Flow Control Power is Nondecentralizable

JEFFREY M. JAFFE, MEMBER, IEEE

Abstract—**Flow control in store-and-forward computer networks is appropriate for decentralized execution. A formal description of a class of "decentralized flow control algorithms" is given. The feasibility of maximizing power with such algorithms is investigated.**

On the assumption that communication links behave like $M/M/1$ servers it is shown that no "decentralized flow control algorithm" can maximize network power. Power has been suggested in the literature as a network performance objective. It is also shown that no objective based only on the users' throughputs and average delay is decentralizable. Finally, a restricted class of algorithms cannot even approximate power.

I. INTRODUCTION

FLOW CONTROL in data communication networks protects the network from excessive user demands, and allocates resources between competing users [1]. It has also been proposed as a means for providing "optimal performance" [2]. In this paper a class of flow control algorithms is formally defined. This class encompasses algorithms that work on the design principles of dynamic behavior, use of local information, and decentralized execution [2]. It is then shown that no such "decentralized" algorithm can optimize several measures of network performance. These measured are variants of network power [3].

It is assumed that data flowing from one node to another is transmitted on fixed, virtual circuits. That is, all messages sent by a source node to a destination node, that are from the same logical "session" follow the same path. Section II describes a queueing model used to evaluate network performance as a function of the offered load. In Sections III–V, we develope a formalism needed to define decentralizability. Various measures are proved to be nondecentralizable in Sections VI–VIII.

There are two important contributions contained in this work. The first is a *technical result* which states that certain desirable functions are not optimizable in a distributed manner. The second, perhaps more important, contribution is a *formalism* for discussing the question of what can be accomplished in a distributed system. We use flow control as a basis for this study—but other network functions should also be subjected to this type of analysis.

The approach taken by this paper has the following general structure. We first develop an informal description of design principles to be followed by flow control algorithms. We next abstract from these principles in order to obtain a simple formal description of "decentralized flow control algorithms." Central to this description is that two users with the same local network view operate identically.

The method of proving nondecentralizability of power then follows from the above formalism. In particular, examples are given which explicitly demonstrate the inadequacy of the local view. It is shown that a global view is indeed required to optimize the global objective of power.

Paper approved by the Editor for Computer Communication of the IEEE Communications Society for publication without oral presentation. Manuscript received August 19, 1980; revised January 30, 1981.

The author is with the IBM Thomas J. Watson Research Center, P. O. Box 218, Yorktown Heights, NY 10598.

II. NETWORK MODEL

A packet-switched computer network can be modeled as a graph (N, L) with vertex (or node) set N and edge (or link) set L. Each link $l \in L$ has a *service rate* of $s(l)$ bits per second. A *path* p in (N, L) is a sequence $p = (n_1, \cdots, n_k)$ with $n_i \in N$, and for $i = 1, \cdots, k-1$ $l_i = (n_i, n_{i+1}) \in L$. The set $\{l_1, \cdots, l_{k-1}\}$, denoted $l(p)$, is the *link set of p*. A path models a fixed route used by one of the virtual circuits (or users) of a network.

To model network performance and relate message rate to delay, it is assumed that each link behaves as an $M/M/1$ queue. To do this, it is assumed that the average message length is b bits/message, there is no nodal processing time, and Kleinrock's independence assumption [4] applies. In that case, a user's path may be modeled as an $M/M/1$ queueing system. Specifically, define the *capacity of* l, $c(l)$ as $c(l) = s(l)/b$ (this is the average rate of messages processed by l). Let $\gamma(l)$ denote the average rate of messages sent by all users whose paths include link l. Then the *average delay* experienced by the messages that go through l is $d_l = 1/(c(l) - \gamma(l))$ [2]. The average delay D_i of packets sent by user i is the sum of the average delays experienced at the individual links of path i. In this model, message rate and throughput are identical [2].

III. PERFORMANCE CRITERIA

Given a network of m users and a message rate γ_i for user i, the rate assignment $\gamma = (\gamma_1, \cdots, \gamma_m)$ determines the average delay D_i for user i. For any pair of rate assignments, a performance criterion simply specifies which rate assignment is better. We assume for simplicity that throughput and delay are the only properties of interest.

