We consider a distributed algorithm for dynamically adjusting the input rate of each session of a voice or data network so as to exercise flow control. Each session origin receives periodically information regarding the level of congestion along the session path and iteratively corrects its input rate. In this paper we place emphasis on voice networks but the ideas involved are also relevant for dynamic routing and flow control in data networks.
DYNAMIC CONTROL OF SESSION INPUT RATES
IN COMMUNICATION NETWORKS*

by

Eli Gafni† and Dimitri Bertsekas‡‡

Abstract

The authors consider a distributed algorithm for dynamically adjusting the input rate of each session of a voice or data network so as to exercise flow control. Each session origin receives periodically information regarding the level of congestion along the session path and iteratively corrects its input rate. In this paper we place emphasis on voice networks but the ideas involved are also relevant for dynamic routing and flow control in data networks.

---

*This research was conducted at the M.I.T. Laboratory for Information and Decision Systems with partial support provided by the Defense Advanced Research Projects Agency under Contract No. ONR/N00014-75-C-1183.

†The author is with the University of California at Los Angeles, Computer Science Dept., Los Angeles, California 90024.

‡‡The author is with the Massachusetts Institute of Technology, Laboratory for Information and Decision Systems, Cambridge, MA 02139.
1. **INTRODUCTION**

The purpose of this paper is to propose and investigate a flow control algorithm for adjusting session rates in a data or voice network. This algorithm is motivated by a Voice Coder scheme introduced in [1] and called "embedded coding". In this scheme a segment of talkspurt is coded into packets of different priority levels. The higher priority packets contain the "core" of the speech while the lower priority packets contain the information that "fine tunes" it. Traditional voice flow control mechanisms either block the initiation of a call or discard small segments of it while it is already in progress. By contrast the embedded coding scheme dynamically trades off between voice quality and congestion by discarding the lower "priority" packets either at the point of congestion or the point of entry. The level of congestion at which the gaps between the segments, delivered by the traditional schemes, render the speech unintelligible is much lower than the one at which the embedded coding scheme delivers unintelligible information. This flexibility in exercising flow control makes the embedded coding scheme attractive.

Alleviation and prevention of congestion by discarding lower priority packets at the point of entry seems to be superior to discarding them at the point of congestion. The latter amounts to a waste of network resources. But, it would not be advisable to forgo the capability of discarding lower priority packets at the point of congestion, because of the time delay involved in making the entry points aware of downstream congestion. Based on this we believe that both capabilities should be used. The rates at
the entry points will be determined upon longer time averages of congestion levels while the capability of discarding packets at the point of congestion will serve to alleviate intolerable momentary bottlenecks. The rates at the entry point will be adjusted so that the capability of discarding packets at the point of congestion will not be exercised too often.

In this paper we discuss a method of determining the input rates at the entry point. To this end we will ignore the capability of discarding packets within the network in order to simplify the analysis. As in quasistatic routing we employ an "on-line" iterative algorithm that will solve a static problem. The hope is that the algorithm converges fast enough relative to the session initiation and termination process, and as a result will be able to "track" its variation keeping the rates in the ballpark of the optimal rates at all times.

The criterion used to determine input rates is based on the notion of "fair allocation" introduced in Section 2. Roughly speaking the objective is to maximize the smallest session rate, and once this is achieved to maximize the second smallest rate, etc. In Section 3 we introduce the algorithm and describe its convergence properties.

The idea of the algorithm is to adjust the input rates of sessions on the basis of the current level of congestion along the session path. The necessary information is collected by a control packet sent periodically by each session origin along the path similarly as in flow control methods investigated by simulation in [1]. This method of adjusting input rates seems also suitable for other situations where fast reaction to momentary congestion is needed. For example when the number of users
in the network is small but some of these users can overload the network if left uncontrolled to transmit at maximum rate, then a dynamic method of routing and flow control is needed. The ideas of this paper can provide an alternative to existing techniques [7], [8] in such situations. Further work is required along this direction.
2. PROBLEM FORMULATION

Consider a network with nodes 1, 2, ..., N and a set of directed links \( L \). Each link \( a \in L \) has a capacity \( C_a \) associated with it—a positive number. Let \( S \) denote a set of sessions taking place between nodes. Each session \( s \in S \) has an origin node associated with it and traverses a subset of links denoted by \( L_s \). Note that we do not restrict the session to have a single destination, so the set of links \( L_s \) may be for example a tree rooted at the origin node of \( s \) and used for broadcasting messages throughout the network. We denote by \( S_a \) the set of sessions traversing a link \( a \in L \). If \( r_s \) is the input rate of session \( s \) (in data units/sec), then the flow \( F_a \) of a link \( a \in L \) is given by

\[
F_a = \sum_{s \in S_a} r_s.
\] (1)

The problem broadly stated is to choose a vector of session input rates \( r = (\ldots, r_s, \ldots) \) which results in a set of "satisfactory" link flows \( \{F_a|a \in L\} \), and at the same time maintains a certain degree of "fairness" for all sessions.

