THe free access stack algorithm beyond infinity

Philippe Jacquet

To cite this version:

Philippe Jacquet. THe free access stack algorithm beyond infinity. RR-1255, INRIA. 1990. <inria-00075303>

HAL Id: inria-00075303
https://hal.inria.fr/inria-00075303

Submitted on 24 May 2006

UNITÉ DE RECHERCHE
INRIA-ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P. 105
78153 Le Chesnay Cedex
France

Tél.:(1) 39 63 55 11

Rapports de Recherche

N° 1255

Programme 1
Programmation, Calcul Symbolique
et Intelligence Artificielle

THE FREE ACCESS STACK ALGORITHM BEYOND INFINITY

Philippe JACQUET

Juin 1990

THE FREE ACCESS STACK ALGORITHM BEYOND INFINITY

Philippe JACQUET[†]
INRIA
FRANCE

Abstract. *We give an analytical evaluation of the behaviour of the free access stack algorithm when the input load, λ packet per slot, is above the maximum throughput achievable by the protocol (within* 0.360177 *packet per slot) in infinite population model. In particular we analytically and quantitatively derive the marginal output stream that the system sustains on the channel.*

LE PROTOCOLE EN ARBRE AU DELA DE L'INFINI

Résumé. *Nous donnons une évaluation analytique du comportement du protocole en arbre à arrivées libres quand le taux d'arrivée, de λ paquet par slot, est supérieur au débit maximum que peut soutenir le protocole (à savoir* 0.360177 *paquet par slot) dans le modèle de la population infinie. En particulier nous dérivons de manière analytique et quantitative le débit de sortie marginal que le système continue de fournir sur le canal.*

[†] This research was partially supported by the ESPRIT II Basic Research Actions Program of the EC under contract No. 3075 (project ALCOM).

1 INTRODUCTION

We analyze the performance of a protocol for managing the use of a *single-channel packet switching communication network* such as that used in the Ethernet. We adopt the following model, commonly taken as the basis of mathematical studies of multiple access channel [3]. The time is slotted, and stations can start transmitting only at the beginning of slots. Each packet is supposed to be exactly one slot long. Each transmission is within the reception range of every user. When more users than one transmit simultaneously, packet are said to *collide*, none is correctly transmitted; these collisions are treated as transmission errors and each user strives to retransmit its colliding packet till it is correctly received. The retransmission policy is the *collision resolution algorithm.*

In this paper we consider the stack algorithm with free access [1,2,3,4,5]. We adopt the *infinite* population model. This probabilistic model assume a potentially infinite number of users, among them at each slot new active users (users with a new packet to be transmitted) follows a Poisson distribution equivalent to λ packet per slot. It has been shown [2,4] that this model implies that the network cannot be stable if quantity λ exceeds some critical value $\lambda_{\max}$. Theoritical value of $\lambda_{\max}$ can be computed with an arbitrary accuracy, for instance $\lambda_{\max} = 0.360177...$

Fayolle *et al.* [1,2] published a complete analysis of this protocol in infinite population model when $\lambda < \lambda_{\max}$. Under this condition the network is stable and packets experience finite delay with probability one. [1] gives a complete characterization of the packet delay distribution *versus* quantity λ. The question remained about the behaviour of the network when $\lambda \geq \lambda_{\max}$. The purpose of this paper is to give a qualitative and quantitative answer to this question. It will be shown that when $\lambda \geq \lambda_{\max}$ packets can experience infinite delay with a non-zero probability, which increases when λ increases. A key result is the fact that finite delays are also finite in mean (and in other moment), and that the network reaches a kind of stable state with a continuous exits rate from packets that experience finite delay.

We give an exact evaluation of this exit rate, or *marginal output* stream, of λ_o packet per slot. We show that $\lim_{\lambda\to\infty} \lambda_o = 0$ but we always have $\lambda_o > 0$ when $\lambda > 0$. This property illustrates the robustness and flexibility of the stack algorithm when submitted to extreme conditions. That property strongly counterparts the weakness of some other algorithms like ALOHA or the exponential back-off algorithm (the one used in standard Ethernet): it is known [ref] that both of them are unstable under infinite population model as soon as $\lambda > 0$, and $\lambda_o = 0$ in every case.

A. Description of the protocol

We stress the reader to refer to [1,2,4] for a complete description of the stack algorithm and its properties. Follows a short description of the algorithm.

The protocol we consider is free access. Free access means that every station that generates a new packet immediately transmits at the beginning of the next slot, say slot number s. Doing so it initializes a counter $C(s)$ at zero and follows the rules below.