Formally, we associate the rate assignment γ with a $2m$-tuple $(\gamma_1, \cdots, \gamma_m, D_1, \cdots, D_m)$. Our informal notion of a performance criterion translates into a total ordering G on $2m$-tuples. If $x, x^* \in R^{2m}$ (where $x = (x_1, \cdots, x_{2m})$ and $x^* = (x_1{}^*, \cdots, x_{2m}{}^*)$) then $x\,G\,x^*$ means that the criterion G specifies that it is at least as good to have rates of $x_1, \cdots, x_m$ with resulting delays $x_{m+1}, \cdots, x_{2m}$ as it is to have rates $x_1{}^*, \cdots, x_m{}^*$ with delays $x_{m+1}{}^*, \cdots, x_{2m}{}^*$.

IV. FLOW CONTROL ALGORITHMS

The goal of flow control algorithms is to adjust each user's message rate in accordance with current conditions on its path. In order to meet this goal the design principles of decentralized execution, use of current local information, and dynamic behavior must be adhered to. These principles are needed if the flow control algorithm is to effectively react to *current* network conditions.

The meaning of decentralized execution is that each user calculates its *own* message rate. Local information consists of some set of network properties conveniently available to a user. Only local information is allowed since the algorithm must react quickly to network changes. Dynamic execution requires that each user *constantly* react to changes in the value of the local information. Combining these requirements yields the following structure for each user's flow control algorithm:

1) determine the current conditions on the user's path;
2) update the message rate (as a function of these conditions);
3) go to 1.

Thus in step 2, the user applies an update function U, to the information set I obtained in step 1, to obtain a new message rate $U(I)$.

The question of what information set is conveniently available to a user is debatable and probably implementation dependent. We use one fairly comprehensive definition of local information and prove nondecentralizability results with respect to it. The information set consists of:

1) the capacity of each link on the user's path;
2) the number of users sharing each link on the path;
3) the current message rate of each interfering user.

We assume one additional characteristic of flow control algorithms. The problem with the above outline is that users may cleverly encode global topology into their message rates, permitting interfering users to learn more than local information. Use of such coding tricks is restricted by requiring that the updating function U, used in the rate update step be a *continuous* function of its inputs. Thus if two information sets differ by small amounts, the resulting message rates must be reasonably close. Intuitively, such a requirement is not too restrictive—the responses to similar network conditions *should* be similar.

V. ACHIEVING A PERFORMANCE CRITERION WITH A FLOW CONTROL ALGORITHM

In this section, we define the concept of achieving performance criteria with flow control algorithms which leads to the definition of nondecentralizability. We then prove the main theorem used to demonstrate nondecentralizability of certain performance criteria.

First we define the *final message rate* of a user executing a flow control algorithm. Fix a network with m paths. A *history* $h = h_0 h_1 \cdots$ is an infinite sequence of integers between 1 and m; each integer appearing infinitely often. A history specifies the order in which the users execute their update algorithms. In particular, starting with an initial rate assignment, user h_0 updates its rate based on network conditions. Then user h_1 changes its rate, using its new value of "current information" (i.e. considering that user h_0 has changed its rate). Given an algorithm, an initial rate assignment, and a history, the sequence of rate changes for each user may be determined. If the sequence of rates for user j is $\gamma_1, \gamma_2, \cdots$ then the *final message rate for user* j *is* $\lim_{i\to\infty}\gamma_i$ if the limit exists, and undefined otherwise.

The main definitions may now be given.

Definition 1: A flow control algorithm *achieves* a performance criterion G if for any network and any sets of paths in the network there is *some* history h such that when users update their message rates in the order specified by h, the final rates of all users (paths) yield optimum performance (according to G).

Definition 2: A performance criterion is *nondecentralizable* if no flow control algorithm achieves it.

Thus a nondecentralizable criterion cannot be achieved even with total control over when each user performs its updates. The main technique to prove nondecentralizability is to show that for certain users. local information is not enough. Showing this fact is facilitated by the following definition and theorem.

Definition 3: Two users have *identical viewpoints* if the value of their respective local information sets are the same.

The importance of the above definition is that whenever two users have identical viewpoints, they choose the same message rate if they use the same flow control algorithm.

Theorem 1: Assume that in two networks (each with a set of users) the optimum rate assignment according to G causes two distinguished users (one in each network) to have identical viewpoints. Assume further, that an optimum G these two users have different rates. Then G is nondecentralizable.