It is customary to consider as one of the characteristics of a fair allocation of resources in a network the feature that it is indifferent to the geographical separation of the session's origin and destinations. Although there might be different priorities assigned to sessions, these priorities are not assigned on the basis of geographical distance. Moreover, two sessions of the same priority should obtain the same rate, if the rate of one can be traded for the rate of the other without overloading the network or reducing the rate of any other session. This is in the spirit of making the network "transparent" to the user.

To capture the notion of fairness and priority as presented above we define the notion of fair allocation:
For a vector \( x = (x^1, x^2, \ldots, x^n) \) in the Euclidean space \( \mathbb{R}^n \), we consider the vector \( \overline{x} = (\overline{x}^1, \overline{x}^2, \ldots, \overline{x}^n) \) the coordinates of which are the same as those of \( x \) but are rearranged in order of increasing value, i.e. we have \( \overline{x}^1 \leq \overline{x}^2 \ldots \leq \overline{x}^n \) and with each \( i = 1, \ldots, n \) we can associate a distinct \( i' \) such that \( \overline{x}^i = x^{i'} \). We call \( \overline{x} \) the increasing permutation of \( x \). Given a subset \( X \) of \( \mathbb{R}^n \) we will say that a vector \( x \in X \) is a fair allocation over \( X \) if for every vector \( y \in X \) the increasing permutation \( \overline{x} \) of \( x \) is lexicographically greater or equal to the increasing permutation \( \overline{y} \) of \( y \), i.e. if \( \overline{y}^j > \overline{x}^j \) for some \( j \), then there exists an \( i < j \) such that \( \overline{x}^i < \overline{x}^j \).

If we view \( X \) as a "feasible" set, a fair allocation vector \( x \) over \( X \) solves a hierarchy of problems. The first problem is to maximize the minimal coordinate of vectors in \( X \). The second problem is to maximize the second minimal coordinate over all vectors which solve the first problem, etc.

Hayden [2] proposed an algorithm which results in a rate vector \( r = (\ldots, r_s, \ldots) \) which is a fair allocation over the set defined by

\[
F_a \leq \rho C_a, \quad \forall a \in L, \quad (2)
\]

where \( \rho \) is some constant between 0 and 1. Jaffe [3] proposed an algorithm which obtains a rate vector \( r \) such that the vector \( (\ldots, \beta_s r_s, \ldots) \) is a fair allocation over the set defined by

\[
\beta_s r_s \leq C_a - F_a, \quad \forall s \in S, \quad \forall a \in L_s, \quad (3)
\]

\[
F_a \leq C_a, \quad \forall a \in L, \quad r_s \geq 0, \quad \forall s \in S, \quad (4)
\]
where $\beta_s$ is some positive constant that characterizes the priority of session $s$.

The rationale behind the fair allocation problem based on (2) is quite simple: we maximize the minimum session rate while not allowing the flow of any link to be more than some given fraction of its capacity. The rationale behind (3), (4) is somewhat more sophisticated. Primarily it enables us to establish preferences among sessions, and to accommodate fluctuations of a session rate which depend linearly on the rate as we will demonstrate shortly.

While Jaffe's algorithm is not iterative and as a result is somewhat unsuitable for distributed operation, Hayden's algorithm may result in transient flows that are larger than some link capacities (for an example see [4], p. 39).

