



**HAL**  
open science

# LIGHTNET : Un protocole d'accès à haut débit pour bus unidirectionnel

Philippe Jacquet, Paul Mühlethaler

► **To cite this version:**

Philippe Jacquet, Paul Mühlethaler. LIGHTNET : Un protocole d'accès à haut débit pour bus unidirectionnel. RR-0654, INRIA. 1987. inria-00075899

**HAL Id: inria-00075899**

**<https://hal.inria.fr/inria-00075899>**

Submitted on 24 May 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# INRIA

UNITÉ DE RECHERCHE  
INRIA-ROQUENCOURT

Institut National  
de Recherche  
en Informatique  
et en Automatique

Domaine de Voluceau  
Roquencourt  
BP 105  
78153 Le Chesnay Cedex  
France

Tél (1) 39 63 55 11

## Rapports de Recherche

N° 654

### **LIGHTNET : UN PROTOCOLE D'ACCÈS À HAUT DÉBIT POUR BUS UNIDIRECTIONNEL**

**Philippe JACQUET  
Paul MUHLETHALER**

Avril 1987

**LIGHTNET : UN PROTOCOLE D'ACCES A HAUT DEBIT  
POUR BUS UNIDIRECTIONNEL**

*Philippe Jacquet  
Paul Muhlethaler*

INRIA  
Rocquencourt  
78153 Le Chesnay cedex

**ABSTRACT**

*Nous présentons un protocole simple de communication haut débit s'adaptant à la topologie EXPRESSNET. Nous comparons les performances à partir de modèles analytiques simples et des simulations.*

**LIGHTNET : A HIGH PERFORMANCE MEDIA ACCESS PROTOCOL  
FOR UNIDIRECTIONNAL BUS NETWORK**

*Philippe Jacquet  
Paul Muhlethaler*

INRIA  
Rocquencourt  
78153 Le Chesnay cedex

**ABSTRACT**

*We introduce a simple high-speed multiple access protocol based on the topology of EXPRESSNET. We compare the performances through simple analytic models and simulations.*

# LIGHTNET : UN PROTOCOLE D'ACCES A HAUT DEBIT POUR BUS UNIDIRECTIONNEL

*Philippe Jacquet  
Paul Muhlethaler*

INRIA  
Rocquencourt  
78153 Le Chesnay cedex

## ABSTRACT

*Nous présentons un protocole simple de communication haut débit s'adaptant à la topologie EXPRESSNET. Nous comparons les performances à partir de modèles analytiques simples et des simulations.*

## 1. Introduction

Le développement des fibres optiques et de la technologie *optronique* ouvre de nouvelles perspectives pour les communications haut débit. Notre étude est dirigée sur le domaine des réseaux à accès *multiple*. On sait que la méthode du temps partagé (TDMA) se révèle en générale dommmageable en ce qui concerne les performances des réseaux. En effet le pourcentage de la bande passante utilisée est toujours inférieure à

$$\lambda_{\max} = \frac{N_a}{N_t}$$

où  $N_a$  est le nombre de stations actives à un instant donné et  $N_t$  la population totale. Or en utilisation normale, on ne peut guère compter que sur une population active de l'ordre de 10% de la population totale. Les réseaux FASNET et EXPRESSNET constituent une avancée notable à ce problème. Néanmoins le pourcentage utilisé de la bande passante subit encore une limitation, en ce sens que

$$\lambda_{\max} = \frac{N_a}{R + N_a}$$

où  $R$  est le délai de propagation exprimé en nombre de longueur moyenne de message. Ainsi pour  $N = 50$  et  $R = 20$  on obtient

$$\lambda_{\max} \leq 0.72..$$

Dans ce qui suit on s'attache à décrire des algorithmes (LIGHTNET1, LIGHTNET2) vérifiant les propriétés suivantes:

- (i) l'accès au canal est égal pour tous;
- (ii) l'utilisation optimale de 100% est accessible quelque soit la distribution de la population active;

(iii) l'accès est déterministe : c'est à dire une station passe son message en un délai fini fixé par les caractéristiques du réseau.

Le corps principal de cette note s'articule autour de trois sections principales :

- une présentation conjointe d'EXPRESSNET et de LIGHTNET;
- quelques commentaires sur un plan qualitatif;
- une évaluation comparée de performance sur la base de modèles simples.

## 2. Présentation des réseaux et des algorithmes.

### 2-1. EXPRESSNET.

Le réseau EXPRESSNET a été introduit en [1], et repose sur une technologie originale sur bus unidirectionnel.

#### 2-1-1. Le réseau EXPRESSNET.

Un même bus unidirectionnel est replié de façon à servir à la fois de canal d'écriture et de canal de lecture (*fig 1*). Une station est donc connectée deux fois au câble : une fois au *point d'écriture*, une autre fois au *point de lecture* (*fig 2*).

La topologie en Z donne un délai de propagation compris entre  $2a$  et  $3a$ ,  $a$  étant le délai de propagation que l'on aurait avec un câble simple. Ce délai peut être ramené entre  $a$  et  $2a$  grâce à l'adoption d'une topologie en *spirale* (*fig 3*). Dans les deux cas le délai écriture-lecture est fixe,  $2a$  et  $a$  respectivement, et c'est cette quantité que désignera désormais  $R$ .

#### 2-1-2. Les messages.

Les messages sont émis sur le canal d'écriture et lus sur le canal de lecture. Ils sont composés de deux parties consécutives : le pare-choc, consommable, et la partie vitale du message (*fig 4*).

#### 2-1-3. Principe : détection de collision sur le canal d'écriture.

Pour de plus amples détails on peut se référer à [1] ou [2]. Grossièrement, il s'agit de circulation de trains de messages sur le canal. Un train est constitué d'abord d'une locomotive (petite impulsion sur le canal d'écriture) ensuite d'une succession de wagons (messages) qui se sont accrochés les uns derrière les autres grâce à la procédure *accrochage*.

Une station effectue la procédure accrochage de la manière suivante.

