation. Les efforts d'implémentation du modèle étaient surtout "bureaucratiques" à l'inverse, TCP/IP est venu d'Unix et a été tout de suite utilisé, qui plus est par des centres de recherches et les universités, c'est-à-dire les premiers à avoir utilisé les réseaux de manière poussée. Le manque de normalisation de TCP/IP a été contre- balancé par une implémentation rapide et efficace, et une utilisation dans un milieu propice à sa propagation.

I.7 L'avenir d'OSI et les protocoles standards de communication

Au niveau de son utilisation et implémentation, et ce malgré une mise à jour du modèle en 1994, OSI a clairement perdu la guerre face à TCP/IP. Seuls quelques grands constructeurs dominant conservent le modèle mais il est amené à disparaître d'autant plus vite qu'Internet (et donc TCP/IP) explose.

Le modèle OSI restera cependant encore longtemps dans les mémoires pour plusieurs raisons. C'est d'abord l'un des premiers grands efforts en matière de normalisation du monde des réseaux. Les constructeurs ont maintenant tendance à faire avec TCP/IP, mais aussi le WAP, l'UMTS etc. ce qu'il devait faire avec OSI, à savoir proposer des normalisations dès le départ. OSI marquera aussi les mémoires pour une autre raison : même si c'est TCP/IP qui est concrètement utilisé, les gens ont tendance et utilisent OSI comme le modèle réseau de référence actuel. En fait, TCP/IP et OSI ont des structures très proches, et c'est surtout l'effort de normalisation d'OSI qui a imposé cette "confusion" générale entre les 2 modèles. On a communément tendance à considérer TCP/IP comme l'implémentation réelle d'OSI [3], [4].

I.8 Le modèle TCP/IP

I.8.1 Les caractéristiques du protocole TCP

Le TCP (Protocole de Contrôle de Transmission) est un des principaux protocoles de la couche transport du modèle TCP/IP. Il permet, au niveau des applications, de gérer les données en provenance (ou à destination) de la couche inférieure du modèle (c'est-à-dire le protocole IP). Lorsque les données sont fournies au protocole IP, celui-ci les encapsule dans des datagrammes IP, en fixant le champ protocole à 6 (Pour savoir que le protocole en amont est TCP...). TCP est un protocole orienté connexion, c'est-à-dire qu'il permet à deux machines qui communiquent de contrôler l'état de la transmission. Les caractéristiques principales du protocole TCP sont les suivantes :

- Permet de remettre en ordre les datagrammes en provenance du protocole IP
- Permet de vérifier le flot de données afin d'éviter une saturation du réseau
- Permet de formater les données en segments de longueur variable afin de les "remettre" au protocole IP
- Permet de multiplexer les données, c'est-à-dire de faire circuler simultanément des informations provenant de sources (applications par exemple) distinctes sur une même ligne
- Permet enfin l'initialisation et la fin d'une communication de manière courtoise

Le modèle TCP/IP, inspiré du modèle OSI, reprend l'approche modulaire (utilisation de modules ou couches) mais en contient uniquement quatre :

<table>
<tr><th>Modèle TCP/IP</th><th>Modèle OSI</th></tr>
<tr><td rowspan="3">Couche Application</td><td>Couche Application</td></tr>
<tr><td>Couche Présentation</td></tr>
<tr><td>Couche Session</td></tr>
<tr><td>Couche Transport (TCP)</td><td>Couche Transport</td></tr>
<tr><td>Couche Internet (IP)</td><td>Couche Réseau</td></tr>
<tr><td rowspan="2">Couche Accès réseau</td><td>Couche Liaison données</td></tr>
<tr><td>Couche Physique</td></tr>
</table>

Figure (I.9) : comparaison entre modèles TCP/IP et ISO

Comme on peut le remarquer, les couches du modèle TCP/IP ont des tâches beaucoup plus diverses que les couches du modèle OSI, étant donné que certaines couches du modèle TCP/IP correspondent à plusieurs couches du modèle OSI. Les rôles des différentes couches sont les suivants :

- Couche Accès réseau : elle spécifie la forme sous laquelle les données doivent être acheminées quel que soit le type de réseau utilisé
- Couche Internet : elle est chargée de fournir le paquet de données (datagramme)
- Couche Transport : elle assure l'acheminement des données, ainsi que les mécanismes permettant de connaître l'état de la transmission
- Couche Application : elle englobe les applications standard du réseau (Telnet, SMTP, FTP, ...) Voici les principaux protocoles faisant partie de la suite TCP/IP :

I.8.2 Encapsulation des données

Lors d'une transmission, les données traversent chacune des couches au niveau de la machine émettrice. A chaque couche, une information est ajoutée au paquet de données, il s'agit d'un en-tête, ensemble d'informations qui garantit la transmission. Au niveau de la machine réceptrice, lors du passage dans chaque couche, l'en-tête est lu, puis supprimé. Ainsi, à la réception, le message est dans son état originel...

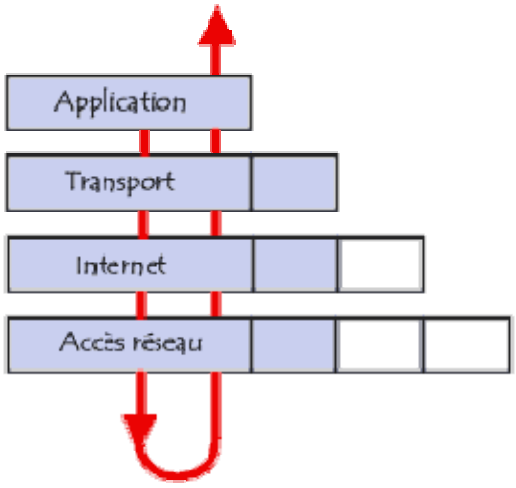

Figure (I.10) : Modèle TCP/IP

A chaque niveau, le paquet de données change d'aspect, car on lui ajoute un en-tête, ainsi les appellations changent suivant les couches :

- Le paquet de données est appelé **message** au niveau de la couche Application
- Le message est ensuite encapsulé sous forme de **segment** dans la couche Transport
- Le segment une fois encapsulé dans la couche Internet prend le nom de **datagramme**
- Enfin, on parle de **trame** au niveau de la couche Accès réseau

* La couche Accès réseau

La couche accès réseau est la première couche de la pile TCP/IP, elle offre les capacités à accéder à un réseau physique quel qu'il soit, c'est-à-dire les moyens à mettre en œuvre afin de transmettre des données via un réseau. Ainsi, la couche accès réseau contient toutes les spécifications concernant la transmission de données sur un réseau physique, qu'il s'agisse de réseau local (Anneau à jeton - token ring, Ethernet, FDDI), de connexion à une ligne téléphonique ou n'importe quel type de liaison à un réseau. Elle prend en charge les notions suivantes :

- Acheminement des données sur la liaison
- Coordination de la transmission de données (synchronisation)
- Format des données
- Conversion des signaux (analogique/numérique)
- Contrôle des erreurs à l'arrivée

Toutes ces spécifications sont transparentes aux yeux de l'utilisateur, car l'ensemble de ces tâches est en fait réalisé par le système d'exploitation, ainsi que les drivers du matériel permettant la connexion au réseau (ex : driver de carte réseau).

* La couche Internet

La couche Internet est la couche "la plus importante" (elles ont toutes leur importance) car c'est elle qui définit les datagrammes, et qui gère les notions d'adressage IP. Elle permet l'acheminement des datagrammes (paquets de données) vers des machines distantes ainsi que de la gestion de leur fragmentation et de leur assemblage à réception. La couche Internet contient 5 protocoles :

- Le protocole IP
- Le protocole ARP
- Le protocole ICMP
- Le protocole RARP
- Le protocole IGMP

Les trois premiers protocoles sont les protocoles les plus importants de cette couche.

* La couche Transport

Les protocoles des couches précédentes permettaient d'envoyer des informations d'une machine à une autre. La couche transport permet à des applications tournant sur des machines distantes de communiquer. Le problème consiste à identifier ces applications. En effet, suivant la machine et son système d'exploitation, l'application pourra être un programme, une tâche, un processus..

De plus, la dénomination de l'application peut varier d'un système à un autre, c'est la raison pour laquelle un système de numéro a été mis en place afin de pouvoir associer un type d'application à un type de données, ces identifiants sont appelés ports. La couche transport contient deux protocoles permettant à deux applications d'échanger des données indépendamment du type de réseau emprunté (c'est-à-dire indépendamment des couches inférieures...), il s'agit des protocoles suivants :

- TCP, un protocole orienté connexion qui assure le contrôle des erreurs
- UDP, un protocole non orienté connexion dont le contrôle d'erreur est archaïque

* La couche Application

La couche application est la couche située au sommet des couches de protocoles TCP/IP. Celle-ci contient les applications réseaux permettant de communiquer grâce aux couches inférieures.

Les logiciels de cette couche communiquent donc grâce à un des deux protocoles de la couche inférieure (la couche transport) c'est-à-dire TCP ou UDP. Les applications de cette couche sont de différents types, mais la plupart sont des services réseau, c'est-à-dire des

applications fournies à l'utilisateur pour assurer l'interface avec le système d'exploitation. On peut les classer selon les services qu'ils rendent :

- Les services de gestion (transfert) de fichier et d'impression
- Les services de connexion au réseau
- Les services de connexion à distance
- Les utilitaires Internet divers

I.8.3 Le but de TCP

Grâce au protocole TCP, les applications peuvent communiquer de façon sûre (grâce au système d'accusés de réception du protocole TCP), indépendamment des couches inférieures. Cela signifie que les routeurs (qui travaillent dans la couche Internet) ont pour seul rôle l'acheminement des données sous forme de datagrammes, sans se préoccuper du contrôle des données, car celui-ci est réalisé par la couche transport (plus particulièrement par le protocole TCP).

Lors d'une communication à travers le protocole TCP, les deux machines doivent établir une connexion. La machine émettrice (celle qui demande la connexion) est appelée client, tandis que la machine réceptrice est appelée serveur. On dit qu'on est alors dans un environnement Client-serveur. Les machines dans un tel environnement communiquent en mode connecté, c'est-à-dire que la communication se fait dans les deux sens.

Pour permettre le bon déroulement de la communication et de tous les contrôles qui l'accompagnent, les données sont encapsulées, c'est-à-dire qu'on ajoute aux paquets de données un en-tête qui va permettre de synchroniser les transmissions et d'assurer leur réception [3] , [4].

Une autre particularité de TCP est de pouvoir réguler le débit des données grâce à sa capacité à émettre des messages de taille variable, ces messages sont appelés *segments*.

I.8.4 La fonction de multiplexage

Le TCP permet d'effectuer une tâche importante: le multiplexage/démultiplexage, c'est-à-dire faire transiter sur une même ligne des données provenant d'applications diverses ou en d'autres mots et mettre en série des informations arrivant en parallèle (Fig.(I.10)).

Figure (I.11) : Opération Multiplexage/démultiplexage

Ces opérations sont réalisées grâce au concept de ports (ou sockets), c'est-à-dire un numéro associé à un type d'application, qui, combiné à une adresse IP, permet de déterminer de façon unique une application qui tourne sur une machine donnée.

I.8.5 Le format des données sous TCP

Un segment TCP est constitué comme suit :

										0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Port Source																Port destination															
Numéro d'ordre																															
Numéro d'accusé de réception																															
Décalage données			Réservée			URG		ACK		PSH		RST		SYN		FIN		Fenêtre													
Somme de contrôle																				Pointeur d'urgence											
Options																												Remplissage			
Données																															

Signification des différents champs :

- Port Source (16 bits): Port relatif à l'application en cours sur la machine source
- Port Destination (16 bits): Port relatif à l'application en cours sur la machine de destination
- Numéro d'ordre (32 bits): Lorsque le drapeau SYN est à 0, le numéro d'ordre est celui du premier mot du segment en cours. Lorsque SYN est à 1, le numéro d'ordre est égal au numéro d'ordre initial utilisé pour synchroniser les numéros de séquence (ISN)

- Numéro d'accusé de réception (32 bits): Le numéro d'accusé de réception également appelé numéro d'acquittement correspond au numéro (d'ordre) du prochain segment attendu, et non le numéro du dernier segment reçu.
- Décalage des données (4 bits): il permet de repérer le début des données dans le paquet. Le décalage est ici essentiel car le champ d'options est de taille variable
- Réservé (6 bits): Champ inutilisé actuellement mais prévu pour l'avenir
- Drapeaux (flags) (6x1 bit): Les drapeaux représentent des informations supplémentaires :
  - URG: si ce drapeau est à 1 le paquet doit être traité de façon urgente.
  - ACK: si ce drapeau est à 1 le paquet est un accusé de réception.
  - PSH (PUSH): si ce drapeau est à 1, le paquet fonctionne suivant la méthode PUSH.
  - RST: si ce drapeau est à 1, la connexion est réinitialisée.
  - SYN: Le Flag TCP SYN indique une demande d'établissement de connexion.
  - FIN: si ce drapeau est à 1 la connexion s'interrompt.
- Fenêtre (16 bits): Champ permettant de connaître le nombre d'octets que le récepteur souhaite recevoir sans accusé de réception
- Somme de contrôle (Checksum ou CRC): La somme de contrôle est réalisée en faisant la somme des champs de données de l'en-tête, afin de pouvoir vérifier l'intégrité de l'en-tête
- Pointeur d'urgence (16 bits): Indique le numéro d'ordre à partir duquel l'information devient urgente
- Options (Taille variable): Des options diverses
- Remplissage: On remplit l'espace restant après les options avec des zéros pour avoir une longueur multiple de 32 bits

I.8.6 Fiabilité des transferts

Le protocole TCP permet d'assurer le transfert des données de façon fiable, bien qu'il utilise le protocole IP, qui n'intègre aucun contrôle de livraison de datagramme.

En réalité, le protocole TCP possède un système d'accusé de réception permettant au client et au serveur de s'assurer de la bonne réception mutuelle des données. Lors de l'émission d'un segment, un **numéro d'ordre** (appelé aussi *numéro de séquence*) est associé. A réception d'un segment de donnée, la machine réceptrice va retourner un segment de donnée dont le drapeau ACK est à 1 (afin de signaler qu'il s'agit d'un accusé de réception) accompagné d'un

numéro d'accusé de réception égal au numéro d'ordre précédent comme le montre la figure (I.12).

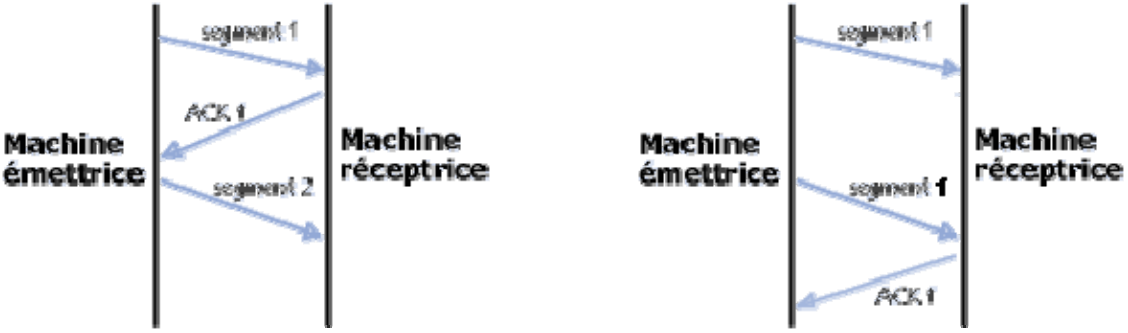

Figure (I.12 :système d'accusé de recxeption du modèle TCP

De plus, grâce à une minuterie déclenchée dès réception d'un segment au niveau de la machine émettrice, le segment est réexpédié dès que le temps imparti est écoulé, car dans ce cas la machine émettrice considère que le segment est perdu.

Toutefois, si le segment n'est pas perdu et qu'il arrive tout de même à destination, la machine réceptrice saura grâce au numéro d'ordre qu'il s'agit d'un doublon et ne conservera que le dernier segment arrivé à destination.

I.8.7 Etablissement d'une connexion

Etant donné que ce processus de communication, qui se fait grâce à une émission de données et d'un accusé de réception, est basé sur un numéro d'ordre (appelé généralement *numéro de séquence*), il faut que les machines émettrices et réceptrices (client et serveur) connaissent le numéro d'ordre initial de l'autre machine.

L'établissement de la connexion entre deux applications se fait souvent selon le schéma suivant :

- Les ports TCP doivent être ouverts
- L'application sur le serveur est passive, c'est-à-dire que l'application est à l'écoute, en attente d'une connexion
- L'application sur le client fait une requête de connexion sur le serveur dont l'application est en ouverture passive. L'application du client est dite "en ouverture active"

Les deux machines doivent donc synchroniser leurs séquences grâce à un mécanisme communément appelé *three ways handshake* (*poignée de main en trois temps*), que l'on retrouve aussi lors de la clôture de session.

Ce dialogue permet d'initier la communication, il se déroule en trois temps, comme sa dénomination l'indique :

- Dans un premier temps la machine émettrice (le client) transmet un segment dont le drapeau SYN est à 1 (pour signaler qu'il s'agit d'un segment de synchronisation), avec un numéro d'ordre N, que l'on appelle numéro d'ordre initial du client
- Dans un second temps la machine réceptrice (le serveur) reçoit le segment initial provenant du client, puis lui envoie un accusé de réception, c'est-à-dire un segment dont le drapeau ACK est à 1 et le drapeau SYN est à 1 (car il s'agit là encore d'une synchronisation). Ce segment contient le numéro d'ordre de cette machine (du serveur) qui est le numéro d'ordre initial du client. Le champ le plus important de ce segment est le champ accusé de réception qui contient le numéro d'ordre initial du client, incrémenté de 1
- Enfin, le client transmet au serveur un accusé de réception, c'est-à-dire un segment dont le drapeau ACK est à 1, dont le drapeau SYN est à zéro (il ne s'agit plus d'un segment de synchronisation). Son numéro d'ordre est incrémenté et le numéro d'accusé de réception représente le numéro d'ordre initial du serveur incrémenté de 1. (voir Fig.(I.12)).

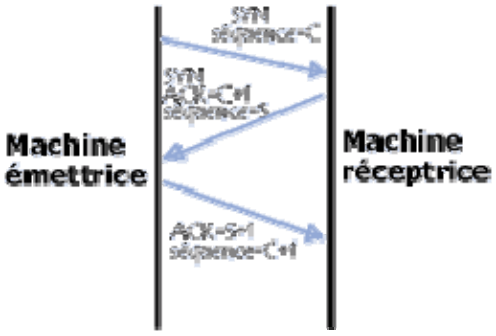

Figure (I.13) : Echange entre émetteur et récepteur

Suite à cette séquence comportant trois échanges les deux machines sont synchronisées et la communication peut commencer.

I.8.8 Méthode de la fenêtre glissante

Dans de nombreux cas, il est possible de limiter le nombre d'accusés de réception, afin de désengorger le réseau, en fixant un nombre de séquence au bout duquel un accusé de réception est nécessaire. Ce nombre est en fait stocké dans le champ *fenêtre* de l'en-tête TCP/IP.

On appelle effectivement cette méthode "*méthode de la fenêtre glissante*" car on définit en quelque sorte une fourchette de séquences n'ayant pas besoin d'accusé de réception, et celle-ci se déplace au fur et à mesure que les accusés de réception sont reçus (voir Figure (I.13)).

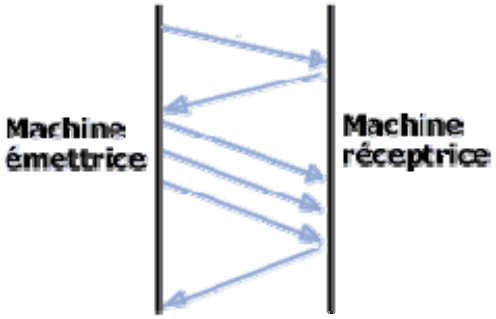

Figure (I.14) : Limitation des accusés de réception

De plus, la taille de cette fenêtre n'est pas fixe. En effet, le serveur peut inclure dans ses accusés de réception en stockant dans le champ fenêtre la taille de la fenêtre qui lui semble la plus adaptée. Ainsi, lorsque l'accusé de réception indique une demande d'augmentation de la fenêtre, le client va déplacer le bord droit de la fenêtre.

Par contre, dans le cas d'une diminution, le client ne va pas déplacer le bord droit de la fenêtre vers la gauche mais attendre que le bord gauche avance (avec l'arrivée des accusés de réception).

I.9 Conclusion

En fait, le modèle OSI se compose de 07 couches, cependant, nous mettons plus d'attention sur la couche du niveau 2 : couche liaison de données, celle-ci se caractérise par :

- L'utilisation de la couche physique pour offrir à la couche réseau une liaison sans erreurs.
- Elle fractionne les données en trame et en paquet
- Elle gère les trames d'acquittement et les réémissions ainsi que les collisions à partir de mesures effectuées par la couche physique.
- Elle assure la transmission fiable d'un flux de bits entre deux nœuds adjacents

En effet, le protocole Aloha relevant de cette couche qui est un protocole d'accès multiple selon lequel, toutes les stations émettent et reçoivent sur la même bande de fréquence, sera analysé au chapitre suivant.

CHAPITRE II

Techniques d'accès aux canaux satellitaires et Analyse du protocole Aloha

II.1 Introduction

Les performances des systèmes de radiocommunications sont fortement liées aux choix techniques qui permettent à des utilisateurs multiples d'accéder à un canal de transmission. Ce chapitre aborde les mécanismes d'allocations des ressources physiques que ces dernières soient des fréquences, des times slots ou des codes d'étalement.

Une des principales caractéristiques de la transmission de données par satellite est l'importance de la distance de transmission. En effet l'éloignement entre l'émetteur terrien et le récepteur spatial (satellite) est trés important : 35800 Km pour le cas des satellites géostationnaires. Cette distance implique donc non seulement un retard de transmission important (de l'ordre de 240 ms pour atteindre le satellite et rejoindre la terre couvrant ainsi un trajet de 71600 Km), mais cela a aussi pour conséquence un affaiblissement considérable du signal hertzien émis [1].

