Decentralized Adaptive Flow Control of High Speed Connectionless Data Networks

by

Felisa J. Vázquez-Abad [1]

Department of Computer science and Operations Research, University of Montreal

and

Lorne G. Mason[2]

INRS-Telecommunications, University of Québec.

March 20, 1998

Abstract

We introduce a permit-based adaptive control scheme for regulating traffic admission in high speed connectionless data networks, such as the internet. Permits are awarded to potential customers arriving from outside and travel with them towards their destinations, where the permits are assigned to the local controllers. The controllers randomly distribute the permits among the entry gates at the nodes. Customers from outside are not allowed to enter the network unless there are permits available at the entrance node, thus the model is that of a closed queueing network if we model the dynamics of the permits. The goal is to find the permit distribution strategy that maximizes network performance subject to the restrictions of the network topology. A traffic balance approach is used to establish non-uniqueness of the optimal distribution probabilities for the decentralized operation. We exploit non-uniqueness introducing the concept of the automata *actions*, focusing on two strategies for the actions. For each strategy, a learning automaton is implemented at the controllers using the Kuhn-Tucker conditions for optimality. Our first learning algorithm converges weakly to a unique limit point, which is optimal, while the limit behaviour of our second learning algorithm may be suboptimal. We illustrate our results using computer simulations in order to compare the two strategies for the same network.

Key Words: Flow control, decentralized control, network optimization, adaptive control, learning automata, connectionless networks.

[1] The work of this author was partially supported by a grant from the Canadian Institute for Telecommunications Research under the NCE program of the Government of Canada, and partially supported by NSERC grant WFA 0139015 and FCAR-Québec grant 95-NC-1375.

[2] This work was supported by a grant from the Canadian Institute for Telecommunications Research under the NCE program of the Government of Canada.

§1. Introduction

Problems of congestion arise in communication and transportation networks when subjected to traffic overload unless adequate measures are put in place to control access to the network(s) in question. When congestion occurs, the effective throughput drops and delays become excessive. To mitigate such unwanted effects, flow controls have been employed and continue to be an active area of research as new and advanced networks are being deployed. In the case of low speed packet data communication networks, flow control has been exercized at a number of levels, refered to as hop level, entry-to exit level, access level and transport level. the different levels of flow control have different motivations and techniques for implementation. For an early survey of flow controls in conventional data networks see [7]. More recently, with the advent of high speed links such as arise in ATM networks and the high speed internet for example, the flow control problem is exacerbated due to the large delay bandwidth product, whereby significant amounts of traffic enters the network, before congestion can be detected at the source where a remedial action can be undertaken. This has resulted in the flow control problem being revisited by many telecommunication network researchers.

In this paper we address the problem of access flow control for high speed connectionless data networks, such as a high speed internet for example. While access flow controls have been implemented for virtual circuit based data networks, the same cannot be said for connectionless networks such as the celebrated internet. An early proposal for control of connectionless networks due to Davies [4], was never implemented, as it was found by simulation to be effective only in balanced traffic conditions. To avoid the shortcomings of the original approach, Mason and Gu [12] suggested an adaptive version of the isarithmic scheme where learning automata are employed to actively control the permit distribution, thereby making the scheme adaptive to changing traffic conditions, and avoiding the problems of permit starvation and permit excess associated with the original scheme under unbalanced traffic conditions. Various extensions of this adaptive isarithmic flow control architecture have been reported for high speed data and ATM networks in [3] [16], [10], [23] and [9].

The work presented here has applications in a high speed internet, where for example IP frames are independently routed over an underlying transport network such as SDH (Synchronous Digital Hierarchy). We consider a decentralized adaptive isarithmic control scheme for optimizing the performance in a high speed packet switching network. We shall now briefly explain these terms.

In a packet-switched data communications network, the data arrives at the network layer at the designated entry ports. Each "packet" consisist of a random number of bits (generally assumed to have a geometric distribution) and its header contains information about its origin and intended destination. Normally, packets that enter the network will be processed at the entry port of each visited node, where its origin and destination are determined from the header, as well as the next node to visit in its path towards destination. At each switch (node), the packet is placed in the outport queue leading to the next scheduled node. There are as many such queues as links from the given node. Routing is usually fixed and determined uniquely by the destination, as we shall assume here, although we have also applied our learning algorithms to admit dynamic routing [16]. If the server at the outgoing link is free, the packet enters service (transmission), which takes an amount of time proportional to its length. Inmediately afterwards, the packet is sent through a fiber optic cable to cover the (assumed) long distance until it reaches the entry port of the next node in its trip. The term "high speed" refers to this modeling of the links, where we have explicitly added the propagation delay through the cables, assuming that relatively long distances must be covered from one node to another. When a packet arrives at the entry port of its own destination, no further routing is required and the packet is released from the network layer towards the user layers.

Flow control refers to a strategy to control the incoming rate of arrivals into the network. If no flow control is present, then heavy traffic could, of course, cause bottlenecks and long end-to-end delays (total travel times). The word "isarithmic" has the greek roots "$\acute{\iota}\sigma o\nu$" (equal) and "$\alpha\rho\iota\theta\mu\acute{o}\varsigma$" (number). As its name indicates, this strategy for flow control is based on the idea of having always a limited number of packets in the network. A fixed number W, call the "window size" is specified and W "permits" are distrubuted to the different nodes in the network. Packets can only enter the network if there are permits available at the entrance node. Once the packets are released to their destination, the corresponding permits are routed toward the source nodes randomly, according to a distribution probability vector. The control parameters for this strategy are thus the window size and the distribution probabilities. A potential application is LAN interconnection, where the controllers are located at the gateway nodes and regulate the rate at which traffic enters the backbone network. The scheme optimizes the product of powers and achieves an appropriate compromise between throughput and delay, while being fair to users and Pareto efficient. The basic architecture was first proposed by [12] in the context of low speed packet networks, where propagation delay did not play a significant role. In [14] a performance model was developed, the question of optimal window size was studied by [2] and the effect of propagation delay was considered in [3], where a centralized control architecture was used. The present work considers a decentralized single chain architecture of the distribution probabilities for a high speed network, where the propagation delays are explicitly considered. Any decentralized scheme that dynamically adjusts the control variables would have to concurrently update the window size and the distribution probabilities in a hierarchical structure. For such a scheme to converge to the global optimal flow strategy, it is necesssary to study first the convergence of the distribution probabilities for fixed window size, which is the subject of this work.

Section 2 presents the mathematical formulation of the problem. We introduce the notation and a particular model is presented along with its solution. This tractable model is used later for validation of computer simulations, but our main results are robust and apply to more complex models. Under isarithmic flow control [4], the process is a closed network [18] and the optimization goal is the maximization of the steady state average product of powers, for the fixed control process. We show that the problem is equivalent to the optimization problem for a centralized control strategy. The analytical results of this particular model are used in Section 6 to assess the performance of our algorithms in the simulations performed.

We explore in section 3 the relationship between the optimal throughputs and the corresponding decentralized control. This section is general and our results are independent of the model presented in Section 2. We show that even if there is a unique set of optimal throughputs, there is non-uniqueness of the decentralized optimal distribution probabilities. We then propose several schemes introducing the concept of the actions at each controller, focusing on two different action strategies: the one we call "$N-1$-action scheme" and the "2-action scheme". These results are new and represent the first contribution of the paper. Since they are stated in a general framework, they can be aplied to similar permit-routing problems outside the telecommunications model studied here.

In Section 4 we describe the updating algorithms for the distribution probabilities as learning algorithms. Our scheme is decentralized: a controller at each node reads relevant information from the permits that it receives and sends it to other controllers. It then calls its automaton to update the distribution probabilities and decides to which source node the permits are to be routed. We use the Kuhn-Tucker conditions for optimality in order to update the controls at the different automata. The proof of convergence of the algorithms is based on the weak convergence method of [8] and is included in the appendix. This establishes asymptotic optimality of the $N-1$-action strategy and allows us to identify the conditions for the 2-action strategy to be asymptotically optimal, which are not always satisfied. Before [23], no theoretical proof of convergence was available for this type of algorithms, and we apply this technique to identify the behaviour of the not always optimal 2-action scheme. This is an important contribution of our paper, yielding the basis for the construction of the learning algorithms in practical cases.

The updating scheme is quite general and can be implemented in other types of models. The feedback functions used for the updating require estimation of certain derivatives, which are in general unknown. In our simulations, reported in Section 5, we implemented an approximation of these derivatives following [3], based on the tractable model of section 2 operating at the (known) optimal window size. We verified that the algorithms converge to the predicted limits. We believe it is interesting to note that in the convergence proofs the propagation delays are explicitly incorporated, and our simulation results indeed show that these do not alter the asymptotic behaviour of the algorithms, contrary to what many practicioners belive (in particular dealing with Satellite communciations).

§2. Problem Formulation

In this section we introduce notation and present a tractable model where we can explicitly write the stationary distribution of the process. We then proceed to establish that given a decentralized model, there exists a unique equivalent central controller model, in the sense that the stationary distribution of the process is identical. The solution of the corresponding central optimization problem was given in [3] and determines uniquely the optimal window size and throughputs of the decentralized model. However, as we shall see in Sextion 3, even if the optimal throughputs are unique, the distribution probabilities are not.