Proof: It will be shown that if *any* history for a given algorithm causes the correct final message rates in one network, then *no* history for the same algorithm causes the correct final message rates in the other.

Consider the sequence of information sets seen by one of the two distinghished users; $I_1, I_2, \cdots$ in a history h. If the rate update function is $U: I \to R$, then the resulting sequence of message rates for the user is $U(I_1)$, $U(I_2)$, $\cdots$. Now if the algorithm with history h obtains the correct final message rates for all users, then the sequence I_1, I_2, $\cdots$ converges to $I^* = \lim_{j\to\infty} I_j$ since it is an infinite subsequence of a convergent sequence [5]. (The convergent sequence is the total set of information sets, whether or not seen by the users.) Since U is continuous, $\lim_{j\to\infty} U(I_j) = U(I^*)$ [5], i.e., $U(I^*)$ is the final rate for the user of interest.

Assume that a history h' causes the correct final message rates in the second network of interest. Due to the identical viewpoint hypothesis, the limit of the information sets seen by the second distinguished user is I^*. Thus the final message rate of the user is $U(I^*)$ which is the incorrect final message rate for this user. Thus no history for the algorithm achieves the correct final message rates in the second network.

VI. NETWORK POWER

In [3], "power" is suggested as a performance criterion. Power is defined as the ratio of throughput to delay—one rate assignment is better than another if it results in greater power. In the remainder of the paper we show that a number of variants on the power criterion are nondecentralizable.

To formally define power fix a network with m users, rates $\gamma_1, \cdots, \gamma_m$ and resulting delays $D_1, \cdots, D_m$. The *total throughput* is $T = \Sigma_{i=1}{}^m \gamma_i$, the *average delay is* $\bar{D} = (1/T)\Sigma_{i=1}{}^m \gamma_i D_i$, and the *power* is $P = T/\bar{D}$.

Corollary 1: Power is nondecentralizable.

Proof: Two users with identical viewpoints and different desired rates will be constructed. Thus by Theorem 1, power is nondecentralizable. The first network is particularly simple—a network of a single link l and a single user on the link. The optimum rate is $c(l)/2$ [6].

Consider a network with two users each using one link (l_1 and l_2, respectively), and each sharing its own link with no one. They both trivially share the viewpoints of users in one user, one link networks with link capacities $c(l_1)$ and $c(l_2)$, respectively. To prove nondecentralizability, it suffices to show that when the two users are in one network, rates of $c(l_1)/2$ and $c(l_2)/2$ will not in general maximize power. In particular, the throughput is $T = \frac{1}{2}(c(l_1) + c(l_2))$, average delay is

$$\bar{D} = \frac{1}{(\frac{1}{2})(c(l_1) + c(l_2))} \sum\left(\frac{c(l_i)}{2} \cdot \frac{1}{c(l_i) - (c(l_i)/2)}\right) = \frac{4}{(c(l_1) + c(l_2))} \tag{1}$$

and power is $P = T/\bar{D} = (c(l_1) + c(l_2))^2/8$. If $c(l_1) \gg c(l_2)$, then the throughput assignment $(c(l_1)/2, 0)$ provides $T = c(l_1)/2, \bar{D} = 1/c(l_1)$, and $P = c(l_1)^2/4 > (c(l_1) + c(l_2))^2/8$.

The fact that power "encourages" a user to change throughput based on *noninterfering* traffic points out the inadequacy in the basic definition. Variants of power will soon be discussed, but first a stronger result about criteria that depend only on T and $\bar{D}$ is proved.

A *performance criterion G depends only on T and* $\bar{D}$, if whenever $T_x \geq T_y$ and $\bar{D}_x \leq \bar{D}_y$ then $x\,G\,y$ ($T_x, T_y, \bar{D}_x, \bar{D}_y$ are the throughputs and average delays of vectors x and y) and if $x\,G\,y$, then $y \not G\, x$, unless $T_x = T_y$ and $\bar{D}_x = \bar{D}_y$.

Corollary 2: No performance criterion based on T and $\bar{D}$ is decentralizable.