I1 If $C(s) = 0$, the station transmits on slot s. If a collision occurs, counter $C(s+1)$ assumes one of the values zero and one with respective probability p and q, $p + q = 1$ (the station has to follow the retransmission policy). Otherwise if no collision is detected, the transmission was successful and the counter $C(s)$ becomes idle.

I2 When $C(s) > 0$, the station modifies the counter either when slot s is a collision, whereupon $C(s+1) = C(s) + 1$, or when slot s is a blank or a success, in which case $C(s+1) = C(s) - 1$.

From a global point of view this protocol consists of monitoring a virtual *stack* containing the station which are waiting for (re)transmission. The stack is a sequence of cells, indexed by $\{0, 1, 2, \cdots\}$: during the slot s, the elements of cell number i are the stations such that $C(s) = i$. According to the protocol, the stack levels are incremented in the event of a collision (and colliding users split over the two first levels), and decremented in the event of a blank or success slot.

B. Parameters of interest

The key parameter of this protocol is the session. The session is the set of slots involved in the resolution of a given initial collision. To be precise we define the notion of n-session by the minimal set of slots separating the following two events:

1) there are exactly n station inserted in the global stack, and each of them is stored at level 0;
2) counting from this initial slot 1) the number of non collision slots just exceeds the number of collision slots by one (i.e., the stack is emptied).

The length of an n-session is usually termed the *collision resolution interval* (CRI) and will be here noted by $\mathbf{L}_n$.

Another parameter of interest involves deeper insight in the retransmission process. We call $\mathbf{S}(u)$ the output polynom of an n-session. Supposing that the n-session starts at slot 1, we have the identity

$$\mathbf{S}_n(u) = \sum_{k=1}^{k=\mathbf{L}_n} \delta_k u^k ,$$

with $\delta_k = 1$ if slot k is a success, and $\delta_k = 0$ otherwise. The variable u is a complex number of modulus *a priori* stricly less than unity, as supposed in classic generating functions.

2 PROBABILISTIC MODEL AND CONSIDERATIONS

We assume that the number of station is so large as to be effectively infinite. Moreover, the number A of newly created active users in each slot is assumed to be independent of the state of the stack and its history and to be Poisson with a fixed rate λ,

$$\Pr\{A = n\} = \frac{\lambda^n}{n!} e^{-\lambda} .$$

According to this model let $P_n(u)$ be the characteristic function of the (finite) length of an n-session, namely $P_n(u) = \sum_{k=1}^{\infty} \Pr\{\mathbf{L}_n = k\} u^k$. In the latter formula quantity u refers to a complex number of modulus less than or equal to unity. From obvious considerations we have $P_0(u) = P_1(u) = u$. Of course $P_n(1) = 1 - \Pr\{\mathbf{L}_n = \infty\}$. Let $P_n = P_n(1)$ be the probability for an n-session to be finite. When $\lambda < \lambda_{\max}$, we have $P_n = 1$ for all values of n. When $\lambda \geq \lambda_{\max}$, we generally have $P_n < 1$, except for the trivial cases which lead to $P_0 = P_1 = 1$ whatever be λ.

Let us introduce $S_n(u)$ as the mean output polynom of an n-session. We have $S_n(u) = \mathrm{E}[\mathbf{S}_n(u)]$ or, in other words, $S_n(u) = \sum_{k=1}^{\infty} u^k \Pr\{\mathbf{L}_n \geq k \,\&\, \text{slot } k \text{ is a success}\}$. Of course

$S_0(u) = 0$ and $S_1(u) = u$. The output polynom is a special feature that captures the output process sustained by the protocol. Note that the definition of $S_n(u)$ fits the case where the n-session is infinite ($\mathbf{L}_n = \infty$). In particular, by Cesaro,

$$\lim_{u \to 1} (1-u) S_n(u) = (1 - P_n)\lambda_o ,$$

where λ_o is the marginal output stream allowed by the protocol when $\lambda \geq \lambda_{\max}$. Thus determining $S_n(u)$ leads to the determination of quantity λ_o when $\lambda \geq \lambda_{\max}$ (when $\lambda < \lambda_{\max}$ we have $\lambda_o = \lambda$ but the previous equation does not find any application since $P_n = 1$).

The next section of paper is dedicated to the statement of functionnal equations that determine parameters $P_n(u)$ and $S_n(u)$. A special section develops the resolution of such equations. Quantitative results are presented and discussed in the last section.