Our purpose in this paper is to propose and analyze an iterative algorithm that solves a problem that is more general than Jaffe's [3], maintains at all times feasibility of link flows with respect to capacities, and is suitable for distributed operation. To this end we generalize the set defined by (3), (4) as follows:

For each link $aeL$ and session $seS$ let $g_a: R^+ \rightarrow R^+$ and $\beta_s: R^+ \rightarrow R^+$ be functions mapping the nonnegative portion of the real line $R^+$ into itself. We are interested in finding a rate vector $r$ such that the vector $(..., \beta_s(r_s), ...)$ is a fair allocation over the set defined by

$$\beta_s(r_s) \leq g_a(C_a-F_a), \quad \forall \ seS, \ aeL_s$$

$$F_a \leq C_a, \quad \forall \ aeL, \ r_s \geq 0, \ \forall \ seS.$$
A vector $r$ with this property will be called a **fair allocation rate**.

We make the following assumptions regarding the functions $g_a(\cdot)$ and $b_s(\cdot)$:

**Assumption A:** For all $a \in L$, $g_a(\cdot)$ is monotonically nondecreasing, and, for all $s \in S$, $b_s(\cdot)$ is continuous, monotonically increasing, and maps $\mathbb{R}^+$ onto $\mathbb{R}^+$. (This implies also that the inverse $b_s^{-1}(\cdot)$ exists, is continuous, monotonically increasing and maps $\mathbb{R}^+$ onto $\mathbb{R}^+$).

**Assumption B:** The function $H_{sa}(\cdot)$ defined by

$$H_{sa}(f) = b_s^{-1}[g_a(f)], \quad \forall s \in S, \forall a \in L, f \in \mathbb{R}^+$$

is convex and differentiable on $\mathbb{R}^+$ and satisfies

$$H_{sa}(0) = 0.$$

Assumption B is not very restrictive. It is satisfied in particular if both $b_s^{-1}(\cdot)$ and $g_a(\cdot)$ are convex, differentiable and monotonically increasing on $\mathbb{R}^+$, and $g_a(0) = 0$. Also the convexity assumption in Assumption B can be replaced by a concavity assumption without affecting the convergence result of the next section, but this will not be pursued further.

The introduction of the nonlinear function $b_s(\cdot)$ allows us to assign different priorities to different sessions in a more flexible manner than in (3), and allows additional freedom in mathematically expressing algorithmic design objectives. As an example let us provide justification for the use of a particular form for $g_a$ in the case where each session is a voice conversation.
Suppose that the length of time over which each session rate is averaged is short relative to the "time constant" of the counting process of the number of off-hook speakers which are currently in talkspurt mode. Since about 30% of a talkspurt is silence and some segments of the talkspurt need more encoding than others, we view the bit rate generated by the Vocoder for session \( seS \) as a stochastic process with mean \( r_s \) --the rate assigned to user \( seS \). We thus implicitly assume that the Vocoder has the means of dynamically reconfiguring to the demands of the voice to achieve the desired average rate. Suppose that we want to reserve excess capacity on each link so as to be able to accommodate a variation at least as large as the standard deviation of the flow on the link. Assume that the standard deviation of the rate of each session \( seS \) is \( \gamma \cdot r_s \) where \( 0 < \gamma < 1 \). For a fixed link \( aeL \) let \( s' \in S \) be such that

\[
s' = \arg \max_{seS_a} r_s.
\]

Then, by the independence of the rates of different sessions, we have, assuming \( F_a \leq C_a \), that the standard deviation \( \sigma(F_a) \) of the flow \( F_a \) satisfies

\[
\sigma(F_a) = \sqrt{\sum_{seS_a} [\sigma(r_s)]^2} = \gamma \sqrt{\sum_{seS_a} r_s^2} \\
\leq \gamma \sqrt{\left( \sum_{seS_a} r_s \right) r_s'} \leq \gamma \sqrt{C_a r_s'}
\]

Suppose we take in (5)
Then from (5), (7) and (8) we obtain

\[ \sigma(F_a) \leq \gamma \sqrt{C_a r_s} \leq \gamma \sqrt{C_a g_a (C_a - F_a)} = C_a - F_a. \]

We are thus guaranteed to be able to accommodate the standard deviation of the flow resulting from the fair allocation.

In the second interpretation, the length of time over which the rate is averaged is relatively long with respect to the "time constant" of the counting process of the number of off-hook speakers in talkspurt mode. In this case we deal concurrently with all the off-hook sessions and want to be able to accommodate the standard deviation around the mean of the process (i.e., the instantaneous effect of the number of speakers at the talkspurt mode is washed out by the long time average).