Dans un premier temps, elle attend la fin du wagon courant (ou de la locomotive) détecté sur le point d'écriture. Dans le second temps, elle commence l'émission de son message. En cas de collision détectée sur le canal d'écriture elle interrompt sa transmission : un autre wagon se trouve derrière le premier, heureusement seuls les pare-choc ont été endommagés. La station reprend le processus à la fin de ce nouveau wagon, et ainsi de suite. Quand il n'y a plus de collision, c'est qu'on a réussi à accrocher son wagon à la fin du train courant.

La locomotive du train suivant sur le canal d'écriture est synchronisée avec la fin du train précédent sur le canal de lecture afin d'éviter des "télescopages". En conséquence, deux trains successifs sont séparés d'un blanc égal au délai de propagation  $R$ .

## 2-2. LIGHTNET

Le protocole LIGHTNET est introduit par les auteurs par cette présente note. Il est possible que le nom LIGHTNET ait déjà été utilisé auparavant et pour un autre usage. Cependant l'algorithme décrit ici est, à la connaissance des auteurs, totalement inédit.

### 2-2-1. Le réseau LIGHTNET.

On reprend les mêmes topologies en  $Z$  ou en spirale d'EXPRESSNET. On suppose en supplément que le canal est *sloté*, c'est à dire que le temps est divisé en tranches élémentaires appelées *slots* (*fig 5*). Ce découpage est *simultané* pour toutes les stations. Deux événements, ayant lieu respectivement dans les stations  $i$  et  $j$  ( $i < j$ ,  $i$  est en amont de  $j$ ), sont dits *simultanés* dans le cas où s'ils avaient correspondu à des tentatives d'émission sur leur site respectifs, ces tentatives auraient été entendues au même instant par un auditeur du canal situé en aval des deux stations.

On est donc en présence d'un protocole synchrone, en opposition avec l'algorithme précédent qui lui a la propriété d'être asynchrone. Cette synchronisation peut être maintenue par une station "gendarme" qui envoie sur le canal soit de temps en temps une impulsion de recalage, soit une modulation de référence en basse fréquence.

### 2-2-2. Les messages.

Le message a même configuration que dans EXPRESSNET. Une contrainte supplémentaire réside dans le fait que sa longueur doit être égale au slot au tout au moins ne pas l'excéder. D'autre part toute tentative d'émission ne peut se faire qu'au début d'un slot afin d'éviter qu'un message chevauche deux slots.

### 2-2-3. Principe : détection de collision sur le canal d'écriture et de lecture.

Un slot est soit *libre* soit *en session*.

On ne peut tenter une première émission que sur un slot *libre*. Si cette tentative avorte (par collision sur le pare choc) la station se met alors *en session*. Elle ne retentera une nouvelle émission de son message que lorsqu'elle verra passer ce premier slot sur le canal de *lecture*. Le slot immédiat en *écriture* est alors réservé aux stations impliquées dans la *session*, aucune nouvelle station ne peut intervenir. En effet, le fait de lire sur le canal de lecture un pare choc endommagé met le slot immédiat en écriture au statut *en session* ce qui le protège des nouveaux arrivants. Les stations en conflit réappliquent récursivement la procédure *mise en session* tant qu'elles subissent des collisions.

Une session est donc un ensemble de stations qui ont tenté leur première émission de message sur un même slot. Plusieurs sessions peuvent se

dérouler en parallèle sur le canal. Une même station peut traiter en même temps plusieurs messages qui sont donc classés dans des sessions distinctes.

Dans chaque session les stations sortent dans l'ordre de leur placement sur le réseau. Les sorties se succèdent tous les délais de propagation  $R$ .

### 2-3 Quelques remarques techniques.

On a donc essentiellement besoin de reconnaître en lecture un slot qui a été essayé en écriture; c'est à dire, par exemple, savoir compter  $R$  slots d'écart. On a aussi besoin de savoir détecter une collision sur le canal de lecture, ce qui peut être moins simple que sur le canal d'écriture. Pour ce faire, on peut utiliser une technique *violation* de code, tout en sachant que la durée de cette détection n'est pas pénalisante car elle peut être comprise dans le délai de propagation totale.

En revanche, la détection de collision sur le canal d'écriture tout en étant plus facile (détection de porteuse en amont) est cruciale en ce qui concerne les performances car elle détermine la longueur du pare-choc et par conséquent la fraction utile du corps des messages. Si le pare-choc s'allonge la fraction efficace de la bande passante diminuera d'autant (\*).

Dans ce qui suit nous considérons la longueur des pare-choc comme négligeable, ou tout au moins comme ne faisant pas partie de la bande passante. Ainsi il nous sera possible de monter, en tout cas pour LIGHTNET, jusqu'à 100% d'utilisation de la bande passante. Dans la réalité, il suffira de déduire de cette quantité la fraction consommée en pare-choc.

### 3. Quelques commentaires.

Nous articulons nos commentaires sur les trois exigences que nous avons explicitées en introduction à l'égard des algorithmes présentés.

#### 3-1. L'accès au canal est égal pour tous.

En ce qui concerne EXPRESSNET chaque locomotive équivaut, pour chacune des stations, à un droit de passage pour un message. Quant à LIGHTNET, c'est chaque slot *libre* qui équivaut à ce droit de passage toujours au bénéfice de toutes les stations sans distinction. Donc la propriété 1-(i) est vérifiée.

Néanmoins, en ce qui concerne les deux protocoles EXPRESSNET et LIGHTNET, un élément de déséquilibre est introduit entre les stations à l'égard des délais. En effet à l'intérieur d'un même train ou d'une même session, les stations amont sont susceptibles d'être favorisées par rapport aux stations aval.