Ces deux conséquences amènent donc à considérer la télécommunication par satellite comme une télécommunication spécifique possédant d'une part ses propres contraintes qui est la réserve à certains types de communications et d'autre part ses techniques propres de transmission :

- Les télécommunications par satellite sont en très grande partie réservées à des transmissions sans acquittements et supportant un retard de transmission important.
- L'affaiblissement du signal dû à la distance de transmission implique l'utilisation de technique d'amplification, de modulation et de transposition de fréquences.

II. 2 Les politiques d'accés aux canaux satellites

Aujourd'hui, de nombreuses stations terrestres veulent accéder aux satellites par l'intermédiare de fréquences spécifiques pour pouvoir transporter leurs informations. S'il n'y avait pas de politique d'accés au support bien définie, les signaux transmis par une station se confondent avec d'autres provenant d'une source différente ; ces signaux seraient alors incompréhensibles et impossibles à décoder. Ceci engendrerait donc leur perte et il serait nécessaire de les retransmettre. De plus, il n'est pas envisageable d'avoir des canaux de satellite dédiés à une station, car ce système serait beucoup trop coûteux. La mise en place d'une politique d'accés aux canaux satellites a donc été réalisée tout d'abord pour permettre à plusieurs stations ou utilisateurs d'accéder à un même canal de transmission, ensuite pour avoir

une exploitation maximale des transpondeurs du satellite tout en garantissant qu'il y ait le moins de collisions possibles.

II. 3 Les politiques d'accés aléatoires

Les politiques d'accés aléatoires pour les réseaux satellites sont sensiblement les mêmes que celles définies pour les réseaux locaux. Nous définissons parmi elles ; la technique Pur Aloha et Aloha en tranches (Discrétisé) [5]-[7].

II.3.1 La technique PUR ALOHA

Selon cette méthode, des stations émettent de façon inconditionnelles, des paquets dés qu'ils sont en leur pocession, il n'y a pas d'écoute du support avant la transmission. De plus, le temps de propagation des signaux sur le canal satellite est un facteur contraignant, car les stations sont averties d'une collision seulement 270 ms après l'émission des données. Dans le cas où la transmission des données ne s'est pas bien passée, la station va retransmettre les paquets après un retard aléatoire [1].

La probabilité pour que k trames soient générées pendant un temps égal à la durée d'une trame est donnée par la formule de distribution de Poisson :

$$P(k)= G^K e^{-G}/ k! \qquad (II.1)$$

Où : G est le pourcentage moyen du nombre de trames transmises pendant un temps égal à la durée d'une trame.

Dans un intervalle de durée égale à 2 «durée de trame », le nombre moyen de trames générées est 2G. La probabilité pour qu'aucun trafic supplémentaire ne soit généré pendant cette période vulnérable est donnée par :

$$P_0 = e^{-2G} \qquad (II.2)$$

Donc en posant $S= G.P_0$, nous obtenons :

$$S= Ge^{-2G} \qquad (II.3)$$

Où S est le nombre moyen de trames par durée de trame. Le trafic maximum est obtenu pour G=0.5 , ce qui donne S= ½, donc une valeur de 0.184.

Cette méthode d'accés a donné un taux d'utilisation du canal satellite faible, s'approchant de 20 % (voir Figure (III.1)), d'où l'apparition de techniques similaires mais avec des modificatins qui apportent de meilleurs performences.

II.3.2 ALOHA en tranches ou discrétisé

L'idée principale de cette méthode est de découper le temps en tranches correspondant chacune au temps de transmission d'un paquet, les émissions sont alors synchronisées en début de tranches. Grâce à cette méthode, s'il y a détection de collisions, c'est sur l'ensemble de la tranche de temps, et non plus sur une partie d'un paquet [7]- [10].

Etant donné que la durée de la période de vulnérabilité est divisée par deux, on obtient [1]:

$$S= Ge^{-G} \qquad (II.4)$$

Avec un trafic maximal égal à 1/e pour G=1.

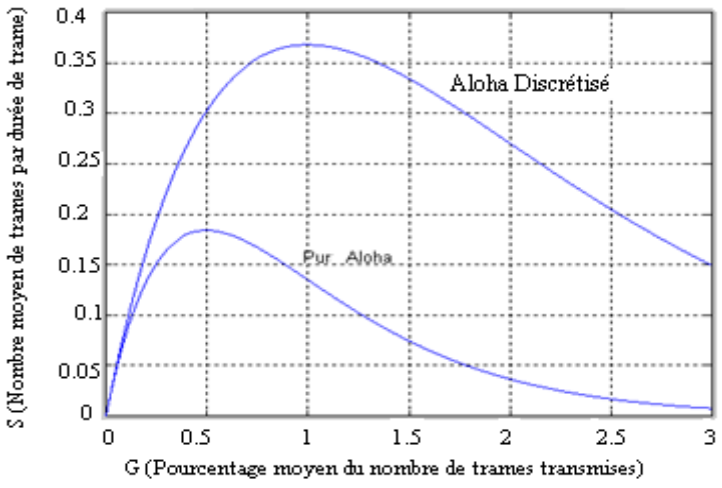

Figure (II.1) : Système Aloha, trafic offert et trafic écoulé

Cette méthode de découpage de temps en tranches, tout en gardant le système de Aloha, améliore le taux d'utilisation du canal et le ramène à 37% d'intervalles de temps utilisés, 37% de transmissions réussies et 26% de collisions (voir Figure (II.1)).

III.3.3 ALOHA avec une population finie

Les résultats précedants sont obtenus pour un nombre infini d'utilisateurs, nous étudiérons après le cas d'une population infinie.

Soit S_i la probabilité d'une transmission réussite par l'utilisateur i, et G_i la probabilité de transmission de l'utilisateur i. La probabilité pour qu'un intervalle de temps contienne une trame transmise avec succès par l'utilisateur i qui est lié au N-1 autres utilisateurs est donnée par [1] :

$$S_i = G_i \prod_{i \neq j} (1 - G_j) \qquad \text{(II.5)}$$

Dans le cas où les N utilisateurs sont identiques, chacun d'eux ayant un trafic écoulé : S_i= S/N trame par intervalle de temps et une transmission totale G_i= G/N trame par intervalle de temps, et $G = \Sigma G_i$, se qui donne:

$$S = G(1 - G/N)^{N-1} \qquad \text{(II.6)}$$

Si N tend vers l'infini, on aura:

$$S = G\, e^{-G} \qquad \text{(II.7)}$$

La condition pour que le débit soit maximal est:

$$\sum G_i = 1 \qquad \text{(II. 8)}$$

Si nous considérons deux types différents d'utilisateurs faisant des transferts de fichiers et ceux qui exécutent des applications intéractives. Soient N_1 pour les transferts de fichiers et N_2 pour l'interactif, avec les débits S_1 et S_2, l'équation (II.8) donne:

$$S_1 = G_1(1 - G_1)^{N_2 - 1}(1 - G_2)^{N_2} \qquad \text{(II.9)}$$

$$S_2 = G_2(1 - G_2)^{N_2 - 1}(1 - G_1)^{N_1} \qquad \text{(II.10)}$$

(II.9)

et

$$S_2 = G_2 (1-G_2)^{N_2-1} (1- G_1)^{N_1} \quad \text{(II.10)}$$

Pour un débit écoulé maximum, l'équation (II.8) conduit à :

$$N_1 G_1 + N_2 G_2 = 1 \quad \text{(II.11)}$$

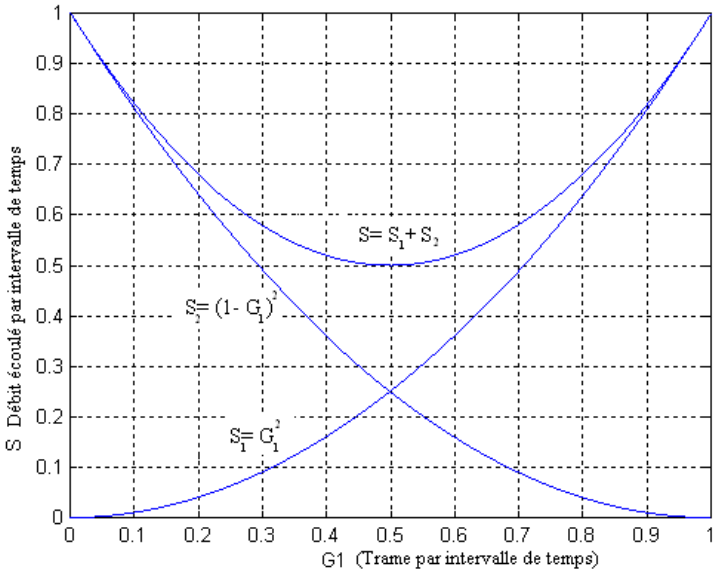

Figure (II.2) : Débit écoulé pour deux utilisateurs d'un système Aloha

- Dans le cas où : $N_1 = N_2 = 1$, $S_1=G_1^2$ et $S_2= (1-G_1)$. Les courbes des paramètres S_1, S_2 and $S= S_1+S_2$ sont présentées sur la figure (II.2).

- Lorsque G_1 est voisin de 0, l'utilisateur 1 essaie d'émettre alors que l'utilisateur 2 est pratiquement libre d'utiliser n'importe quel intervalle, de telle manière que le débit total écoulé soit voisin d'une trame par intervalle de temps. Le cas le plus défavorable apparaît lorsque les deux utilisateurs essaient d'émettre dans chaque intervalle de temps avec une probabilité de 0.5. Si cette condition se produit, il y'a 25% de chances pour que l'utilisateur 1 essaie de transmettre et que l'utilisateur 2 s'abstienne. En conclusion, une situation asymètrique entre les deux utilisateurs conduit à un débit écoulé supérieur à celui qu'on peut obtenir pour un trafic homogène.

- Dans le cas où $N_1=1$ and N_2 tend vers l'infini, l'utilisateur 1 éffectue le transfert d'un très grand fichier, tandis que les autres utilisateurs exécutent des transactions intéractives, donc on fait tendre G_2 vers zéro pour que $N_2 G_2$ soit fini. En posant $S=N_2 S_2$ en obtient:

$$S_1=(G-1)\, e^{-G}$$

et

$$S=G^2\, e^{-G} \qquad \text{(II.12)}$$

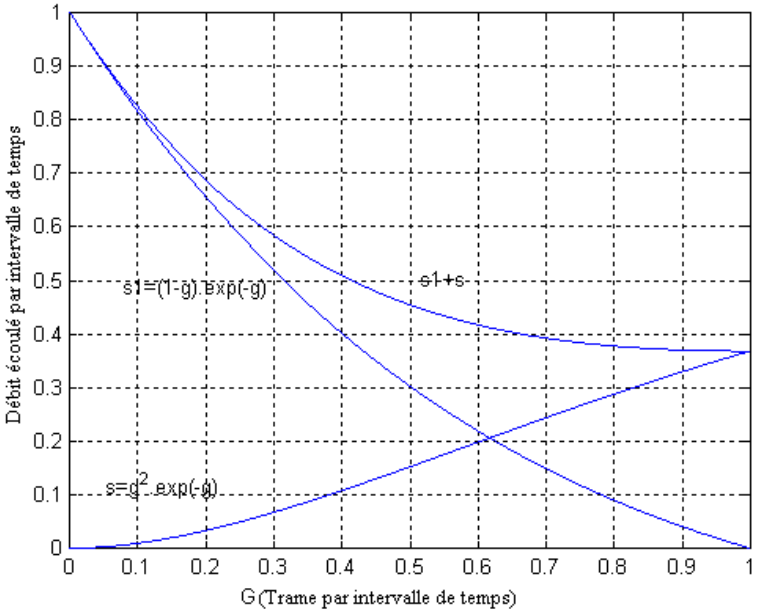

Figure (II.3) : Débit écoulé pour un gros utilisateur et plusieurs petits utilisateursd'un système Aloha

Pour avoir un débit maximal : $G_1+G=1$, la figure (II.3) représente les courbes des paramètres S_1, S ainsi que S_1+S en fonction de G. Lorsque G est petit, les utilisateurs interactifs ont une faible activité et l'utilisateur unique effectuant un transfert de fichier peut transmettre de nombreuses trames sans être victime de collisons, ce qui conduit à une très forte utilisation du canal. Lorsque G augmente, le trafic interactif demande une part plus importante du canal et l'utilisateur unique est dans l'obligation de réduire son activité pour maintenir le trafic à une trame par intervalle de temps.

II.3.4 Amélioration du protocole ALOHA discrétisé

Le problème du protocole Aloha est la faible efficacité d'utilisation d'un canal. Dans le cas du protocole Aloha Pur l'efficacité est égale à 1/2e soit 0.184, et avec le protocole Aloha synchronisé elle est de 1/e soit 0.368 [1]. Une solution simple pour mettre en œuvre un protocole

Aloha synchronisé dans un réseau satellite, consiste à utiliser un canal dans le sens terre-Satellite (uplink) et un autre dans le sens satellite-terre (downlink). Chaque canal offre un débit nominal de B bits/s pour un débit global de 2B bits/s. Le débit efficace est de B/e bits/s. L'efficacité est (B/e)/2B ou 1/e soit 0.184 , l'obligation que les trames soient rediffusées par le satellite coûte deux fois plus en efficacité comparativement à un système Aloha où les stations entendent l'émission originale de chaque trame.

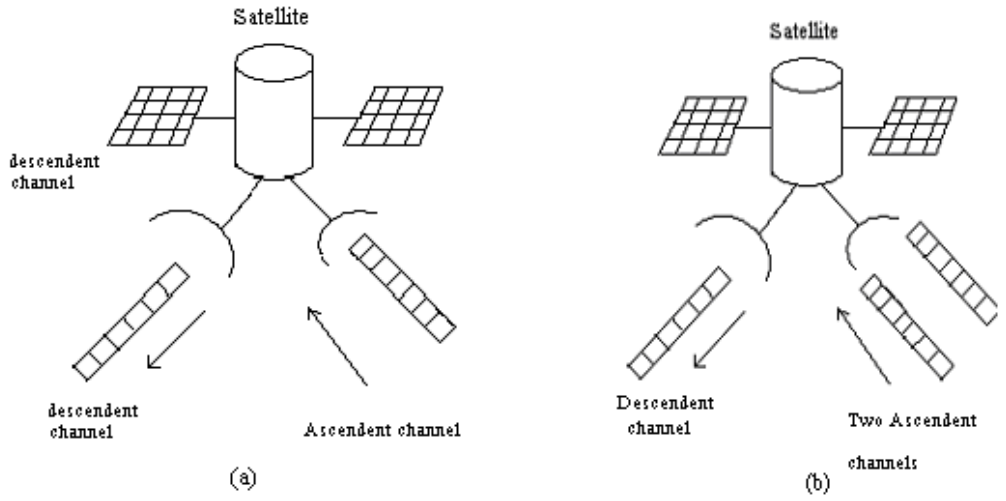

Figure (II.4) : (a) Système Aloha à deux canaux : l'un dans le sens terre-satellite, l'autre dans le sens inverse
(b) Système Aloha à trois canaux : deux dans le sens terre-satellite, un en sens inverse.

Si l'on suppose qu'un système utilisant deux canaux dans le sens terre-satellite et un canal dans le sens satellite-terre (Figure II.4). Une station désirant émettre choisit l'un des deux canaux à sa disposition, chacun d'eux étant un canal Aloha synchronisé, si l'on considère que l'utilisation de chaque canal est optimale (G=1), alors d'après l'équation (II.1), la probabilité qu'un intervalle de temps soit disponible est de 0.368, la probabilité de succès d'une transmission est de 0.368 et la probabilité d'une collision est de 0.264, avec deux canaux neuf cas peuvent se présenter.

Canal 1				
Canal		Vide	Succès	Collision
	Vide	0.135	0.135	0.097
	Succès	0.135	0.135	0.097
	Collision	0.097	0.097	0.070

Tableau (II.1)

A partir du tableau, on constate que la probabilité pour avoir un réel succès est de 0.464. En plus, dans 0.137 des cas les deux canaux transportent une trame, le satellite choisit l'une d'elle puis la retransmet, il abandonne la seconde trame. La station dont la trame a été abandonnée traite cet évènement de la même manière qu'une collision. La somme de deux probabilités montre que l'utilisation du canal de retransmission (sens satellite-terre) par le satellite est de 0.599. Le débit binaire nominal global nécessaire est de 3B et le débit efficace est de 0.599B, ce qui donne une efficacité de 0.2 soit 9% de plus que le système à deux canaux.

II.4 Région de stabilité d'Aloha discrétisé dans le cas de deux utilisateurs

II.4. 1 Systèmes de files d'attentes

Les systèmes de files d'attente modèlisent l'arrivée des clients (paquets) à une station, l'attente de ces clients de leurs tours de service, le temps de service puis le départ. Les systèmes de files d'attente sont caractérisés par cinq composantes [11]- [15].

1- La loi de probabilité de l'intervalle de temps qui sépare deux arrivées consécutives.
2- La loi de probabilité du temps de service.
3- La discipline d'attente.
4- Le nombre maximal de places réservées à l'attente.

La loi de probabilité des inter-arrivées est la loi de l'intervalle qui sépare deux arrivées consécutives.

Le temps de service varie d'un client à un autre. Pour caractériser ces variations , on a besoin de connaître les fréquences des temps de service. Il est possible d'avoir plusieurs serveurs pour une seule file d'attente, c'est l'exemple de beaucoups de banques qui n'ont qu'une seule file d'attente pour tous les clients. Lorsqu'un guichet est libre, le client en tête de file rejoint le guichet libre. Dans d'autres banques, chaque guichet a sa propre file d'attente. Dans ce cas, on a un ensemble de systèmes indépendants, composés d'une file d'attente et d'un serveur chacun, tandis que dans le premier cas, on était en présence d'un file d'attente avec plusieurs serveurs.

La discipline d'attente décrit l'ordre dans lequel les clients sont servis, par exemple la discipline FIFO (first in first out), c'est à dire le premier arrivé est le premier servi. Certaines files d'attente ont une capacité finie. Les clients qui arrivent alors que la file est pleine sont soit perdus définitivement, soit amenés à revenir plus tard [1].

On adoptera les notations de Kendall A/B/C/K/m/Z qui sont très utilisées dans la littérature des files d'attente. A désigne la loi des inter-arrivées. B la loi du temps de service, C le nombre de serveurs, K la capacité maximale de la file d'attente, m la population des usagers et Z la discipline de service. Lorsque K,m et Z ne sont pas précisés, Z est proposée FIFO et la capacité de la file d'attente ainsi que le nombre d'usagers sont supposés infinis($K=m=\infty$). Souvent A et B prennent leurs valeurs parmi :

M : loi exponentielle (M pour Markov)

D : loi déterministe

G : loi générale

Souvent la file d'attente est supposée M/M/1. Le fait que la loi des inter-arrivées soit supposée exponentielle est raisonnable pour un système avec un grand nombre de clients indépendants. Cela est dû à l'approximation d'une loi binomiale par la loi de Poisson :

$$P_n(t) = \frac{(\lambda t)}{n!} e^{-\lambda t} \qquad \text{(II.13)}$$

Où λ est le taux d'arrivée des clients

A partir de la distribution de Poisson, on déduit la densité de probabilite $a(t)\Delta t$ de l'intervalle de temps aléatoire qui sépare deux arrivées consécutives. La densité $a(t)\Delta t$ est la probabilité qu'il y ait juste un client à l'instant Δt sachat qu'à l'instant 0 il n'y avait pas de clients, c'est à dire pendant l'itervalle $[t, t+\Delta t]$ un seul client est arrivé.

$$a(t)\Delta t = P_0(t)P_1(t) \qquad \text{(II.14)}$$

Où $P_0(t)$ est $\exp(-\lambda t)$ et $P_1(\Delta t)$ est $\lambda \Delta t \exp(-\lambda t)$. En faisant tendre Δt vers 0 , on obtient :

$$a(t)dt - \lambda e^{-\lambda t} dt \qquad \text{(II.15)}$$

De la même façon, on peut montrer que la loi de l'intervalle de temps qui sépare deux départs suit une loi exponentielle de paramètre μ (la moyenne de l'intervalle de temps est $1/\mu$) si le nombre aléatoire des départs sur un intervalle de temps de longueur t suit une loi de Poisson de paramètre μt.

L'intérêt des lois exponentielles est que le système obtenu est Markovien. Un système est un processus de Markov si les états futurs du système ne dépendent du passé que par l'intermédiaire de l'état présent.

II.4.2 Etat stationnaire de la file M/M/1

Le nombre de clients dans la file d'attente M/M/1 est suffisant pour décrire l'état et l'évolution du système (Figure III.5). Dans le cas où la loi de service est quelconque, il est nécessaire de connaître, outre le nombre de clients présents dans le système, le temps déjà passé au service par le client en train d'être servi. Ceci découle du fait que la loi exponentielle n'a pas de mémoire. Les lois exponentielles sont les seules à posséder cette propriété.