Proof: As in the proof of Corollary 1, consider two users having one link each with capacities c_1 and c_2. Assume that in a single user network, when a user is confronted with a single link path of capacity $c_1 (c_2)$ and no interfering traffic, it chooses message rate $x(y)$. Then when two noninterfering users in one network have capacities c_1 and c_2, the sum of throughputs is $T = x + y$, and

$$\bar{D} = \frac{1}{c_1 - x}\,\frac{x}{x+y} + \frac{1}{c_2 - y}\,\frac{y}{x+y}$$

Our object is to show that there must be some values for c_1 and c_2, for which the assignment x, y is nonoptimal.

Consider the assignment that user 1 (with link capacity c_1) chooses rate $x - \epsilon$, and user 2 chooses rate $y + \epsilon$. By abuse of notation, let T_ϵ and $\bar{D}_\epsilon$ be the resulting throughputs and delays. In that case $T_\epsilon = x + y$, and

$$\bar{D}_\epsilon = \frac{1}{c_1 - (x - \epsilon)}\,\frac{x - \epsilon}{x + y} + \frac{1}{c_2 - (y + \epsilon)}\,\frac{y + \epsilon}{x + y}. \tag{2}$$

Since $T = T_\epsilon$, to get the desired contradiction it suffices to show that $\bar{D}_\epsilon < \bar{D}$. That would prove that the throughputs x and y are chosen inappropriately for the two user network. Let

$$f(z) = \frac{z}{c_1 - z} \tag{3}$$

and

$$g(w) = \frac{w}{c_2 - w}\,. \tag{4}$$

Note that $\bar{D}_\epsilon < \bar{D}$ iff

$$\left(\frac{1}{c_1 - (x - \epsilon)}\,\frac{x - \epsilon}{x + y} + \frac{1}{c_2 - (y + \epsilon)}\,\frac{y + \epsilon}{x + y}\right) < \left(\frac{1}{c_1 - x}\,\frac{x}{x + y} + \frac{1}{c_2 - y}\,\frac{y}{x + y}\right) \tag{5}$$

iff

$$\frac{y + \epsilon}{c_2 - (y + \epsilon)} - \frac{y}{c_2 - y} < \frac{x}{c_1 - x} - \frac{x - \epsilon}{c_1 - (x - \epsilon)} \tag{6}$$

iff

$$g(y + \epsilon) - g(y) < f(x) - f(x - \epsilon). \tag{7}$$

As $\epsilon \to 0$, (7) holds iff $f'(z)|_{z=x} > g'(w)|_{w=y}$. Thus we have the desired contradiction if $(f'(z)|_{z=x}) > (g'(w)|_{w=y})$. Conversely, by choosing throughputs $x + \epsilon$ and $y - \epsilon$, we obtain the desired contradiction as long as $(f'(z)|_{z=x}) < (g'(w)|_{w=y})$. Thus given any flow control policy based on T and $\bar{D}$, and any algorithm that purports to implement the policy, we have shown that the algorithm does not implement the policy, unless the throughput as a function of capacity is chosen, so that for every c and resulting x

$$\left.\frac{d\left(\dfrac{z}{c - z}\right)}{dz}\right|_{z=x} \tag{8}$$

is some constant (i.e.,

$$\left.\frac{d\left(\dfrac{z}{c_1 - z}\right)}{dz}\right|_{z=x} = \left.\frac{d\left(\dfrac{w}{c_2 - w}\right)}{dw}\right|_{w=y} \tag{9}$$

for every c_1, c_2).

The proof is completed by showing that $d(z/(c-z))/dz$ cannot be a constant. Note that

$$d\left(\frac{\frac{z}{c-z}}{dz}\right)\Bigg|_{z=x} = \frac{c}{(c-x)^2}. \tag{10}$$

As

$$c \to 0, \quad \frac{c}{(c-x)^2} \to \infty \tag{11}$$

irrespective of the choice of $x (0 < x < c)$. Thus any finite value for $d(z/(c-z))/dz)$ cannot be achieved by sufficiently small values of c.

Three alternative definitions of multiple-user power are now explored. A nondecentralizability proof is presented for only one of these as we proceed to explain.

Consider a set of paths, and for a path p, let $c_{\min} = \min_{l \in l(p)} c(l)$. Let $d_l(\gamma)$ be the delay on l at throughputs γ, and $D_i(\gamma)$ be the total average delay suffered by the ith user at γ. Then we have the following.