Let \( q \) be the fraction of time a speaker is in the talkspurt mode and assume his rate while in the talkspurt mode is constant. Then using notations as before

\[ \sigma(F_a) = \left[ \sum_{s \in S_a} \frac{(r_s / q)^2 \cdot q(1-q)} {q(1-q)} \right]^{1/2} \]

\[ \leq \left( \frac{1-q} {q} C_a r_s' \right)^{1/2} \]

\[ \leq \left( \frac{1-q} {q} C_a \right)^{1/2} \left[ g_a (C_a - F_a) \right]^{1/2}. \]
Again, by choosing \( g_a \) as in (8) with \( \gamma = \left( \frac{q}{1-q} \right)^{1/2} \) we obtain \( \sigma(F_a) \leq C_a - F_a \).

The point we want to make by the above arguments is that there is often a need to allow \( g_a \) to be a nonlinear function, which may depend also on \( C_a \), rather than only on the excess capacity as (3) implies. The exact role of \( g_a \) is up to the network designer to decide, and our formulation allows him a great deal of flexibility in this regard.

It is possible to show that Assumptions A and B guarantee existence and uniqueness of a fair allocation rate. The proof given below is constructive and is based on a finitely terminating algorithm. However this algorithm, in contrast with the one of the next section, is not suitable for distributed, on-line operation since it must be restarted each time an old session is terminated or a new one is initiated.

Consider first the problem of finding a vector \( r = (\ldots, r_s, \ldots) \) that maximizes
\[
\min_{s \in S} \beta_s(r_s)
\]
over the feasible set
\[
R_0 = \{ r \mid (5) \text{ and (6) are satisfied} \}
\]
This is the first problem in the hierarchy of problems solved by a fair allocation rate, and can be solved simply by observing that its optimal value [i.e. max \( \min_{s \in S} \beta_s(r_s) \)] is equal to
\[
\omega^*_1 = \max \{ w \mid w \leq g_a[C_a - \sum_{s \in S_a} \beta_s^{-1}(w)], a \in L \}.
\]
This follows easily from the fact that both $g_a$ and $\beta_s^{-1}$ are monotonically nondecreasing. Denote

$$L^*(1) = \{a \in L \mid w_1^* = g_a [C_a - \sum_{s \in S} \beta_s^{-1}(w_1^*)]\}$$

$$S^*(1) = \{s \in S \mid L \cap L^*(1) \neq \emptyset\}.$$

For any fair allocation rate $(\ldots, r_s, \ldots)$ the rate of the sessions in $S^*(1)$ is equal to $\beta_s^{-1}(w_1^*)$, i.e.

$$r_s = \beta_s^{-1}(w_1^*), \quad \forall s \in S^*(1),$$

while $L^*(1)$ may be viewed as the set of bottleneck links the presence of which does not allow us to increase $\min_{s \in S} \beta_s(r_s)$ beyond the level $w_1^*$. Therefore for the purposes of determining further a fair allocation rate vector, the rates of the sessions $s \in S^*(1)$ are fixed at $\beta_s^{-1}(w_1^*)$ and we can consider a reduced network whereby the links $a \in L^*(1)$ and sessions $s \in S^*(1)$ are eliminated while the capacity $C_a$ of each link $a \in L^*(1)$ is replaced by

$$C_a - \sum_{s \in S^*(1) \cap S} \beta_s^{-1}(w_1^*).$$

If $S^*(1) = S$ we are done; otherwise we can consider the problem of maximizing the minimal value of $\beta_s(r_s)$ in the reduced network similarly as earlier. This will determine a new optimal value $w_2^*$ with $w_2^* > w_1^*$, a
a new set of bottleneck links \( L^*(2) \), and a set of sessions \( S^*(2) \) such that
\[
\beta_s(r_s) = w^*_2, \quad \forall s \in S^*(2)
\]
in any fair allocation vector. If \( S^*(1) \cup S^*(2) = S \), we are done; otherwise we can proceed by constructing a reduced network and continue in the same manner as earlier until we exhaust all sessions. This argument constructs a fair allocation rate \( r^* \) and shows that it is uniquely defined in terms of the scalars \( w^*_1, w^*_2, \ldots \), and the corresponding sets \( S^*(1), S^*(2), \ldots \). Note also that the session rates \( r^*_s, s \in S \) and associated link flows \( F^*_a, a \in L \) satisfy
\[
r^*_s = \min_{a \in L_s} \min_{s \in S} g_a(C_a - F_a), \quad \forall s \in S.
\]
The algorithm of the next section is based on this property.