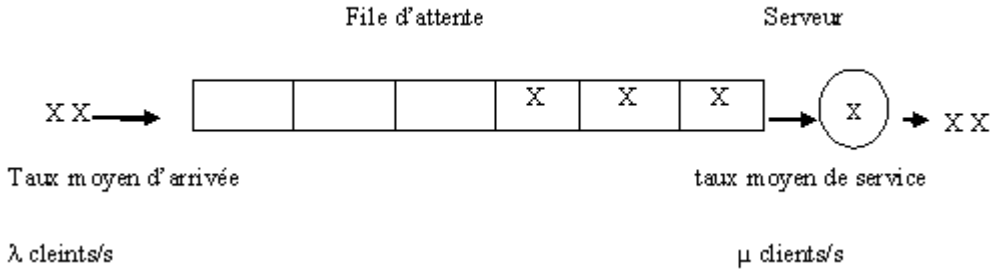

Figure (II.5) : Une file d'attente simple à un serveur et quatre clients. Trois clients Sont en attente, un en service

Soit p_k la probabilité à l'état stationnaire d'avoir exactement k clients dans le système (file d'attente+serveur). A partir des probabilités p_k, on peut déduire le nombre moyen de clients dans le système, le temps moyen d'attente et d'autres propriétés statistiques du réseau. Des transitions d'un état à un autre se produisent même lorsque le système est à l'état stationnaire. Si le réseau contient n clients, et si un nouveau client arrive, le système passe à l'état n+1. De même, lorsqu'un client reçoit le service demandé, il quitte le système qui passe alors d'un certain état k à l'état inférieur k-1. Un réseau dans lequel les transitions sont +1 ou -1 est appélé un état de naissance (+1) ou de mort (-1).

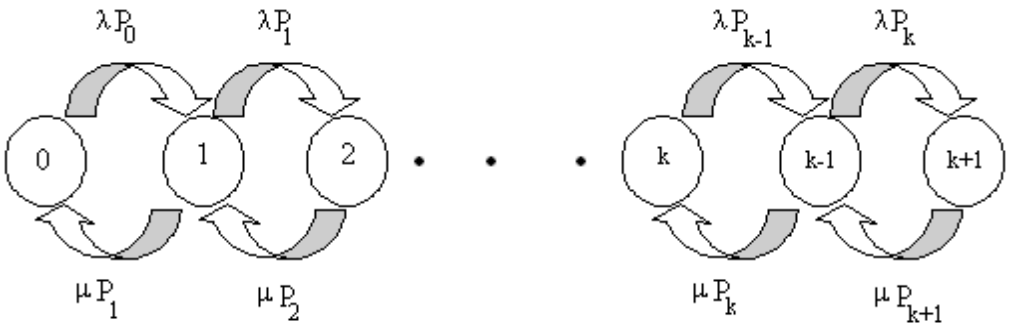

Figure (II.6) : Diagramme d'état d'un file d'attente simple à un serveur

La figure (II.6) montre les états pour une file d'attente à un serveur, avec les transitions possibles indiquées par des flèches. Pour analyser le système, on doit connaître le nombre de transitions par seconde qui se produisent d'un état donné à un état adjacent. Si le taux d'arrivée des clients est 1, le nombre moyen des transitions par seconde de l'état 0 à l'état 1 est λp_0, celui de l'état 1 à l'état 2 est λp_1 et donc celui de l'état k à l'état k+1 est λp_k.

A l'état stationnaire, la probabilité de trouver le système dans un état donné ne dépend pas du temps. En particulier, la probabilité d'avoir plus de k clients dans le système est constante. Les transitions de k à k+1 augmentent cette probabilité et les transitions de k+1 à k diminuent. Si les transitions de l'état k+1 à l'état k sont plus nombreuses que les transitions de l'état k à l'état k+1, le nombre moyen des clients dans les états inférieurs à k augmentera avec le temps, ce qui contredit l'hypothèse d'un état stationnaire. Ce principe est la clé du calcul de probabilités dans les réseaux à l'état stationnaire. La figure (II.6) illustre ces explications traduises par les équations de balance :

$$\lambda p_0 = \mu p_1$$

$$\lambda p_1 = \mu p_2 \qquad \text{(II.16)}$$

$$\lambda p_k = \mu p_{k+1}$$

Par récurrence, à partir des équations précédantes, on obtient

$$p_k = \rho^k p_0 \qquad \text{(II.17)}$$

Où ρ désigne l'intensité du trafic λ/μ. L'intensité ρ doit être inférieure à 1 pour que le nombre moyen des clients n'augmente pas sans cesse.

p_0 est déterminée en utlisant l'équation précédante et le fait que la somme des probabilités est égale à 1.

Or la somme d'une série géométrique est :

$$\sum_{k=0}^{\infty} \rho^k = \frac{1}{1-\rho} \qquad \text{(II.18)}$$

p_0 est alors égale à 1-ρ et finalement :

$$p_k = (1-\rho)\rho^k \qquad \text{(II.19)}$$

On remarque que $\rho=1-p_0$ est la probabilité que le système ne soit pas vide (que le serveur soit occupé). Le nombre moyen de clients dans le système N, peut se calculer par les probabilités p_k :

$$N = \sum_{k=0}^{\infty} k p_k = (1-\rho) \sum_{k=0}^{\infty} k \rho^k \qquad \text{(II.20)}$$

En dérivant l'équation (II.18) par rapport à ρ et en multipliant par (1-ρ) ρ le nombre moyen de clients dans le système s'écrit :

$$N = \rho / (1-\rho) \qquad \text{(II.21)}$$

Lorsque ρ tend vers 1, N tend vers $+\infty$.

On définit le temps de séjour d'un client dans le réseau comme l'intervalle de temps (aléatoire) qui sépare l'arrivée du client de son départ du réseau. Le temps de séjour moyen T d'un client est déterminé par la formule de Little :

$$N = \lambda T \qquad \text{(II.22)}$$

Ce qui permet de retrouver,

$$T = \frac{N}{\lambda} = \frac{1}{\mu - \lambda} \qquad \text{(II.23)}$$

L'analyse du temps de séjour dans les files d'attente se fonde essentiellement sur ce résultat.

Les réseaux à commutation de paquets sont modélisés par des réseaux de files d'attente M/M/1 : à l'état stationnaire, chaque nœud réagit comme une file d'attente M/M/1 isolée. Il convient d'adapter les notations à ces réseaux. Le temps de service est mesuré en nombre de secondes/paquet , alors que la longueur des paquets est de 1/μ bit.

II.5 Analyse du système à deux files

Considérons le systeme S de files, et Q_i la taille de la file i, où i=1,2 et considérons la transition $(Q_1, Q_2) \rightarrow (Q_1\text{-}1, Q_2)$, si les probabilités de transitions sont identiques même lorsque Q_2=0, le système à deux files permet d'analyser chaque file séparément. Considérons un nouveau système S^2 consistant de copies de deux files du système S avec les propriétés suivantes [16]-[22]:

- Les arrivées à la file i du nouveau système apparaissent aux mêmes instants que dans les systèmes d'origine, i=1,2 .
- Les transitions dans une file i ont les mêmes résultats que dans les deux systèmes.
- Lorsque Q_i= 0, i=1,2. Le terminal i continue à transmettre des paquets dupliqués avec la même probabilité p_i, provoquant des interférences avec l'autre système. Nous définissons les probabilité de succès dans le cas de deux utilisateurs transmettant avec des probabilités p_1 et p_2.

La dominance stochastique de S^2 sur S, qui implique que la région de stabilité de S^2 est incluse dans la région de stabilité de S [18] , [23] ,[24] ,[25].

La région de stabilité dans le système S^2 représenté par la Figure (II.7) est donnée par :

$$\lambda_i < p_i \prod_i (1 - p_j), \text{ avec i\#j} \qquad \text{(II.24)}$$

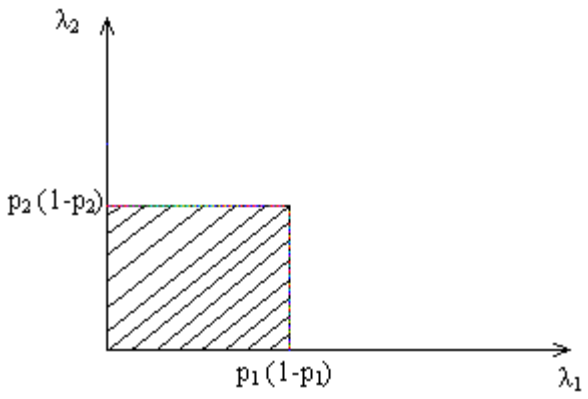

Figure (II.7) : Région de stabilité de S^2

Considérons un système auxilaire S^1 similaire à S et S^2 (mêmes prorpiètés 1 et 2), seulement il est différent lorsque les files sont vides. Dans ce système, la file 2 se comporte comme dans S^2 tandis que la file 1 continue à transmettre des paquets dupliqués lorqu'elle est vide. Ce qui fait que la probabilité de succès du système oscille entre les valeurs p_1 et $p_1(1-p_2)$ selon le cas où le canal de l'utilisateur 2 est vide ou non. Alors que la file 2 a une probabilité de succès $p_2(1-p_1)$ quelque soit l'état de la file. La stabilité dans S^1 de la file 2 (où S^1 est un système M/M/1 à temps discrétisé) est stable si et seulement si :

$$\lambda_2 < p_2(1-p_1) \qquad \text{(II.25)}$$

Supposons que la condiction (II.14) est satisfaite, et cherchons le critère de stabilité de la file 1. La probabilité de succès de la file 1 est p_1 lorsque la file 2 est vide (qui a la probabilité 1- $\lambda_2 / p_2 (1-p_1)$ selon la loi de Markov M/M/1) et la valeur $p_1(1-p_2)$ lorsque la file 2 n'est pas vide (ayant la probabilité $\lambda_2 / p_2 (1-p_1)$). La condition suffisante de stabilité de la file 1 est que le taux d'arrivée λ_1 Soit inférieur à sa probabilité moyenne de succès [19], [26], [27] :

$$\lambda_1 < p_1(1- \lambda_2 / p_2 (1-p_1)) + p_1 (1-p_2)\lambda_2 / p_2 (1-p_1) \qquad \text{(II.26)}$$

Qui donne aussi:

$$\lambda_1 < p_1(1- \lambda_2 / (1-p_1)) \qquad \text{(II.27)}$$

D'où la stabilité de S1 est donnée par la Figure (II.8).

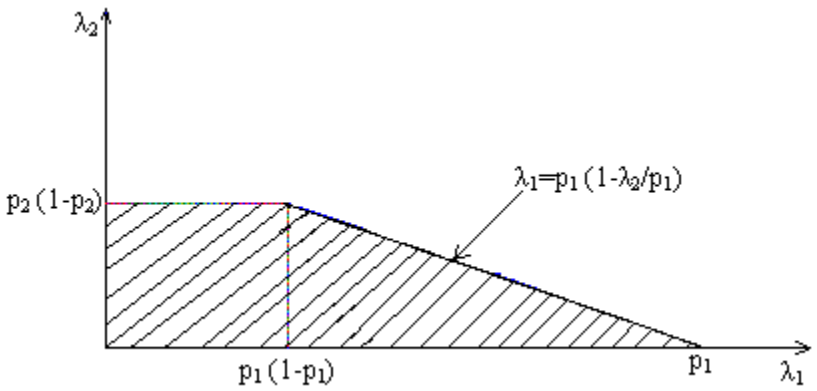

Figure (II.8) : Région de stabilité de S^1

La limite de stabilité de S^1 donnée par (II.27) n'est pas seulement une limte de la région de stabilité de S mais coincide aussi avec la région définie par : $\lambda_2 < p_2 p_1$, si pour un λ_1 la file 1 est stable dans le système S^1 , elle est ausssi stable dans le système S en vertu de la dominance stochastique de S^1 sur S, et inversement, si pour un λ_1 la file 1 est instable dans S^1 elle est aussi instable dans le système S. On supposant que la file 2 est celle qui transmet des paquets dupliqués lorsqu'elle se vide, on obtient selon [19] , la région de stabilité donnée par :

$$\lambda_2 < p_2(1-\lambda_1/(1-p_2)) \qquad \text{(II.28)}$$

L'équation (II.28) donne une partie de la région de stabilité de S. L'union des équations (II.27) et (II.28) donne la région de stabilité de S comme le montre la Figure (II.9).

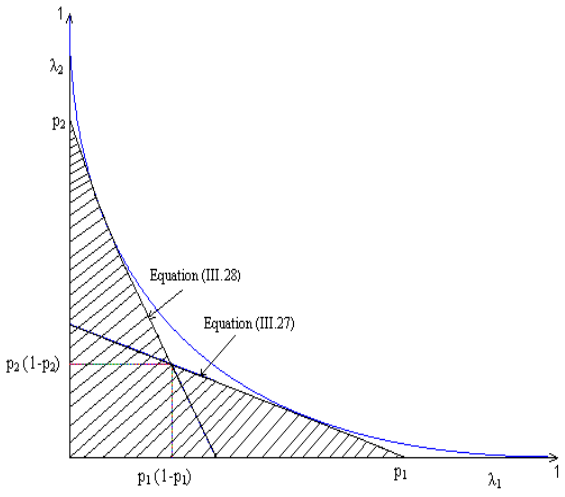

Figure (II.9) : Région de stabilité de S

En tenant compte de l'enveloppe de ces régions, sachant que p_1 and p_2 varient dans l'intervalle [0,1], nous obtenons la courbe de stabilité de S, qui est déduite par :

$$\sqrt{\lambda_1}+\sqrt{\lambda_2}=1 \qquad \text{(II.29)}$$

Où

$$\lambda_1 = p_1(1-p_2)$$

$$\lambda_2 = p_2(1-p_1)$$

$$p_1+p_2=1 \qquad \text{(II.30)}$$

Cette courbe défini la région de stabilité du système S.

II.6 Conclusion

Le protocole Aloha discrétisé consiste à découper le temps en tranches correspondant chacune au temps de transmission d'un paquet. Donc les émissions sont alors synchronisées en début de tranches. Grâce à cette méthode, s'il y a détection de collisions, c'est sur l'ensemble de la tranche de temps, et non plus sur une partie d'un paquet.

Etant donné que le protocole Aloha est un protocole à accès aléatoire, dont le débit est régi par la distribution de Poisson définie par une loi exponentielle, le système obtenu est Markovien. A cet effet, le Modèle de Markov appliqué au protocole Aloha discrétisé fera l'objet d'une analyse étalée au chapitre III.

CHAPITRE III

Analyse du modèle de Markov du protocole Aloha discrétisé en communication par satellite

III.1 Introduction

Pur Aloha et Aloha discrétisé ont été utlisés comme protocoles d'accès aléatoire en communication par satellite. Selon ses protocoles, les paquets sont transmis par plusieurs utilisateurs. Si ces paquets sont transmis simultannément par plus d'un utilisateur, il y aura une collision. Après la transmission, l'émetteur reçoit une information concernant les paquets reçus et ceux qui ont été objet d'une collision. Ces derniers vont être retransmis après un temps aléatoire.Nous considérons que le temps est divisé en unités, et à chaque unité de temps un paquet peut être transmis. A la fin de chaque unité de temps, les sources auront une information d'aucune transmission, d'une transmission ou de plusieurs transmissions (collision). Un paquet arrivant à la source est immédiatement transmis. Les paquets sujet de collisions seront refoulés et programmés pour une retransmission dans un intervalle aléatoire de temps [28]- [31].

III.2 Modèle et formulation du problème

Nous utilisons un modèle Markovien dans lequel, l'état du système change régulièrement dans le temps, le flux des paquets arrivés à la source i suit un processus de Bernoulli avec un paramètre q_a, qui signifie qu'à chaque unité de temps, il y a une probabilité q_a d'un nouveau paquet arrivant à la source, et tous les paquets arrivés sont indépendants. Un paquet refoulé à la source i sera retransmis avec une probabilité q_r^i, qui ne dépend pas du temps, et pout une restriction d'une solution symétrique optimale, nous considérons que les probabilités de retransmission sont indépendantes de i. Nous supposons que si plus d'une source tente une transmission dans un intervalle de temps, tous les paquets seront perdus, et la distribution de probabilité d'un état après une transition dépend seulement de l'état présent.

Il est important de signaler que l'état du système est représenté par une matrice stochastique de transition [31]-[35].

Nous supposons qu'il y a un nombre fini de sources sans buffers. Les distributions de probabilité binomiales de j transmissions ou arrivées parmi i paquets (terminaux ou sources) refoulés (backlog).

Posant Q_r (i,j) la probabilité de j parmi i paquets refoulés sont retransmis. Et supposant que N est le nombre de sources et posant aussi Q_a(i,j) la probabilité d'arrivée de j paquets parmi i paquets refoulés.

III.3 Analyse des transitions

Pour Aloha discrétisé il est possible d'écrire la chaîne de Markov qui caractérise l'évolution exacte.

III.3.1 Hypothèses

- Nombre fini de nœuds ou sources (N)
- Tranches de durée unitaire
- Pas de stockage de paquets: chaque nœud a un buffer de capacité 1
- Probabilité de retransmission au début de chaque trame q_r
- Arrivée de nouveaux paquets à retransmettre caractérisé par un processus de Poisson. Chaque nœud disponible va transmettre un nouveau paquet avec la probabilité $q_a = 1 - e^{-\lambda}$

Où λ est le taux d'arrivée de paquets.

III.3.2 Caractérisation de la chaîne

- L'état de la chaîne est le nombre de paquets à retransmettre (backlog).
- Les transitions entre deux états i et k correspond à des événements de type : '' le nombre de trames à retransmettre augmente (ou diminue) de $|k - i|$ ''.

Etat 0. Les transitions possibles sont :

- On reste dans le même état s'il y a une arrivée ou pas d'arrivée (on ne pas avoir de retransmission à l'état 0)
- Vers un état k > 1 s'il a k arrivées.
- On ne peut pas avoir une transition vers l'état 1 (un paquet ne peut pas entrer en collision avec lui-même).

Etat k > 0. Les transitions possibles sont :

- On reste dans le même état si
- Il y a une seule arrivée et on n'a pas de retransmission
- On n'a pas d'arrivée et on a au moins deux retransmissions
- On n'a pas d'arrivée ou de retransmission
- Vers un état i > k+1 si on a i-k arrivées
- Vers l'état k+1 si on a une arrivée et au moins une retransmission
- Vers l'état k-1 si on n'a pas d'arrivée et on a une seule retransmission.

Donc on note ;

Probabilité d'avoir i nœuds qui transmettent un nouveau paquet à l'état j [32], [33] :

$$Q_r(i,j) = \binom{i}{j} q_r^j (1-q_r)^{i-j} \tag{III.1}$$

j=0,1,…,i

Probabilité d'avoir i nœuds qui retransmettent un paquet à l'état j

$$Q_a(i,j) = \binom{N-i}{j} q_a^j (1-q_a)^{N-i-j} \tag{III.2}$$

j=0, 1… N-i

qui ont des valeurs maximales pour q_r=j/i et q_a =j/(N-i)

évidemment Qa(i,j)=0 pour i > N-j et Qr(i,j)=0 pour i > j, et $Q_r(0,1)=0$ and $Q_a(N,1)=0$

Dans le cas où tous les nœuds ont la même valeur q pour q_r, et avec ces notations, la chaîne de Markov discrétisé est caractérisée par les transitions suivantes [33] -[36]

$$T_{i,i+k}(q) = \begin{cases} Q_a(i,k), 2 \le k \le N-i \\ Q_a(i,1)\left[1-Q_r(i,0)\right], k=1 \\ Q_a(i,1)Q_r(i,0) + \\ Q_a(i,0)\left[1-Q_r(i,1)\right], k=0 \\ Q_a(i,0)Q_r(i,1), k=-1 \end{cases} \tag{III.3}$$

Le débit du système qui est le nombre moyen des paquets transmis avec succès est donné par:

$$Thp(q) = q_a \sum_{i=0}^{N} P_i(q)(N-i) = \sum_{i}^{N} P_i(q)\left[Q_a(i,0)Q_r(i,1) + Q_a(i,1)Q_r(i,0)\right]$$

$$= q_a(N - S(q)) \tag{III.4}$$

On peut numériquement résoudre les équations d'évolution et on peut calculer le nombre moyen de paquets en attente et le temps moyen d'attente. Pour ce qui de la stabilité du protocole, on est intéressé par le calcul de la dérive, c'est à dire de caractériser pour chaque état de la chaîne la tendance d'évolution (augmentation ou diminution du nombre de paquets en attente).

La dérive sera égale en moyenne à la différence entre le trafic en entrée (les arrivées) et le trafic en sortie (trafic réel moins les collisions) [34], [36] :

$$D_i = (N-i)q_a - P_s \tag{III.5}$$

Nous supposons que dans la première ligne de la matrice de transition d'état stationnaire, nous n'avons pas de paquets refoulés, et le système est vide dans ce cas.

$$T_{00} = \prod_{i=1}^{N}(1-q_{ai}) + \sum_{i=1}^{N} q_{ai} \prod_{\substack{j=1 \\ j\neq i}}^{N}(1-q_{aj})$$

Où $q_{a1} = q_{a2} = ... = q_{aN} = q_a$,

Donc

$$T_{00} = (1-q_a)^N + Nq_a(1-q_a)^{N-1} \tag{III.6}$$

Si deux ou plusieurs utilisateurs du canal, transmettent dans le même intervalle de temps, il y aura une collision. Donc le reste des valeurs de la première ligne est donné par:

$$\begin{aligned} T_{0j} &= \binom{N}{j} q_a^j (1-q_a)^{N-j} \\ j &= 2,3,...,N \end{aligned} \tag{III.7}$$

Si on a deux ou plusieurs arrivées dans un intervalle de temps (slot), nous avons;

$$\begin{aligned} T_{ii} &= Q_a(i,1)Q_r(i,0) + Q_r(i,0)Q_a(i,0) + \\ &Q_a(0,0)\left[1-(Q_r(i,0)+Q_r(i,1)\right] \\ &(N-i)q_a(1-q_a)^{N-i-1}(1-q)^i + \\ &= (1-q)^i(1-q_a)^{N-i} + \\ &(1-q_a)^N\left[1-((1-q)^i + iq(1-q)^{i-1})\right] \end{aligned} \tag{III.8}$$

Où: $0 < i \leq N-1$

Donc on obtient:

$$P_i = \begin{cases} \sum_{j=0}^{i+1} P_j T_{ij,} \; 0 \le i \le N-1 \\ \sum_{j=0}^{N} P_j T_{iN,} \; i = N \end{cases} \qquad \text{III.9)}$$

Et les autres probabilités sont données par :

$$P_{i+1} = (P_i - \sum_{j=0}^{i} P_j T_{ij}) / T_{ii+1} \qquad \text{(III.10)}$$

Où P_i (i=1,2,…N) sont les probabilités de l'état stationnaire pour les paquets refoulés dans le système.