The *link normalized delay of p is*

$$\sum_{l \in l(p)} \frac{d_l(\gamma)}{(1/1/c(l))} = \sum_{l \in l(p)} c(l) d_l(\gamma). \tag{12}$$

The delay of a link is normalized by its minimal delay (transmission delay). The next two measures weigh each of p's link's delay by the same amount, by the transmission delay of the smallest capacity link, and by the transmission delay of the entire path. Thus we have the following.

The *link-route normalized delay of user i* is

$$\frac{1}{(1/c_{\min})} D_i(\gamma) = c_{\min} D_i(\gamma) \tag{13}$$

The *route normalized delay* of user i is

$$\frac{1}{\sum_{l \in l(p)} \frac{1}{c(l)}} D_i(\gamma). \tag{14}$$

Using these three different definitions of delay, one has three different definitions of power (T/D). These interpretations of power are meant to be typical but not exhaustive.

These three definitions are introduced to account for an anomalous property of the original definition. In order to maximize the original notion of power, a user of a single link had to choose different optimal rates, based on noninterfering traffic. While the new definitions solve this problem somewhat, they have anomalies of their own. For example, consider the simple network of Fig. 1.

The labels of the links specify their capacity. Let $p_1 = (A, B, C)$ and $p_2 = (B, C)$. In this simple example, both of the first two new measures of power have a terribly unfair optimum. User 1 must have zero throughput at optimum

Fig. 1.

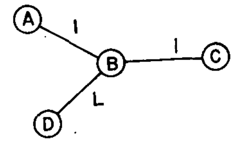

Fig. 2.

power. The third new measure of power seems to handle such situations somewhat more equitably. Thus we restrict attention to the third new measure.

In order to find two users with identical viewpoints, the optimum rates for certain networks must be computed. This was done by calculation as analytic determination of optimum power based on route normalized delay is quite difficult.

Corollary 3: Power, when defined on the basis of route normalized delay is not decentralizable.

Proof: (Sketch.) Consider again the example of Fig. 1, with $p_1 = (A, B, C)$ and $p_2 = (B, C)$. The optimum rates for the two users are (x, y) where $x \approx 0.32$ and $y \approx 0.22$. Using Theorem 1 we have that if in a different network a user had an identical viewpoint to user 1, but at optimum power had throughput different from x, then "route-normalized power" is not decentralizable.

Consider Fig. 2 with $p_1 = (A, B, C)$ and $p_2 = (D, B, C)$. Assume that for some value of L optimal throughput for user 2 is equal to y. Then user 1 in Fig. 2 would have an identical viewpoint to that of user 1 in Fig. 1 and would have to choose throughput x. For some value of $L \approx 0.62$, the optimum for user 2 is y, but the optimum for user 1 is about $0.36 \neq x$.

VII. PRODUCT OF POWERS

A deficiency in many of the performance measures considered is that to optimize a measure it may be necessary for some user to have zero throughput. A measure is now discussed [2] that avoids this difficulty.

Let $P_i(\gamma) = \gamma_i / D_i(\gamma)$ be the *individual power of user i*, where γ_i is the throughput of user i (recall throughput = message rate), and $D_i(\gamma)$ the delay of user i at throughput assignment γ. The *product of powers at* γ is $\Pi_{\text{users}, i} P_i(\gamma)$. The performance criterion based on product of powers is that γ is better than γ^* if it produces a larger product of powers. User i cannot have zero throughput at optimum product of power, since then P_i and thus $\Pi_i P_i$ would be zero. Product of powers, however, does have the following deficiency.

Corollary 4: Product of powers is not decentralizable.

Proof: Consider the single link network of Fig. 3, with $p_1 = p_2 = (A, B)$. $P_1(\gamma) = \gamma_1(1 - \gamma_1 - \gamma_2)$, $P_2(\gamma) = \gamma_2(1 - \gamma_1 - \gamma_2)$, and $\Pi P = \gamma_1 \gamma_2 (1 - \gamma_1 - \gamma_2)^2$. To maximize ΠP, compute $\partial(\Pi P)/\partial \gamma_1 = 0$ and get $\gamma_1 = (1 - \gamma_2)/3$ and $\partial(\Pi P)/\partial \gamma_2 = 0$ or $\gamma_2 = (1 - \gamma_1)/3$. Solving the two equations yields $\gamma = (1/4, 1/4)$, and $\Pi P(\gamma) = 1/64$.