3 THE FUNCTIONAL EQUATIONS

A. Equations about CRI length

The classic recursion formula for $n \geq 2$, see [1,2] $\mathbf{L}_n = 1 + \mathbf{L}_{n_1 + A_1} + \mathbf{L}_{n_2 + A_2}$, (with A_i of Poisson λ) holds, even with infinite value. Introducing $P(z,u) = \sum_{n=0}^{\infty} P_(u) z^n e^{-z}/n!$, we get the functional equation:

$$\frac{P(z,u)}{u} = (1+z)e^{-z} + P(pz+\lambda,u)P(qz+\lambda,u) - P(\lambda,u)(P(\lambda,u)(1+z) + P_z(\lambda,u)z)e^{-z} , \quad (1)$$

with P_z expressing the first derivative of $P(z,u)$ with respect to variable z. Expresing $P(z) = P(z,1)$ we get

$$P(z) = P(pz+\lambda)P(qz+\lambda) + (1+z)e^{-z} - P(\lambda)(P(\lambda)(1+z) + P'(\lambda)z)e^{-z} , \quad (2)$$

($P'(z)$ is the first derivative of $P(z)$). Note that $P(z) = 1$ (the P_ns are all equal to 1) is a trivial solution to (2); in fact this is the *real* solution when $\lambda < \lambda_{\max}$. The problem is to find the other solution which becomes the real one when $\lambda \geq \lambda_{\max}$. This will be treated in section 2.

Let us introduce $X_n = \mathrm{E}[\mathbf{L}_n$ when finite], or in other words $X_n = P'_n(1)$. Using the generating function $X(z) = \sum_{n=0}^{\infty} X_n z^n e^{-z}/n! = P_u(z,1)$, we get the functional equation

$$\begin{aligned} X(z) =& P(z) + P(qz+\lambda)X(pz+\lambda) + P(pz+\lambda)X(qz+\lambda) - \\ &- X(\lambda)(2P(\lambda)(1+z) + P'(\lambda)z)e^{-z} - X'(\lambda)P(\lambda)ze^{-z} . \end{aligned} \quad (3)$$

The knowledge of $P(z,u)$ or, in particular of P_n and X_n, allows us to determine the temporary behaviour of the protocol before meeting an infinite CRI. But, of course some works remain to determine the behaviour of the algorithm during an infinite CRI, namely its marginal output. This is the object of the following section.

B. Equations about the marginal output stream

When $n \geq 2$ the basic recursion on $\mathbf{L}_n$ can be translated for the use of the output polynom $\mathbf{S}_n(u)$, namely

$$\mathbf{S}_n(u) = u[\mathbf{S}_{n_1+A_1}(u) + u^{\mathbf{L}_{n_1+A_1}} \mathbf{S}_{n_2+A_2}(u)] . \quad (4)$$

Using the expectation $S_n(u)$ and its generating function $S(z,u) = \sum_{n=0}^{\infty} S_n(u) z^n e^{-z}/n!$, we get the following functionnal equation

$$\begin{aligned}\frac{S(z,u)}{u} - ze^{-z} = {} & S(pz+\lambda,u) + P(pz+\lambda,u)S(qz+\lambda,u) - \\ & - S(\lambda,u)((1+P(\lambda,u))(1+z) + P_z(\lambda,u)pz)e^{-z} - \\ & - S_z(\lambda,u)z(p+qP(\lambda,u))e^{-z} \ .\end{aligned} \tag{5}$$

And the considerations of section **2** lead to the identity $\lim_{u \to 1}(1-u)S(z,u) = (1-P(z))\lambda_o$.

4 RESOLUTION OF THE FUNCTIONAL EQUATIONS

A. Resolution for $P(z)$

Let us call T the non-linear operator defined on analytical functions $f(z)$ by $Tf(z) = f(pz+\lambda)f(qz+\lambda) - f(\lambda)(f(\lambda)(1+z) + f'(\lambda)z)e^{-z} + (1+z)e^{-z}$. According to (2), the function $P(z)$ is a fixed point of the operator T. The function 1 is also a fixed point of T. We use *brute force* to obtain the result: we start with $(1+z)e^{-z}$, which has all its coefficients less or equal to those of $P(z)$ (the formal identity $P(z) = (1+z)e^{-z}$ should lead to all P_n's equal to 0 when $n \geq 2$), and then to iterate T. We get

$$\lim_{n \to \infty} T^n(1+z)e^{-z} = P(z) \ . \tag{6}$$

As a numerical illustration, it is interesting to notice that when $\lambda < \lambda_{\max}$ we get $\lim_{n \to \infty} T^n(1+z)e^{-z} = 1$ as expected. When $\lambda > \lambda_{\max}$ the iterations converge to something else which is suspected to be $P(z)$. Note that the iterations converge with difficulty when λ is close to $\lambda_{\max}$, since it is easy to check that the gradient of T at $f = 1$ when $\lambda = \lambda_{\max}$ is exactly the identity: $T(1+h) = 1 + h + O(h^2)$ when function $h \to 0$.