Et le nombre moyen des paquets (nœuds) refoulé dans le système est donné par:

$$S(q) = \sum_{i=0}^{N} iP_i \qquad \text{(III.11)}$$

Le nombre moyen des paquets dans le système est S (q)+Thp(q) (somme des paquets refoulés et des nouveaux paquets arrivés).

A partir de la formule de Little, on obtient le délai moyen des paquets du canal Aloha discrétisé :

$$\begin{aligned} D(q) &= (Thp(q) + S(q)) / S(q) \\ &= 1 + S(q)/(q_a(N - S(q)) \end{aligned} \qquad \text{(III.12)}$$

III.4 Résultats et analyse

Les distributions binomiales de probabilités de retransmission et d'arrivée de paquets sont illustrées pour plusieurs valeurs de q_r, le débit total moyen ainsi que le délai moyen des paquets sont calculés pour les cas de 5 et 10 nœuds. (Voir Figures (III.1), (III.2), (III.3) et (III.4)

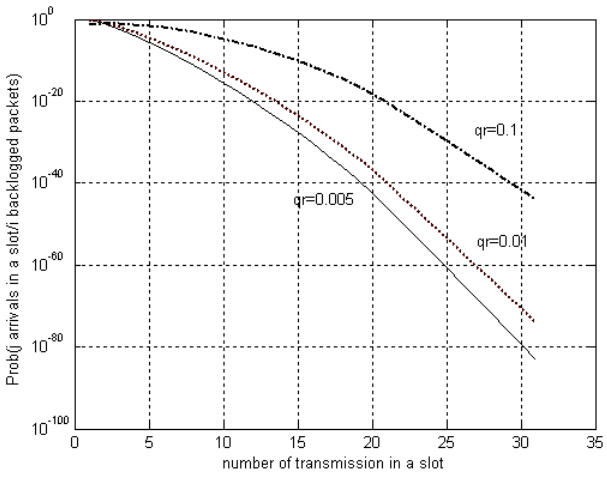

Figure (III.1) : Distribution Binomiale de probabilité de retransmission en fonction du nombre de retransmissions dans une unité de temps (slot)

Des valeurs optimales pour la distribution de retransmission et arrivées sont obtenues lorsque q_r =j/i, and qa=j/(N-i) respectivement.

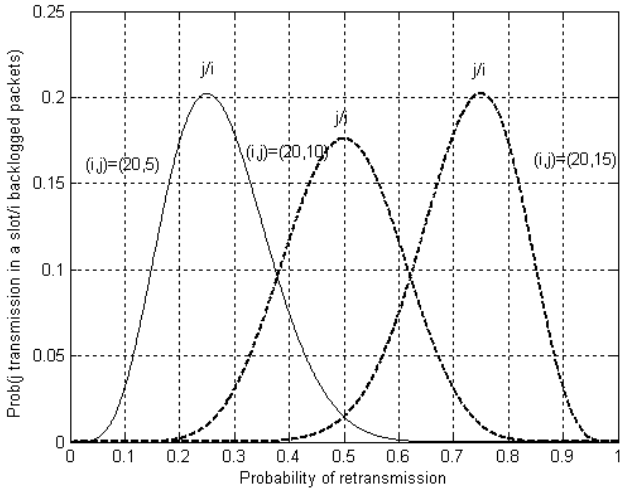

Figure (III.2) : Distribution Binomiale de probabilité de retransmission
Ayant des valeurs optimales pour qr=j/i

Le parametre de performance des protocoles aléatoires à accès multiple est le délai moyen des paquets, qui est évalué pour 5 et 10 noeuds, qui donne comme résultat; l'acroissement du nombre de noeuds, donne un délai plus important et un débit moindre comme le montre les figures (III.5), (III.6), (III.7) et (III.8).

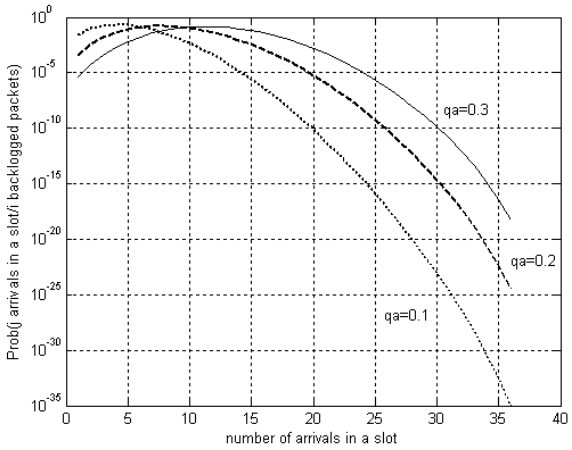

Figure (III.3) : Distribution de probabilité binomiale en fonction de nombre d'arrivées dans un slot

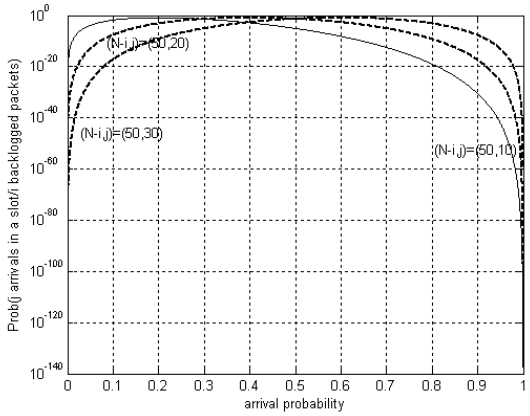

Figure (III.4) : Distribution de probabilité binomiale des arrivées ayant des valeurs max pour qa=j/(N-i)

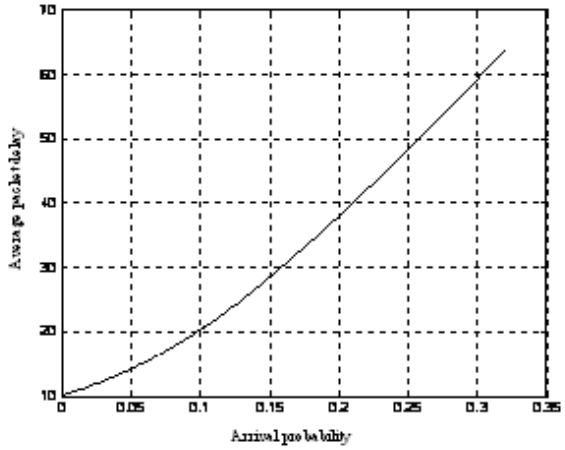

Figure (III.5): Délai Moyen de paquet en fonction de la probabilité d'arrivée, N=10 nœuds

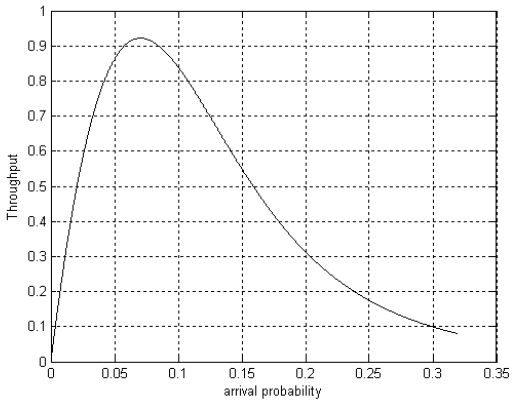

Figure (III.6) : Débit en fonction de la probabilité d'arrivée N=5 nœuds

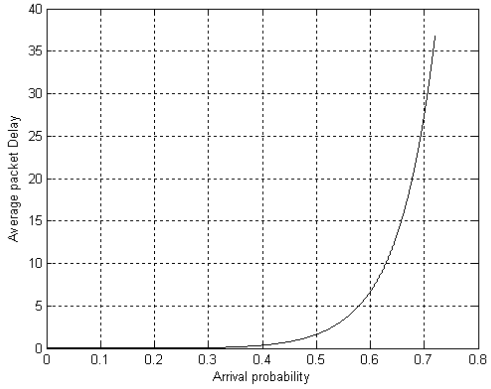

Figure (III.7) : Délai moyen de paquet en fonction de la probabilité d'arrivée q_a, N=5 nœuds

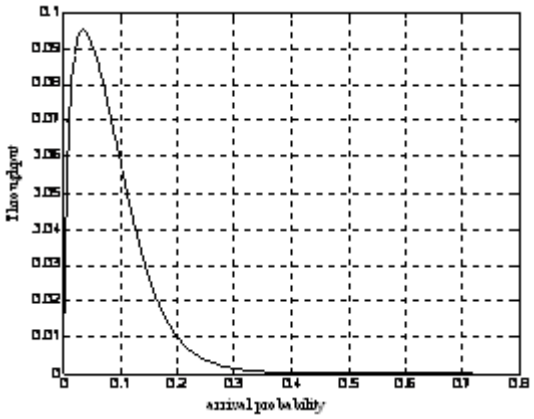

Figure (III. 8) : Débit en fonction de la probabilité d'arrivée q_a, N=10 nœuds

La probabilité de succés correspond à l'évènement "un seul paquet émis" c'est à dire une seule arrivée et pas de retransmission, soit une retransmission et pas d'arrivées. Donc :

$$P_{succès} = Q_a(1,j)Q_r(0,j) + Q_a(0,j)Q_r(1,j) \quad \text{(III.13)}$$

En utilisant les expressions de Q_a et Q_r et l'approximation $(1-x)^y \approx (1-x)^{y-1}$ pour x suffisement petit, on obtient :

$$P_{succès} \approx G(j)e^{-G(j)} \quad \text{(III.14)}$$

Où G(j) est le trafic à l'état j (le nombre de tentatives de transmission dans une unité de temps (slot) lorsque le système est à l'état j) est donné par : $G(j)= (N-j)q_a + j\, q_r$

Nous avons la probabilité d'une unité de temps vide est $e^{-G(j)}$.

- $G(j)e^{-G(j)}$ a une valeur maximale 1/e pour G(j)=1

En traçant simultanément les courbes $P_{succès}$ (G(j)) et $(N-j)q_a$ en fonction de j, on constate qu'il y a trois intersections qui correspondent au trois points d'équilibre possibles du système (D(j)=0). On remarque aussi que la région définie par $P_{succès} < G(j)e^{-G(j)}$ correspond à des dérivées négatives (j diminue) et la région définie par $P_{succès} > G(j)e^{-G(j)}$ correspond à des dérivées positives (j augmente).

Si on analyse les trois points d'équilibre on constate que le point 2 est un point d'équilibre instable (dérivée négative à gauche et positive à droite). Le point 1 correspond à un état désiré du système (peu de collisions, utilisation maximale du support de transmission. Le point 3 correspond à un comportement non désiré du système (utilisation très faible à cause du grand nombre de collisions).

Les techniques de stabilisation se proposent justement d'éviter ce point de fonctionnement non désiré. Le principe de stabilisation est de faire varier dynamiquement le paramètre q_r. Pour j grand on choisit un q_r petit afin d'éviter des collisions. Pour un j petit on choisist q_r élevé car on a des bonnes chances de transmettre avec succès des paquets en attente.

En pratique, la difficulté d'implémentation est l'estimation de j. On ne peut pas faire parvenir l'information sur l'état global du système à chaque nœud en temps réel. On utilise alors, soit des techniques d'estimation statistiques de j, soit des techniques de variation de q_r.

Soit D_n la dérive; le changement d'état de refoulement (backlog) dans une unité de temps à l'état n, exprimée par $D_n = (m - n)q_a - P_s$, et $P_s = G(n)e^{-G(n)}$ est la probabilité d'une transmission réussite, et aussi le nombre de transmissions réussites

Alors que $G(n) = (m - n)q_a + nq_r$ est le taux de tentatives de transmission et le nombre de tentatives des transmission dans une unité de temps (slot) lorsque le système est à l'état n.

La probabilité d'avoir une unité de temps vide est donnée par $e^{-G(n)}$ et lorsque $G(n)e^{-G(n)}$ ait une valeur maximale de 1/e pour $G(n) = 1$, on a un taux de départ maximal de 1/e.

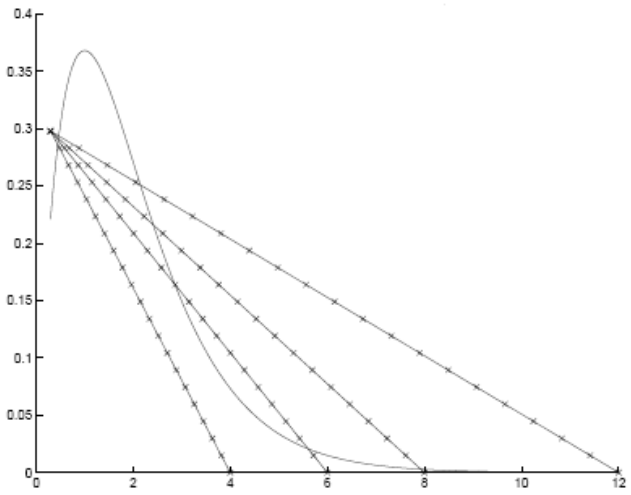

Figure (III.9) : Taux de départ et taux d'arrivée en fonction du taux de tentative pour qr=0.2,0.3,0.4,0.6

On peut aussi voir que le système possède deux points stables, et le taux de départ est presque nul pour l'autre point stable, donc si le système passe au point stable indésirable on aura un taux de départ presque nul tout le temps.

Si on augmente q_r, le délai de retransmission des paquets perdus suite collision, et la linéarité qui existe entre n et $G(n) = mq_a + n(q_r - q_a)$ change (G(n) croît rapidepement en fonction de n lorsque q_r est élevé), donc l'augmentation de q_r mène à peu de paquets refoulés et éviter le point d'équlibre instable.

Alternativement, si q_r diminue le délai augmente et il sera difficile d'éviter le point d'équlibre instable, et lorsque cette diminution est importante, il ne reste qu'un seul point sable, dans ce cas le refoulement est signifiant et un grand nombre de paquets arrivés sera refoulé et le délai de transmission sera excessivement important.

Dans le cas d'un nombre inifini de nœud le taux de rentatives de transmission devient G(n)=$\lambda + nq_r$ et la dérive devient $D_n = \lambda - P_s$ donc la droite inclinée sur la figure (IV.9) devient horizontale, par conséquent, le point stable disparaît, mais lorsque le système passe à l'équilibre instable il croît indéfiniment. Dans ce cas, il n'existe pas d'état stationnaire pour notre modèle de Markov. A cet effet, le refoulement augmentera sans limite.(voir figure (IV.9).

Du point de vue pratique, si $\lambda << 1/e$ et q_r est modéré de telle manière que le système peut rester dans l'état de stabilité pour de longues durées, dès que $P_s = G(n)\, e^{-G(n)}$ qui est optimale lorque G(n) = 1, l'approche pour atteindre la stabilité est de changer q_r pour maintenir le taux de tentatives de transmission G(n) égal à 1. Le problème qui existe est que le nombre de noeud de n est inconnu et ne peut qu'être estimé par le feedback.

Si on suppose qu'il n'y pas de stockage de paquets, le système adandonne un nombre important de paquets arrivés et aura un délai important mais fini. Par contre la supposition d'un nombre infini de nœuds, pemet d'avoir un délai infini mais pas de paquets abandonnés.

Si maintenant on utilise la supposition d'un système de nœuds infinis, on aura un système sable à accès multiple si le délai par paquet est fini. Pour Aloha discrétisé, le débit de l'état stable est 0. En estimant G(n)=1, il est possible d'avoir un système vide avec une porbabilité de 1/e=0.368 et une probabilité similaire pour avoir une transmission avec succès, et une probabilité de 1-2/e=0.264 en cas de collisons, la procédure de minimisation de q_r permettra d'avoir peu de collisions que des vides.

Lorsque tous les nœuds utiilsent la même probabilité de retransmission, un maximum débit à l'état stable est obtenu égal à 1/e. Donc le taux de succès est imité à 1/e et la dérive est positve pour $\lambda > 1/e$.

III.5 Conclusion

Nous avons utilisé le Modèle de Markov dans lequel, l'état du système change régulièrement dans le temps, le flux des paquets arrivés aux sources suit un processus de Bernoulli, cela nous a permis de définir les performances du protocole Aloha discrétisé, à savoir ; le débit et le délai moyen d'un paquet. Cependant, ces paramètres sont altérés par des erreurs et des effacements lors des transmissions, suite au bruit et aux collisions. A cet effet, le code de Reed- Solomon qui est un standard potentiel pour la correction de ces erreurs et effacement est analysé au cours du chapitre IV.

CHAPITRE IV

Correction d'erreurs et d'effacement

IV. 1 Introduction

La tendance dans les missions spatiales est d'intégrer de plus en plus des instruments de mesure à bord des satellites, ce qui implique une demande en lageur de bande plus importante. Malheureusement la voie descendante (satellite-terre) est limitée et un traitement de données s'avère nécessaire. De plus l'augmentation des données transférées attire potentiel hackers, surtout lorsque ces dernières dont d'ordre potentiellement stratégique. Un standard pour la transmission des données télémetriques existe au niveau du codage. Un codeur –décodeur de Reed-Solomon est utilisé dans ce contexte et doit pouvoir être capable de fonctionner à très haute vitesse.

Les codes de Reed-Solomon sont des codes correcteurs d'erreurs utilisés dans tous les domaines requérant des données fiables. Typiquement dans les communications spatiales.

Ces codes permettent de corriger des erreurs et des effacements grâce à des symboles de contrôle ajoutés après l'information, ils sont des codes non binaires qui travaillent dans les « champs de Galois » .

IV. 2 Applications mathématiques

Les champs de galois font partie d'une branche particulière des mathématiques qui modélisnet les fonctions du monde numérique. Ils sont très utilisés dans la cryptographie ainsi que la reconstruction des données.

Un champs de galois consiste en un ensemble de nombre, ces nombres sont constitués à l'aide de l'élément base α comme suit : 0,1, α, α^2, α^3 …α^{N-1}, en prenant : N=2^m-1 , on forme 2^m é léments. Le champs est alors noté GF(2^m).

GF(2^m) est formé à partir du champ de base GF(2) et contiendra des multiples éléments de GF(2). Sur les champs de galois on peut effectuer les opérations de base. L'addition dans le champ fini GF(2) correspond à faire une addition modulo 2, donc l'addition de tous les éléments d'un champs de Galois dérivés du champ de base sera une addition modulo 2 (XOR). La soustraction effectuera la même opération qu'une addition, c'est-à-dire la fonction logique (XOR). La multipplication et la division seront des opérations modulo « grandeur du champ, donc mod(2^m-1).

Tous les éléments non nuls du champ peuvent être construits en utilisant l'élément α comme racine du polynôme primitif. Chaque m peut avoir un polynôme primitif (tableau (IV.1)) [37]- [42].

M	P(X)	m	(X)
3	$1+X+X^3$	14	$1+X+X^6+X^{10}+X^{14}$
4	$1+X+X^4$	15	$1+X+X^{15}$
5	$1+X^2+X^5$	16	$1+X+X^3+X^{12}+X^{16}$
6	$1+X+X^6$	17	$1+X^3+X^{17}$
7	$1+X^3+X^7$	18	$1+X^7+X^{18}$
8	$1+X^2+X^3+X^4X^8$	19	$1+X+X^2+X^5+X^{19}$
9	$1+X^4+X^9$	20	$1+X^3+X^{20}$
10	$1+X^3+X^{10}$	21	$1+X^2+X^{21}$
11	$1+X^2+X^{11}$	22	$1+X+X^{22}$
12	$1+X+X^4+X^6+X^{12}$	23	$1+X^5+X^{23}$
13	$1+X+X^3+X^4+X^{13}$	24	$1+X+X^2+X^7+X^{24}$

Tableau (IV.1) : Polynômes primitifs dans $GF(2^m)^4$

A partir du polynôme du $GF(2^4)$: $p(x)=x^4+x+1$, l'élément α racine de ce polynôme donne ; $\alpha^4=\alpha+1$, ce qui permet d'avoir les éléments d'un champ de Galois de $GF(2^4)$:

Eléments	Formes polynomiales	Formes binaires	Formes décimales
0	0	0000	0
1	1	0001	1
α	A	0010	2
α2	α2	0100	4
α3	α3	0011	8
α4	1+ α	0110	3
α5	α2 + α	0110	6
α6	α3 + α2	1100	12
α7	α3+ α+1	1011	11
α8	α2 +1	0101	5
α9	α3+ α	1010	10
α10	α2 + α+1	0111	7
α11	α3 + α2 +1	1110	14
α12	α3 + α2 + α+1	1111	15
α13	α3 + α2 + 1	1101	13
α14	α3+ 1	1001	9

Tableau (IV.2) : Eléments de $GF(2^4)$

Les éléments d'un champ de Galois peuvent être aussi calculés avec matalb selon les instructions suivantes :

```
P=2 ; % Nombre de base du champ
M=4 ; %Elément
```

Champ=gftuple([-1 :p^m-2]',m,p) ;

IV.3 Principe du code Reed-Solomon

IV.3.1 Introduction

Les codes de Reed-Solomon sont des codes de détection et de correction des erreurs. Se sont des codes particuliers des codes BCH. Les messages sont divisés en blocs dont on a ajouté des informations redondantes à chaque bloc permettant ainsi de diminuer la possibilité de retransmission. La longueur des blocs dépend de la capacité du codeur [42]-[46].

Le décodeur traite chaque bloc et corrige les éventuelles erreurs. A la fin de ce traitement les données originelles seront restaurées.