Donc, en bref, toutes les stations ont la potentialité de passer le même nombre de messages, mais une dissymétrie subsiste sur le plan des délais d'acheminement de ces messages. On verra dans la section consacrée aux

---

(\*) On peut se passer de pare-choc dans les conditions suivantes. (i) Utilisation d'un canal auxiliaire de moindre débit légèrement en avance sur le canal principal et sur lequel se font les collisions. Le décalage entre les deux canaux joue maintenant le rôle de pare-choc. (ii) Medium tel que seul le message écrit en dernier sur un slot soit lisible.

résultats numériques que la stratégie de LIGHTNET est sensiblement meilleure que celle d'EXPRESSNET sur le plan des délais, même pour le cas "le pire", en l'occurrence le cas de la station aval extrême.

### 3-2. Introduction au parallélisme: 100% accessible d'utilisation du canal.

La nouveauté essentielle introduite par les protocoles LIGHTNET est le *parallélisme*. Plusieurs sessions peuvent se dérouler en parallèle dans le temps. En réalité c'est plutôt du multiplexage, ou parallélisme *mou*, les événements se déroulant toujours d'une manière séquentielle au niveau du canal. Ce parallélisme est encouragé au niveau même des stations afin qu'elles puissent traiter plusieurs messages en même temps. Ainsi l'utilisation optimale du canal est accessible même dans le cas extrême où la population active est réduite à une seule station (au lieu de  $1/R+1$  pour EXPRESSNET). En effet, pour obtenir 100% d'utilisation du canal il suffit qu'à chaque nouveau slot libre il existe une station avec un nouveau message à transmettre. En d'autres termes, il suffit que chaque session ne soit pas vide au sens de la théorie des ensembles. Donc la propriété 1-(ii) est vérifiée par LIGHTNET. Elle ne l'est pas par EXPRESSNET comme on l'a vu dans l'introduction.

Cette remarque entraîne une autre: tout trafic sur le réseau passe à concurrence de la capacité du canal. Nous étudierons, dans la section consacrée aux résultats numériques, un modèle poissonnien de création de message uniformément répartie sur tous les sites actifs. On constate que les délais d'acheminement des messages restent finis même quand l'effectif de la population active croît indéfiniment, du moment que la charge demandée cumulée n'excède pas la capacité du canal. La conséquence de cette propriété est que les phénomènes de files d'attente disparaissent quand la population active est potentiellement infinie.

On appelle généralement cette propriété la *transparence canal* ([4] et [5]). Le réseau EXPRESSNET possède aussi cette propriété, mais à un stade moins accompli. En effet EXPRESSNET réagit moins bien lorsque la population active est relativement faible, en vertu de la formule

$$\lambda_{\max} = \frac{1}{1 + \frac{R}{N_a}}$$

Donc si les phénomènes de files d'attente disparaissent bien quand la population active est infinie, ils risquent, à charge cumulée égale, de saturer le système en cas de *pénurie* de sites actifs. En bref, EXPRESSNET s'adapte moins bien aux trafics à "géométrie variable".

**Remarque:** Le défaut le plus important de LIGHTNET (et aussi de FASNET) par rapport à EXPRESSNET est le *slottage*. *Grosso modo* le slot doit être de la dimension du plus long message, ou du plus long morceau de message, admis sur le canal. En conséquence la longueur moyenne d'un message est toujours inférieure au slot, la différence étant irrémédiablement perdue en bande passante. Néanmoins on peut parier que dans un proche avenir les communications informatiques se feront entre des postes de travail, des terminaux "intelligents", sur la base de transferts de fichier en pièces de format standard; donc le slot s'imposera de lui-même.

### 3-3. Accès déterministe au canal.



En ce qui concerne EXPRESSNET, il est évident que le temps d'attente maximale d'une station est de deux trains: un train en lecture puis un train en écriture. La longueur d'un train ne peut excéder  $R+N_t$  (en comptant le blanc de séparation). Donc le délai maximal est borné par  $2(R+N_t)$ . Le protocole EXPRESSNET vérifie la propriété 1-(iii).

Quant à LIGHTNET, la propriété 1-(iii) est toujours vérifiée mais d'une manière moins évidente. Le délai peut être distingué en deux parties: une station désireuse d'émettre doit d'abord attendre d'avoir un slot *libre* puis doit attendre son *tour* dans sa session.

**Proposition 3-3-1:** *Le délai maximal est borné supérieurement par  $2RN_t$ .*

**Lemme 3-3-2:** *Une session dure au maximum  $RN_t$ .*

**Preuve:** évidente.

**Lemme 3-3-3:** *Une session se termine toujours par un slot libre.*

**Preuve:** évidente

**Corollaire 3-3-4:** *Il y a au plus  $R$  sessions non vides en déroulement au même instant sur le canal.*

**Preuve:** Regardons évoluer le nombre de sessions courantes, y compris les sessions vides. Une session se termine toujours par un slot libre. Or un slot libre est le point de départ d'une nouvelle session. Donc le nombre de sessions courantes est rigoureusement constant. Il est aisé de voir que ce nombre est exactement  $R$ .

**Preuve de la proposition:** Le premier lemme prouve que le délai en session n'excède pas  $RN_t$ . Donc au bout de  $RN_t$  une des sessions en cours (en fait toutes) aura terminé. En vertu du second lemme cette terminaison se concrétisera par un slot libre. Donc le délai d'attente pour un slot libre est aussi borné supérieurement par  $RN_t$ . La proposition est démontrée.

**Remarque:** On ne peut tout à fait se cacher le fait qu'EXPRESSNET présente un meilleur déterminisme que LIGHTNET, en vertu de la relation

$$R+N_t \leq RN_t .$$

Dans un premier temps on peut répondre que le déterminisme est davantage recherché pour une garantie de non autobloquage des systèmes que pour des contraintes de performance.

Dans un second temps nous reconnaissons que pour certains réseaux à forte contrainte temps réel (réseau de surveillance et d'alerte), LIGHTNET peut s'avérer moins adapté qu'EXPRESSNET. Néanmoins on peut envisager d'allier les qualités des deux protocoles par une technique d'*interruption* pour les messages urgents. Le montage serait le suivant. L'utilisation normale (trafic de routine) serait effectuée sous le mode LIGHTNET. En cas d'urgence, une station brouillerait le canal en écriture. Le signal, lu sur le canal de lecture, permuterait le système sur le mode EXPRESSNET (sloté). Le même signal constituerait la locomotive du train des messages urgent. La fin de ce train lue sur le canal d'écriture permuterait de nouveau le réseau sur le mode LIGHTNET.