B. Resolution for $X(z)$

Now we have function $P(z)$. Equation (3) is simpler to deal with since it is linear with a more familiar form. Let us note σ_1 and σ_2 the linear operators defined by $\sigma_1 f(z) = f(pz+\lambda)$ and $\sigma_2 f(z) = f(qz+\lambda)$. Let us introduce the linear operator Π_2 defined on analytical functions $f(z)$ by $\Pi_2 f(z) = f(z) - f(0) - zf'(0)$. We define the operator R by $Rf(z) = \sigma_2 P(z)\Pi_2\sigma_1 f(z) + \sigma_1 P(z)\Pi_2\sigma_2 f(z)$. For instance $\Pi_2\sigma_1 f(z) = f(\sigma_1(z)) - f(\lambda) - f'(\lambda)pz$. The interesting fact about the operator R is the fact that $\sum_{n=0}^{\infty} R^n$ converges like a geometric sequence of rate $p^2+q^2 < 1$ (basically the square comes from the application of operator Π_2) whatever be λ. Therefore equation $f - Rf = g$ of unknown f and parameter g has the unique solution $f = \sum_{n=0}^{\infty} R^n g = (1-R)^{-1}g$.

The equation (3) can be tuned to the form

$$X(z) - RX(z) = g_0(z) + X(\lambda)g_1(z) + X'(\lambda)g_2(z) \ .$$

For instance

$$\begin{cases} g_0(z) = P(z) \\ g_1(z) = P(pz+\lambda) + P(qz+\lambda) - (2P(\lambda)(1+z) + P'(\lambda)z)e^{-z} \\ g_2(z) = z(qP(pz+\lambda) + pP(qz+\lambda) - P(\lambda)e^{-z}) \ . \end{cases}$$

In this perspective we can use the operator $(1-R)^{-1}$, and obtain $X(z) = (1-R)^{-1}g_0(z) + X(\lambda)(1-R)^{-1}g_1(z) + X'(\lambda)(1-R)^{-1}g_2(z)$.

By using elementary identification at $z = \lambda$ we get the following linear system in $X(\lambda)$ and $X'(\lambda)$:

$$\begin{cases} aX(\lambda) + bX'(\lambda) = x \\ cX(\lambda) + dX'(\lambda) = y \end{cases}$$

with

$$\begin{array}{rl} a = 1-(1-R)^{-1}g_1(\lambda) & b = -(1-R)^{-1}g_2(\lambda) \\ c = -((1-R)^{-1}g_1)'(\lambda) & d = 1-((1-R)^{-1}g_2)'(\lambda)\ , \\ x = (1-R)^{-1}g_0(\lambda) & y = ((1-R)^{-1}g_0)'(\lambda)\ . \end{array}$$

Therefore $X(\lambda) = (dx - by)/\det$ and $X'(\lambda) = (-cx + ay)/\det$ with $\det = ad - bc$.

C. Resolution for $S(z,u)$

Let us call $H(u)$ the linear operator defined by $H(u)f(z) = u(\Pi_2\sigma_1 f(z) + P(pz+\lambda, u)\Pi_2\sigma_2 f(z))$. This operator is interesting about two points. First, the serie $\sum_{n=0}^{\infty} H^n(u)$ converges like a geometric serie of rate $u(p^2+q^2)$ thus is absolutely convergent when $|u| < (p^2+q^2)^{-1}$. Second, equation (5) translates into $S(z,u) - H(u)S(z,u) = h_0(z,u) + S(\lambda,u)h_1(z,u) + S_z(\lambda,u)h_2(z,u)$ with

$$\begin{cases} h_0(z,u) = uze^{-z} \\ h_1(z,u) = u(1 + P(pz+\lambda,u) - ((1+P(\lambda,u))(1+z) + P_z(\lambda,u)pz)e^{-z}) \\ h_2(z,u) = uz(p + qP(pz+\lambda,u) - (p+qP(\lambda,u))e^{-z})\ . \end{cases}$$

We use the operator $(1-H(u))^{-1} = \sum_{n=0}^{\infty} H^n(u)$, and obtain $S(z,u) = (1-H(u))^{-1}h_0(z,u) + S(\lambda,u)(1-H(u))^{-1}h_1(z,u) + S_z(\lambda,u)(1-H(u))^{-1}h_2(z,u)$.