Typiquement, la transmission des données dans un canal avec cette méthode est effectuée ainsi :

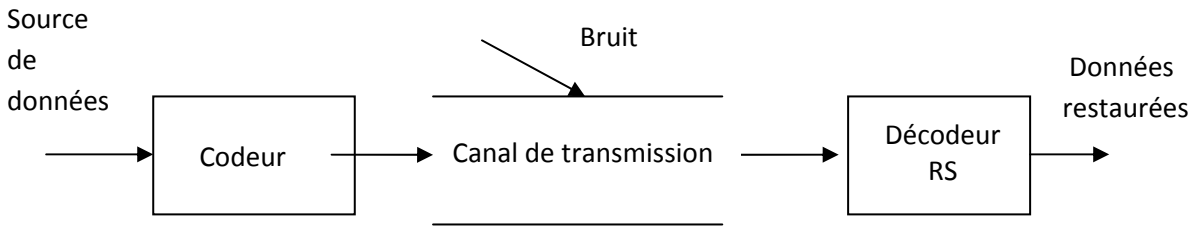

Figure (IV.1) : Schéma général

Le codeur prend k symboles de données (chaque symbole contenant s bits) et calcule les informations de construire n symboles, ce qui donne n-k symboles de contrôle. Le décodeur peut corriger au maximumu t symboles, où 2t=n-k

Le diagramme ci-dessous montre une trame constituée avec le codeur Redd-Solomon :

K données	2t

Figure (IV.2) : mot-code de Reed-Solomon

La longueur maximale d'un code de Reed-Solomon noté RS(n,k) est définie comme :

$$n=k+2t=2s-1 \qquad \text{(IV.1)}$$

Avec :

k : nombre de symboles d'information

2t : nombre de symboles de contrôle

S : nombre de bits par symbole

La distance minimale d'un code Reed-Solomon est :

$$d_{min}=2t+1 \qquad \text{(IV.2)}$$

Autre propriété des codes Reed-Solomon, ils sont cycliques, c'est-à-dire, que chaque mot-code décalé engendre un autre mot-code. Tous les codes cycliques peuvent être réduits en gardant la même capacité d'érreur, mais le nouveau code formé n'est alors pas cyclique.

De plus, les codes de Reed-Solomon sont des codes non binaires.Les codes sont représentés sur des champs de Galois de GF(2^m) et non pas sur des champs de GF(2). Les symboles définis comme les coefficients du polynôme et le degré de x indique l'ordre. Ainsi, le symbole avec l'ordre le plus élévé est reçu/envoyé en premier et le dernier symbole est celui dont l'ordre est moindre.

IV.3.2 Codage

Le codage avec les codes de Reed-Solomon est effectué de la même façon que le codage à l'aide du CRC. La seule différence est que les codes de Reed-Solomon sont non binaires.

L'équation clé définissant le codage systématique de Reed-Solomon (n,k) est :

$$c(x) = i(x)x^{n-k} + [i(x)x^{n-k}] mod(g(x)) \qquad \text{(IV.3)}$$

Avec :

c(x) : polynôme du mot-code, degré n-1

i(x) : polynôme d'information, degré k-1

$[i(x)x^{n-k}] mod(g(x))$: polynôme de contrôle, degré n-k-1

g(x) : polynôme générateur, degré n-k

Le codage systématique signifie que l'information est codée dans le degré élévé du mot-code et que les symboles de contrôle sont introduits après les mots d'information.

Les symboles de contrôle sont générés à l'aide de polynômes particuliers appelés polynômes générateurs. Tous les codes Reed-Solom sont valables si et seulement si ils sont divisibles par leur polynôme générateur, c(x) doit être divisible par g(x).

Pour la génération d'un correcteur d'érreurs de t symboles, on devrait avoir un polynôme générateur de puissance α^{2t} . La puissance maximale du polynôme est déterminée grâce à la distance minimale qui est $d_{min}=2t+1$. On devrait avoir 2t+1 termes du polynôme générateur [37].

Le polynôme générateur est sous la forme :

$$g(x) = \prod_{i=1}^{i=2t}(x \quad \alpha^i)$$

$$= g_{2t}x^{2t} + g_{2t-1}x^{2t-1} + \cdots + g_1 + g_0 \qquad \text{(IV.4)}$$

Les coefficents du polynôme générateur peuvent être calculés en prenant l'équivalence décimale à partir du tableau IV.2.

Concernant la partie hardware, le codage systématique consiste à effectuer un décalage pour placer les informations dans le degré élevé du mot-code de sortie, ce décalage est effectué seon la fonction :

$$t(x)x^{n-k} \qquad \text{(IV.5)}$$

Où x^{n-k} est un déclage du polynôme d'information de n-k positions vers la gauche. Le deuxième terme de l'équation (IV.3) est le reste de la division de $t(x)x^{n-k}/g(x)$. Cette division donnera les symboles de contrôle. Donc l'implémentation du codeur demandera deux opérations : un décalage et une division. Ces deux opérations peuvent être effectuées grâce à des registres et à des multiplexeurs.

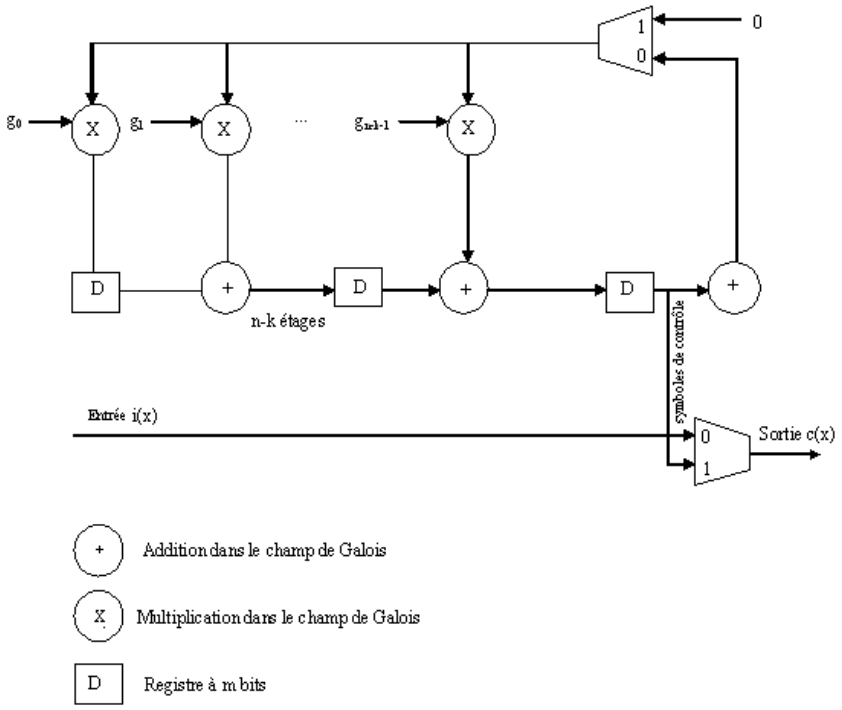

Figure (IV.3) :schéma de codage

Les additions sont définies dans le champ de Galois GF(2^m), donc pour additionner deux éléments, on prendra la notation binaire de chaque élément et les additionnera modulo 2. L'addition modulo 2 est une opération logique définie par l'opérateur logique « XOR » bit à bit. L'addition entre deux éléments d'un champ fini donnera toujours un élément dans le même champ. Les multiplications dans les codes de Reed- Solomon, sont des multiplications dans le champ de Galois GF(2^m) . Cette multiplication est une opération modulaire, ce qui fait qu'une mlutiplication entre deux éléments d'un champ donnera toujours un élément dans le même champ. Il s'agit de developper la multiplication entre deux polynpomes et de normaliser le résultat par rapport au polynôme primitif du champ choisi.

IV.4 Décodage

IV.4.1 Théorie du codage

L'idée de base du décodeur de Reed- Solomon est de détecter une séquence erronée avec peu de termes, qui sommée aux données reçues, donne lieu à un mot code valable.

Plusieurs étapes sont nécessaires pour le décodage de ces codes [37], [38], [44]:

- Calcul du syndrome
- Calcul des polynômes de localisation des erreurs et d'amplitude

- Calcul des racines et évaluation des deux polynômes
- Sommation du polynôme constitué et du polynôme reçu pour reconstituer l'information de départ sans érreur.

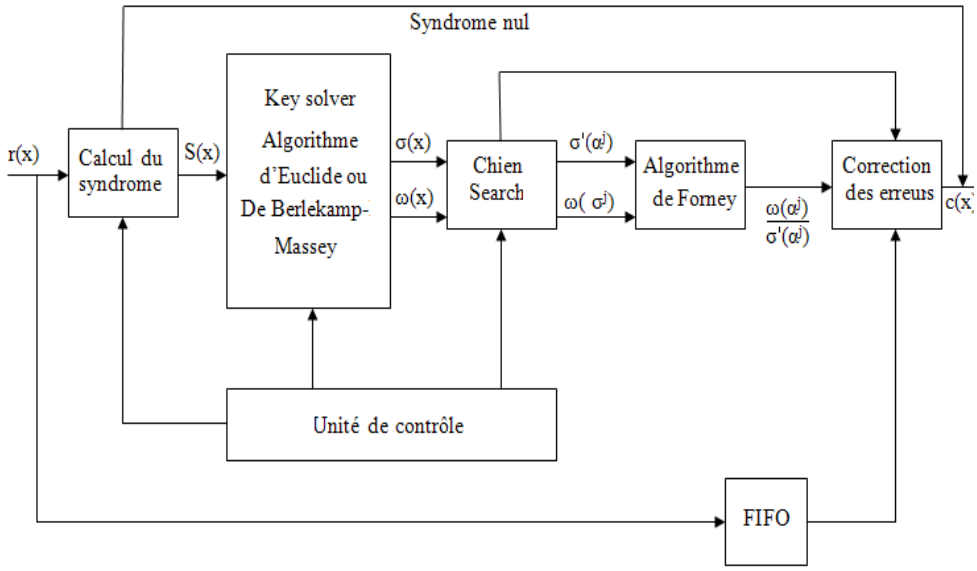

Figure (IV.4) : schéma du décodage

Où :

r(x) : mot-code reçu

S(x) : Syndrome calculé

ω(x) : polynôme d'amplitude des erreurs

$\omega(\alpha^i)$: polynôme d'amplitude des erreurs évalués pour tous les éléments compris dans $GF(2^m)$

σ(x) : polynôme de localisation des erreurs

$\sigma(\alpha^i)$: polynôme de localisation des erreurs évalué pour tous les éléments compris dans$GF(2^m)$

$\sigma'(\alpha^i)$: dérivée du polynôme de localisation des erreurs évalué pour tous les éléments ompris dans $GF(2^m)$

$\omega(\alpha^i) / \sigma'(\alpha^i)$: division entre le polynôme d'amplitude et la dérivée du polynôme de localisation des erreurs

c(x) : mot-code reconstitué

Considérons un code de Reed-Solomon c(x) correspondant au code transmis, et soit r(x) le code que l'on reçoit. Le polynôme d'erreur introduit par le canal est défini comme :

$$e(x) = r(x) - c(x) \qquad \text{(IV.6)}$$
$$= r(x) + c(x)$$
$$= \sum_{i=0}^{n} e_i x^i$$

Supposons que le polynôme des erreurs contienne v erreurs aux positions $x^{j}_{1,}$ x^{j}_{2} …, x^{j}_{v}

Avec $0 \leq j_1 < j_2 < \ldots j_v \leq n-1$. On peut donc définir le polynôme des erreurs comme :

$$e(x) = e_{j_1} x^{j_1} + e_{j_2} x^{j_2} + \cdots + e_{j_v} x^{j_v} \qquad \text{(IV.7)}$$

Où : $e_{j_1}, e_{j_2}, \ldots e_{j_v}$: Valeurs d'amplitude des erreurs

$x^{j_1}, x^{j_2}, \ldots, e_{j_v}$: Emplacement des erreurs

A partir du polynôme r(x) reçu, on peut calculer le polynôme du syndrome S(x), qui indiquera la présence d'éventuelles erreurs.Si tous les coefficients du syndrome sont nuls, alors les étapes suivantes n'ont pas lieu d'être car le mot-code reçu ne condiendra pas d'erreurs. Par contre, si le syndrome n'est pas nul, on devra calculer le polynôme de localisation des erreurs et le polynôme d'amplitude des erreurs. Le calcul de ces deux polynômes se fait par ; le décodage selon l'algorithme d'Euclide et le décodage selon l'agorithme de Berlekamp- Massey. Et puis en utilisant l'algorithme de Forney, on calculera les valeurs à soustraire pour obtenir le mot-code sans erreurs.

IV.4.2 Calcul du syndrome

Le calcul du syndrome est défini par le reste de la division entre polynùome reçu r(x) et le polynôme générateur g(x). Le reste indiquera la présence d'erreurs. Come l'opération de division est toujours complexe par rapport à des sommes et des additions, on utiilise une autre méthode pour le calcul du syndrome [37].

Le calcul du syndrome peut aussi être effectué par un processus itératif. Si tous les éléments du polynôme r(x) sont reçus, comme :

$$r(x)=c(x) + e(x) \qquad \text{(IV.8)}$$

on aura:

$$S_i = r(\alpha^i) = c(\alpha^i) + e(\alpha^i) \qquad \text{(IV.9)}$$

Par la relation (V.9) on peut définir les différentes équations qui lient le polynôme d'erreus au syndrome.

$$S_i = \sum_{i=1}^{i=2t} \sum_{k=1}^{k=v} e_{j_k} \alpha^{ij_k}$$

$$S_1 = e_{j_1}\alpha^{j_1} + e_{j_2}\alpha^{j_2} + \cdots + e_{j_v}\alpha^{j_v} \qquad \text{(IV.10)}$$

$$S_2 = e_{j_1}\alpha^{2j_1} + e_{j_2}\alpha^{2j_2} + \cdots + e_{j_v}\alpha^{2j_v}$$

....

$$S_{2t} = e_{j_1}\alpha^{2tj_1} + e_{j_2}\alpha^{2tj_2} + \cdots + e_{j_v}\alpha^{2tj_v}$$

Le syndrome sous forme polynomiale sera donc :

$$S(x) = \cdots + S_{2t+1}x^{2t} + S_{2t}x^{2t-1} + \cdots + S_2 x + S_1 \qquad \text{(IV.11)}$$

Seuls les premiers 2t symboles du syndrome sont connus. Si le code reçu r(x) n'est pas affecté par des erreurs, tous les coefficients du syndrome seront nuls (r(x)=c(x)).

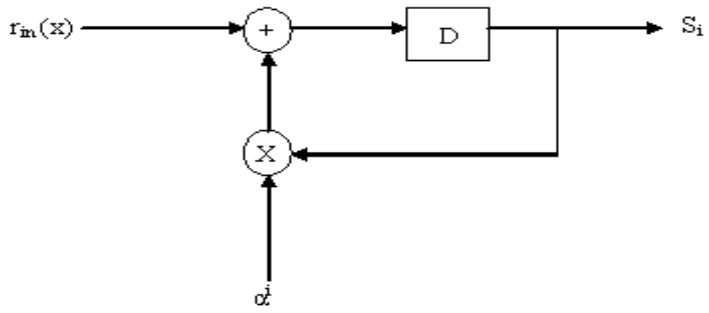

Figure (IV.5) : schéma pour le calcul du syndrome

On aura besoin de 2t schémas, comme celui de la figure (IV.5) , pour avoir le syndrome complet.

IV.4.3 Euclide

L'algorithme d'Euclide est un algorithme récursif qui permet de trouver le plus grand diviseur commun de deux polynômes $r_0(x)$ et $r_1(x)$ dans le champ de Galois GF(q) [47] -[50].

Il existe deux polynômes a(x) et b(x) en GF(q) tels que :

$$MCD(r_0(x),r_1(x))=a(x)\ r_0(x) + b(x)\ r_1(x) \qquad \text{(IV.12)}$$

Où : a(x) et b(x) peuvent être calculés selon l'algorithme d'Euclide.

En donnant deux polynômes non nuls $r_0(x)$ et $r_1(x)$ en GF(q), l'algorithme d'Euclide fonctionne de la façon suivante :

$$deg(r_1(x)) \leq deg(r_0(x))$$

$$a_0(x)=1,\ b_0(x)=0 \qquad \text{(IV.13)}$$

$$a_1(x)=0,\ b_1(x)=1$$

Avec: $\deg(r_1(x))$: degré du polynôme $r_1(x)$

$\deg(r_0x))$: degré du polynôme $r_0(x)$

Pour $i\geq 2$ on calcule le quotient $q_i(x)$ et le polynôme restan $r_i(x)$, la division est effectuée sur $r_{i-2}(x)$ et $r_{i-1}(x)$ où :

$$r_{i-2}(x)=q_i(x)\ r_{i-1}(x)+r_i(x) \qquad \text{(IV.14)}$$

avec :

$$0 \leq deg(r_i(x)) < deg(r_{i-1}(x))$$

$$a_i(x)=a_{i-2}(x)-q_i(x)\ a_{i-1}(x) \qquad \text{(IV.15)}$$

$$b_i(x)=b_{i-2}(x)-q_i(x)\ b_{i-1}(x)$$

Les calculs se terminent lorsque deg(ri)=0, le dernier polynôme non nul indique le plus grand diviseur commun.

IV.4.4 Correction d'erreurs avec Euclide

Le polynôme de localisation des erreurs est defini comme :

$$\sigma(x) = \prod_{i=k}^{v}(1 - \alpha^k x)$$

$$= \sigma_v x^v + \sigma_{v-1} + \cdots + \sigma_1 x + 1 \qquad \text{(IV.16)}$$

Le polynôme d'amplitude des erreurs se calculera da la façon suivantes :

$$\omega(x)=S(x)\ \sigma(x) \qquad \text{(IV.17)}$$

où : σ(x) : Polynôme de localisation des erreus, inconnu à ce stade
ω(x) : Polynôme d'amplitude, inconnu à ce stade
S(x) : Polynôme syndrome, connu

Comme on connaît, seulement 2t symboles du polynôme du syndrome ($x^0 \ldots x^{2t-1}$), on devrait limiter le résultat à 2t :

$$S(x)\ \sigma(x) = \omega(x)\ \mathrm{mod}(x^{2t}) \qquad \text{(IV.18)}$$

Cette expression est l'équation clé pour les codes de Reed-Solomon [37] [39] [40] . Si le nombre d'erreurs **v** dans le mot-code transmis c(x) est plus petit ou égal à t, l'équation clé à une seule paire de solution σ(x) et ω(x).

Les deux degrés des polynômes doivent respecter la contraite qui suit :

$$\deg(\omega(x))<\deg(\sigma(x)) \qquad \text{(IV.19)}$$

L'équation clé peut être résolue selon l'algorithme d'Euclide en appliquant $r_0(x)=x^{2t}$ et $r_1(x)=S(x)$. Le calcul donnera comme solution le polynôme de localisation des erreurs et le polynôme d'amplitude.

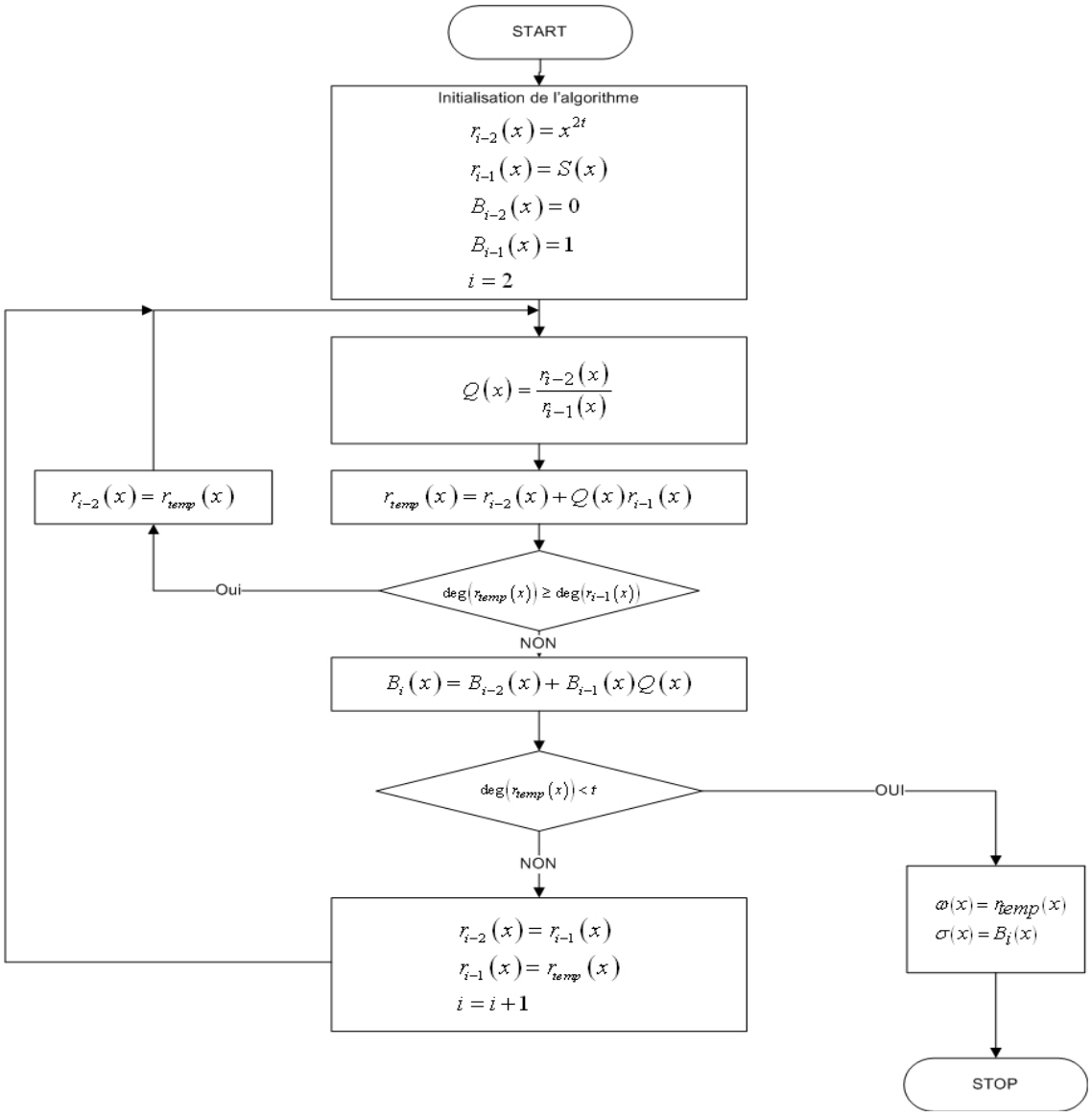

Figure (IV.6) : Algorithme d'Euclide pour la détermination des polynôles de localisation et d'amplitude

Le dernier reste de la division donnera le polynôme d'amplitude, et le polynôme de localisation des erreurs est donné par :

$$\sigma_i(x)= \sigma_{i-2}(x) + \sigma_{i-1}(x)\, Q_i(x) \qquad \text{(IV.20)}$$

avec : $\sigma_i(x) = B_i(x)$

La théorie montre que l'on est obligé d'avoir deux blocs dans l'implémentation hardware. Un bloc qui effectue la division et qui donnera le polynôme d'amplitude des erreurs, et également un bloc de multiplication qui donnera le polynôme de localisation des erreurs.