## 4. Evaluation de performances

### 4-1. Modèle général.

Dans ce qui suit nous adoptons le modèle de réseau et de trafic suivant.

- (i) La population d'émetteurs est infinie.
- (ii) Les émetteurs sont tous identiques et indépendants. Ils sont soumis à une même loi poissonnienne de génération de message, dont la charge cumulée par unité de temps est  $\lambda$ .

On remarque immédiatement le fait que les hypothèses favorisent le protocole EXPRESSNET puisque ce dernier manifeste de certaines faiblesses quand le nombre de sites actifs est trop réduit. Les hypothèses permettent donc aux protocoles évalués de supporter 100% de charge du canal.

L'unité de temps est prise égale à la longueur moyenne des messages qui elle-même est prise égale au slot.

### 4-2. Paramètres évalués.

On s'intéresse au délai d'un message aléatoire généré sur un site quelconque. Le délai est classiquement la durée qui sépare la date de génération du message et la date de son émission avec succès sur le canal. On ne prend donc pas en compte la durée du transit sur le canal lui-même car cette quantité est le délai de propagation  $R$  (à quelques corrections dues à la topologie du réseau). Par compte il est de grand intérêt de comparer ces deux quantités.

Afin de serrer au plus près les finalités des réseaux évalués --en clair, les hauts débits --on considère comme très grande la quantité  $R$ , et on introduit le paramètre  $W$  dépendant du taux de génération  $\lambda$ :

$$W(\lambda) = \frac{\text{délai}}{R}$$

D'une manière plus formelle :

$$W(\lambda) = \lim_{R \rightarrow \infty} \frac{E[\text{délai} / \lambda, R]}{R}$$

où  $E[\text{délai}]$  est l'espérance du délai d'un message aléatoire généré avec le taux  $\lambda$  sur un réseau dont le délai de propagation est  $R$ .

### 4-2. Analyse d'EXPRESSNET.

Soit  $L$  la longueur moyenne d'un train, blanc de séparation compris. Il est aisé de voir que la fraction utilisée de la bande passante est

$$1 - \frac{R}{L}$$

Donc, en régime stationnaire, une équation simple d'équilibre est

$$1 - \frac{R}{L} = \lambda$$

D'où

$$L = \frac{R}{1-\lambda}$$

L'étude effectuée dans [3] montre que, lorsque  $R$  est grand, la distribution de la longueur des trains est pratiquement confinée sur sa moyenne (l'écart type est en  $\sqrt{R}$ ). En conséquence la longueur moyenne du train durant lequel un message aléatoire est généré est aussi  $L$  (\*). Donc le temps d'attente moyen d'un message est d'un demi train courant, soit

$$E[\text{delai}] = \frac{L}{2},$$

et le résultat asymptotique

$$W(\lambda) = \frac{1}{2} \cdot \frac{1}{1-\lambda}.$$

#### 4-3. LIGHTNET1.

Dans cette première version de LIGHTNET on suppose que lorsqu'une station est à la première tentative de transmission pour un message elle choisit le premier slot libre à venir.

De toute manière, quelque soit les versions, on a toujours

$$\text{delai} = \text{recherche slot libre} + \text{delai en session}.$$

**Modèle simplifié:** Les slots sont indépendants et ont une probabilité  $\mu$  d'être en session.

Donc une station avec un nouveau message attend son premier slot libre suivant une loi géométrique de raison  $\mu$ :

la station attend 1 slot avec une probabilité  $1-\mu$ ;

la station attend 2 slots avec une probabilité  $(1-\mu)\mu$ ;

la station attend  $k$  slots avec une probabilité  $(1-\mu)\mu^{k-1}$ .

Dans un premier temps on a donc

$$E[\text{recherche slot libre}] = \frac{1}{1-\mu}.$$

Reste l'évaluation du délai en session. La distribution du nombre de messages dans une session aléatoire est fonction de la distribution du nombre de slots "en session" qui la précède (et durant lesquels les nouvelles stations actives ont dû se retenir de transmettre). Donc la distribution de l'effectif d'une session est donnée par le *mélange* des lois poissonniennes de taux  $k\lambda$  ( $k$  entier supérieur à 1) avec les coefficients de pondération respectifs  $(1-\mu)\mu^{k-1}$ .

Donc le nombre  $N$  moyen de messages dans une session a pour formule

$$N = \sum_{k \geq 1} (1-\mu)\mu^{k-1} k\lambda = \frac{\lambda}{1-\mu}.$$

Les délais cumulés moyens par session  $D$  sont

$$D = R \sum_{k \geq 1} (1-\mu)\mu^{k-1} \sum_{n \geq 0} \frac{n(n-1)}{2} \frac{(k\lambda)^n}{n!} e^{-k\lambda}$$

(\*) Ce qui n'est pas évident *a priori*. En effet, si la distribution de la longueur du train avait été poissonnienne de moyenne  $L$  (et non pas asymptotiquement *quasi ponctuelle* sur  $L$ ), la longueur moyenne du train contemporain de la génération d'un message aléatoire aurait été là  $2R$ , comme l'énonce un résultat classique sur les cycles poissonniens.

$$= R \frac{\lambda^2}{2} \left( \frac{2\mu}{(1-\mu)^2} + \frac{1}{1-\mu} \right).$$

Donc le délai moyen en session ramené au message vérifie

$$E[\text{délai en session}] = \frac{D}{N} = R \left[ \frac{\lambda\mu}{1-\mu} + \frac{\lambda}{2} \right].$$

Par conséquent

$$E[\text{délai}] = \frac{1}{1-\mu} + R \left[ \frac{\lambda\mu}{1-\mu} + \frac{\lambda}{2} \right].$$