By using elementary identification at $z = \lambda$ we get the following linear system in $S(\lambda,u)$ and $S_z(\lambda,u)$:

$$\begin{cases} a(u)S(\lambda,u) + b(u)S_z(\lambda,u) = x(u) \\ c(u)S(\lambda,u) + d(u)S_z(\lambda,u) = y(u) \end{cases}$$

with

$$\begin{array}{rl} a(u) = 1-(1-H(u))^{-1}h_1(\lambda,u) & b(u) = -(1-H(u))^{-1}h_2(\lambda,u) \\ c(u) = -((1-H(u))^{-1}h_1)_z(\lambda,u) & d(u) = 1-((1-H(u))^{-1}h_2)_z(\lambda,u)\ , \\ x(u) = (1-H(u))^{-1}h_0(\lambda,u) & y(u) = ((1-H(u))^{-1}h_0)_z(\lambda,u)\ , \end{array}$$

where $(f)_z(z,u) = f_z(z,u)$, the derivative with respect to variable z. Therefore $S(\lambda,u) = (d(u)x(u) - b(u)y(u))/\det(u)$ and $S_z(\lambda,u) = (-c(u)x(u) + a(u)y(u))/\det(u)$ with $\det(u) = a(u)d(u) - b(u)c(u)$.

5 DETERMINATION OF THE MARGINAL OUTPUT STREAM AND RESULTS

We have $\lambda_o(1-P(z)) = \lim_{u\to 1}(1-u)S(z,u)$, therefore, by identification $z=\lambda$ and using the results of the last section we obtain

$$\lambda_o = \frac{1}{1-P(\lambda)} \lim_{u\to 1} \frac{d(u)x(u)-b(u)y(u)}{\det(u)} .$$

Since the functions, $a(u)$, $b(u)$, $c(u)$, $d(u)$, $x(u)$ and $y(u)$ are continuous for $|u| < (p^2 + q^2)^{-1}$, and $\det(1) = 0$, we get, by Liouville,

$$\lambda_o = -\frac{d(1)x(1)-b(1)y(1)}{(1-P(\lambda))\det'(1)} ,$$

where $\det'(u)$ is the first derivative of function $\det(u)$ with respect to the variable u. Now we have an exact expression for λ_o, it remains to compute the result.

The expressions of $a(1)$, $b(1)$, *etc*, include application of operator $H(1)$, which is well defined with function $P(z)$, iterated on functions $h_0(z,1)$, $h_1(z,1)$ and $h_2(z,1)$:

$$\begin{cases} h_0(z,1) = ze^{-z} \\ h_1(z,1) = 1 + P(pz+\lambda) - ((1+P(\lambda))(1+z) + P'(\lambda)pz)e^{-z} \\ h_2(z,1) = z(p + qP(pz+\lambda) - (p+qP(\lambda))e^{-z}) . \end{cases}$$

The derivative of $\det(u)$ includes the first derivatives of functions $a(u)$, $b(u)$, *etc.* The derivative of $(1-H(u))^{-1}f(z,u)$ with respect to u is exactly $(1-H(u))^{-1}H'(u)(1-H(u))^{-1}f(z,u) + (1-H(u))^{-1}f_u(z,u)$, where $H'(u)$ is the derivative of $H(u)$ with respect to u. We have $H'(1)f(z) = H(1)f(z) + X(pz+\lambda)\Pi_2\sigma_2 f(z)$ and

$$\begin{cases} \frac{\partial}{\partial u} h_0(z,1) = h_0(z,1) \\ \frac{\partial}{\partial u} h_1(z,1) = h_1(z,1) + X(pz+\lambda) - (X(\lambda)(1+z) + X'(\lambda)pz)e^{-z} \\ \frac{\partial}{\partial u} h_2(z,1) = h_2(z,1) + qz(X(pz+\lambda) - X(\lambda)e^{-z}) . \end{cases}$$

This allow us to compute the numerical results which are illustrated by the following figures shown in appendix.

REFERENCE

[1] G. FAYOLLE, P. FLAJOLET, M. HOFRI, P. JACQUET [1985]. "Analysis of a Stack Algorithm for Random Multiple Access Communication", in Special Issue on Random-Access Communication, *IEEE Transactions on Information Theory* **IT-31**, March 1985, 244-254.