The Closed Network Model and Notation. Throughout the paper we shall be using the notation introduced here, which has been condensed for easy reference in Appendix C.

There are N nodes in the network, connected by trunks, which are directed links. Whenever there is a link between nodes i and j, there are two trunks with same capacity $C_{i,j}$ and distance $D_{i,j}$. Routing within the network is random: the probability that a node i routes a packet with destination d to node j is given by $r^d_{i,j}$, and $\sum_j r^d_{i,j} = 1$, for $i \neq d$. At each node o we model a *source queue*. The W permits are initially allocated among the N source queues. If an arrival from outside occurs at node o when the source queue is empty, the incoming arrival is lost. Otherwise, the packet is awarded a permit and the packet-permit combination enters the network. Arrivals at node o of packets with destination d follow a Poisson process with rate $\Lambda_{o,d}$. The total size of the packet, permit and information header is exponentially distributed with mean α for all origin destination pairs. Upon an arrival at a node i, the identity of the destination d is read from the header. If the packet has arrived to its destination, the permit is sent to the controller at node d. Otherwise, a node j is chosen with probability $r^d_{i,j}$ and the packet-permit combination is sent to the trunk (i,j). After service completion at a trunk (i,j), the permit-packet must travel the distance $D_{i,j}$ before arriving at node j. We thus model each trunk by two tandem queues, one in which the service time is exponentially distributed with rate $C_{i,j}/\alpha$ and one (representing the propagation) in which the delay is deterministic and equal to $p_{i,j} = D_{i,j}/c$, where c is the speed of light. The function of the controller at each node d is to assign a source queue to which the permit is sent. We assume that permits travel along a higher level network from any controller d to a source queue j, and a delay p^d_j is involved in this transmission. During the time that a permit is travelling through the controller loops towards its source queue, it is not available for packets. Let π^d_i denote the probability that controller d distributes a permit to source queue i. Figure 1 shows schematically the dynamics of such a system.

Consider the process under fixed values of π^c_i and W. The network is described as a closed network with respect to the permits. Following the notation in [3], we call

$$\begin{cases} F_s: & \text{the set of source queues of permits,} \\ F_a: & \text{the set of processor queues} \\ F_p: & \text{the set of trunk delay queues with propagation delay } p_{j,l}, \\ F_c: & \text{the set of queues for the delays at the controllers with propagation delay } p^c_j \end{cases}$$

where we have introduced the latter "queues" to model the fixed time trip of permits from any destination to the source nodes $j \in F_c$ (refer to Figure 1). Permits at the source queues $i \in F_s$ obviously have a service every time that an arrival form outside demands access to the network, provided this queue is not empty, thus its service rate is $\mu_i = \Lambda_i = \sum_d \Lambda_{id}$

for $i \in F_s$. Processor queues have service rate $\mu_{j,l} = C_{j,l}/\alpha$ for $(j,l) \in F_a$. To model the propagation delay, we introduce deterministic service queues $(j,l) \in F_p$ with an infinite number of servers, which will accurately describe the passage through a fiber optics, and similarly, for each source queue $j \in F_c$, we introduce an infinite server, deterministic service time queue to respresent the travel time p^c_j from any destination node towards node j.

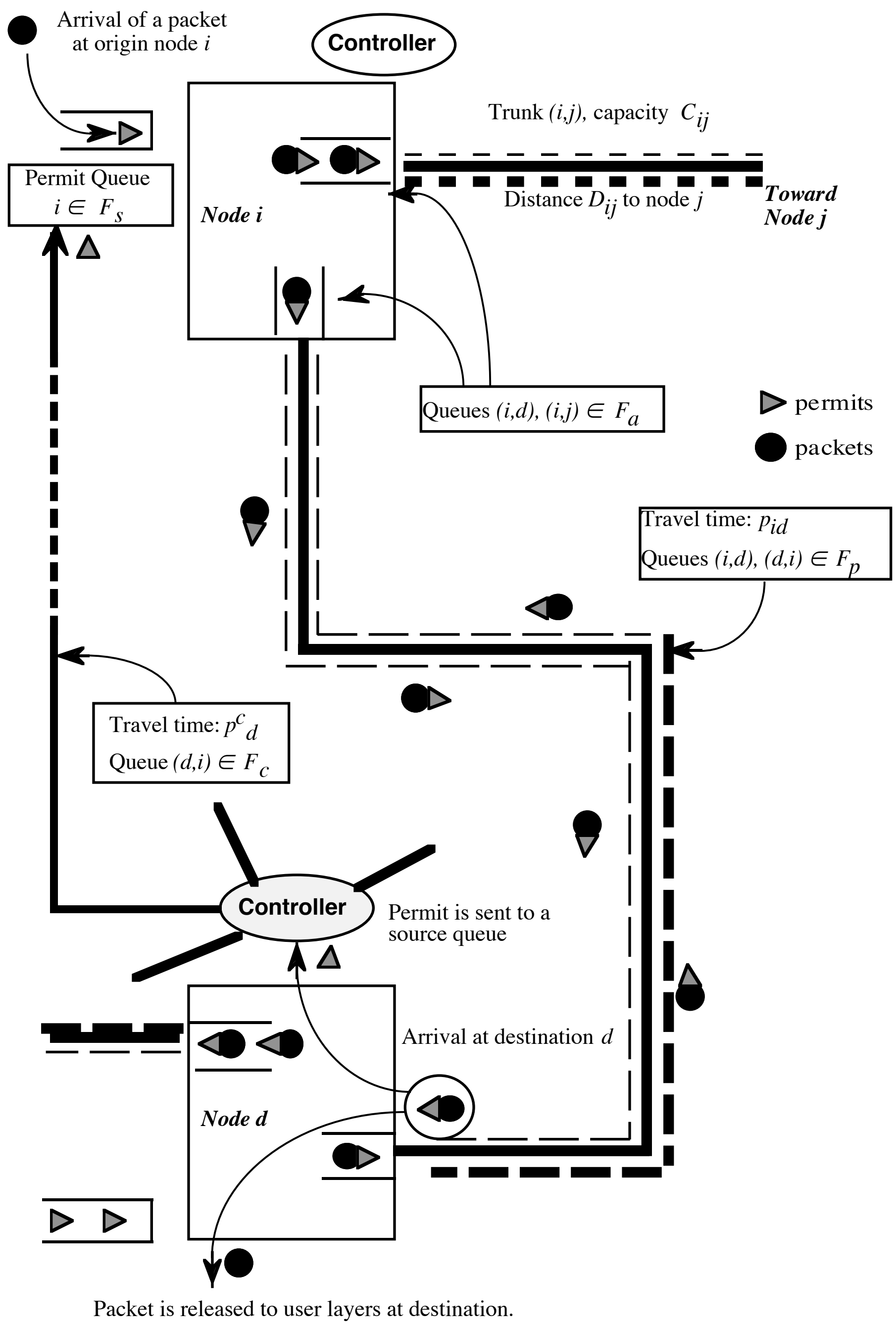

Figure 1: Representation of the Isarithmic Flow Control.

Call $s_i, i \in F_s$, $c_j, j \in F_c$ and $f_{j,l}$ the relative number of visits of a permit to source queue i, to controller queue j and to the trunk (j,l) respectively. Then the utilization factors for each queue are given by:

$$\begin{aligned}
\rho_i^s &= \frac{s_i}{\mu_i},\ i \in F_s, \\
\rho_{j,l}^a &= \frac{f_{j,l}}{\mu_{j,l}},\ (j,l) \in F_a, \\
\rho_{j,l}^p &= f_{j,l} p_{j,l},\ (j,l) \in F_p, \\
\rho_j^c &= c_j p_j^c,\ j \in F_c
\end{aligned}$$

The Product Form Solution.

In a closed network (a network having a fixed number of customers traveling through) the service distributions and the topology determine completely the arrival processes at any given node, and it is customary to use the notation $\cdot/M/1$ to represent nodes such as the source nodes, with one exponential server. The BCMP ("Basket, Chandy, Muntz and Palacios") theorem (see [5], and [6]) states the conditions for a closed network to posses a "product form" solution. The latter refers to the distribution that would result from having independent queues instead of a network. For our model, the nodes are either $\cdot/M/1$ or $\cdot/D/\infty$ and one can apply the BCMP theorem to obtain the steady state probability of the system as:

$$P(x) = \frac{1}{G(W)} \prod_{i \in F_s} (\rho_i^s)^{x_i} \prod_{(j,l) \in F_a} (\rho_{j,l}^a)^{x_{j,l}} \prod_{(j,l) \in F_p} \frac{(\rho_{j,l}^p)^{x_{j,l}}}{x_{j,l}!} \prod_{j \in F_c} \frac{(\rho_j^c)^{x_j}}{x_j!} \tag{1}$$

where $x_i, i \in F_s$ is the number of permits in the source queue i and analogously for the other queues. We must have for the closed network that

$$\sum_{i \in F_s} x_i + \sum_{(j,l) \in F_a} x_{j,l} + \sum_{(j,l) \in F_p} x_{j,l} + \sum_{j \in F_c} x_j = W$$

Remark: Product form approximations can be extended to more realistic models. As will be seen later, the adaptive scheme proposed in section 5 does not depend on this representation of the invariant measure, and our formulas are justified for more general models.