We can also show the reverse property, namely that if a rate vector \( r^* \) satisfies (10) then it is a fair allocation rate. To see this let \( \tilde{r} = (\ldots, r^*_s, \ldots) \) satisfy (10) or equivalently
\[
\beta_s(\tilde{r}_s) = \min_{a \in L_s} g_a(C_a - F^*_a), \quad \forall s \in S.
\]
Observe that from the definition (9) of \( w^*_1 \) and (11) we obtain
\[
w^*_1 \geq \tilde{w}_1
\]
where
\[
\tilde{w}_1 = \min_{s \in S} \min_{a \in L_s} g_a(C_a - F^*_a).
\]
Let $\tilde{a} \in L$ be any link such that

$$g_a(C_a - \tilde{F}_a) = \tilde{w}_1.$$ 

Then from (11) we have

$$\tilde{r}_s = \beta_s^{-1}(\tilde{w}_1), \quad \forall s \in S_a.$$ 

The inequality $\tilde{w}_1 < w^*_a$ implies that $\tilde{F}_a = \sum_{s \in S_a} \tilde{r}_s \leq \sum_{s \in S_a} r^*_s = F^*_a$

where $r^*$ is the fair allocation rate. But this implies that

$$\tilde{w}_1 = g_a(C_a - \tilde{F}_a) \geq g_a(C_a - F^*_a) \geq w^*_1.$$ 

Therefore we must have $\tilde{w}_1 = w^*_1$ and it follows that the vector $\tilde{r}$ solves the first problem in the hierarchy of the fair allocation problem.

Proceeding similarly as earlier we can show that $\tilde{r}$ solves all the problems in the hierarchy of the fair allocation problem and is therefore a fair allocation rate.

We summarize the conclusions from the preceding arguments in the following proposition.

**Proposition 1:** Let Assumptions A and B hold.

a) There exists a unique fair allocation rate.

b) $r^* = (\ldots, r^*_s, \ldots)$ is a fair allocation rate if and only if it satisfies

$$\beta_s(r^*_s) = \min_{a \in L_s} g_a(C_a - r^*_s), \quad \forall s \in S$$

or equivalently

$$\beta_s(r^*_s) = \min_{a \in L_s} g_a(C_a - r^*_s), \quad \forall s \in S$$

(12)
In some situations it may be reasonable to consider, in addition to (5) and (6), the constraint

\[ r_s \leq R_s, \; \forall \; s \in S \]  \hspace{1cm} (14)

where \( R_s \) is given upper bound to the rate of session rate \( s \). We may view \( R_s \) either as a limit on rate imposed by technological restrictions or as a maximum desired rate by session \( s \). The problem of finding a fair allocation rate over the set defined by (5), (6), and (14) can be reduced to the problem considered earlier by introducing for each \( s \in S \), an artificial link \( a_s \) traversed only by session \( s \) by setting the capacity \( C_{a_s} \) of that link equal to

\[ C_{a_s} = R_s + \beta_s(R_s), \]

and by selecting the function \( g_{a_s} \) to be the identity. Then the constraint

\[ \beta_s(r_s) \leq g_{a_s}(C_{a_s} - r_s) = C_{a_s} - r_s \]

becomes \( r_s + \beta_s(r_s) \leq C_{a_s} = R_s + \beta_s(R_s) \) and is equivalent to (14).
3. **THE ALGORITHM**

Let \( r^k = (\ldots, r_s^k, \ldots) \) be the rate vector obtained after \( k \) iterations and let \( \{ F_a^k \} \) be the corresponding set of total link flows. Assume that

\[
0 < F^k_a < C_a, \quad \forall a \in L. \tag{15}
\]

The new rate vector \( r^{k+1} = (\ldots, r_s^{k+1}, \ldots) \) obtained at the \((k+1)\)st iteration is given by

\[
r_s^{k+1} = \min_{a \in L_s} \{ r_s^k + \gamma_a^k [H_{sa}(C_a - F_s^k) - r_s^k] \}, \quad \forall s \in S \tag{16}
\]

where \( \gamma_a^k \) is given by

\[
\gamma_a^k = \frac{1}{1 + \sum_{s \in S_a} H'_{sa}(C_a - F_a^k)} \tag{17}
\]

and \( H'_{sa}(\cdot) \) denotes the first derivative of \( H_{sa}(\cdot) \). In a practical implementation of the algorithm the link flows \( F_a^k \) can either be measured (by taking a time average), or they can be mathematically computed as the sum of the session rates \( r_s^k, s \in S_a \). The session rates computed via (16), (17) will have to be translated into physical rates by software residing at the session origin nodes.