Une inconnue subsiste:  $\mu$ . La quantité  $\mu$  est une fonction de  $\lambda$ . On peut réécrire sur les sessions l'équation d'équilibre évoquée au 4-2. Soit  $S$  le nombre moyen de slots utilisés pour la résolution d'une session aléatoire. On a en régime stationnaire

$$\frac{N}{S} = \lambda.$$

Or le nombre de slots d'une session est égal au nombre de messages impliqués dans celle-ci, sauf dans le cas précis où la session est vide; dans ce dernier cas il y a un et un seul slot utilisé. Donc

$$S = N + \sum_{k \geq 1} (1-\mu) \mu^{k-1} e^{-k\lambda} = \frac{\lambda}{1-\mu} + \frac{(1-\mu)e^{-\lambda}}{1-\mu e^{-\lambda}}.$$

Avec l'équation d'équilibre on obtient une équation quadratique en  $\mu$  que l'on résout:

$$\mu = 1 - \frac{1-\lambda + \sqrt{(1-\lambda)^2 + 4(e^\lambda - 1)(1-\lambda)}}{2}.$$

Pour en revenir à  $W(\lambda)$  on a l'expression

$$W(\lambda) = \frac{\lambda}{1-\mu} - \frac{\lambda}{2}$$

Soit

$$W(\lambda) = \frac{2\lambda}{1-\lambda + \sqrt{(1-\lambda)^2 + 4(e^\lambda - 1)(1-\lambda)}} - \frac{\lambda}{2}.$$

**Commentaires:** Quand  $\lambda \rightarrow 0$  on a le comportement

$$W(\lambda) = \frac{\lambda}{2} + O(\lambda^2).$$

Alors que le protocole EXPRESSNET induit un développement de la forme

$$W(\lambda) = \frac{1}{2} + \frac{\lambda}{2} + O(\lambda^2),$$

D'ailleurs le délai moyen d'EXPRESSNET ne saurait de toute manière descendre en dessous de  $\frac{1}{2}$  compte tenu du délai de propagation incompressible qui sépare deux trains consécutifs.

Quand  $\lambda \rightarrow 1$  EXPRESSNET donne

$$W(\lambda) = \frac{1}{2} \cdot \frac{1}{1-\lambda}.$$

tandis que LIGHTNET1 affiche

$$W(\lambda) = \frac{1}{\sqrt{e-1}} \cdot \frac{1}{\sqrt{1-\lambda}} + O(1).$$

#### 4-4. LIGHTNET2.

Nous présentons une nouvelle version de LIGHTNET. La différence avec la précédente version réside dans la recherche du slot libre. Maintenant chaque station tire à *pile ou face* à chaque slot en écriture. Les tirages sont tous indépendants mais sont effectués avec la même loi de probabilité. On appelle  $p$  la probabilité de tirer *face*. On suppose  $p$  petit mais tel que  $pR$  soit grand (par exemple  $p = \frac{1}{\sqrt{R}}$ ). Une station choisit son slot libre quand la coïncidence slot libre et tirage=*face* est réalisée.

**Modèle simplifié:** En plus des hypothèses sur le statut des slot présentées pour LIGHTNET1 on suppose que la loi du nombre de stations tentant de transmettre sur un slot libre aléatoire est une loi de Poisson de paramètre  $\gamma$ .

Donc la loi du nombre de messages impliqués dans une session aléatoire est aussi la loi de Poisson de paramètre  $\gamma$ . C'est pour des raisons évidentes la loi du nombre de stations avec lesquelles un message aléatoire entre en compétition dans sa session. Il n'est pas besoin de s'étendre sur les égalités suivantes.

$$\begin{aligned} E[\text{délai}] &= E[\text{recherche slot libre}] + E[\text{délai en session}] \\ &= \frac{1}{p(1-\mu)} + \frac{\gamma R}{2}. \end{aligned}$$

Donc

$$W(\lambda) = \frac{\gamma}{2}.$$

Il faut évidemment évaluer les inconnues  $\gamma$  et  $\mu$ . Pour  $\gamma$  on utilise l'équation d'équilibre. Il est aisé d'évaluer

$$\begin{aligned} N &= \gamma, \\ S &= \gamma + e^{-\gamma}. \end{aligned}$$

L'équation d'équilibre est toujours

$$\frac{N}{S} = \lambda;$$

donc on obtient la relation simple

$$\gamma e^{\gamma} = \frac{\lambda}{1-\lambda}$$

qui est aisé à résoudre numériquement.

Reste à exprimer  $\mu$ . Pour ce faire on effectue le raisonnement suivant. Pour un slot libre il y a en moyenne  $\gamma - 1 + e^{-\gamma}$  slots en session. En effet chaque slot libre marque le début d'une session. Soit  $n$  l'effectif de la session. Si  $n \geq 2$  il y a un conflit et par conséquent  $n - 1$  slots sont induits en session (le dernier slot de la session est toujours libre). Si  $n \leq 1$  il n'y a pas de slot en session induit. La loi de  $n$  étant de Poisson de paramètre  $\gamma$  le résultat est

évident.

Donc

$$\mu = \frac{\gamma - 1 + e^{-\gamma}}{\gamma + e^{-\gamma}}.$$

Ceci permet d'exprimer complètement le délai moyen, par exemple quand  $p = \frac{1}{\sqrt{R}}$ ,

$$E[\text{délai}] = (\gamma + e^{-\gamma})\sqrt{R} + \frac{\gamma}{2}R.$$

**Commentaires:** Quand  $\lambda \rightarrow 0$  LIGHTNET2 a le même comportement que LIGHTNET1 à savoir

$$W(\lambda) = \frac{\lambda}{2} + O(\lambda^2).$$

Par contre, quand  $\lambda \rightarrow 1$ , on constate une nette amélioration des délais qui s'exprime par le développement suivant

$$W(\lambda) = \frac{1}{2} \log\left(\frac{1}{1-\lambda}\right) + O\left[\log\log\left(\frac{1}{1-\lambda}\right)\right].$$

On rappelle que LIGHTNET1 présente le développement

$$W(\lambda) = \frac{1}{\sqrt{e-1}} \cdot \frac{1}{\sqrt{1-\lambda}} + O(1),$$

et qu'EXPRESSNET fournit

$$W(\lambda) = \frac{1}{2} \cdot \frac{1}{1-\lambda}.$$

Dans tous les cas, d'après les formules exposées, le double du délai moyen de LIGHTNET2 est toujours inférieur au logarithme du double du délai moyen d'EXPRESSNET.