The relative number of visits to source queue i in steady state is $s_i = \sum_{j \in F_c} c_j \pi_i^j$. Call s_i^d the fraction of permits that carry a packet with origin i and destination d. Since there is only one source queue per node, then we have $s_i^d = s_i \frac{\Lambda_{i,d}}{\Lambda_i}$. For packets with origin

i and destination d, call

$$\begin{aligned} \lambda_{i,d} &: \text{stationary average throughput} \\ T_{i,d} &: \text{stationary average delay} \\ y_{i,j}^d &: \text{average number of visits to node } j \text{ for an } (i,d) \text{ packet} \\ t_{j,l} &: \text{average delay at queue } (i,j) \in F_a \end{aligned}$$

The throughputs and delays can be readily obtained using the methods in [6], as was done in [3]:

$$\lambda_{id} = s_i^d \frac{G(W-1)}{G(W)} \tag{2}$$

$$T_{id} = \sum_{(j,l)\in F_a} t_{j,l} y_{i,j}^d r_{j,l}^d + \sum_{(j,l)\in F_p} t_{j,l} y_{i,j}^d r_{j,l}^d \tag{3}$$

The Optimization Problem.

It is shown in [14] that the product of powers is a performance criterion that satisfies Nash's axioms of fairness and is Pareto optimal for a simple network model. They conjecture that the result is true for other cases as well. Following [3], the optimization problem is stated as:

$$\max_{\pi_i^d, W} P(\pi, W) = \max_{\pi_i^d, W} \prod_{(i,d):\Lambda_{i,d}>0} \frac{\lambda_{i,d}}{T_{i,d}}$$

$$\text{subject to:} \quad \sum_{(i,d):\Lambda_{i,d}>0} \lambda_{i,d} y_{i,j}^d r_{j,l}^d \leq \frac{C_{j,l}}{\alpha} \quad \text{for all } (j,l) \in F_a$$

and $\sum_{i=1}^N \pi_i^d = 1$ for all $d = 1, \ldots, N$, $W > 0; \pi_i^d \geq 0$ for all $i, d = 1, \ldots, N$.

We shall now present the arguments that establish the decentralized optimization problem above in terms of a central controller problem, which in turn simplifies the calculation of the solution. As mentioned before, the importance of this calculation is to assess the behavior of the algorithms in section 5. For more general models it is practically impossible to evaluate an analytical solution, and that is the main motivation for the construction of good adaptive schemes that require only measured data in order to search for the optimal operating point. Let $\sum c_j = 1$ in the definition of the relative number of visits. We aggregate the probability (1) in the controller queues by adding up over $x_j, j \in F_c$. Call $x^c = \sum_{j\in F_c} x_j$ the total number of permits travelling in such queues. An extension of Newton's binomial for any number R of terms is given by:

$$\left(\sum_{r=1}^R \rho_r\right)^x = \sum_{x_1+\ldots+x_R=x} x! \prod_{r=1}^R \left(\frac{\rho_r^{x_r}}{x_r!}\right)$$

and therefore, direct application of this formula for the aggregate controller queue gives: $\sum \prod_{j\in F_c} \frac{(\rho_j^c)^{x_j}}{x_j!} = \frac{(\rho^c)^x}{x!}$, where the sum is over all $x_1, \ldots, x_N$ such that $\sum x_j = x$ and: $\rho^c = \sum_{j\in F_c} \rho_j = \sum_{j\in F_c} c_j p_j^c$. In the aggregate system, the stationary probability (1) is equivalent to:

$$P(x) = \frac{1}{G(W)} \prod_{i\in F_s} (\rho_i^s)^{x_i} \prod_{(j,l)\in F_a} (\rho_{j,l}^a)^{x_{j,l}} \prod_{(j,l)\in F_p} \frac{(\rho_{j,l}^p)^{x_{j,l}}}{x_{j,l}!} \frac{(\rho^c)^{\bar{x}}}{\bar{x}!} \tag{4}$$

where now $\bar{x}$ is the number of permits travelling in any of the controller queues, so that $\sum_{i\in F_s} x_i + \sum_{(j,l)\in F_a} x_{j,l} + \sum_{(j,l)\in F_p} x_{j,l} + \bar{x} = W$. Equation (4) corresponds to an equivalent centralized operation if we interpret s_i as the probability of sending a permit to source queue i and ρ^c as the corresponding delay from the controller. Therefore the centralized and the decentralized operation of the network are equivalent in distribution, in the steady state. The normalizing function $G(W)$ can be evaluated using the same computations as in the central control case, which obviously cuts down considerably the computational effort, especially for large networks.

Remark: It is clear that in particular, if $p_j^c = p^c$ is a constant for all controller queues, then since $\sum c_j = 1$, we would have $\rho^c = p^c$. This choice involves delaying some of the permits that are assigned to source queues in order for all the controller queues to have the same delay and we used this simple implementation for our simulations.

Once the expression of $P(x)$ has been rendered to the form (4) we can use the results from [3] in order to reformulate the problem as the maximization of the product of powers with repect to an aggregate throughput $\lambda_i = \sum_{d=1}^N \lambda_{i,d}$:

$$\max_{\lambda_i} P(\lambda_i) = \max_{\lambda_i} \prod_{(i,d):\Lambda_{i,d}>0} \lambda_i \frac{\Lambda_{i,d}}{\Lambda_i} \frac{1}{T_{i,d}} \tag{5.1}$$

subject to:

$$\sum_{(i,d):\Lambda_{i,d}>0} \lambda_i \frac{\Lambda_{i,d}}{\Lambda_i} y_{i,j}^d r_{j,l}^d \leq \frac{C_{j,l}}{\alpha} \quad \text{for all } (j,l) \in F_a \tag{5.2}$$

and $0 \leq \lambda_i \leq \Lambda_i$, where $\Lambda_i = \sum_{d=1}^N \Lambda_{i,d}$. This formulation views the aggregate throughputs as the flow control variables. Given the optimal values λ_i^*, the optimal throughputs are $\lambda_{i,d}^* = \lambda_i^* \frac{\Lambda_{i,d}}{\Lambda_i}$. It is shown in [3] that, given λ_i^*, there is a unique corresponding value of W^*. If the window size $W \neq W^*$ is fixed and the optimization is carried only over the distribution probabilities, the transformation of the problem is still valid with (2) and the corresponding $\lambda_{i,d}^*$ determine uniquely the distribution probabilities that maximize the product of powers, given that window size. In this work we focus on the decentralization of the control of the distribution probabilities. The following section explores the relationship between the optimal throughputs $\lambda_{i,d}^*$ of the problem and the corresponding optimal decentralized controls $(\pi_i^c)^*$.

§3. Non-Uniqueness of the Optimal Control

This section explores a design problem that is general for other network models and does not require a product form solution. We focus here on a general network under isarithmic flow control where we want to maximize the product of powers by chosing the optimal throughputs, when the window size is fixed. These throughputs are to be achieved by appropriately assigning the values of the distribution probabilities π_i^c. Based on a mean flow approach, an equilibrium argument for the stationary quantities shows that at each node i the output permit rate must equal the input permit rate, or:

$$\sum_{d:\Lambda_{i,d}>0} \lambda_{i,d} = \sum_{c=1}^{N} \sum_{o:\Lambda_{o,c}>0} \lambda_{o,c}\pi_i^c \quad \text{for } i = 1, \ldots, N \tag{6}$$

where $\sum_{i=1}^{N} \pi_i^c = 1 \quad$ for $c = 1, \ldots, N$. The left hand side of (6) corresponds to the output rate at node i, and the right hand side is the input from the controllers to the source queue i. If the network operates at any fixed value of $\{\pi_i^c\}$, then the corresponding stationary throughputs $\lambda_{i,d}$ of (2) fulfill (6). We are interested, however, in the *optimal* values of the parameters. Suppose that the $\lambda_{i,d}$ of (6) are the optimal throughputs. Since there are $2N$ equations in N^2 unknowns, then it is clear that the optimal control values for π_i^c are not unique. Non-uniqueness of the decentralized distribution probabilities can be exploited through the concept of the *controller actions*, that is, we are free to restrict the number of actions at a particular controller. This reduces the number of unknown variables in the equilibrium equations. However, the choice of those actions has to be made in such a way that the solution exists: restricting the actions may cause an input to a particular source node to decrease below the optimal output level, in which case (6) would not be satisfied for $\lambda_{i,d}^*$ and thus the particular action scheme would not be able to realize the optimal throughputs.

In this section we address the questions of existence and uniqueness of the optimal solution in the decentralized operation, that is, we study equations (6) when $\lambda_{i,d}$ are the optimal throuputs of problem (5) and the distribution probabilities π_i^c are regarded as the unknowns, although we shall omit the superscripts in $\lambda_{i,d}^*$. The framework that we use here is extremely simple and provides an approach to answer these questions that is undoubtedly simpler than the analysis of the steady state distribution for each of the possible action models.