IV.5 Berlekamp- Massey

IV.5.1 Généralité de l'algorithme de Berlekamp- Massey

L'algorithme de Belekamp –Masey permet de résoudre les identités de Newton de façon itérative. Dans le cadre des codes de Reed-Solomon, cet algorithme permet de calculer le plolynôme de loclaisation des erreurs [51]- [53].

L'identité de Newton à résoudre pour les codes de Reed- Solomon est :

$$\begin{aligned} &S_{v+1} + \sigma_1 S_v + \sigma_2 S_{v-1} \dots + \sigma_v S_1 = 0 \\ &S_{v+2} + \sigma_1 S_{v+1} + \sigma_2 S_v \dots + \sigma_v S_2 = 0 \\ &\quad \dots \qquad\qquad \dots \\ &S_{2t} + \sigma_1 S_{2t-1} + \sigma_2 S_{v2t-2} \dots + \sigma_v S_{2t-v} = 0 \end{aligned} \qquad \text{(IV.21)}$$

Sachant que : $S_j = \sum_{j=1}^{v} e_{j_i} \alpha^{j_i}$

$$= \sum_{j=1}^{v} \sigma_i S_{j-i} \qquad \text{(IV.22)}$$

Cet algorithme est basé sur la synthèse du polynôme de localisation des erreurs en utlisant les registres à décalage pour la résolution des identités de Newton.

Connaissant les syndromes, on peut appliquer le diagramme ci-dessous pour le calcul de $\sigma(x)$.

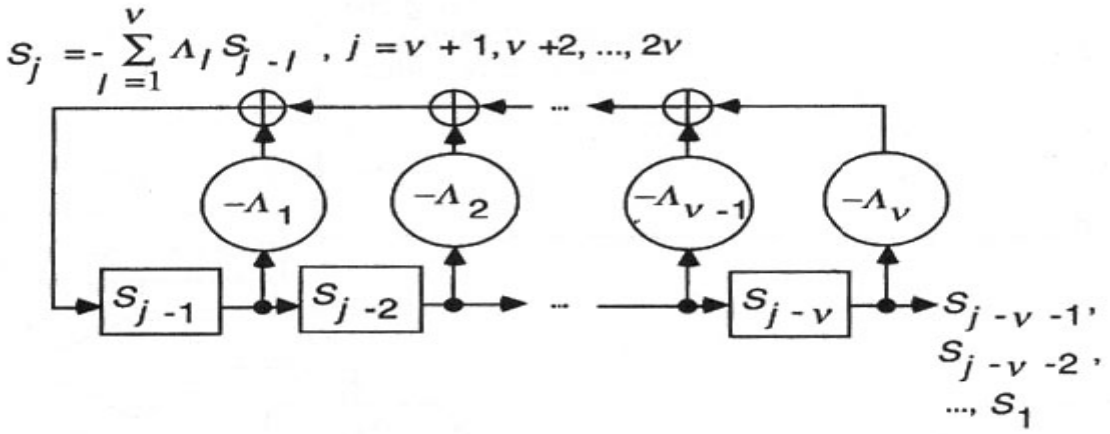

Figure (IV.7) : Registres à décalage de Berlekamp-Massey

Avec : $\Lambda^{(\mu)} = \sigma^{(\mu)}$: polynôme de localisation des erreurs après μ étapes

L_μ : degré du polynôme de localisation des erreurs après μ étapes

d_μ : incohérence après μ étapes

T(x) : polynôme de connexion annulant l'incohérence

IV.5.2 Correction d'erreurs avec Berlekamp- Massey

Le polynôme de localisation des erreurs est définit comme :

$$\sigma(x) = \prod_{i=1}^{v}(1 + \alpha^{j_i}) = 1 + \sigma_1 x + \sigma_2 x^2 + \cdots + \sigma_v x^v \qquad \text{(IV.23)}$$

Avec :

$j_1, j_2, \ldots, j_v$: indiquant l'emplacement des erreurs

$\sigma_1, \sigma_2, \ldots, \sigma_v$: coefficents du polynôme de localisation des erreurs.

A partir des relation (V.10) et (V.23), il est possible de voir la liaison entre le syndrome et le polynôme de localisation des erreurs par l'identité de Newton :

$$\begin{aligned} &S_{v+1} + \sigma_1 S_v + \sigma_2 S_{v-1} + \cdots + \sigma_v S_1 = 0 \\ &S_{v+2} + \sigma_1 S_{v+1} + \sigma_2 S_v + \cdots + \sigma_v S_2 = 0 \\ &\ldots. \\ &S_{2v} + \sigma_1 S_{2v-1} + \sigma_2 S_{2v-2} + \cdots + \sigma_v S_{2v-v} = 0 \end{aligned} \qquad \text{(V.24)}$$

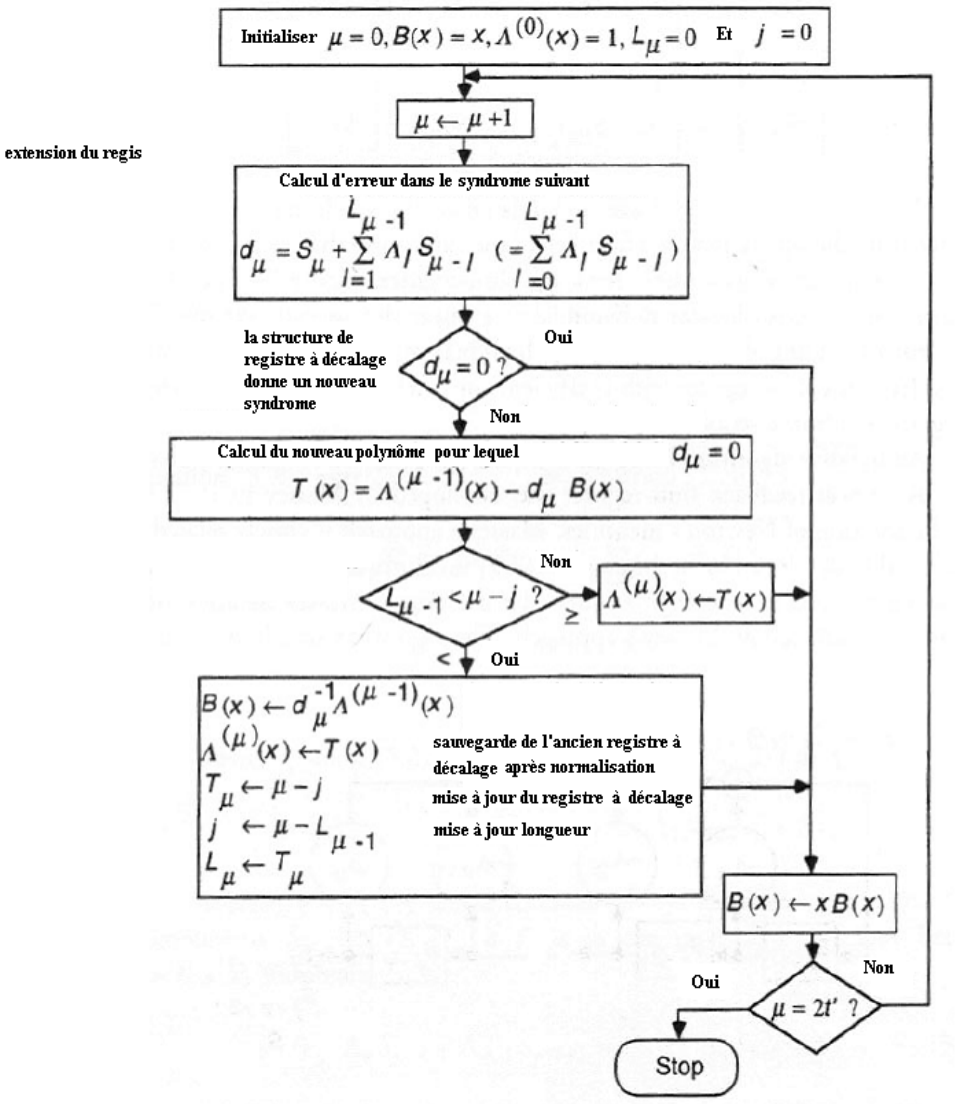

Figure (IV.8) : Algorithme de Berlekamp-Massey pour la détermination des polynômes de localisation et d'amplitude

L'objectif de l'algorithme de Berlekamp- Massey est de trouver le polynôme σ(x) de degré le plus petit possible et satisfaisant l'identité de Newton. Le calcul de σ(x) est effectué itérativement avec 2t étapes selon le diagramme .

Le polynôme calculé sera sous la forme :

$$\sigma(x) = \sigma_0 + \sigma_1 x + \cdots + \sigma_t x^t \qquad \text{(IV.25)}$$

La relation (V.11) , permet d'avoir :

$$S(x) = \cdots + S_{2t+1}x^{2t} + S_{2t}x^{2t-1} + \cdots + S_2x + S_1$$
$$= \sum_{j=1}^{\infty} S_j x^{j-1} \quad \text{(IV.26)}$$

En sachant que: $S_j = \sum_{j=1}^{v} e_{j_i} \alpha^{j_i}$, le polynôme d'amplitude sera :

$$\omega(x) = [S(x)\sigma(x)] mod(x^{2t}) \quad \text{(IV.27)}$$
$$=[(S_{2t}x^{2t-1} + \cdots + S_2x + S_1)\ (\sigma_0 + \sigma_1 x + \cdots + \sigma_i x^{2t})]mod(x^{2t})$$

IV.5.3 Chien Search

Une fois le polynôme de localisation des erreurs calculé, on doit évaluer ses racines et sa dérivée. L'évaluation des racines est effectuée avec l'algorhthme appélé « Chien Search » qui permet d'évaluer toutes les possibilités. A la sortie de ce bloc, une séquence de symboles est obtenue, et lorsque ces derniers sont nuls, sela indiquera qu'une racine a été détectée. Le schéma (IV.9) permet de calculer les racines pour un polynôme de localisation des erreurs et pour sa dérivée. Pendant n coups d'horloge, on évaluera le polynôme de localisation des erreurs et son polynôme dérivé. Chaque coup d'horloge indiquera une valeur diférente de α^i , où i est le numéro du coup d'horloge [37], [53].

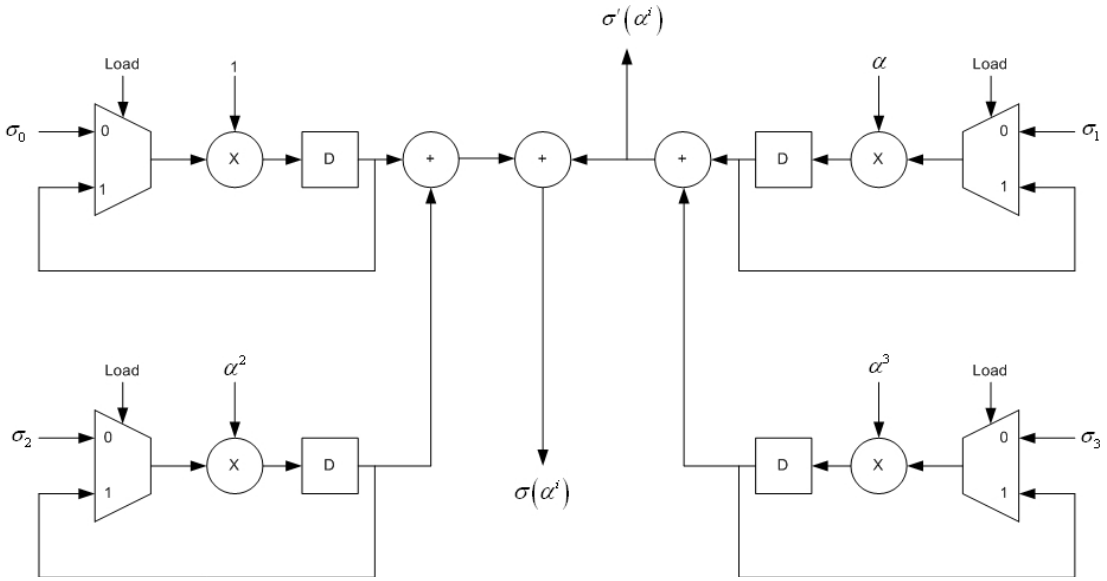

Figure (IV.9) : Orgarigramme de Chien Search

IV.5-4 Algorithme de Forney

Cet algorithme permet de construire le polynôme d'erreurs e(x) à additionner avec le polynôme reçu r(x) pour reconstituer le polynôme c(x). Pour le calcul du polynôme e(x), les polynômes $\sigma(\alpha^i)$, $\sigma'(\alpha^i)$ et $\omega(\alpha^i)$ sont nécessaires [37], Le polynôme de localisation des erreurs et sa dérivée sont déjà évalués pour les différentes valeurs de α, il reste à évaluer $\omega(\alpha^i)$.
On défini e_i comme :

$$e_i = \omega(\alpha^i) / \sigma'(\alpha^i) \qquad \text{(IV.28)}$$

Avec :

$\omega(\alpha^i)$: polynôme d'amplitude évalué pour les valeurs de $GF(2^4)$

$\sigma'(\alpha^i)$: dérivée du polynôme de localisation des erreurs pour les valeurs de $GF(2^4)$

Le schéma pour le calcul de l'algorithme de Forney et pour la correction des erreurs est :

L'évaluation du polynôme d'apmplitude $\omega(\alpha^i)$ peut être effectuée avec un schéma semblable à celui du « Chien Search ». Le schéma (IV.9) évalue le polynôme d'amplitude pour un code RS(15,9) (voir Annexes B et C).

Le calcul de l'algorithme de Forney requiert une division entre deux valeurs, qui peut être calculée en faisant une multiplication par l'élément inverse du dénominateur. On crée une ROM avec tous les éléments inverses de $GF(2^m)$ de manière à puvoir effectuer la division avec une multiplication par un symbole inverse. Le calcul de l'inverse dans un champ de Galois est défin comme suit :

$$\alpha^{-t} = \alpha^{élément_{max}-t} \qquad \text{(IV.29)}$$

La multiplication entre la valeur inverse de $\sigma'(\alpha^i)$ et $\omega(\alpha^i)$ donne la valeur de e_i

$$e_t = \frac{\omega(\alpha^i)}{\sigma'(\alpha^i)} = \omega(\alpha^i)\ \sigma'(\alpha^{-i}) \qquad \text{(IV.30)}$$

La détection du zéro est effectuée avec une porte logique « NOR » . Lorsque l'on aura un élément nul à l'entrée, donc une racine, on aura un « 1 » à la sortie. Cette détection sert uniquement à selectionner les éléments pour la correction des erreurs.

La porte logique « AND » sert à selectionner seulement les symboles qui devraient être corrigés. De cette façon on éliminera les erreurs du mot-code reçu. L'addition modulo 2 donnera toujours le symbole reçu lorsque la valeur de e_i=0. Quand la valeur de e_i # 0, on additionnera le symbole de e_i et de r_i pour éliminer l'erreur.

IV.6 Correction des erreurs et des effacements

Les codes de Reed- Solomon sont non seulement utilisés pour la correction des erreurs, mais permettent aussi de corriger les effacements. Un effacement suit le même principe que lorsqu'on efface une lettre dans un mot à l'aide d'un effaceur. La lettre effacée dans le mot n'est pas connue, mais la position de celle-ci l'est. Les codes de Reed-Solomon permettent de corriger deux fois plus d'effacements que d'erreurs [51], [53].

La séquence de décodage est presque la même que celle utilisée pour la correction des erreurs, la seule différence est qu'avant de calculer les syndromes, on doit substituer dans le polynôme reçu r(x) les effacements avec des « 0 » avant de procéder au calcul du syndrome lui-même. La première opération à effectuer pour le décodage des erreurs et des effacements, est l'évaluation du polynôme de localisation des effacements.

L'équation clé comme pour le décodage simple des erreurs peut être résolue selon les algorithmes de ;

1- Algorithme d'Euclide
2- Algorithme de Berlekamp- Massey.

IV.6.1 Capacité de Correction

La capacité de correction des erreurs d'un code de Reed- Solomon est au maximum de :

$$v \leq t \qquad \text{(IV.31)}$$

Cette règle n'est plus valable lorsqu'on doit aussi corriger des effacements, car on devrait ajouter f effacements aux v erreurs. En supposant que le polynôme des erreurs ait $\eta = v+f$ non nulles valeurs aux positions $j_1, j_2, ..j_n$, le polynôme des erreurs sera défini comme :

$$e(x) = e_{j_1}x^{j_1} + e_{j_n}x^{j_n} + \cdots e_{j_{\eta-1}}x^{j_{\eta}-1} + e_{j_\eta}x^{j_\eta} \qquad \text{(IV.32)}$$

Avec : $0 \leq j1 < j2 \ldots < j_\eta$ et $2v+ f \leq t$

IV.6.2 Résolution selon l'algorithme d'Euclide

On défini le polynôme de localisation des erreurs comme :

$$\sigma(x) = \prod_{k=1}^{v}(1 - \alpha^{i_k}x)$$
$$= \sigma_v x^v + \sigma_{v-1}x^{v-1} + \cdots + \sigma_1 x + 1 \qquad \text{(IV.33)}$$

Avec : v : nombre d'erreurs dans le polynôme reçu r(x)

Comme la position des effacements est connue, le polynôme des effacements est défini comme :

$$\beta(x) = \prod_{k=1}^{v}(1 - \alpha^{i_l}x)$$
$$= \beta_v x^f + \beta_{f-1}x^{f-1} + \cdots + \beta_1 x + 1 \qquad \text{(IV.34)}$$

Avec : f : nombre d'effacements dans le polynôme reçu r(x)

Le polynôme de localisation des effacements est défini de manière à ce que l'on puisse substituer les symboles effacés par des symboles aléatoires dans le polynôme reçu r(x). Pour raison de simplicité, on utilisera toujours le symbole nul « 0 » . Car la substitution des symboles effacés par des symboles aléatoires introduit des erreurs.

En mettant ensemble le polynôme de localisation des erreurs et le polynôme des effacements, on obtient un nouveau polynôme appélé, polynôme de localisation des erreurs et des effacements. Ce polynôme est défini comme :

$$\gamma(x) = \sigma(x)\beta(x) \qquad \text{(IV.35)}$$

Avec :

$\gamma(x)$: polynôme de localisation des erreurs et des effacements

$\sigma(x)$: polynôme de localisation des erreurs, inconnu à ce stade

$\beta(x)$: polynôme de localisation des effacements, connu à ce stade

On calcule le nouveau polynôme du syndrome avec les symboles effacés remplacés par des symboles nuls :

$$S(x) = S_{2t}x^{2t-1} + \cdots S_2x + S_1 \qquad \text{(IV.36)}$$

La nouvelle équation clé à résoudre sera :

$$\sigma(x)\beta(x)S(x) = \omega(x)mod(x^{2t}) \qquad \text{(IV.37)}$$

Avec :

$\omega(x)$: polynôme d'amplitude, inconnu à ce stade

$\sigma(x)$: polynôme de localisation des erreurs, inconnu à ce stade

$\beta(x)$: polynôme de localisation des effacements, connu à ce stade

S(x) : syndrome modifié, connu à ce stade.

Le problème cette fois est de calculer $\sigma(x)$ et $\omega(x)$ de façon à ce que $\sigma(x)$ ait le plus petit degré v et que deg($\omega(x)$) $\leq$ v+f.

Etant donné qu'on connaît $\beta(x)$ et S(x), on peut les assembler pour former le nouveau polynôme E(x) :

$$E(x) = [\beta(x)S(x)]mod(x^{2t}) \qquad \text{(IV.38)}$$

Ici, on peut appliquer l'algorithme d'Euclide pour $r_0(x)= x^{2t}$ et $r_1(x)=E(x)$. L'algorithme d'Euclide est appliquée jusqu'à :

$$\deg(r_i(x)) \leq \begin{cases} t+\frac{f}{2} & pour\ f\ pair \\ t+\frac{f-1}{2} & pour\ f\ impair \end{cases} \qquad \text{(IV.39)}$$

Avec la même technique décrite précedemment, on peut évaluer l'amplitude des erreurs et des effacements selon la relation :

$$e_{j_i} = \frac{\omega(\alpha^{j_i})}{\gamma'(\alpha^{j_i})} \qquad \text{(IV.40)}$$

IV.6.3 Résolution selon l'algorithme de Berlekamp- Massey

Définition des polynômes de localisation des erreurs, des effacements, d'amplitude et du syndrome avec les symboles effacés remplacés par des symboles nuls,n est la même qu'à la section V.2. Connaissant les 2t syndromes, le problème cette fois est de calculer le polynôme de localisation des erreurs qui satisfait la relation suivante [37] [41] [42], :

$$\omega(x) = [S(x)\gamma(x)] mod(x^{2t}) \qquad \text{(IV.41)}$$

Où : $\omega(x)$: polynôme d'amplitude

$\gamma(x)$: polynôme de localisation des erreurs, inconnu à ce stade

$S(x)$: syndrome modifié, connu à ce stade

Comme $\beta(x)$ est un facteur de $\gamma(x)$, le calcul du polynôme des erreurs et des effacements, peut se faire on initialisant l'algorithme de Berlekamp- Massey à la valeur $\mu=f=\deg(\beta(x))$.