Consider the scheme of networks of Fig. 4, parameterized by the variable n. Let $p_1 = (B, C)$ and $p_2 = (A_1, A_2, A_3, \cdots, A_n, B, C)$.

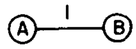

Fig. 3.


Fig. 4.

First we determine the optimal γ as $n \to \infty$. Note that

$$P_1(\gamma) = \gamma_1(1-\gamma_1-\gamma_2)$$
$$P_2(\gamma) = \frac{\gamma_2}{1/(1-\gamma_1-\gamma_2)+n/(1-\gamma_2)} \tag{15}$$
$$\Pi P = \frac{\gamma_1\gamma_2(1-\gamma_1-\gamma_2)}{1/(1-\gamma_1-\gamma_2)+n/(1-\gamma_2)}.$$

Denote ΠP by P, γ_1 by y, and γ_2 by x. To maximize

$$P = \frac{xy(1-x-y)}{1/(1-x-y)+n/(1-x)} \tag{16}$$

first compute

$$\frac{\partial P}{\partial x} = \frac{\left(\frac{1}{1-x-y}+\frac{n}{1-x}\right)(-xy+(1-x-y)(y)) - (xy)(1-x-y)\left(\frac{1}{(1-x-y)^2}+\frac{n}{(1-x)^2}\right)}{\left(\frac{1}{1-x-y}+\frac{n}{1-x}\right)^2}. \tag{17}$$

Setting $\partial P/\partial x = 0$ and simplifying yields

$$\left(\frac{(1-x)^2}{n} + (1-x)(1-x-y)\right)(y-2xy-y^2) - xy\left(\frac{(1-x)^2}{n} + (1-x-y)^2\right) = 0. \tag{18}$$

Since $0 \leqslant x, y \leqslant 1$ for every n, the $(1-x)^2/n$ can be ignored for large n. Dividing through by y yields

$$3x^2 + x(2y-x) + (1-y) = 0 \tag{19}$$

Similarly, by solving $\partial P/\partial y = 0$, one obtains

$$(1-x-y)(1-x-2y) = 0. \tag{20}$$

At $y = 1-x$, the solution is not feasible (infinite delay), and at $y = (1-x)/2$, solving into (19) yields

$$2x^2 - \tfrac{5}{2}x + \tfrac{1}{2} = 0 \tag{21}$$

Thus $x = 1/4$ and $y = 3/8$.

If a decentralized algorithm implements product of powers, the scheme of Fig. 4 tells us that if a user shares one unit capacity link with one other user, then as the second user's rate approaches 1/4, the first user's rate must approach 3/8. This contradicts the fact that in the network of Fig. 3, the user had to choose 1/4 with an identical viewpoint (using continuity).

VIII. APPROXIMATING POWER

The nondecentralizability of variants of power prompts the question of whether one could approximate power with decentralized algorithms. We show that no algorithm can approximate power if it uses a very restrictive information set. The available information is: the user can determine how much delay will result from a given choice for throughput. That is, the user is "given a function D" and may use that for any choice of throughput γ, to calculate the delay $D(\gamma)$.

An additional assumption is also needed. Assume that the two users have similar one-link paths, p_1 and p_2, except that the capacity of l_1 (the link of p_1) is some number c times that of l_2. For example, assume $c(l_2) = 1$ and $c(l_1) = c$. Then by using different sets of units to measure capacity (e.g., message/min, messages/h, etc.), one may arrive at capacities $c(l_2) = 1/c$, $c(l_1) = 1$. It is desired that the throughput chosen with capacity 1 should be fixed, irrespective of the units used. But this may only be accomplished if a user with capacity c chooses throughput c times that of a unit capacity user.

To incorporate the above notion of ignoring units into the flow control algorithms, the following formal definition is introduced. An algorithm that uses only a "delay function D" as input to its update function is *unitless* if whenever it is given two delay functions, D and D^*, such that $D(x) = (1/c)D^*(x/c)$, for every x then the throughput chosen with D is c times that chosen with D^*. Informally, $D(x) = (1/c)D^*(x/c)$ in terms of a single link means that the link represented by D has capacity c times larger than that of D^*.