Define $C_i = \sum_{o=1}^{N} \lambda_{o,i}$, and $\lambda_i = \sum_{d=1}^{N} \lambda_{i,d}$. For each c, let $A_c(\cdot)$ be the indicator function of the actions taken by controller c, that is, $A_c(i) = 1$ if π_i^c is not identically zero, and $A_c(i) = 0$ otherwise. Then the equilibrium equations require:

$$\sum_{c=1}^{N} C_c A_c(i)\pi_i^c = \lambda_i \quad \text{for } i = 1, \ldots, N \tag{7}$$

$$\sum_{i=1}^{N} A_c(i)\pi_i^c = 1 \quad \text{for } c = 1, \ldots, N \tag{8}$$

which in matrix form can be written as $Ex = b$, with $x_c(i) = A_c(i)C_c\pi_i^c$. The form of the equilibrium matrix E in general is given by:

$$E = \begin{pmatrix} E_1 & E_1 & \dots & E_1 \\ P_1 & P_2 & \dots & P_N \end{pmatrix}$$

where E_1 is the $N \times N$ identity matrix, P_n is the $N \times N$ matrix with rows of zeroes except for the n^{th} row, which is the vector $(1, \dots, 1)$ and $b' = (\lambda_i, \dots, \lambda_N, C_1, \dots, C_N)$. Notice that adding the rows of the upper part of E yields the same result as adding the rows of its lower part. The $N^2 \times N^2$ action matrix A has the effect of cancelling some components of the variable x. We shall focus on two important cases for the action matrix: the $(N-1)$-action scheme, where we establish, for each controller c, $\pi_c^c \equiv 0$ while all other components can be positive, and the two-action scheme, where only two components of π_i^c (with $i \neq c$) are allowed to take positive values.

The results of this section are independent of the learning algorithms used to update the distribution probabilities $\pi^c(t)$, described later. However, in order to understand why we focus on these particular action schemes we anticipate some characteristics of the updating methods. An automaton at controller c will respond to arrivals of permits that carry (i, c) packets to update the probability π_i^c of distributing the permit back to the origin i. Since there are no packets with same origin and destination, there are $(N-1)$ possible values of i. It has been shown in [11] that the rate of learning depends on the number of actions at each automaton. Therefore the two-action scheme may yield faster convergence under the appropriate conditions.

Under the n-action scheme strategy, for $2 \leq n < N$, even if the solution π exists, it is not unique. Indeed, the linear system has $2N$ equations in $N \times n$ variables. Non-uniqueness follows whenever $n \geq 2$. For $n = 2$ the matrix EA has only $2N$ columns which are not identically zeroes, and the remaining columns are exactly those of the original equilibrium matrix E. The determinant of the resulting $2N \times 2N$ vanishes, which yields non-uniqueness.

The linear system always possesses a solution in the $(N-1)$-action case, as follows by direct substitution of

$$\pi_i^c = \frac{\lambda_{i,c}}{C_c} \tag{9}$$

However, not all action matrices give rise to well posed problems. As a trivial example, consider a three node network with two actions, where all controllers send permits only to nodes 1 and 2. Then $A_c(0) = 0$ for all c and it follows from (6) that the first equation requires $0 = \lambda_0$. Recall that according to our notation, λ_0 is not the steady state total throughput out of origin 0, which would be zero under this action scheme, but the optimal throughput of problem (5) that we wish to achieve. We have developped in the Appendix the basic method for modeling the problem of the n-action scheme in general via a formulation akin to transportation models. This method has been further studied in [17].

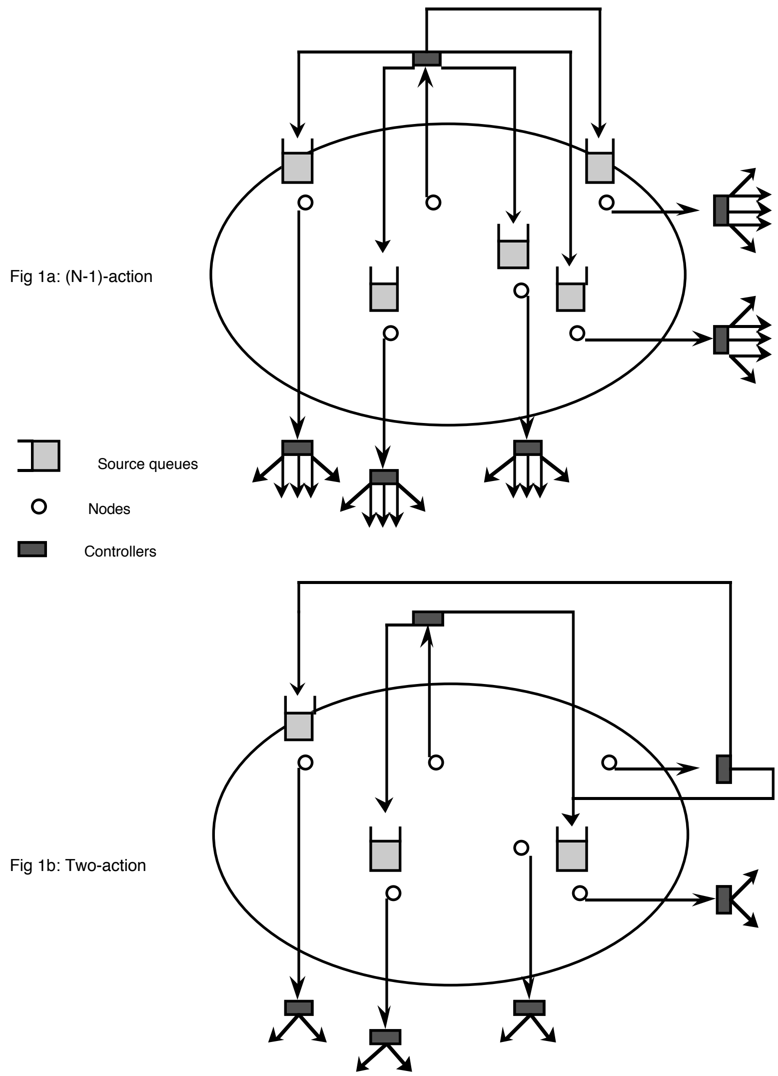

Figure 2: Representation of (a) the $(N-1)$ and (b) the two-action schemes.

§4. The Adaptive Control Algorithms

We shall now construct the adaptive control algorithms for the permit distribution probabilities π_i^c for fixed, arbitrary window size. From the optimization problem (5), the throughputs λ_i^* that maximize the product of powers satisfy $\frac{\partial}{\partial \lambda_i} P(\lambda^*) = 0$, which follows from the Kuhn-Tucker conditions for optimality, as in [3]. A simple calculation yields:

$$\frac{\partial}{\partial\lambda_i}P(\lambda) = P(\lambda)\sum_{d:\Lambda_{i,d}>0}\frac{1}{\lambda_i} - P(\lambda)\sum_{(o,d):\Lambda_{o,d}>0}\frac{1}{T_{o,d}}\frac{\partial T_{o,d}}{\partial\lambda_i}$$

so that the condition for optimality is an equilibrium condition of the form:

$$H_i(\lambda^*) = H_j(\lambda^*) \quad \text{for all } i, j = 1, \ldots, N \tag{10}$$

where the H function is given by:

$$H_i(\lambda) = \sum_{(o,d):\Lambda_{o,d}>0}\frac{\lambda_i}{T_{o,d}}\frac{\partial T_{o,d}}{\partial\lambda_i} = (N-1) - \frac{\lambda_i}{P(\lambda)}\frac{\partial P(\lambda)}{\partial\lambda_i}$$

We assume that the network is operating under heavy traffic conditions, that is, $\Lambda_i > \lambda_i^*$ for all i. The case where $\Lambda_i < \lambda_i^*$ for all i represents light traffic and does not require any flow control. The mixed case, where some nodes are lightly loaded and others are heavily loaded, was treated in [3] using a central controller. The decentralization presented here can be extended to cover the mixed traffic cases to yield a more robust algorithm, but we omit the details for lack of space.

The estimation of derivatives using measured data gathered from the system is a hard problem in the present context, but many methods have been proposed (see [20] and references therein) which could be applied to estimate the H function for more general network models using only pathwise information. Such methods are amenable to the distribution of the computation of the global quantities via local sensitivities. The general condition required for the estimation (see appendix) is that the stationary average of the feedback function for fixed control values be unbiased for the sensitivity functions H. In order to simplify the present analysis and focus on the construction of the algorithms themselves, we have chosen an approximation for the model of Section 2 that requires only the estimation of the steady state average delays $T_{o,d}$ and throughputs $\lambda_{o,d}$. We introduce an information vector $X_i(t)$ at each node i at time t of the simulation, which is defined as a vector with components $\{T^i_{o,d}(t), TB^i_{o,d}(t); o, d = 1, \ldots, N\}$. Its values are calculated as follows: at the time t of arrival of an (o, d) packet to its destination node d the controller d evaluates the time elapsed since the start of the packet at o and updates $T^d_{o,d}(t)$ as a moving average. Analogously, it computes the time elapsed since the previous (o, d) packet arrived at d and updates the time between packets $TB^d_{o,d}(t)$ also by a moving average. At this time the controller sends the updated entries $T^d_{o,d}$ and $TB^d_{o,d}$ for that origin o to the other controllers at nodes $i \neq o$. We assume that the transmission of information is performed in a higher level network and that it arrives at each controller $i \neq d$ after a possibly random delay.