The following lemma shows among other things that property (15) is maintained by the algorithm at each iteration and therefore if the initial link flows \( F_a^0 \) are within the link capacities \( C_a \) the same is true for all link flows generated by the algorithm.

**Lemma 1:** Let Assumptions A and B hold and assume that the initial rate vector \( r^0 \) is such that
0 ≤ r_s^0, V \in S, 0 ≤ F_a^0 < C_a, V \in \Lambda. \quad (18)

Then if \{r_k\} is a rate sequence generated by the algorithm (16) with \gamma_k given by (17) we have for all k

0 ≤ r_s^k, V \in S, 0 ≤ F_a^k < C_a, V \in \Lambda. \quad (19)

Furthermore

F_a^k ≤ \sum_{s \in S_a} H_s(C_a - F_a^k), V \in \Lambda, k ≥ 1. \quad (20)

**Proof:** See Appendix A.

The idea behind the choice of expression (17) as well as the intuition behind Lemma 1 can be best explained by the use of Figure 1. Let the function

\[ G_a(\cdot): R^+ \to R^+ \]

given by

\[ G_a(F_a) = \sum_{s \in S_a} H_s(C_a - F_a). \quad (21) \]

The figure depicts the relations between \( F_a^k \) and \( F_a^{k+1} \), as if the network consisted of the single link a. \( F_a^{k+1} \) is determined by intersection the tangent to the graph of \( G_a(F_a) \) at the point \( (F_a^k, G_a(F_a^k)) \), with the line \( y = F_a \). The reader can easily convince himself that \( \limsup_{k \to \infty} F_a^k \) must lie in the area where

\[ F_a \leq G_a(F_a) \]

which gives rise to the lemma. Figure 2 shows why just monotonicity of \( G_a(\cdot) \) is not sufficient for the lemma to hold.

We can now state the main result of this paper.
Figure 1
Figure 2
Proposition 2: Under the assumptions of Lemma 1 the sequence \( \{r_k\} \) converges to the fair allocation rate.

Proof: See Appendix A.

A Variation of the Algorithm.

In iteration (16) we have assumed that updating of all the rates \( r_s \) takes place simultaneously. It is possible to consider other related algorithms whereby a single session rate \( r_s \) is updated using (16), then the flows \( F_a \) are updated to reflect the change in \( r_s \), then another session rate is updated using (16) and so on until all the session rates are taken up cyclically in a fixed order. This one-session-at-a-time mode of operation is reminiscent of the Gauss-Seidel method for solving systems of equations and is perhaps better suited for distributed implementation. It is possible to show that all the results of this section hold for this modified algorithm as well.

A question of considerable interest is whether a totally asynchronous distributed version of algorithm (16) will work satisfactorily (compare with algorithms investigated in [5], [6]). In such an algorithm each session origin sends at arbitrary times along the session path a control packet containing the current rate of the session. As the packet travels to its destination the information needed to compute the right side of (16) is collected. (We assume here a single destination per session and that each link \( a \) on the session path maintains the current value of \( F_a \) as the sum of all currently assigned session rates \( r_s \), \( s \in S_a \), and
the form of the function $H_{sa}(\cdot)$ for each $seS_a$). The destination returns the new rate to the session origin and the links along the session path. This type of algorithm is very attractive from the practical point of view since it does not require a session synchronization protocol. Its convergence properties however are as yet unclear and are currently under investigation. It is interesting to note that some of the algorithms investigated by simulation in [1] are of similar nature.
REFERENCES


APPENDIX A

Proof of Lemma 1:

It suffices to show (19)-(20) for \( k = 1 \). Consider the function

\[ G_a : \mathbb{R}^+ \rightarrow \mathbb{R}^+ \text{ defined by} \]

\[ G_a(F_a) = \begin{cases} \sum_{s \in S_a} H_{sa}(C_a - F_a) & \text{if } 0 \leq F_a \leq C_a \\ 0 & \text{if } F_a > C_a. \end{cases} \]

From (16) and (17) we have

\[ r^1_s = \min_{a \in L_s} \left\{ r^0_s + \frac{1}{1 - G'_a(F^0_a)} [H_{sa}(C_a - F^0_a) - r^0_s] \right\}, \forall s \in S \]

or

\[ r^1_s = \min_{a \in L_s} \left\{ \frac{H_{sa}(C_a - F^0_a) - r^0_s G'_a(F^0_a)}{1 - G'_a(F^0_a)} \right\}, \forall s \in S. \] (A.1)