[2] G. FAYOLLE, P. FLAJOLET, M. HOFRI [1986]. "On a Functional Equation Arising in the Analysis of a Protocol for a Multi-Access Broadcast Channel", *Advances in Applied Probability* **18**, 1986, 441-472.

[3] J. L. MASSEY [1981]. "Collision Resolution Algorithms and Random-Access Communications", in *Multi-User Communication Systems*, G. Longo Editor, *CISM Courses and Lectures no.* **255**, Springer Verlag, Wien-New York, 1981, 73-137.

[4] B. S TSYBAKOV, N. VVEDENSKAYA [1980] "Random multiple access stack algorithms", *Probl. Inform. Transmission* **16**, 1980, 230-243.

[5] P. JACQUET, E. MERLE [1988]. "Analysis of a stack algorithm for random length packet communication," *IEEE Transaction on Information Theory* **IT-36**, March 1990.

APPENDIX

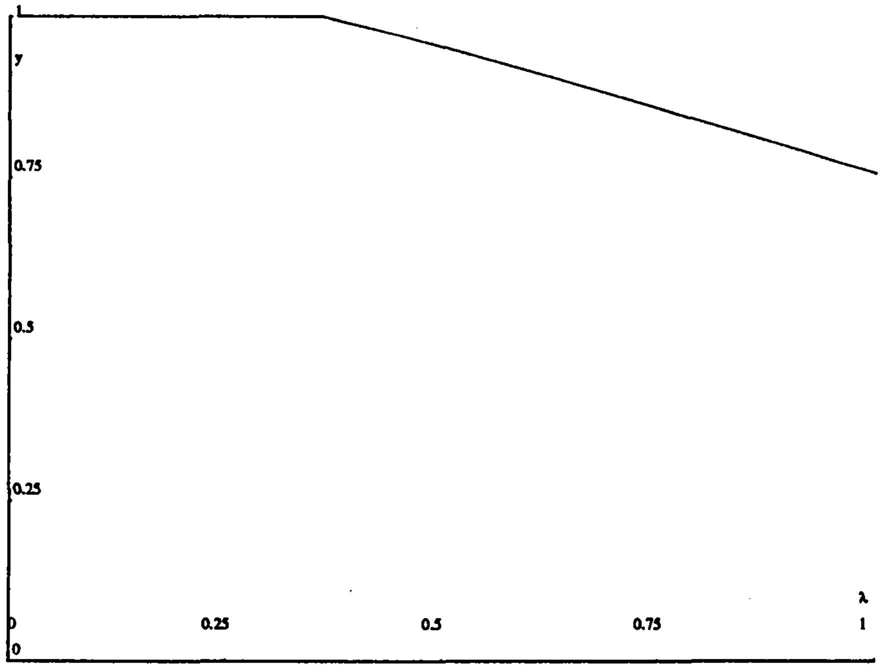

Figure 1: unconditional probability for a CRI to be finite ($P(\lambda)$) versus λ, between 0 and 1.

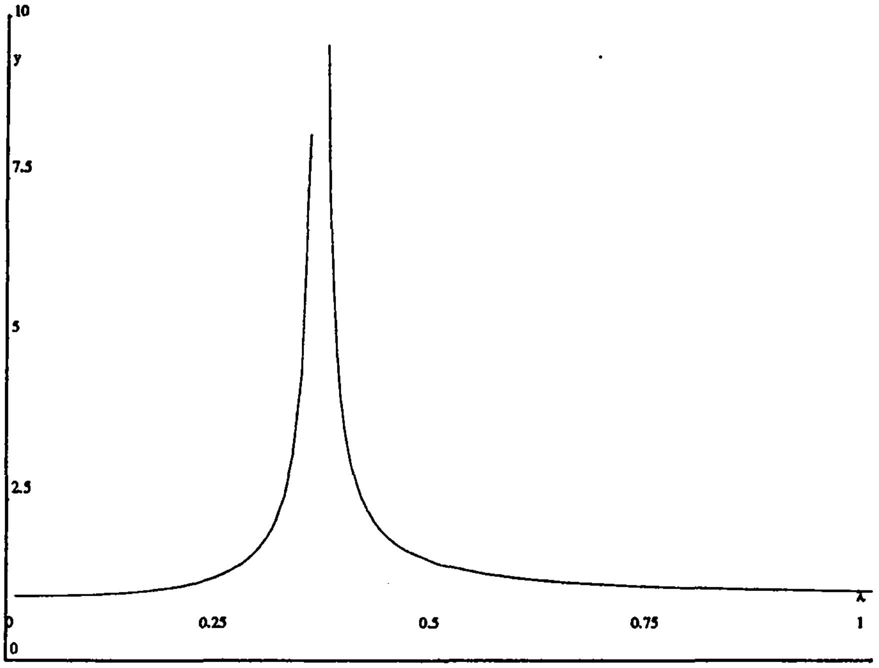

Figure 2: mean CRI length conditioned by the fact it is finite ($X(\lambda)/P(\lambda)$), versus λ, between 0 and 1.