Upon arrival of the information $T^d_{o,d}$ and $TB^d_{o,d}$ at time t, controller i updates $T^i_{o,d}(t)$ and $TB^i_{o,d}(t)$ by setting them equal to the values just received. Since the entries of the information vector $X_c(t)$ are constructed as moving averages, this vector converges $a.s$ to a limit, provided that the delay of the information is uniformly integrable. In particular, for the fixed parameter process with W and π_i^c fixed at constant values, the moving averages converge to the stationary averages and thus all the vectors $X_c(t)$ possess the same limit:

$$X(\pi) = \lim_{t\to\infty} X_c(t) = \left\{ T_{o,d}, \frac{1}{\lambda_{o,d}}; o, d = 1, \ldots, N \right\} \tag{11}$$

Remark: Naturally, this implementation requires an intense exchange in data between nodes in the network. An alternative approach is given by the following scheme. Every destination node d computes its moving averages $T_{o,d}^d(t), TB_{o,d}^d(t)$ as packets arrive. From time to time, and not necessarily at every epoch of arrival of a packet to its destination, the nodes comunicate to their neighbors the latest estimates. A loopless cascade or information "tree" can be defined so that a node is in charge of sending periodically its latest update on various $T_{o,d}^i$, inlcuding $i = d$, to its neighboring nodes, so that eventually all nodes have updated their local information vector in this fashion. Such a scheme can be constructed so that equation (11) still holds, but it would obviously minimize the amount of exchange of information required. Since our results depend on the limiting behaviour, we focus without loss of generality on the scheme mentioned above, which simplifies the notation.

The nodes can calculate the H functions with the most recent available information and use these to update their parameters, as shown below. We shall see in section 6 that the delays in information broadcasting are negligible and do not affect the performance of the algorithms.

The $(N-1)$-action Automata. When an automaton d is called from the controller at time t, after receiving a permit with origin o, the corresponding reward function $b_o^d(t)$ is evaluated as:

$$b_o^d(t) = 1 - \frac{\arctan H_o(X_d(t))}{\pi/2}$$

and the probabilities are updated according to an $L_{R\text{-}I}$ learning automaton:

$$\pi_i^d(t) = \pi_i^d(t^-) + \epsilon\,(\delta_{io} - \pi_i^d(t^-))\, b_o^d(t) \quad i = 1 \ldots, N \tag{12}$$

where ϵ is a gain parameter, and δ_{io} is the Krönecker delta.

Remark: The idea behind the L_{R-I} learning automata model is to adjust the parameter according to a reward function that reflects the feedback from the actions of the controller. If π_i^c increases, then H_i increases. Since controller d updates all its components π_i^d according to the measured feedbacks, it is not hard to visualize that (12) tries to reach an equilibrium between the H_o functions. For more details on this type of scheme the reader is referred to [15], [19], [23] and references therein.

Call $\{\tau_n^d, n > 0\}$ the sequence of updating epochs for $\pi^d(t)$ as it follows the recursion (12). Let $F_i^d(\tau_n^d) = E\{b_i^d(\tau_n^d) \mid \text{permit carries an } (i,d) \text{ packet}, \pi(\tau_n^c)\}$, and P_i^d the probability that a permit carries an (i,d) packet. Then, from (12):

$$E\{\pi_i^d(\tau_{n+1}^d) \mid \pi_i^c(\tau_n^d)\} = \pi_i^d(\tau_n) + \epsilon \left[F_i^d(\tau_n^d) P_i^d - \pi_i^d(\tau_n^d) \sum_{o=1}^{N} F_o^d(\tau_n^d) P_o^d \right] \tag{13}$$

The basic proof of convergence of our algorithms is in Appendix A. It is shown there that the limiting control processes related to the distribution probabilities approach a limiting ODE. The stable points of the limiting ODE coincide with the fixed points of the conditional expectations in (13), that is, the values of π_i^c such that the expression in square brackets vanishes. The limit point of (13) is unique [23] and corresponds to the optimal control values (9): $(\pi_i^c)^* = \frac{\lambda_{o,c}^*}{C_c^*}$. Therefore, all initial conditions drive the control variables to this limit.

The Two Action Automata. Each controller c will only send permits to the source nodes i with $A_c(i) = 1$. However, it may still receive packet-permit combinations fom another origin, say o, with $A_c(0) = 0$. The usual implementation requires the learning automaton to respond to the feedback from the system as a result of its action, updating the corresponding probabilities. Here we can no longer implement this update procedure because the reception of a packet-permit combination with origin o cannot be assigned directly as the impact of the two actions, which exclude source o. Therefore, a different feedback function has to be calculated that accurately reflects the effect of the system's performance in terms of the actions defined at each controller. We describe such a scheme as follows. When a permit arrives at controller d with an (o,d) packet, the controller updates $X_d(t)$ and sends the information as described before. If $A_d(o) = 0$, then the permit is routed to the source queues according to the current values of π_i^d. If $A_d(o) = 1$ then the H function is evaluated and the parameters are updated by:

$$\pi_i^d(t) = \pi_i^d(t^-) + \epsilon\, \pi_i^d(t^-) \left[b_i^d(t) - \sum_{o=1}^{N} b_o^d(t) A_d(o) \pi_o^d(t^-) \right] \quad \text{for } i\colon A_d(i) = 1 \tag{14}$$

where $b_i^d(t)$ is defined as before. In the case of a two-action scheme, fix the controller d and call $o(1)$ and $o(2)$ the two source nodes to which d sends permits, so that $\pi_{o(1)}^d + \pi_{o(2)}^d = 1$. Then the updating equation (14) becomes:

$$\pi_{o(1)}^d(t) = \pi_{o(1)}^d(t^-) + \epsilon\, \pi_{o(1)}^d(t^-)(1 - \pi_{o(1)}^d(t^-)) \left[\frac{\arctan H_{o(2)}(X_d(t)) - \arctan H_{o(1)}(X_d(t))}{\pi/2} \right]$$

which has the form of a "shortest path" algorithm if we view the H function as the cost associated with the permit path from the controller toward the source nodes. The idea behind this type of algorithm is the following: if $\pi^d_{o(1)}$ increases then the stationary average throughtput $\lambda_{o(1)}$ increases as well as $H_{o(1)}(X)$. If this increased value is larger than $H_{o(2)}(X)$ then $\pi^d_{o(1)}$ will decrease in the next update, as will $H_{o(1)}(X)$. The result is that the algorithm will seek for an equilibrium of the H function values at $o(1)$ and $o(2)$. This pairwise decentralized effort will globally seek the optimal equilibrium, provided that the two-action scheme is one that admits the optimal solution, as mentioned in Section 3.

Using the same notation for the conditional expectations, we have:

$$E\{\pi_i^d(\tau_{n+1}^d) \,|\, \pi_i^d(\tau_n^d)\} = \pi_i^d(\tau_n^d) + \epsilon\, \pi_i^d(\tau_n^d) \left[F_i^d(\tau_n^d) - \sum_{o=1}^{N} A_d(o) F_o^d(\tau_n^d) \pi_o^d(\tau_n^d) \right]$$

In the limiting stationary operation (see Appendix A) $F_i^d(\tau_n) \to F_i$ for all d and now the fixed point will satisfy:

$$F_i = \sum_{o=1}^{N} A_d(o) F_o\, \pi_o^d \quad \text{for } A_d(i) = 1 \tag{15}$$

which implies that each controller d will set its distribution probabilities such that the value of F_i for $A_d(i) = 1$ is constant for all origins o with $A_d(o) = 1$. The chosen action scheme defines equivalence classes of nodes as follows: node i is in the class K_c if $A_c(i) = 1$, if $i, j \in K_c$ and $i \in K_d$ then $j \in K_d$. Then the limit point satisfies $F_i = k_c$ for all $i \in K_c$. In Figure 3 we give an example of a two-action scheme with two different equivalence classes. The darker circles in the left represent the controllers with their two actions. The lighter circles in the right represent the source nodes where the permits are sent by the controllers.

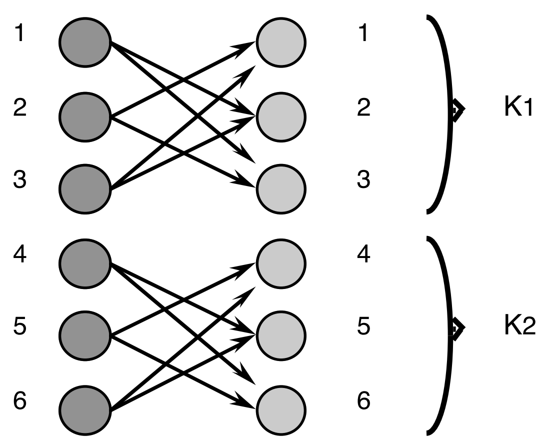

Figure 3: Different Equivalent Classes.