**Remarque:** La version LIGHTNET2 semble ne pas convenir au cadre que nous avons défini en introduction pour les protocoles haut débit. En effet LIGHTNET2 n'est pas déterministe (condition 1-(iii)) puisque la recherche du *slot libre* se fait au moyen d'un processus par tirages au sort successifs dont la durée n'est pas bornée d'une manière absolue. En réalité cette présentation est uniquement une vue de l'esprit destinée à simplifier *a priori* la présentation de LIGHTNET2 et le véritable processus de recherche du premier slot libre est déterministe en ce sens que sa durée est bornée d'une manière absolue. Ce processus est le suivant. La station qui vient juste d'avoir un message choisit aléatoirement un nombre entier  $F$  compris entre 1 et  $F_{\max} = \frac{2}{p}$  (donc, par exemple, entre 1 et  $2\sqrt{R}$ ). Ensuite elle attend que le  $F$ -ème slot libre se présente pour tenter la première transmission de son message et d'entrer éventuellement en session. La durée entre deux slots libres étant bornée d'une manière absolue par théorème, le processus est donc finalement bien déterministe. D'autre part ce processus donne le même nombre moyen de slot libres à attendre que dans le cas de la première description, à savoir  $E[F] = \frac{1}{p}$ .

Comme on peut s'y attendre cette nouvelle méthode d'accès au slot libre ne change pas grand chose dans le cadre de nos évaluations simplifiées. En effet la motivation principale de LIGHTNET2 consiste à répartir le plus uniformément possible le nombre de tentatives d'émission par slot libre. Notre modèle quantitatif suppose que cette répartition se fait en fait d'une manière parfaite et oublie le véritable processus de répartition.

Une dernière remarque concernant la condition 1-(ii). On peut se représenter le processus précédent par le fait que chaque station possède une fenêtre *d'attente* de taille  $F_{\max}$  représentant à chaque instant les  $F_{\max}$  slots libres à venir. Quand une station a un message en attente elle le place donc aléatoirement sur sa fenêtre, au numéro  $F$  par exemple. Si on fait l'hypothèse qu'une station ne doit avoir qu'un seul message à la fois sur sa fenêtre on contredit la condition 1-(ii). En effet une station, toute seule sur le réseau, ne peut occuper le canal qu'au taux moyen de  $p$  (donc  $\frac{1}{\sqrt{R}}$ , encore meilleur que EXPRESSNET qui lui n'offre que  $\frac{1}{R}$  pour une station unique). Pour remédier à cela on peut permettre aux stations de mettre autant de messages qu'elles veulent sur leur fenêtre (toujours par affectation aléatoire). A saturation, une station unique peut utiliser toute la bande passante si on lui permet de saturer sa fenêtre d'attente.

### Résultats numériques

Les premières courbes (fig 6) montrent les délais limites  $W(\lambda)$  en fonction de  $\lambda$  calculés selon les modèles présentés.

La figure 7 compare les évaluations analytiques aux simulations. Le réseau a pour configuration  $N_t=50$  et  $R=20$  et les trafics sont poissonniens et symétriques. Pour LIGHTNET les simulations collent bien au modèle analytique. Quant à EXPRESSNET on remarque une divergence quand  $\lambda \rightarrow 70\%$ . Cela s'explique par le fait qu'avec  $R$  de l'ordre de 60% de  $N$  on se trouve sensiblement éloigné des conditions de la transparence canal et, par conséquent, des conditions d'évaluation du modèle analytique. On peut corriger cet écart en appliquant le modèle que nous avons développé pour des circonstances plus générales d'emploi d'EXPRESSNET [3]. La formule appliquée est la suivante.

$$E[\text{delai}] = \frac{R}{2(1-\lambda) \left[ 1 - \frac{\lambda R}{N(1-\lambda)} \right]}$$

Ces courbes illustrent de façon claire l'amélioration qu'apporte la stratégie LIGHTNET par rapport au protocole EXPRESSNET en ce qui concerne les débits et les délais.

### Références

- [1] F. Tobagi, F. Boronovo, L. Fratta, *Expressnet: a high-performance integrated-services local area network*, IEEE Journal on selected areas in

communications, vol sac-1, No 5, November 1983, pp 898-926.

[2] F. Tobagi, M. Fine, *Performance of unidirectional broadcast local area networks: Expressnet and Fasnet*, same issue, pp 913-926.

[3] P. Jacquet, P. Muhletahler, *A simple analysis of expressnet with the most upstream station generating trains*, rapport INRIA à paraître, janvier 1987.

[4] P. Jacquet, *Evaluation of the queues in stations for the deterministic tree protocol*, rapport INRIA à paraître, 1987.

[5] P. Flajolet, P. Jacquet, *Analytic models for tree communication protocol*, invited lecture at the NATO Advanced Study Institute on " *Flow Control of Congested Networks* ", Capri, October 1986, Springer, New-York, 1987.