When considering such severely restrictive algorithms, none of the versions of power considered in Section VI can be approximated.

Theorem 2: Any unitless flow control that uses only a "delay function" as input to its update mechanism has worst case power (for all versions of power) at least $O(n)$ times worse than optimal with n users.

Proof: Consider a user whose path has no interference and consists of a single unit capacity link. Let x be the throughput assigned by the algorithm. Due to the unitless property, if a user executes the algorithm, and has delay function $1/(c-\gamma)$, it must choose $\gamma = xc$.

Let us assume that m users have identical paths consisting of the same unit capacity link. Let γ_i be the throughput that user i converges to in a given history. For every i, $\gamma_i =$

$x(1 - \Sigma_{j \neq i}\gamma_j)$ since $\Sigma_{j \neq i}\gamma_j$ is the "effective capacity" as seen by user i. Let $T = \Sigma_{i=1}{}^m \gamma_i$

$$T = \sum_{i=1}^{m} \left(x - x \sum_{j \neq i} \gamma_j \right)$$

$$= mx - x \sum_{i=1}^{m} \sum_{j \neq i} \gamma_j$$

$$= mx - x(m-1)T. \tag{22}$$

The first equality follows from $\gamma_i = x - x\Sigma_{j \neq i}\gamma_j$, the second is just a rewrite of the first, and the third uses the fact that in the double summation, each γ_i appears $m - 1$ times.

From (22), it follows that

$$T = \frac{mx}{1 + x(m-1)}. \tag{23}$$

Using $M/M/1$ models the delay (per user) is $1/(1 - T)$, and the power (defined by $T/\overline{D}$) is

$$\frac{(mx)(1-x)}{(1 + (m-1)x)^2} \tag{24}$$

For any x $(0 < x < 1)$, as $m \to \infty$, P shrinks on the order of $1/m$.

On the other hand, the throughput assignment $\gamma_i = 1/2m$ for $i = 1, \cdots, m$ yields $T = 1/2$ and $\overline{D} = 2$. The resulting power is $O(m)$ better than the unitless algorithm.

IX. CONCLUSIONS

There are three possible approaches to take when trying to optimize power. If optimizing the criterion is truly important, then perhaps some centralization should be introduced. We believe that such an approach is inappropriate for real systems.

One might try to investigate practical algorithms and determine which are better vis-a-vis power. This approach is taken in [2]. Finally one might reevaluate the determination that power is the most desirable criterion. Decentralizable policies are discussed in [7]. In any case, it is hoped that the techniques presented herein should encourage researchers to investigate the potential nondecentralizability of their favorite objectives before expending much effort searching for decentralized algorithms.

REFERENCES

[1] M. Gerla and L. Kleinrock, "Flow control algorithms: A comparative survey," *IEEE Trans. Commun.*, vol. COM-28, Apr. 1980.

[2] K. Bharath-Kumar and J. M. Jaffe, "A new approach to performance oriented flow control," *IEEE Trans. Commun.*, vol. COM-29, Special Issue on Congestion Control in Computer Networks, Apr. 1981.

[3] A. Giessler, J. Hänle, A. König, and F. Pade, "Free buffer allocation—An investigation by simulation," *Computer Networks*, vol. 2, 1978.

[4] L. Kleinrock, *Queueing Systems*, Vol. 2. New York: Wiley, 1976.

[5] W. Rudin, *Principles of Mathematical Analysis*. New York: McGraw-Hill, 1953.

[6] L. Kleinrock, "On flow control in computer networks," presented at Int. Conf. Communications, June 1978.

[7] J. M. Jaffe, "Bottleneck flow control," *IEEE Trans. Commun.*, vol. COM-29, July 1981.

★

Jeffrey M. Jaffe (M'80) received the B.S. degree in mathematics and the M.S. and Ph.D. degrees in computer science from the Massachusetts Institute of Technology, Cambridge, in 1976, 1977, and 1979, respectively. He was a National Science Foundation Fellow while a graduate student.

He is currently employed by IBM Thomas J. Watson Research Center, Yorktown Heights, NY, where he is engaged in research on network algorithms and combinatorial optimization.

Dr. Jaffe is a member of ACM and Phi Beta Kappa.