In the arrangement depicted in the figure there are two equivalence classes K_1 and K_2 defined by the sources which share the same set of controllers sending them permits. If the actions at the controllers equalize the H function for the two sources corresponding to its actions, then the joint action of controllers 1, 2 and 3 (respectively 4, 5 and 6) is to equalize H_1, H_2 and H_3 (respectively H_4, H_5 and H_6).

If we choose a two-action scheme with only one equivalence class and if there is a solution to the equilibrium problem with the optimal probabilities, then the automata will converge to an optimal solution, since this limit satisfies the Kuhn-Tucker conditions. We know, however, that the solution is not unique, so the limit point will depend on the initial values. We shall present numerical examples from simulations.

§5. Computer Simulations

We chose a four node newtwork under the model of Section 2. In order to assess the behaviour of our learning algorithm and compare it to the actual optimum, we used an example from [3], where a numerical procedure was used to find the optimal window size W^* and the optimal centralized distribution probabilities s_i^*. Since our main focus in this work is the decentralization of the controllers that distribute the permits, we ran all the simulations at the optimal window size and studied the behavior of the learning automata of section 5 with $\epsilon = 0.001$, comparing it with the true optimum. Figure 4 shows the four node network with links that define the two-way trunks. The distance $D_{i,j}$ of all the trunks is 25 Kms, and the corresponding capacities $C_{i,j} = C_{j,i}$ in Mb per second are shown in the picture. For each of the six trunks the service rate of the corresponding exponential server queue is $C_{i,j}/\alpha$, where $\alpha = 1024 \times 8$ is the mean packet size for 8 bit "words". Analogously, the constant propagation delay queues associated with each trunk has a delay $p_{i,j} = D_{i,j}/c$, where c is the speed of light.

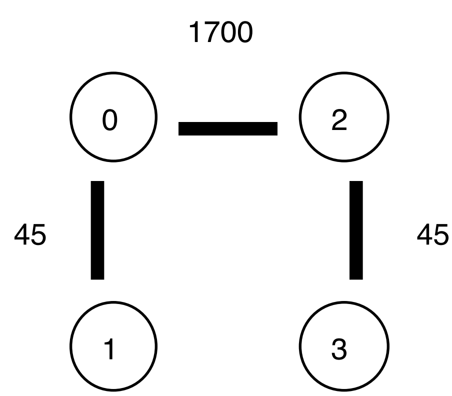

Figure 4

The routing choices are deterministic: packets arriving at node 0 with destinations 1 and 2 are sent to these nodes respectively, and packets with destination 3 are sent to node 2. Similarly, packets arriving at node 2 with destinations 3 or 0 are routed directly toward their destinations, and packets with destination 1 are routed to node 0. Finaly, all packets arriving at node 1 (node 3) are routed towards node 0 (node 2, respectively), regardless of their destinations. The constant propagation delay at the controller loops $p^c = 1000$

Kms./c is considered to be a delay over a much slower link.

The offered traffic matrix $\Lambda_{i,d}$ of incoming packets at origin i and destination d is:

$$\Lambda = \begin{pmatrix} 0.0 & 15000 & 15000 & 30000 \\ 15000 & 0.0 & 15000 & 15000 \\ 20000 & 10000 & 0.0 & 30000 \\ 10000 & 10000 & 20000 & 0.0 \end{pmatrix}$$

The optimal values can be consulted in [22]. In order to study the behavior of the algorithms, we evaluated the relative error of the decentralized scheme using the equilibrium equations $s_i(t) \equiv \sum_{c=1}^{N} \left\{ \frac{\sum_{o=1}^{N} \lambda_{o,c}}{\sum_{o,d=1}^{N} \lambda_{o,d}} \right\} \pi_i^c(t)$, and calculating the relative errors:

$$e_i(t) = \frac{s_i(t) - s_i^*}{s_i^*} \times 100 \tag{16}$$

The $(N-1)$-action automata. We show the results of two experiments using algorithm (12). In the first experiment the information is propagated instantaneously, so that the information vector $X^d(t)$ of section 5 is the same for all destinations d. The second experiment shows the results when the information delay is simulated as an exponential with mean one order of magnitude larger than the average throughputs. We show the plots of the percent error of the normalized throughputs (16), and of the logarithm of the product of powers. We verified that the algorithms do approach the limit value predicted by (9), under various initial conditions and different propagation delays. Although the inherent random fluctuations are present, as seen in the plots, the actual values of the parameters fluctuate very close to the optimal surface. This fact can be seen from the plots of the percent errors. The introduction of delays in information exchange does not affect the limiting behaviour of the automata and the effects in the product of powers are negligible for the scale shown in the plots. Since the information delay is larger than the delays of packets within the networks, the results suggest that the algorithms will also work when the implementation of information exchange is performed by piggy-backing the information to packets in the form of headers.

The two-action automata. We implemented the two-action scheme for the four node network using $A_0(1) = A_0(2) = 1$, $A_1(2) = A_1(3) = 1$, $A_2(0) = A_2(3) = 1$ and $A_3(0) = A_3(1) = 1$. We performed several experiments with different initial conditions, which gave different limiting controls, but show similar convergence rate to the optimal surface, measured by (16). We show the plots of the percent error and the product of powers for two such experiments, under instantaneous information. Similar plots were obtained for other initial conditions. The two-action simulations were 1.75 times faster than the $(N-1)$-action ones. In the $(N-1)$-action scheme, the updates are performed more frequently (every time a packet arrives to destination) whereas in the two-action case the updates take place only when packets arrive from the source nodes to which the controller sends permits. Therefore the computational effort required for the $(N-1)$-action scheme is larger than the corresponding one for the two-action scheme. Also, it is known [11] that under the same variance conditions, the learning rate of the two-action is faster than that of the $(N-1)$-action scheme.

Counterexample. We designed an example using the same topology and two-action scheme as before, where now the input offered traffic is very unbalanced, given by:

$$\Lambda = \begin{pmatrix} 0.0 & 50000 & 15000 & 10000 \\ 20000 & 0.0 & 1000 & 1000 \\ 20000 & 20000 & 0.0 & 5000 \\ 20000 & 10000 & 5000 & 0.0 \end{pmatrix}$$

In [23], we show in detail that this two-action scheme cannot achieve optimality for this system, using the tabloid method presented in Appendix B. We performed two simulations for this network, both under instantaneous information broadcasting, one for the $(N-1)$-action scheme and the other for the two-action scheme. We show the plots of the product of powers for each simulation. As expected, the $(N-1)$-action scheme converges in expectation to the optimal limit point predicted by (9). The two-action scheme shows a faster convergence to a value which is lower than the optimal, as expected.

§6. Concluding Remarks and Future Research

We have developed a decentralized strategy for isarithmic flow control for fast packet-switched networks. Our study focuses on the adaptive routing of permits from controllers to source queues. Non-uniqueness of the optimal controls led us to define different action schemes for sending permits. In the $(N-1)$-action scheme each of the N controllers sends permits to the other $(N-1)$ source queues. In the two-action scheme each of the N controllers sends permits to only two other source queues. The number of operations required by this scheme is less than for the $(N-1)$-action scheme, which updates more frequently. Therefore, in terms of real-time operation, the two-action scheme presents important savings, especially when the number of nodes is large. Since the learning rate also depends on the number of nodes and the number of actions, the faster convergence of the two-action will also be more evident in larger networks. The results given in this work suggest that the two-action scheme works very well when it achieves the optimum, but it degrades performance when the input traffic "shrinks" the feasible solution set to an empty set, in which case its performance is suboptimal. Recent efforts have included the study of the design of n-action schemes that are less sensitive to changes in the input traffic. Future extensions could include the study of robustness with respect to changes in the topology of the network.

In most realistic models, including multiclass customers, several window sizes, bursty arrivals and finite buffer stations, it is practically impossible to evaluate a closed form solution. One of the authors (Felisa Vázquez-Abad) is currently studying methods of distributed derivative estimation for complex systems. These methods do not require a product form solution, and may reduce considerably the amount of information exchange needed for the control updates. One of the authors (Lorne Mason) has studied the concurrent optimization of flow and routing parameters using the centralized version of the algorithms. A natural extension of the present work is to prepare a fully decentralized concurrent on-line optimization version that includes all the control parameters.

Appendix

Appendix A. The Proof of Asymptotic Optimality.

We present the proof of convergence of the algorithms (12) and (14), using the framework and results of the weak convergence method in [8] and [21].