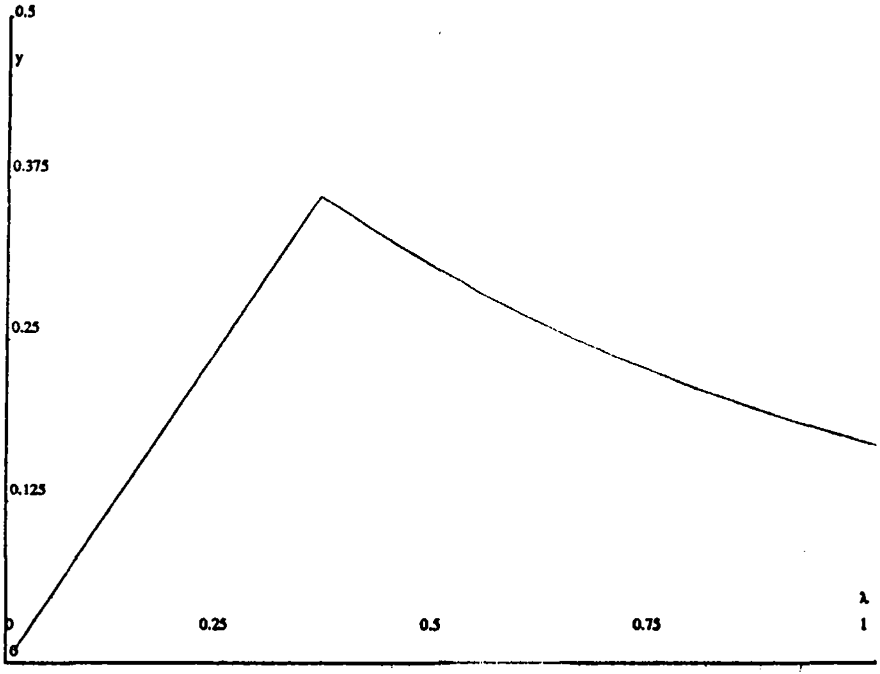

Figure 3: **marginal output stream** (λ_o), versus λ, **between 0 and 1.**

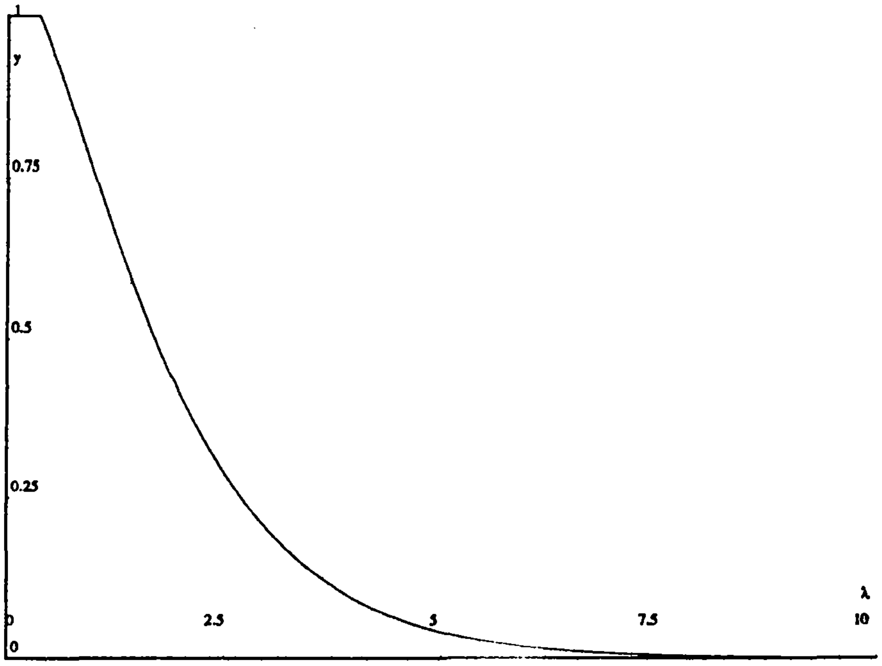

Figure 4: **unconditional probability for a** CRI to **be finite** $(P(\lambda))$ **versus** λ, **between 0 a** 10.