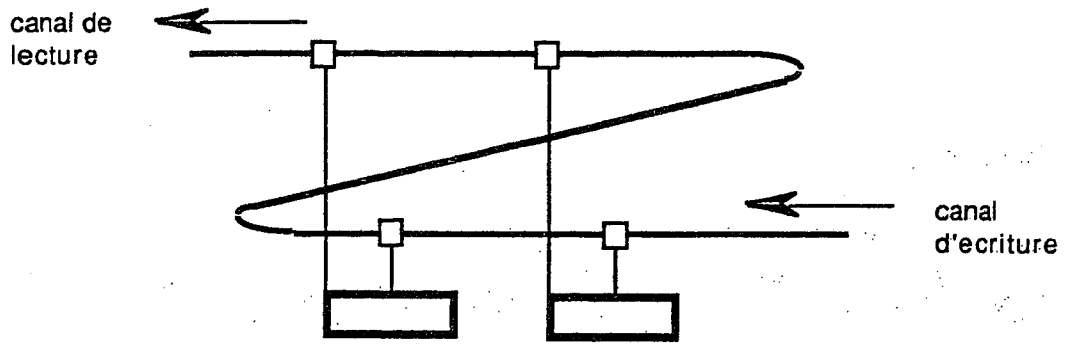


figure 1: topologie en "Z"

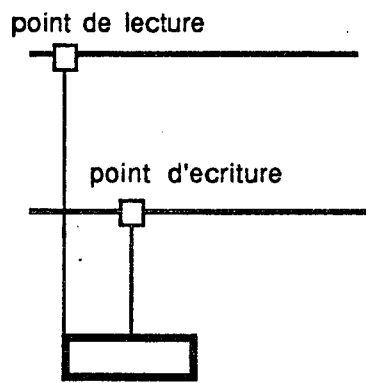


figure 2

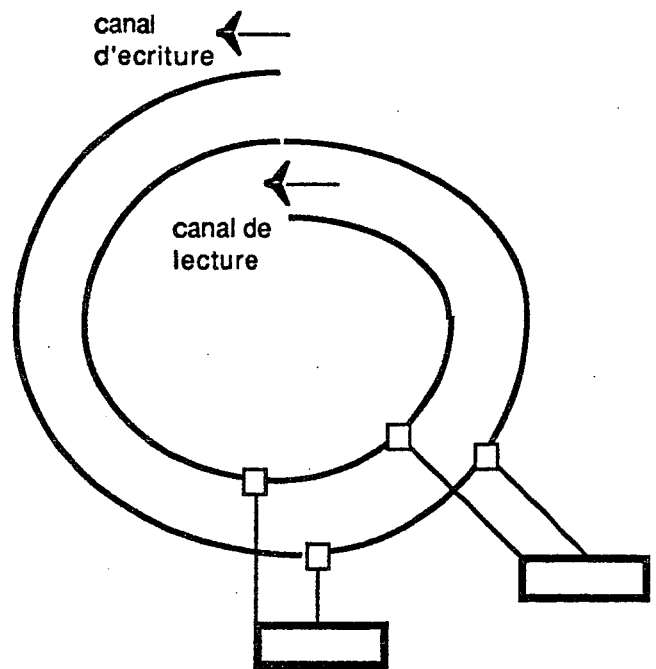


figure 3: topologie en "spirale"