Call τ_n the epoch of the n-th *global* update, regardless of the controller, and τ_n^c the epoch of the n-th local update at c. Denote by θ the vector of all the distribution probabilities π_i^c. Then both algorithms can be expressed as a stochastic approximation:

$$\theta_{n+1}^\epsilon = \theta_n^\epsilon + \epsilon Y_n^\epsilon,$$

if we identify $\epsilon = G$ with the gain parameter, and n with the n-th global update epoch. Clearly, the vector Y_n^ϵ depends on the feedback function $b(\tau_n)$. The process can be imbedded in a Markov Decision Process (MDP) $(\xi_n^\epsilon, \theta_n^\epsilon)$, identifying the state ξ_n^ϵ with the vector of queue lengths, residual service times (if not Markovian) and local information at time τ_n. Following the notation in [8], call $G(\xi, \theta) = E\{Y_n^\epsilon | \xi_n^\epsilon = \xi, \theta_n^\epsilon = \theta\}$. The σ-algebra related to the MDP up to the n-th update will be denoted by $\mathcal{F}_n^\epsilon$. By the Markovian property, $G(\theta_n^\epsilon, \xi_n^\epsilon) = E\{Y_n^\epsilon | \mathcal{F}_n^\epsilon\}$, which is a random variable depending on the distribution of $(\xi_n^\epsilon, \theta_n^\epsilon)$. From the closed network model, it follows that the fixed control process $\xi(\theta)$ Markovian with transition probability $P(dx, x) = P\{\xi_{n+1}(\theta) \in dx | \xi_n(\theta) = x\}$, which is weakly continuous in θ for our model. A closed network with stationary service distributions (independent of θ)) is stable for every possible value of θ. Call $\mu_\theta(dx)$ the invariant measure of the fixed control process, and

$$g(\theta) = \int \mu_\theta(dx) G(x, \theta) = \lim_{m \to \infty} \frac{1}{m} \sum_{n=0}^{m-1} E\{G(\xi_n(\theta), \theta)\}$$

Notice that this latter expectation is w.r.t. the fixed control process. The random variables Y_n^ϵ are uniformly integrable, since they are uniformly bounded by construction of the feedback functions. Furthermore, the sequence $\{(\xi_n^\epsilon, \theta_n^\epsilon), n \geq 0, \epsilon > 0\}$ is tight, that is, for every $\alpha > 0$ there exist a compact set K_α such that $\sup_{\epsilon,n} P\{(\xi_n^\epsilon, \theta_n^\epsilon) \in K_\alpha\} < \alpha$. This follows because θ_n^ϵ are probability vectors, and ξ_n^ϵ are uniformly tight: queue sizes are all bounded by the window size, the residual services are independent of $\epsilon, \theta_n^\epsilon$ and the information vector containing the moving averages is also bounded a.s. by a random variable of maximal delay along feasible paths, which is independent of θ_n^ϵ and n. Tightness is the stochastic analog of compactness. Define the control interpolation process:

$$\vartheta^\epsilon(t) = \theta_n^\epsilon \quad \text{for } t \in [n\epsilon, (n+1)\epsilon) \tag{A.1}$$

From this definition and the stochastic approximation form, for $t = \epsilon n$ we have:

$$\frac{\vartheta^\epsilon(t+\epsilon) - \vartheta^\epsilon(t)}{\epsilon} = Y_n^\epsilon$$

and the conditional expected behaviour of Section 4 is related to:

$$E\left(\frac{\vartheta^\epsilon(t+\epsilon) - \vartheta^\epsilon(t)}{\epsilon} | \mathcal{F}_n^\epsilon\right) = G(\vartheta^\epsilon(t), \xi_n^\epsilon)$$

From Proposition 1 in [21] it follows that every subsequence of $\vartheta^\epsilon(\cdot)$ as $\epsilon \to 0$ has a further weakly convergent subsequence and all weak limits are Lipshitz continuous a.s. All the assumptions of [8] are satisfied, therefore any such limit satisfies the ODE:

$$\frac{d\vartheta(t)}{dt} = g[\vartheta(t)]$$

If the ODE has a unique solution for each initial condition, the limit does not depend on the subsequence and therefore $\vartheta^\epsilon(\cdot)$ converges to $\vartheta(\cdot)$. It is common to assume that $g(\cdot)$ is locally Lipschitz continuous, thus continuous and uniformly bounded on compact sets, which would ensure uniqueness of the solution for each initial condition. If, furthermore, this ODE has asymptotically stable points $\theta^* \in \mathcal{S}$, then the limit points $\lim_{t\to\infty} \vartheta(t)$ of its solutions (which may or may not depend on the initial condition) belong to $\mathcal{S}$ and satisfy $g(\theta^*) = 0$. In particular, for the schemes presented in the present work, it follows from (11) that, for the fixed control process:

$$F_i(\theta) = \lim_{n\to\infty} \frac{1}{n} \sum_{n=0}^{m-1} E\left\{E\{F_i^c(\tau_n^c)|\mathcal{F}_n^c(\theta)\}\right\} = 1 - \frac{\text{arctan} H_i[X(\theta)]}{(\pi/2)} \tag{A.2}$$

for all c, where $\mathcal{F}_n^c(\theta)$ is the σ-algebra generated by $\{\xi_n(\theta)\}$ up to time τ_n^c. Let $\vartheta_i^c(t)$ denote the limit process of the component corresponding to the control variable π_i^c. Then it follows that the limit process corresponding to the scheme in (12) satisfies:

$$\frac{d\vartheta_i^c(t)}{dt} = M_d[\vartheta(t)]\left[F_i[\vartheta(t)] - \vartheta_i^c(t)\sum_{o=1}^{N} F_o[\vartheta(t)]\right] \tag{A.3}$$

where $M_d(\theta)$ is the stationary fraction of updates performed at controller d for the fixed control process (see the proof of Theorem 3 in [21] for the details on the time scaling argument in decentralized schemes). Analogously, for the scheme in (14), the limit processes satisfy:

$$\frac{d\vartheta_i^c(t)}{dt} = M_d[\vartheta(t)]\,\vartheta_i^c(t)\left[F_i[\vartheta(t)] - \sum_{o=1}^{N} A_c(o)F_o[\vartheta(t)]\right] \tag{A.4}$$

Both ODE's are well defined, they have a unique solution for each initial condition and this limit does not depend on the frequency or delays in information broadcasting, as long as (11) holds. The weak convergence approach followed here allows us to interpret the learning schemes as stochastic analogs of numerical approximations of an ODE, whose r.h.s. is constructed using the Kuhn-Tucker conditions for optimality. The asymptotic behaviour of the algorithms is determined by studying the limit points $\lim_{t\to\infty}\vartheta(t)$. From (A.3) and (A.4), these are the fixed points of the equations for the conditional expected behaviour, as mentioned in Section 4.

As a final remark, this proof only requires the construction of apropriate estimators of the sensitivity that satisfy (A.2). We have provided one such method that requires basic information on moving averages, but in order to minimize information exchange, the schemes could be implemented relying on a less frequent transmission of the moving averages between nodes, yielding the same asymptotic behaviour.

Appendix B. Tabloid Method for the n-action Scheme.

One way to study the two-action scheme was introduced in [22] to study the assignment problem in terms of a tabloid solution, as follows. From the equilibrium matrix, we can consider the basic variables $x_c(i) = A_c(i)C_c\pi_i^c$ that appear in the matrix equations instead of the distribution probabilities themselves. The data of the problem in terms of the optimal throughputs can be written in the form of a tabloid, where the rows add up to the numbers in the right hand column, the columns add up to the numbers in the last row and the sum of the right hand column equals the sum of the last row. In a similar way, we can also write down the solution in terms of the variables $x_c(i)$, where the sums of the rows set equal to the corresponding number in the last column represents equation (8) and the sum of the columns set equal to the value in the last row yields equation (7).

In the solution table, some of the entries $x_c(i)$ may be zero entries, depending on the actions at that controller. If we want to find a solution for a given action matrix, we "cross" out the corresponding places in the tabloid and proceed to find a solution. It is clear from this form why the $(N-1)$-action scheme always possesses a solution of the form $x_c(i) = \lambda_{ic}$, although this solution is not unique, as shown in theorem 2.

Data						
0	λ_{10}	λ_{20}	λ_{30}	$\cdots$	λ_{N0}	C_0
λ_{01}	0	λ_{21}	λ_{31}	$\cdots$	λ_{N1}	C_1
λ_{02}	λ_{12}	0	λ_{32}	$\cdots$	λ_{N2}	C_2
λ_{03}	λ_{13}	λ_{23}	0	$\cdots$	λ_{N3}	C_3
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
λ_{0N}	λ_{1N}	λ_{2N}	λ_{3N}	$\cdots$	0	C_N
λ_0	λ_1	λ_2	λ_3	$\cdots$	λ_N	

Solution						
0	$x_0(1)$	$x_0(2)$	$x_0(3)$	$\cdots$	$x_0(N)$	C_0
$x_1(0)$	0	$x_1(2)$	$x_1(3)$	$\cdots$	$x_1(N)$	C_1
$x_2(0)$	$x_2(1)$	0	$x_2(3)$	$\cdots$	$x_2(N)$	C_2
$x_3(0)$	$x_3(1)$	$x_3(2)$	0	$\cdots$	$x_3(N)$	C_3
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$x_N(0)$	$x_N(1)$	$x_N(2)$	$x_N(3)$	$\cdots$	0	C_N
λ_0	λ_1	λ_2	λ_3	$\cdots$	λ_N	

We give now the tabloid solution of two different two-action schemes in a four node example, showing the symbol $\times$ at the zero entries where $A_c(i) = 0$.