Since \( G_a(\cdot) \) is monotonically nonincreasing we have \( G'_a(F^0_a) \leq 0 \), and since also \( H_{sa}(C_a - F^0_a) \geq 0 \) we obtain from the hypothesis \( r^0_s > 0 \) and (A.1)

\[ r^1_s > 0, \quad \forall s \in S, \] (A.2)

and therefore also

\[ F^1_a > 0, \quad \forall a \in L. \] (A.3)
From (A.1) we have

\[ r_s^1 \leq \frac{H_s (C_F) - r_s G'(F)}{1 - G'(F)} \cdot \forall s \in S, a \in L_s \]

and by adding over all \( s \in S \) we obtain

\[ r_s^1 \leq \frac{G_a(F) - r_s G'(F)}{1 - G'(F)} \cdot \forall a \in L. \]

Since \( 1 - G'(F) > 0 \) we obtain from the inequality above

\[ r_s^1 \leq G_a(F) + (r_s^1 - r_s) G'(F) \]

Since \( G_a(\cdot) \) is convex the right side of this inequality is less or equal to \( G_a(F) \) and we obtain

\[ r_s^1 \leq G_a(F). \]  \hspace{1cm} (A.4)

Since \( G_a(\cdot) \) is monotonically nonincreasing and \( G_a(F) = 0 \) for \( F > C_a \) we obtain from (A.4)

\[ r_s^1 < C_a. \]  \hspace{1cm} (A.5)

From (A.2)-(A.5) we obtain (19) and (20). Q.E.D.

Proof of Proposition 2:

Denote

\[ r_s^* = \liminf_{k \to \infty} r_s^k \cdot \forall s \in S. \]
Fix \( s \in S \) and consider a subsequence \( \{r^k_s\}_{k \in K_s} \) converging to \( r^*_s \). We have from (16):

\[
 r^*_s = \lim_{k \to \infty} \min_{a \in L_s} \left\{ r^{k-1}_s + \gamma^{k-1}_a \left[ H_{s_s} (C_a - F^{k-1}_a) - r^{k-1}_s \right] \right\}. 
\]

Since \( L_s \) consists of a finite number of links we may assume (by passing to a subsequence of \( K_s \) if necessary) that there exists a link \( a_s \) such that:

\[
 r^*_s = \lim_{k \to \infty} \{ r^{k-1}_s + \gamma^{k-1}_a \left[ H_{s_s} (C_a - F^{k-1}_a) - r^{k-1}_s \right] \}. 
\] (A.6)

Since \( \{ F^{k-1}_a \} \) is bounded above and below we may assume (by passing to a subsequence of \( K_s \) if necessary) that for some \( a_s \) and some \( \tilde{a}_s \):

\[
 \lim_{k \to \infty} F^{k-1}_{a_s} = \tilde{F}_{a_s}.
\]

Denote also:

\[
 \tilde{\gamma}_{a_s} = \frac{1}{1 - G(a_s)^{-1}} = \lim_{k \to \infty} \gamma^{k-1}_{a_s}. 
\]

We have from (A.6):

\[
 r^*_s > (1 - \tilde{\gamma}_{a_s}) r^*_s + \tilde{\gamma}_{a_s} H_{s_s} (C_a - \tilde{F}_a) + \gamma_{a_s} H_{s_s} (C_a - F^{k-1}_a) - r^{k-1}_s
\]

\[
 \geq (1 - \gamma_{a_s}) r^*_s + \gamma_{a_s} H_{s_s} (C_a - \limsup_{k \to \infty} F^k_a)
\]
and finally
\[ r_s^* \geq H_{s a_s} (C_{a_s} \limsup_{k \to \infty} F_k) . \]  
\[ (A.7) \]

Since the choice of \( s \) was arbitrary we conclude that for every \( s \in S \) there exists \( a_s \in L_s \) such that \( (A.7) \) holds.