L'évaluation de l'amplitude se fait avec la même technique citée précedemment..

IV. 7 Aloha Discrétisé avec code d'effacement

IV.7.1 Description du Modèle

Le but des codes d'effacement est de retrouver les paquets perdus lorsque leurs positions sont connues. Dans un codage d'effacement (N,K), un mot-code (N,K) consiste de N paquets codés où K paquets sont originaux et (N-K) paquets sont redondants [54].

Les K originaux paquets peuvent être recouverts avec succès lorsque K parmi N paquets codés sont reçus. Les (N-K) paquets redondants sont générés suivants certaines fonctions, comme le code de Reed-Solomon pour la correction des effacements. Un exemple pour la génération d'un paquet redondant est donné. Supposant qu'on a K paquets originaux de longueur de m bits chacun. Le paquet redondant est généré par l'équation suivante [55], [56] :

$$c_{k+1,j} = \left(\sum_{j=1}^{k} c_{j,i}\right) \bmod 2 \qquad \text{(IV.42)}$$

Où : cj,i est le i^{ème} bit du j^{ème} paquet.

Pour raison de simplicité, on suppose que tous les paquets ont la même longueur.

Le débit du Aloha discrétisé peut être amélioré en emettant des copies multiples d'un paquet, où on traite Aloha multi-copies comme un système de codage d'effacement (m,1), où chaque paquet est codé en un mot-code consistant de m copies de ce paquet. Les données du paquet original seront reçues avec succès lorsqu'au moins une copie parmi les m copies est correctement reçue.

On propose d'utiliser le code Reed-Solomon pour améliorer la performance du Aloha discrétisé, comme ceci :

- Lorsque K paquets originaux arrivent, un codage correction d'effacement par un mot-code (N,K) est effectué, un identifiant est assigné à chaque paquet codé. Le recepteur utilisera cela pour l'identification du paquet
- Les N paquets codés sont transmis aléatoirement et indépendamment durant les M unités de temps (slots) qui suivent (M est généralement très grand que N).
- Du côté recepteur, seulement les paquets reçus correctement (y compris ceux recouverts par le coadage d'effacement) sont reconnus. Les paquets non reconnus sont considérés perdus, et leur retransmission aura lieu conformément au système Aloha discrétisé.

IV.7.2 Analyse du modèle

On construit un modèle pour étudier la performance de Aloha discrétisé avec le codage d'effacement. Nous supposons :

- Le trafic combiné des nouvelles données et celles retransmises constituent un processus de Poisson d'une moyenne λ de paquets par unité de temps (slot ou durée d'un paquet).

- Le système de file peut avoir un état stationnaire.

- Un codage (N,K) est emplyoé, ce qui fait pour K paquets originaux, (N-K) paquets redondants seront ajoutés.

Donc la charge du trafic actuel est $N\lambda/K$. comme dans le cas du Aloha conventionnel discrétisé, la probabilité pour qu'un paquet codé peut être reçu avec succès sans collision est de :

$$p = e^{-\frac{N\lambda}{K}} \qquad (IV.43)$$

La probabilité P_K d'avoir K paquets codés reçus avec succès est donnée par :

$$R_K = \sum_{i=K}^{N} \binom{N}{i} p^i (1-p)^{N-i} \qquad (IV.44)$$

Posant $P_{n,m}$ la probabilité d'avoir seulement n (n <K) paquets codés reçus, et m paquets parmi n paquets sont originaux, on aura donc :

$$P_{n,m} = \binom{K}{m}\binom{N-K}{n-m} p^n (1-p)^{N-n} \qquad (IV.45)$$

On constate qu'un paquet original peut être reçu avec succès avec les conditions suivantes :

1- Au moins K paquets codés parmi N paquets du mot-code contenant le paquet original sont reçus avec succès.

2- Seulement n paquets parmi N sont reçus, où $n < K$ et m paquets parmi n paquets reçus sont des originaux ; et le paquet original en considération est l'un des m paquets (avec probabilité de m/K).

Par conséquent, la probabilité P_s d'avoir un paquet original reçu avec succès (ou corrigé par le codage d'effacement) est donnée par :

$$P_s = P_K + \sum_{m=1}^{K-1} \sum_{m=1}^{n} P_{n,m} \qquad (IV.46)$$

Et le débit du système est alors :

$$S = \lambda P_s \qquad (IV.47)$$

On constate que lorsque K=1, notre système devient Aloha à multi-copies et le débit devient $S_{multi} = \lambda P_1$ où $P_1 = P_{K(K=1)}$. Lorsque N=K=1, le système se transforme en Aloha conventionnel avec un débit $S = \lambda P_s = \lambda e^{-\lambda}$.

IV.7.3 Résultats

La variation de la probabilité d'une transmission réussie d'un paquet codé en fonction du trafic offert est illustrée par la figure (IV.10), où apparaît l'influence de l'accroissement du taux de codage , et nous notons que lorsque le trafic est faible, la probabilité d'avoir une réception réussie est élevée, ce qui implique que la probabilité de collision est faible.

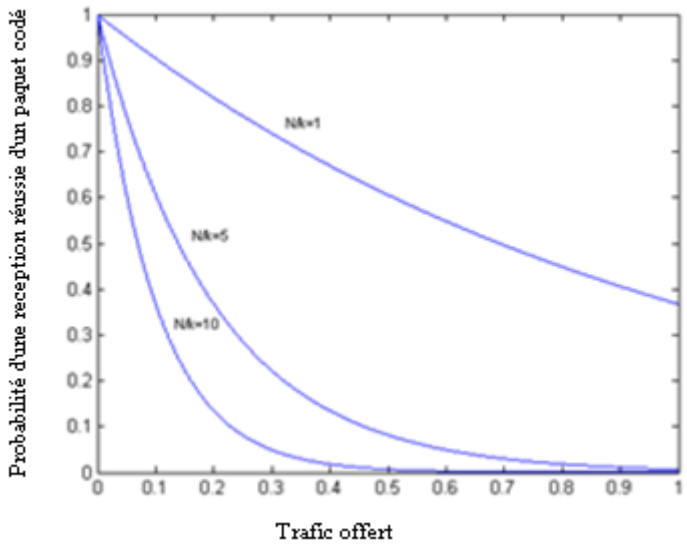

Figure (IV.10) : Probabilité d'une réception réussie d'un paquet codé pour des taux de codage K/N=0.1, K/N=0.2 et K/N=1

Alors que la figure (IV.11) présente la probabilité d'avoir K paquet reçus avec succès pour un codage d'effacement (20,5), cette probabilité est elevée lorsque le trafic offert est faible, et faible dans le cas contraire.

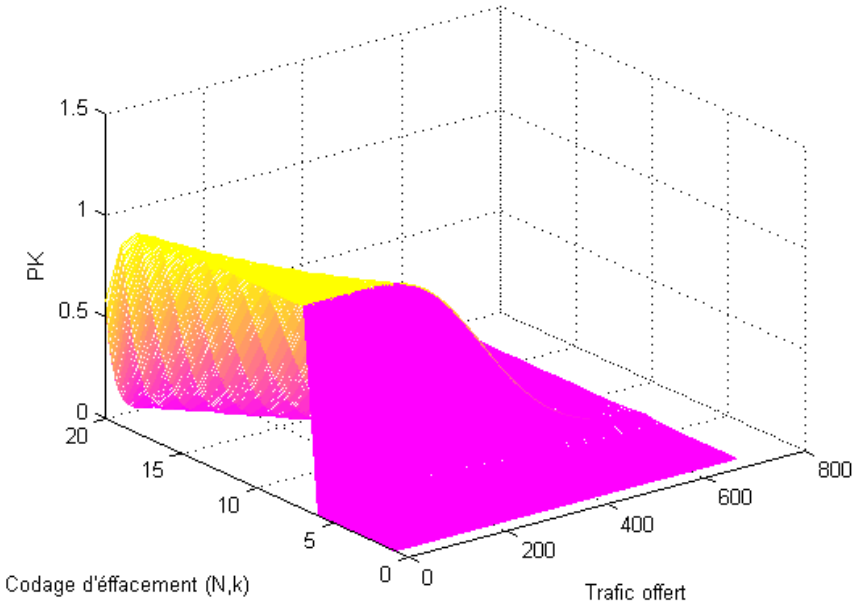

Figure (IV.11) : Probabilité de K paquets pour un codage (20,5)

La figure (IV.12) montre la performance du débit du schéma obtenu à partir de l'équation (IV.47) comparé avec Aloha conventionnel discrétisé et Aloha à multi-copies.

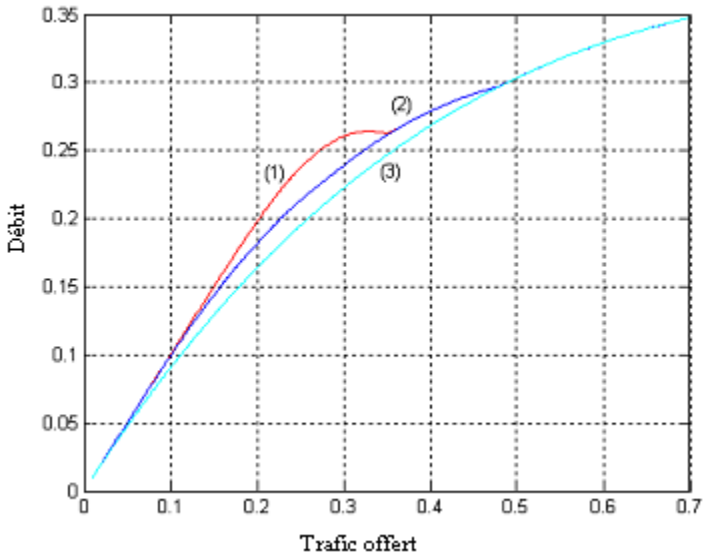

Figure (IV.12): Comparaison des performances : (1) codage d'effacement (N,K) de Aloha discrétisé, (2) Aloha multicopies et (3) Aloha discrétisé conventionnel

Trafic offert λ	Coded'effacement (N,K)	Trafic offert λ	Multi-copies(N,1)
0.01-0.07	(20,5)	0.01-0.12	(6,1)
0.08-0.16	(20,6)	0.13-0.15	(5,1)
0.17-0.26	(20,7)	0.16-0.19	(4,1)
0.27-0.35	(20,8)	0.2-0.28	(3,1)
0.36-0.48	(2,1)	0.29-0.48	(2,1)
0.49-	(1,1)	0.49-	(1,1)

Tableau (IV.3) : Paramètres utilisés pour différent trafic offert de Aloha Discrétisé

Le tableau (IV. 3) montre les parametres utilisés pour les résultats numériques. Il est à remarquer que lorsque le trafic offert est entre 0.1 et 0.35 le débit de ce schéma est plus élevé que celui de Aloha à multi-copies, et lorsque le trafic est faible, les trois schémas sont similaires., car la probabilité de collision est très faible et elle augmente avec l'accroissement du trafic offert, ce qui mène à la dégradation du débit du Aloha conventionnel. Les deux schémas se transforment en Aloha conventionnel à partir d'un trafic offert supérieur à 0.48.

Il est aussi remaqué qu'à partir de la table que lorsque le trafic offert augmente le taux de codage K/N augmente également, cela s'explique par le fait de réduire le trafic redondant permet d'éviter la dégradation de la probabilité de collision.

A cet effet, notre contribution personnelle consiste à l'utilisation de la technique d'interpolation pour définir le comportement du débit en fonction du trafic offert, celle-ci nous a permis d'avoir une meilleure approximation sous forme polynômiale exprimée par l'expression:

$$S = \sum_{i=0}^{i=k} a_i \lambda^i \qquad (IV-48)$$

Avec les parameters suivants

- Polynôme de degree k=19.
- Erreur standard e=0.0008627
- Coefficient de correlation r=0.99995912
- Les coefficients du polynôme sont donnés par le tableau (IV.4)

a_0 = -2.07194728688E-003	a_{10} = -3.52127643333E+005
a_1 = 1.29430268018E+000	a_{11} =-7.61975077188E+005
a_2 = -1.39303209334E+001	a_{12} =1.89963265811E+006
a_3 =3.02414936330E+002	a_{13}=9.69228660073E+005
a_4 =-3.46753945135E+003	a_{14} =-3.46089017210E+006
a_5 =2.24677862291E+004	a_{15} = -2.04153337107E+006
a_6 = -8.40362323725E+004	a_{16} = 1.92587585761E+005
a_7 = 1.81664417873E+005	a_{17} = 1.90305368538E+007
a_8 = -2.54210285902E+005	a_{18} =-2.58489006129E+007
a_9 = 3.61027066101E+005	a_{19}=1.02175647788E+007

Tableau (IV.4) : Paramètres utilisés pour différent trafic offert de Aloha Discrétisé

Supposons qu'on a plusieurs nœuds (utilisateurs), un satellite et un canal duplex entre les nœuds terrestres et le satellite. Les nœuds transmettent des paquets au satellite sur la liaison montante (uplink) suivant le principe du protocole Aloha Discrétisé. La longueur d'une unité de temps (slot) est de 0.004s.

Losqu'un paquet est correctement reçu durant l'unité de temps montante dite u_k, le satellite retransmettra ce paquet durant l'unité de temps descendante correspondante d_k aux nœuds terrestres. Le retard de transmission entre un nœud terrestre et le satellite est de 0.238s, ce qui fait la différence de temps entre u_k et d_k est de 0.238s. le nœud saura si la transmission du paquet est réussite durant u_k en contrôlant l'unité de temps descendante d_k.Si le paquet transmis est présent durant d_k, la transmission durant u_k est réussite.

Donc, le retard minimal pour un paquet est 0.238+0.004=0.242s. L'inter-arrivéé des paquets à chaque nœud est exponentiellement distribuée avec un moyenne de 1s. L'arrivée des paquets est un processus de Poisson avec un nouveau taux d'arrivée 0.004n, où n est le nombre des nœuds terrestres. Losqu'une retransmission est permise après un échec de transmission, cette retransmission est exponentiellement retardée avec une moyenne de 1s.

Considérons un nœud terrestre, pour voir le fonctionnement du code d'effacement (N ,K). au nœud terrestre on dispose d'un compteur de paquets originaux qui initialisé à 0. Losqu'un paquer est prêt à être transmis, une copie de ce paquet est gardée au niveau de ce

nœud, et le compteur est incrémenté à 1.Donc ce paquet original sera transmis durant l'unité de temps suivante.

Lorsque le compteur atteingnera K, (N-K) paquets redondants seront générés pour composer le mot-code d'effacement (N,K) basé sur les K paquets originaux précédants. Ces (N-K) paquets redondants seront transmis en (N-K) unités de temps choisies aléatoirement parmi les unités uplink consucétives qui suivent (supposées au nombre de 50 unités). Le delai d'un transmission réussite d'un paquet est l'intervalle entre le moment de la première transmmsion et celui où le nœud terrestre constate qu'in a été reçu avec succès [56].

La figure (IV.13) montre un exemple de codage d'effacement (3,2) de Aloha discrétisé. Le compteur est initialisé à 1.losqu'un paquet o_1 est transmis par le nœud 1 à l'unité de temps u_1, le compteur est incrémenté à 1 et la copie de ce paquet est gardée au niveau du nœud 1.

Ce nœud contrôlera l'unité de temps descendante d_1 pour vérifier la présence du paquet. Et comme ce n'est pas le cas, la transmission de ce paquet n'est pas encore fine. A l'unité de temps u_2, un autre paquet original o_2 est transmis par le nœud 1. Le compteur pour le nœud 1 passe de 1 à 2 et une copie du paquet est sauvegardée.

A ce moment là, un paquer redondant r_1 est généré pour ces deux paquets originaux. Donc le compteur du nœud 1 est remis à zero et le paquet redondant r_1 est transmis aléatoirement durant les 50 unités de temps montantes (uplink) qui suivent. Supposons que l'unité de temps u_3 est choisie. Après la transmission du deuxième paquet original sur u_2, le nœud 1 contrôle d_2 pour vérifier si la transmission est réussite

Etant donné que le paquet o_2 est présent dans d_2, la transmission de ce dernier est terminée et sa copie est effacée au niveau du nœud 1. Le retard du paquet o2 est 0.238+0.004=0.242s. Après, le paquet redondant r_1 est transmis à u_3 et en trouvant que cette transmission réussite en contrôlant d_3, le nœud 1 saura que le paquet o_1 peut être retrouvé par le satellite. Par conséquent le retard du paquer o_1 est t_4-t_1.

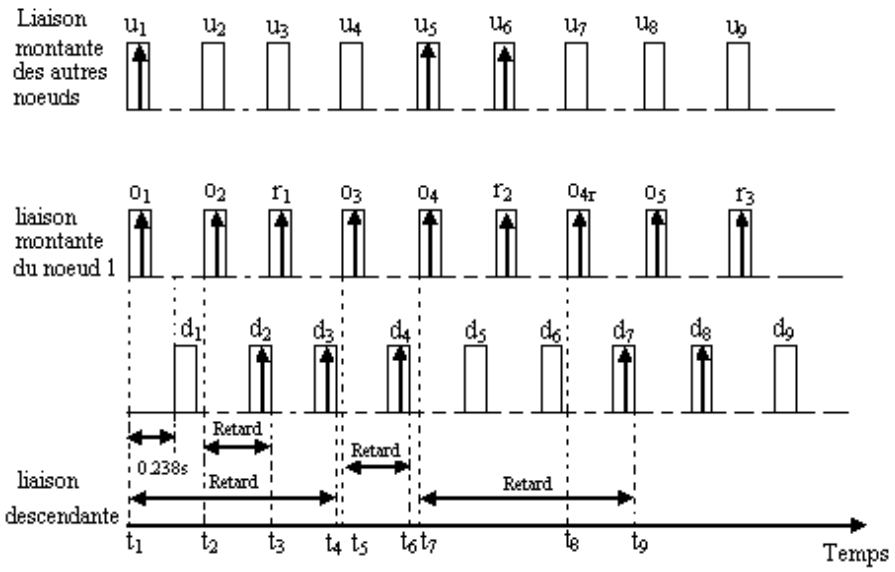

Figure (IV.13) : Aloha Discrétisé avec codage d'effacement (3,2)

Supposons que le troisième paquet original o3 est transmis pendant u_3, et le compteur est incrémenté à 1. Et comme o_3 est présent à l'unité de temps u_4, le retard de ce paquet est t_6-t_5. Après un autre paquet original o_4 est transmis pendant u_5, le second paquet redondant r_2 est généré et transmis pendant une unité de temps choisie aléatoirement.

Supposons que transmission est faite à u_6. Et suite à la collision des paquets o_4 et r_2, le nœud1 notera que le paquet o4 est à retransmettre en vérifiant d_6. Un retard aléatoire est choisi, et le paquet o_4 est retransmis pendant l'unité u_7, et comme o_4 est présent dans d_7, le retard du paquet o4 sera t_9-t_7. Il est aussi noté que le paquet o_4 va joindre un autre nouveau paquet o_5 pour générer un paquet redondant r_3 qui sera utilisé pour protéger simultannément les paquets o_4 et o_5.

IV.8 Conclusion

Le codes de Reed-Solomon est un code correcteur d'erreurs et d'effacement préconisé dans tous les domaines requérant des données fiables telle que les communications spatiales.

L'utilisation de ce code nous a permis de localiser et de corriger les erreurs à l'aide des algorithmes d'Euclide et Berlekamp-Massey, cela par l'introduction des paquers redondants pour la reconstitution des paquets de données perdus. Cette correction nous a permis d'améliorer le débit du protocole Aloha discrétisé, qui suit une loi polynômiale en fonction du trafic offert.

CONCLUSION GENERALE

Les protocoles standards de communications ont connu le passage par le modèle ISO constitué de plusieurs couches, au modèle TCT/IP qui a apporté une certaine amélioration en matière de complexité et d'encombrement. Parmi les protocoles d'accès aux canaux satellite et qui ont fait l'objet de nombreuses recherches, nous notons les protocoles aléatoires à accès multiples, comme les protocoles Pur Aloha et Aloha discrétisé, ce dernier a apporté une amélioration du pur Aloha menant à une performance du débit allant jusqu'au double.

En effet, nous avons analysé ce protocole dans le cas de deux utilisateurs en se basant sur modèle des files d'attente M/M/1, définie et discuté ses zones de stabilité.

En plus, avec l'application du modèle de Markov dans le système Aloha, où les paquets d'information sont sujet de collisions, nous avons obtenu des résultats intéressants concernant les paramètres es de performance d'un protocole à accès aléatoire tels que retard moyen de transmission de paquets et le débit en fonction du nombre d'utilisateurs qui dépasse dans ce cas le nombre de 02 utilisateurs et voir l'influence de l'accroissement du nombre de sources sur la dégradation de ces paramètres.

De nos jours, nous vivons dans un monde où la communication joue un rôle primordial tant par la place qu'elle occupe que par les enjeux économiques et technologiques dont elle fait l'objet. Nous avons sans cesse besoin d'augmenter les débits de transmission tout en gardant la qualité et ceci. Mais sans un souci de fiabilité, tous les efforts d'amélioration seraient vains car cela impliquerait forcément à ce que certaines données soient retransmises. C'est dans la course au débit et à la fiabilité que les codes correcteurs entrent en jeu.

Un code correcteur d'erreur permet de corriger une ou plusieurs erreurs dans un mot-code en ajoutant aux informations des symboles redondants, ou de contrôle. Le code de Reed-Solomon présente le meilleur compromis entre efficacité et complexité.