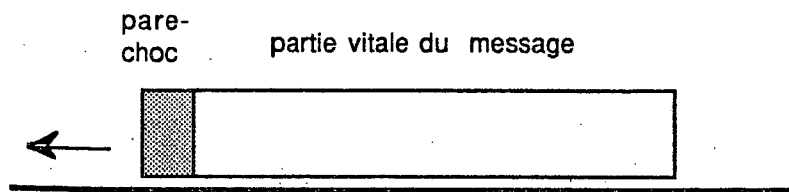


figure 4: configuration de message

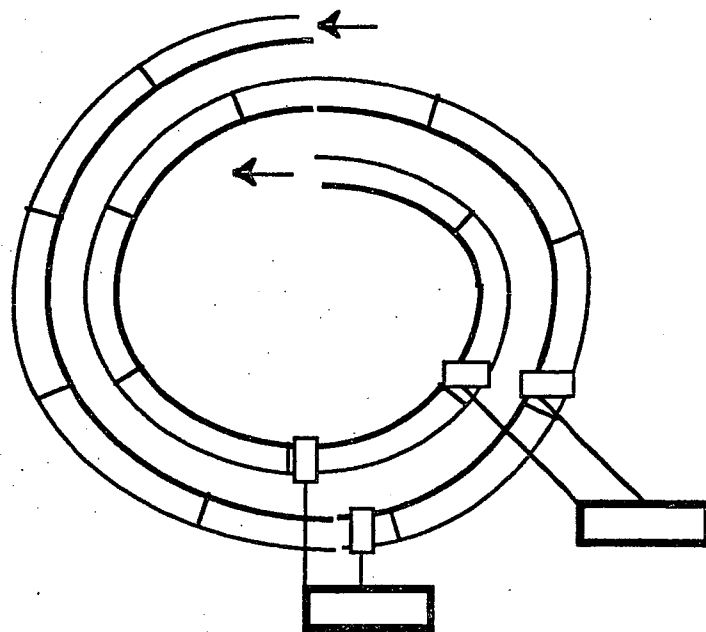


figure 5: canal slotté

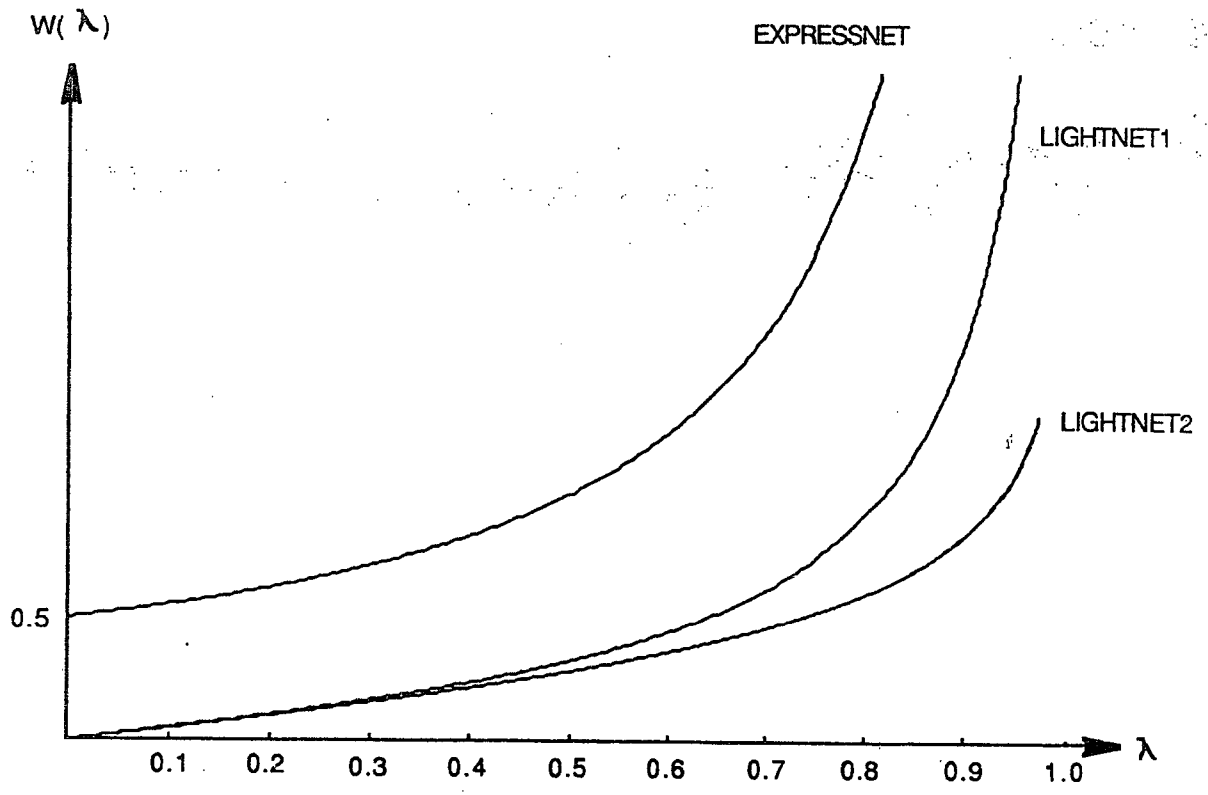


figure 6

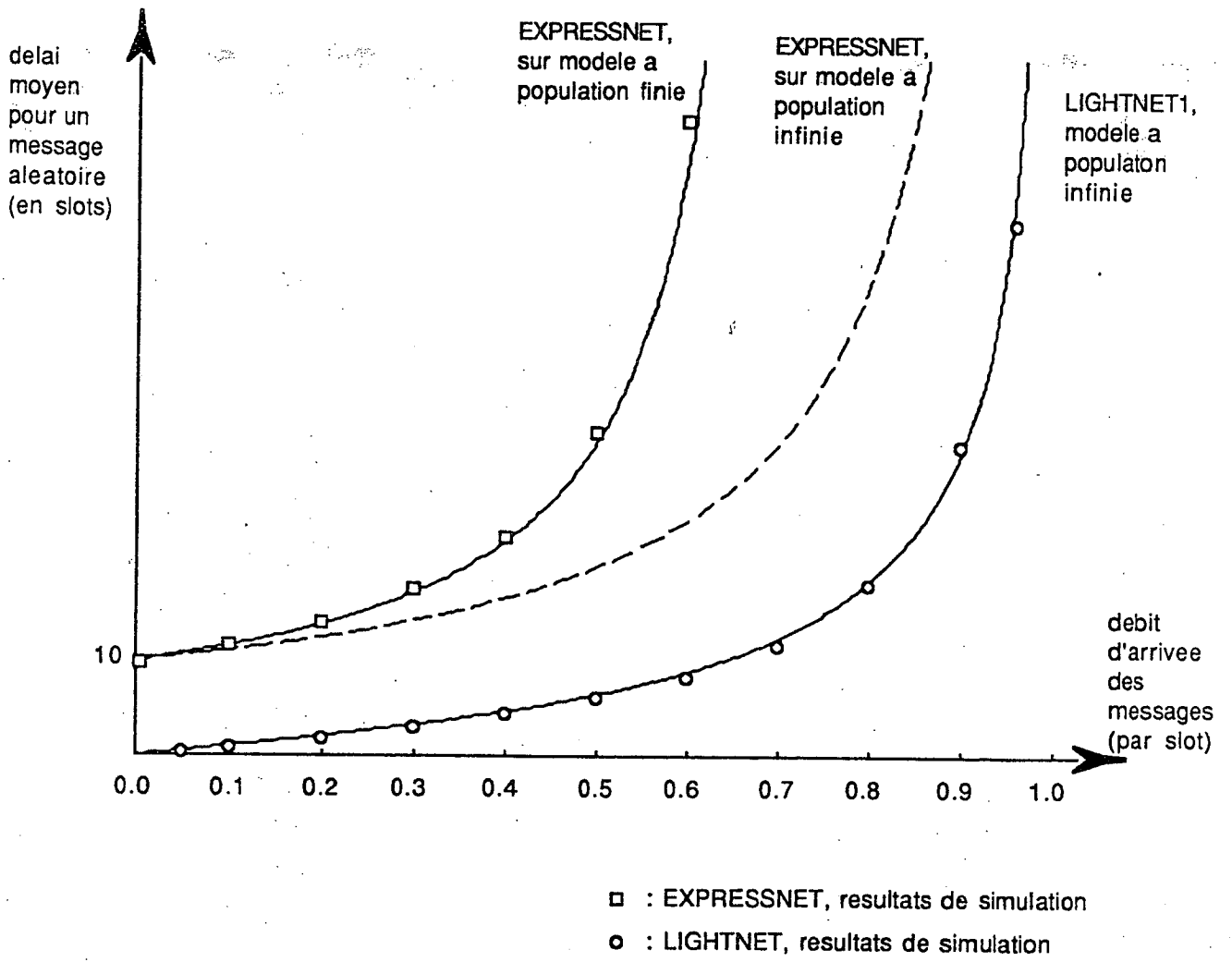


figure 7