In terms of the dimensionality of the problem, the two-action case defines a linear problem in $2N$ variables with $2N$ equations, one of them being linearly dependent on the other $2N-1$. If the problem does not decouple into several independent problems (such as two tabloids put together with zero entries in the diagonal blocks), then the solution is defined as a one dimensional subspace. This condition will later be defined in terms of equivalence classes of source nodes (see section 5). It is clear that in a four node network any two-action scheme will define a tabloid that cannot be decomposed into independent smaller tabloids, since the controllers never send permits to their own source queues.

Example 1				
0	$\lambda_1 + \lambda_0 - C_3 - a$	$C_0 + C_3 - \lambda_0 - \lambda_1 + a$	$\times$	C_0
$\times$	0	$C_1 + C_2 - \lambda_3 - a$	$\lambda_3 - C_2 + a$	C_1
a	$\times$	0	$C_2 - a$	C_2
$\lambda_0 - a$	$C_3 - \lambda_0 + a$	$\times$	0	C_3
λ_0	λ_1	λ_2	λ_3	

In order to find the one-dimensional solution set for these examples we arbitrarily choose one element of the solution matrix as a and work through the tabloid filling the entries that are determined by the requirement of the sums of rows and columns, and using $\sum_i \lambda_i = \sum_c C_c$. The general way of finding a solution in higher dimensional problems with two-action schemes that do not decouple is identical to this procedure and we omit the details of the general algorithm.

In the second example, the entry $x_1(0) = \lambda_0$ is uniquely determined by the choice of the action matrix and does not depend on the value of a. In order for this solution to be feasible in terms of the distribution probabilities, all entries of the solution matrix must be non-negative.

Example 2				
0	$\times$	$\lambda_1 + \lambda_2 - C_3 - a$	$C_0 + C_3 - \lambda_1 - \lambda_2 + a$	C_0
λ_0	0	$\times$	$C_1 - \lambda_0$	C_1
$\times$	a	0	$C_2 - a$	C_2
$\times$	$\lambda_1 - a$	$C_3 + a - \lambda_1$	0	C_3
λ_0	λ_1	λ_2	λ_3	

Since all the expressions are linear in a, the non-negativity condition can be rewritten in terms of an interval of feasibility of the form $a_0 \leq a \leq a_1$, where the limits depend on the data λ_{id}. This set may be empty, in which case the chosen action matrix will not yield the optimal solution for the original problem, therefore given a two-action scheme it does not always possess a solution. Given a problem, we may ask if it is possible to find a two-action scheme with a tabloid solution of non-negative entries. Unfortunately the answer is no, as shown in the tabloid for a counterexample, where we show the data of the problem.

Counterexample				
0	50	20	10	80
60	0	1	1	62
30	1	0	1	32
10	1	1	0	12
100	52	22	12	

In this problem there is no way that we can keep only two nonzero entries of the first row and fill out the rest of the table with non-negative entries. Although this represents an extremely unbalanced traffic where most of the throughput comes in and out of node zero, it shows that in general we cannot always find a two-action scheme which will satisfy the equilibrium equations for the optimal throughputs.

The main problem in assigning the two-action automata scheme is that the tabloids for the data and the solutions cannot be written in terms of the data of the original problem and depend on the unknown optimal throuputs. Therefore we cannot use this algorithm in order to determine if the problem admits a two-action scheme to achieve its optimum performance. This framework has further been investigated in [17] to establish the range of feasibility of the n-action design, for $n < N-1$. This approach introduces a ficticious objective function to implement the simplex method, which finds the corner points of the feasible set, thus finding the solution points of the tabloids.

Appendix C. Notation.

N :	number of nodes in the network
W :	number of permits in the network
C_{ij} :	capacity of link (i,j)
D_{ij} :	distance between node i and node j
r_{ij}^d :	probability of routing a packet with destination d from i to j
p_{ij} :	$= D_{ij}/c$ propagation delay at trunks
p_j^d :	propagation delay from controller d to source node j
π_i^d :	control variable: probability of sending a permit from d to source node i
s_i :	relative number of visits to permit source queue i
c_j :	relative number of visits to controller queue j
f_{jl} :	relative number of visits to trunk queue (j,l)
Λ_{id} :	external arrival rate at node i of packets with destination d
α :	average packet size
λ_{id} :	stationary average throughput of packets with origin-destination (i,d)
λ_i :	aggegate throughput $= \sum_d \lambda_{id}$
C_d :	aggegate throughput $= \sum_i \lambda_{id}$
T_{id} :	stationary average end-to-end delay of packets with origin-destination (i,d)
$A_c(i)$:	action matrix $= \mathbf{1}_{\{\pi_i^c > 0\}}$

References

1. Basket, F., Chandy, K.M., Muntz, R., Palacios, F. (1975) "Open, closed, and mixed networks of queues with different classes of customers", *Jour. of the Assoc. of Computing Machinery*, **vol. 22**, No. 2, pp 248-260.
2. Coderre, J. (1989) *Modèle optimal d'un contrôle global de flux appliqué aux réseaux à commutation par paquets*, M.Sc. thesis, INRS-Télécommunications, Université du Québec, April 1989.
3. Cotton, M., Mason, L.G. (1994) "Adaptive isarithmic flow control in fast packet switching networks", *IEEE Trans. on Comm.*, **vol 42**.
4. Davies, D.W. (1972) "The control of congestion in packet-switching networks", *IEEE Trans. on Comm.*, pp 546-550.
5. Gelenbe, E., Pujolle, G. (1981) *Introduction to queueing networks*, Chapter 4. John WIley & Sons, New York.
6. Gelenbe, E., Mitrani, I. (1980) *Analysis and synthesis of computer systems*, Chapter 3. Academic Press, New York.

7. Gerla and Kleinrock (1980) "Flow Control: a comparative study", *IEEE Trans. on Comm.*, **vol COM-28**, No. 4, pp. 553-574.

8. Kushner, H., Vázquez-Abad, F. (1996) "Stochastic approximation methods for systems of interest over an infinite time horizon", *SIAM J on Control and Optim*, **vol 34**, N0. 2, pp 712-756.

9. E. Létourneau and L.G. Mason (1996) "Integration Strategies for Flow Control Mechanisms in ATM Networks", *Proceedings of ITC Specialists Seminar on Control in Communications*, Lund Sweden, Oct '96, pp 177-188.

10. K.-Q. Liao, L.G. Mason (1993) "Self-optimizing Window Flow Control in High Speed Data Networks", *Computer Communications*, **vol. 16**, No. 11, pp 706-716.

11. Mason, L.G. (1973) "An optimal learning algorithm for S-model environments", *IEEE Trans. on Autom. Control*, **vol. AC-18**, pp. 493-496.

12. Mason, L.G., Gu, X.D. (1985a) "Learning automata models for adaptive flow control in packet-switching networks", in *Adaptive and Learning Systems Theory and Applications*, K.S. Narendra editor. Plenum Press, New York, pp. 213-227.

13. Mason, L.G., Gu, X.D. (1985b) "Adaptive isarithmic flow control" rapport technique de l'INRS-Télécommunications No. 85-18.

14. Mazumdar, R., Mason, L.G., Douligeris, C. (1991) "Fairness in network optimal flow control: optimality of product forms", *IEEE Trans. on Comm.*, pp 775-782.

15. Narendra, K.S., Thatachar, M.A.L. (1989) *Learning Automata: an Introduction*, Englewood Cliffs: Prentice Hall Inc.

16. A. Pelletier, M. Cotton, L.G. Mason (1993) "Combined Adaptive Routing and Flow Control in Fast Packet Switching Networks", *Proceedings of COMCON 4*, Rhodes, Greece, June '93, pp 557-569.

17. Perron, J. (1993) "Conception et évaluation des systèmes à contrôle de flux isarithmique adaptatif decentralisé–approche linéaire", M.Sc. thesis, INRS-Télécommunications, Université du Québec, April, 1993.

18. Reiser, M. (1979) "A queueing network analysis of computer communication networks with window flow control", *IEEE Trans. on Comm.*, **vol COM-27**, No. 8, pp 1199-1209.

19. Srikantakumar, P.R., Narendra, K.S. (1982) "A learning model for routing in telephone networks", *SIAM Jour. Control and Optim.* **vol. 20**, No. 1, pp 34-57.

20. Vázquez-Abad, F. (1992) "Sensitivity Analysis for stochastic DEDS: an Overview", *Proc. II Simposio de Probabilidad y Procesos Estocásticos* Aportaciones Matemáticas, Notas de Investigación No. 7, Caballero M.E., Gorostiza, L.G. Editors, pp 163-182.

21. Vázquez-Abad, F.J., Cassandras, C.G. and Julka, V. (1998) "Centralized and decentralized asynchronous optimization of stochastic discrete event systems", to appear in *IEEE on Automatic Control*, March '98.

22. Vázquez-Abad, F., Mason, L.G. (1992) "Decentralized Adaptive Isarithmic Flow control for Packet-switched Networks", technical report INRS-Telecommunications No. 92-09.

23. Vázquez-Abad, F., Mason, L.G. (1995) "Adaptive Decentralized Control of DEDS under Non-uniqueness of the Optimal Control", *Journal of DEDS, Theory and Applications*, **vol 6**, No. 4, pp 323-359.