Dans le code de Reed-Solomon, le décodage représente la tâche la plus complexe, tant au niveau théorique qu'au niveau hardware. Les ressources hardware utilisées par le décodeur sont beaucoup plus importantes que lors du codage. En choisissant un code performant comme RS(255,223) proposé par les normes de télémétrie spatiale, le débit peut être sensiblement augmenté.

A cet effet, nous avons appliqué ce code au protocole Aloha discrétisé, pour la récupération des informations perdues suite aux erreurs de transmission ou des collisions, produites par la transm ission de plusieurs paquets d'une manière aléatoirement répartie dans le temps, par une m ultitude d'utilisa teurs et de m ettre en évidence la notable amélio ration en matière de débit de transmission.

Nous sommes arrivés ainsi, à déterm iner le comportement du débit en fonction du trafic offert, en utilisant le code correcteur de Reed-Solomon.

Nous notons que le travail prés enté et les r ésultats ob tenus, concordent avec les travaux déjà élaborés dans ce domaine et constituent un apport appréciable.

Il serait avantageux pour les réseaux de communication par sate llite, d'app liquer le modèle de Markov à d'autres protocoles plus pe rformants tel que le TCP/IP associé à un code correcteur pour la récupération des données perdues lors d'une transmission.

Annexe A

Théorie de l'information

Etant donné S L' ensemble S = {m_1 ... m_i ... m_n} des messages que l'on peut former à l'aide d'un alphabet donné constitue une source (discrète) de messages.

L'incertitude, quant à la teneur du message m_i, est donnée par l'expression :

$$I(m_i) = \log_a(1/p_i) = - \log_a p_i$$

Où : p_i est la probabilité d'émission du message m_i

a base logarithmique

On peut utiliser les logarithmes décimaux, et regarder la quantité d'information transmise par un message émis avec une certaine probabilité :

probabilité d'émission	quantité d'information
p_1=1	$I(m_1) = \log_{10}(1/p_1) = - \log_{10} 1 = 0$
p_2=0,1	$I(m_2) = \log_{10}(1/p_2) = - \log_{10} 10^{-1} = +1$
p_3=0,01	$I(m_3) = \log_{10}(1/p_3) = - \log_{10} 10^{-2} = +2$
... ...	

Cette définition probabiliste de la quantité d' information est caractérisée par une quantité d'information (ou incertitude) moyenne, d'après l'expression :

$$H(S) = \sum_{i=1}^{n} p_i \log_a p_i$$

qui permet d'évaluer a priori la quantité moyenne d'information que peut fournir un message ; Cette grandeur est appelée entropie de *S*.

Annexe B

Application de l'algorithme d'Euclide pour la localisation des erreurs

Considérons le code RS(15,9) avec un mot-code c(x)=0, un mot-code reçu $r(x)= \alpha x^{14} + \alpha^2 x^{12} + \alpha^{13} x^4$. Donc avec 6 symboles de contrôle (2t=6), le syndrome S se calcule comme suit :

$S_1=r(x=\alpha)=\alpha (\alpha^{14})+ \alpha^2 (\alpha^{12})+ \alpha^{13} (\alpha^4)= \alpha^6$

$S_2=r(x=\alpha^2)=\alpha ((\alpha^2)^{14}) + \alpha^2 ((\alpha^2)^{12})+ \alpha^{13} ((\alpha^2)^4)= \alpha^7$

$S_3=r(x=\alpha^3)=\alpha ((\alpha^3)^{14}) + \alpha^2 ((\alpha^3)^{12})+ \alpha^{13} ((\alpha^3)^4)= \alpha^{12}$

$S_4=r(x=\alpha^4)=\alpha ((\alpha^4)^{14})+ \alpha^2 ((\alpha^4)^{12})+ \alpha^{13} ((\alpha^4)^4)= 0$

$S_5=r(x=\alpha^5)=\alpha ((\alpha^5)^{14})+ \alpha^2 ((\alpha^5)^{12})+ \alpha^{13} ((\alpha^5)^4)= \alpha$

$S_6=r(x=\alpha^6)=\alpha ((\alpha^6)^{14})+ \alpha^2 ((\alpha^6)^{12})+ \alpha^{13} ((\alpha^6)^4)= \alpha^8$

Donc pour le calcul du polynôme de localisation des erreurs et le polynôme d'amplitude on utilise l'algorithme d'Euclide, on prend :

$r_0 (x) =x^{2t}=x^6$ et $r1(x) =S_6 x^5+ S_5 x^4+ S_4 x^3+ S_3 x^2+ S_2 x+ S_1$

$= \alpha^8 x^5+ \alpha x^4+ 0. x^3+ \alpha^{12} x^2+ \alpha^7 x+ \alpha^6$

La division Euclidienne de $r_0(x)$ par $r_1(x)$ donne:

$$r_0 (x)= r_1 (x)\, Q_1(x) + r_2 (x)$$

Où: $r_2(x)= = \alpha x^4+ \alpha^4 x^3+\alpha^5 x^2+ \alpha^5 x+ \alpha^6$

Et $Q_1(x)= \alpha^7 x+ 1$

Et comme deg $(r_2) \geq 3$, on continue l'algorithme, le polynôme de localisation des erreurs intermédiaire sera :

$$\sigma_2= \sigma_0(x)+ \sigma_1(x)\, Q_1(x) =0+1* (\alpha^7 x+ 1)= \alpha^7 x+ 1$$

La nouvelle opération consiste à diviser $r_1(x)$ par $r_2(x)$, ce qui permet d'avoir :

$$r_1(x) = r_2(x)\, Q_2(x) + r_3(x)$$

où: $r_3(x) = \alpha^8 x^3 + \alpha^{10} x^2 + x + \alpha$

et $Q_2(x) = \alpha^7 x + \alpha^5$

Et comme deg $(r_3) \geq 3$, on continue l'algorithme, le polynôme de localisation des erreurs intermédiaire sera :

$$\begin{aligned} \sigma_3 &= \sigma_1(x) + \sigma_2(x)\, Q_2(x) \\ &= 1 + (\alpha^7 x + 1)(\alpha^7 x + \alpha^{5)} \\ &= \alpha^{14} x^2 + \alpha^2 x^2 + \alpha^{10} \end{aligned}$$

La nouvelle opération est la division de $r_2(x)$ par $r_3(x)$, donc on obtient :

$$r_2(x) = r_3(x)\, Q_3(x) + r_4(x)$$

où: $r_4(x) = \alpha^{14} x^2 + \alpha^8 x + \alpha^{13}$

et $Q_3(x) = \alpha^8 x + \alpha^{14}$

Étant donné que comme deg $(r_4) < 3$, on arrête l'algorithme, le dernier reste de la division est le polynôme d'amplitude cherché :

$$\omega(x) = \alpha^{14} x^2 + \alpha^8 x + \alpha^{13}$$

et le polynôme de localisation des erreurs cherché est donné par :

$$\begin{aligned} \sigma_4 &= \sigma_2(x) + \sigma_3(x)\, Q_3(x) \\ &= \alpha^7 x + 1 + (\alpha^{14} x^2 + \alpha^2 x^2 + \alpha^{10})(\alpha^8 x + \alpha^{14}) \\ &= \alpha^7 x^3 + \alpha^9 x^2 + x + \alpha^7 \end{aligned}$$

Annexe C

Application de l'algorithme de Berlekamp- Massey pour la localisation des erreurs

Nous Considérons le code RS(15,9) avec un mot-code c(x)=0, un mot-code reçu r(x)= $\alpha x^{14} + \alpha^2 x^{12} + \alpha^{13} x^4$. Donc avec 6 symboles de contrôle (2t=6), le syndrome S est calculé en annexe B.

Donc pour le calcul du polynôme de localisation des erreurs et le polynôme d'amplitude selon l'algorithme de Berlekamp-Massey, on procède comme suit :

$\mu=0$

$B(x)=x$, $\sigma_0(x)=1$, $L_\mu=0$ et $j=0$

1ière Etape :

$\mu=1$

$$d_1 = S_1 + \sum_{i=1}^{L_{\mu-1}} \sigma_1 s_{\mu-1} = \alpha^6$$

T(x)= $$T(x) = \sigma_{\mu-1}(x) - d_\mu B(x) = 1 + \alpha^6 x$$

Sachant que la condition $L_{\mu-1} < \mu-j$ est vérifiée donc on aura ::

$$B(x) = d_\mu^{-1}\sigma_{\mu-1}(x) = \alpha^9$$

$\sigma_1(x)=T(x)= 1 + \alpha^6 x$

$T_1=\mu-j=1$

$j= \mu- L_0 =1$

$L_1 =T_1 =1$

B(x)=x B(x)= $\alpha^9 x$

2ème Etape :

$\mu = 1$

$$d_2 = S_2 + \sum_{i=1}^{L_{\mu-1}} \sigma_1 S_{\mu-1} = \alpha^7 + \alpha^6 \alpha^6$$

$= \alpha^2$

T(x)= $T(x) = \sigma_{\mu-1}(x) - d_\mu B(x) = 1 + \alpha \quad x$

Etant donné que la condition $L_{\mu-1} < \mu - j$ n'est vérifiée donc on aura :

$$B(x) = d_\mu^{-1} \sigma_{\mu-1}(x) = \alpha^9$$

$\sigma_2(x)=T(x)= 1 \mid \alpha \quad x$

$j= \mu - L_1 = 1$

B(x)=x B(x)= $\alpha^9 x^2$

La même procédure se poursuit pour les valeurs de μ=3,4,5 et 6 pour avoir le polynôme de localisation des erreurs :

$$\sigma(x) = x^3 + \alpha^2 x^2 + \alpha^8 x + 1$$

Avec : $d_6 = \alpha$

Et B(x)= $\alpha^{13} x^4 + \alpha^8 x^3 + \alpha^4 x^2$

Quant au polynôme d'amplitude, il est calculé à partir de l'équation :

$$\omega(x) = [S(x)\sigma(x)] mod(x^r)$$

$= (S_3 + \sigma_1 S_2 + \sigma_2 S_1)\, x2 + (S_2 + \sigma_1 S_1)\, x + S_1$

$= (\alpha^{12}+1+ \alpha^{8})\, x^2 + (\alpha^7 + \alpha^{14})\, x + \alpha^6$

$= \alpha^7\, x^2 + \alpha\, x + \alpha^6$

REFERENCES

[1] A.Tananbaum , “Réseaux Architectu re, Pr otocoles, Applicati on”, pretience-Hall International, Inc., Englewood Cliffs, 1989.

[2] O.Tharan, “Architecture Réseaux”, Institut Pasteur, IEB, pp.1-86, 2004.

[3] E.Altman, “Communication Satellite”, INRIA Sophia-Antipolis, 2004.

[4] A.Mehaoua, “Réseaux et Télécommunication”, Université Paris 5, 2006.

[5] N.Abramson, “ The Aloha System -Another alternative for communication”, Proc AFIPS Conf., Fall Joint Computer Conferen ce, N.Y, Vol. 3,pp. 281-285, 17-19November, 1970.

[6] W.Turin, “Perform ance Analysis of Digita l tran smission system s”, AT&T Beil Laboratories, 1990.

[7] L. Tong, V. Naware and P. Venkita subramaniam “ Signal Processing in Random Access”, IEEE Signal processing Magazine, pp.29-39, Sept. 2004.

[8] L. Kleinrock and Y. Yemini, «Interfering queuing processes in packet switching broadcast communication”, In Proc. IFIP Congress, Tokyo, pp.557-562, 1980.

[9] V. Na ware and L.Tong, “Stability of Qu eues in Slotted A LOHA with Multip le Antennas”, School of electrical an d Com puter Engineerin g Cornel University, Ithaca, NY 14853, pp.4-8, Oct. 2002.

[10] V.Anantharam, “The Stability Re gion of the Finite-User Slotted Aloha Protocol”, IEEE Transactions on information Theory, Vol. 37, No. 3, pp 535-540, May 1991.

[11] R.Rao and A.Ephrem ids “On the Stability of Interacting Queues in a Multip le- Access System”, IEEE Trans.On Information, Vol. 34, N°5 Sept.1988.

[12] V.Na ware and L.Tong “Using Queues Statistics for Beamfor ming in ALOHA”, Proc. Of the Asilomar Conf. On Signals, System s and Computers, Monterey, NY, USA, 9-12 Nov.2003

[13] V.Naware, G.Mergen and L.Tong, “Stabil ity and Delay of Fin ite-User Slotted Aloha with Multipacket Reception”, IEEE Transactions on Information Theory, Vol.51, No.7, pp.2636-2656, Juil.2005.

[14] Y.Sun and Y.Wang “Stability and Cap acity of Large Aloha System s with Retransmission Gain and Poisson New A rrival”, Conf. On Inform. Science and Systems, the John Hopkins University, March 12-14, 2003

[15] W.Szpankowski,” Stability conditions fo r som e multi queu e distributed system s: Buffered random access systems”, Department of Computer science, Purdue University, USA, pp.1-25, May 1992.

[16] T.Saadaoui and A.Ephrem ids,”Analysis, Stability and Optim ization of Slotted Aloha with finite nu mber of buffered users”, I EEE Trans, Autom at, Contr., Vol.AC-26, No.3, pp.680- 689, June 1981.

[17] S.Ghez, S.Verdu and S.Schwartz, “Stability properties of Slotted Aloha with multipacket reception capability”, IEEE Trans, Automat, Contr., Vol.33,No7, pp.640-649, Juil. 1988.

[18] M.Belatta r, D.Bena tia a nd M. Benslama, “Slotted Alo ha Stab ility in the cas e of two users", 1st International Symposium on Electromagnetism and Cryptography ISECC’05, Jijel, Algeria, pp.49-54, 2005.

[19] M.Bel attar, D.Benatia and M.Benslam a, “Analysis o f Slotted A loha Stab ility Using Spread-Spectrum in Satellite Communication” , Canadian Journal On Electrical and Electronics Engineering, Vol.3, No.2, pp.80-v4, February 2012.

[20] A.Annamalai and V.K.Bhargava, “Throughput performance of Slotted DS/CDMA Aloha with packet com bining over generalized fading channels”, Electronics letters, Vol. 33, No.14, pp.1195-1197, Jul.1997.

[21] H.Saraghi, “Analysis Throughput m ulti-code Multicarri er CDMA S-Aloha”, Interworking Indonesia Journal, Vol.2, No.1, pp.11-15, 2010.

[22] H.Okada, T.Yam azato, M.Katayama and A. Ogawa, “CDMA Slotted Aloha System with finite Buffe rs”, IEICE Trans, Fundam entals, Vol.E81-A, No.7, pp.1473-1478, July 1998.

[23] L.Tong, V.Nawa re and P.Venkitasubram aniam “Stability of Queues in Slotted Aloha with m ultiple antenn as", School of electrical and Com puter Engineering Cornel University, Ithaca, NY 14853, pp.4-8, October 2002.

[24] M.Médard, J.Huang, A.J.Goldsm ith, S.P.Meyn and T.P.Coleman “Capacity of Time – Slotted Alo ha Packetized Multip le-Access Sys tems over the AWGN Channel”, IEEE Trans.On Wireless Communications, Vol.3, No.3, pp.486-491, Mars 2004.

[25] M.S.Do, J.S.Park, H.Jeon and J.Y.Lee “The effect of Spreading gain control on a CDMA Slotted Aloha System ", Elsevier, Co mputer Communications, Vol.26, pp.996-1006, July 2002.

[26] I.E.Telatar and R.G.Gallager, “Com bining queuing theory with information theory for multiple-Access,” IEEE J.Select.Areas Commun., Vol.13, pp.963-969, August 1995.

[27] W.A.Rosenkrantz and D.Towsley, “On the instability of the Slotted Aloha m ulti-access algorithm", IEEE Trans.Automat.Contr., Vol.AC-28, pp.994-996, 1983.

[28] M.Akaplan “A sufficient c ondition for non ergodic ity of Markov chain”, IEE E Trans. Inform, Vol. IT-25, No.4, pp.470-471, July 1979.

[29] B.S.Ts ybagov and V.A.Mikhailov, “Ergodi city of slotted ALOHA system ”, Probl . Peredachi Inf., Vol.15, No.4, pp.73- 87, March 1979.

[30] P.W.De Graaf and J.S.Lehnert, " Performance comparison of a slotted Aloha DS/SSMA network and a multichannel narrow-band slotted Aloha network", IEEE Transactions on Communications, Vol.46, No.4, pp.544-552, 1998.

[31] E.Altman, R.EL Azouzi and T.jiménez, "Slotted Aloha as a game with partial information", Elsevier, Computer Networks, Vol.45, pp.701-713, 2004.

[32] M.R.Ayala, E.Alejandro, A.Gonzalez, J.Alfredo, T.Mendez and H.J.Aguilar, "Markovian Chain Analysis for Satellite Applications", International Journal of Communications, Vol.1, pp.170-173, 2007.

[33] M.Belattar, D.Benatia and M.Benslama, "Analysis of Markov Model of Slotted Aloha protocol in Satellite Communication", International Review of Aerospace Engineering (IREASE), Vol.3, No.3, pp.134-137, June 2010.

[34] E.W.M.Wong, T.S.P.Yum, "The optimal multi-copy Aloha", IEEE Trans. On automatic control, Vol.39, No.6, pp.1233-1236, June 1994

[35] E. Sabir, M.R.El Fennich and M.El Kamili, "Slotted Aloha à la première transmission différée: une nouvelle solution pour supporter les applications sensibles au délai", Majecstic, Avignon, France, du 16 au 18 November 2009.

[36] M.R.Ayala, E.Alejandro, A.Gonzalez, J.Alfredo, T.Mendez and H.J.Aguilar, "Average packet delay in Random Multiple Access for satellite systems", Proceedings of the 7th WSEAS International Conference on telecommunications and informatics, TELE-INFO'08, Istanbul, Turkey, pp.120-124, 27-30 Mai 2008.

[37] S.DIETER, «Implémentation de codes de Reed-Solomon sur FPGA pour communications Spatiales", HEIG.VD, 2006

[38] S.B.Wicker and V.K.Bhargava, "Reed-Solomon codes and their applications", Wiley-IEEE Press, New York, Sept, 1999.

[39] T.Zhang and K.K.Parhi, “On the high Speed im plementation of Errors and Erasures correcting Reed-Solomon decoders”, Proc.12th Great lakes Symposium on VLSI, PP.89-93, NY, USA, 2002.

[40] B.Sklar, “A structured Overview of di gital Communication- A Tutorial Review- part I", IEEE Communications magazine, Vol.21, No5, pp.5-17, August 1983.

[41] O.Sidek and A.Yahia, “Reed-Solomon Coding for Frequency Hopping Spread Spectrum in Jamm ing Environm ent", Am erican Journal of Applied Sciences, Vol.5, pp.1281-1284, 2008.

[42] S.Pasc richa and S .Sharma, “FPGA based design of Reed- Solo mon codes", Indian Journal of Science and Technology, Vol.5, No4, pp.48-52, Mars 2009.

[43] P.Shankar, “Error correcting codes: The Reed-Solomon Codes”, Resonance, Vol.2, No.3, pp.33-47, 1997.

[44] A.Bonnecaze, “Introduction à l’Algèbre pour les codes cycliques”, pp.1-20, 2007.

[45] L.Rizzo, “Effective Erasure Codes for Reliable Computer communication Protocols”, In Computer Communication Review, Vol.32, pp.187.207, April 1997.

[46] S.Ki m, “Efficient Erasure Co de for W ireless Sens or Networks”, Com puter Sc ience Division, University of California at Berkeley, 2003.

[47] P.Shankar, “Decoding Reed-S olomon Code s Using Euclid’s Algorithm ”, Resonance, Vol.7, No.2, pp.37-51, April 2007.

[48] D.V.Sarwate et Z.Yan,” M odified Euclidian Algorithm for D ecoding Reed-Solomon Codes”, Departm ent of Electrical and Co mputer Engineering and the Coordinated Science Laboratory, U niversity of Illinois at Urbana-Champaign, Urbana, USA, pp.1398- 1402 June 2007.

[49] J.H.Ba ek, J.Y.Kang and M.H.Sunwoo, “Design o f high –Sp eed Reed- Solom on decoder", International Sy mposium on Circuits and System s, ISCAS 2002, Arizona, USA, Vol.5, pp.793-796, 26-29 May 2002.

[50] T.K.Truong, W.L.Eastman, I.S.Reed and I.S.Hsu, “Simplified Procedure for correcting Errors and Erasures of Reed-Solomon code using Euclidean algorithm", in proc, IEEE, part E, Vol.135, pp.318-324, November 1988.

[51] A.E.He ydtmann and J.M.Jensen, “On the equivalence between Berlekam p-Massey and the Euclidean algorithm s for decoding", IEEE Trans actions on Information Processing (ISMIP), Vol.46, pp. 2614 – 2624, Hsinchu, Taiwan, December 2002.

[52] N.Ben Atti, G.M.Diaz and H.Lombardi, “The Berlekamp-Massey Algorithme revisited”, CNRS 6623, UFR des Sciences et Techniques, Université de Franche-Comté, 2005.

[53] J.L.Dornstetter, “On the equivalence between Berlekam p and Euclid’s algorithm s”, IEEE Transactions on Information Theory, Vol.IT-33, pp.428-431, May 1987.

[54] K.S.Chan, L.K.Yeung and W.Shao “Cont ention-based MAC prot ocols with erasure coding for wireless data networks,” Ad Hoc Networks, Vol.3, pp.495-506, Jul 2005.

[55] K.S.Chan, L.K.Ye ung and W .Shao, “Cont ention-Based MAC protocols with erasure coding for wireless Data networks “, Ad Hok Networks, Elsevier, Vol.3, pp.495-506, 2005.

[56] M.Belatta r, D.Benatia and M. Benslama, “Slotte d Aloha protocol with Reed-Solomon Erasure Coding", International Review of Aerospace Engineering (IREASE), Vol.4, No.3, pp.168-172, June 2011.