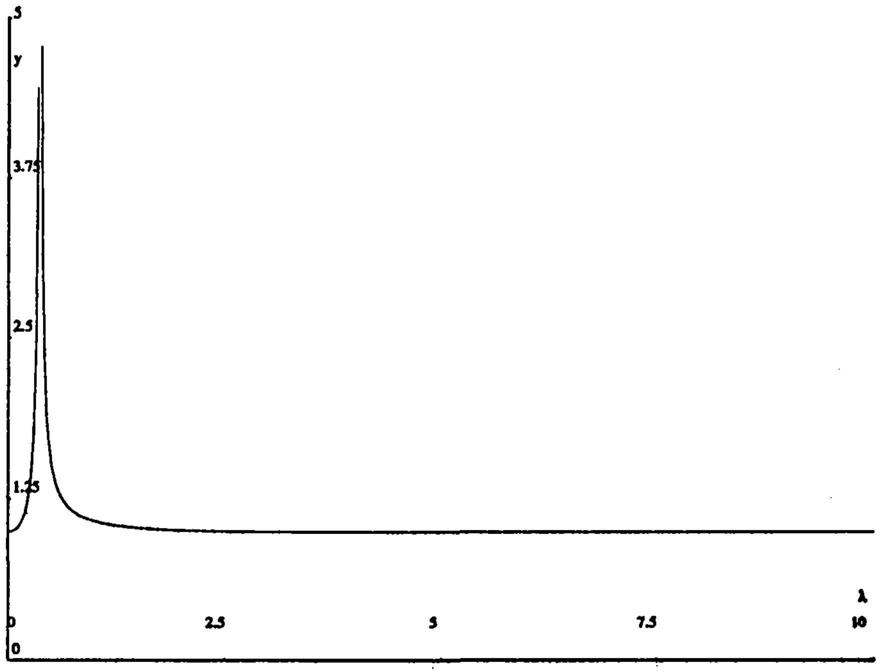

Figure 5: mean CRI length conditioned by the fact it is finite $(X(\lambda)/P(\lambda))$, **versus** λ, between 0 and 10.

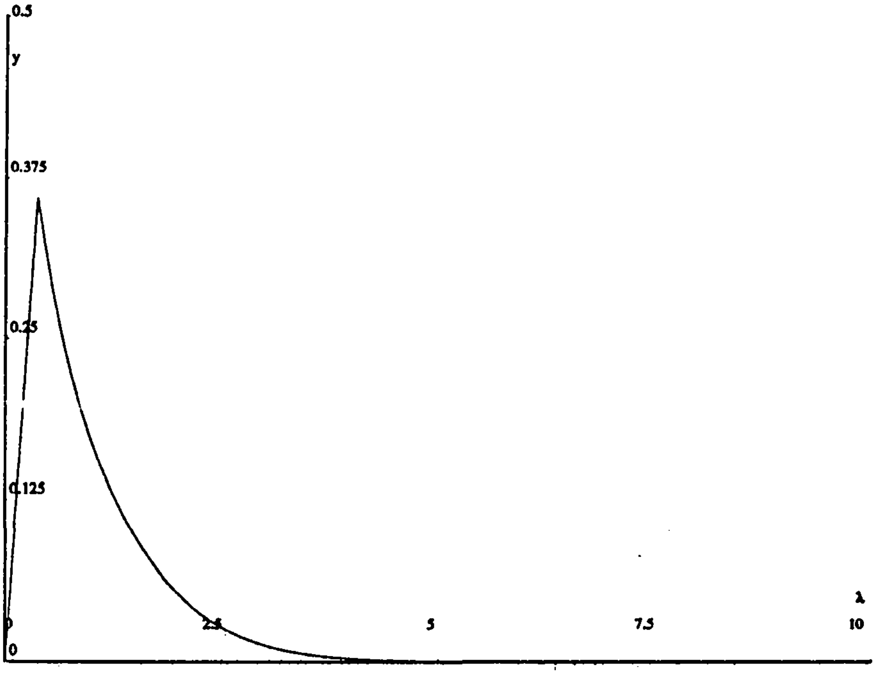

Figure 6: marginal output stream (λ_o), versus λ, between 0 and 10.

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique

ISSN 0249-6399