Let \( a_1 \in L \) be such that
\[ a_1 = \arg \min_{a \in L} g_a (C_a - \limsup_{k \to \infty} F_k) \]

Using the monotonicity of \( \beta_s^{-1} \) we obtain
\[ H_{s a_s} (C_{a_s} \limsup_{k \to \infty} F_k) \geq H_{s a_1} (C_{a_1} \limsup_{k \to \infty} F_k) \]

and therefore from \( (A.7) \)
\[ r_s^* \geq H_{s a_1} (C_{a_1} \limsup_{k \to \infty} F_k), \quad \forall s \in S_{a_1} . \]  
\[ (A.8) \]

Summing \( (A.8) \) over all \( s \in S_{a_1} \) we obtain
\[ \liminf_{k \to \infty} F_k^{a_1} \geq \sum_{s \in S_{a_1}} \liminf_{k \to \infty} r_s^* = \sum_{s \in S_{a_1}} r_s^* \]

\[ \geq \sum_{s \in S_{a_1}} H_{s a_1} (C_{a_1} \limsup_{k \to \infty} F_k) \]

On the other hand from (20) we have
\[
\limsup_{k \to \infty} F^k_{a_1} \leq \sum_{s \in S} H_{sa_1} (C_{a_1} - \limsup_{k \to \infty} F^k_{a_1}).
\]

It follows that the last two inequalities as well as (A.8) hold as equations, the entire sequence \( \{F^k_{a_1}\} \) converges to \( \sum_{s \in S} r_s^* \) while each sequence \( \{r^k_{s}\}, \ s \in S \), converges to \( r_s^* \).

Consider now a new network derived from the previous one by deleting link \( a_1 \), and all the sessions traversing it. We consider the algorithm executed in the same manner as before with the same initial rates for the remaining sessions but with the capacity of each link \( a \in L \) replaced by

\[
C_a - \sum_{s \in S} r^k_s.
\]

This will result in the same rate sequence for the sessions \( s \in S \) as in the original algorithm. A trivial modification of the argument used to show (20) in Lemma 1 shows that we will have

\[
\limsup_{k \to \infty} r^k_a \leq \sum_{s \in S} H_{sa} (C_a - \limsup_{k \to \infty} F^k_a) \quad \forall \ a \in L.
\]

This relation can be used to repeat the argument given earlier in order to show the convergence of the sequences \( \{r^k_{s}\}, \ s \in S \) to \( r_s^* \) for all sessions \( s \in S \) traversing the link \( a_2 \) where

\[
a_2 = \arg \min_{a \in L \setminus \{a_1\}} g_a (C_a - \limsup_{k \to \infty} F^k_a).
\]

By repeating this procedure we will eventually exhaust all links thereby showing that each rate sequence \( \{r^k_s\}, \ s \in S \) converges to \( r_s^* \), each
each flow sequence \( \{F^k_a\} \), \( a \in L \) converges to \( F^*_a = \sum_{s \in S_a} r^*_s \) and each stepsize sequence \( \{\gamma^k_a\} \), \( a \in L \) converges to

\[
\gamma^*_a = \frac{1}{1 + \sum_{s \in S_a} H'(C_a - F^*_a)}. 
\]

By taking limits in (16) we have for all \( s \in S \)

\[
r^*_s = \lim_{k \to \infty} \min_{a \in L_s} \{r^k_s + \gamma^k_a[H_s a (C_a - F^k_a) - r^k_s]\} \tag{A.9}
\]

\[
= \min_{a \in L_s} \lim_{k \to \infty} \{r^k_s + \gamma^k_a[H_s a (C_a - F^k_a) - r^k_s]\}
\]

where the interchange of \( \min \) and \( \lim \) above is valid since all the sequences inside the braces converge and the number of elements of \( L_s \) is finite.

From (A.9) we obtain for all \( s \in S \)

\[
\min_{a \in L_s} \gamma^*_a[H_s a (C_a - F^*_a) - r^*_s] = 0
\]

Since \( \gamma^*_a > 0 \) for all \( a \in L_s \) we obtain

\[
r^*_s = \min_{a \in L_s} H_s a (C_a - F^*_a), \quad \forall s \in S.
\]

The result now follows from Proposition 1 [cf. (13)]. Q.E.D.
## Distribution List

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Copies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defense Documentation Center</td>
<td>12 Copies</td>
</tr>
<tr>
<td>Cameron Station</td>
<td></td>
</tr>
<tr>
<td>Alexandria, Virginia 22314</td>
<td></td>
</tr>
<tr>
<td>Assistant Chief for Technology</td>
<td>1 Copy</td>
</tr>
<tr>
<td>Office of Naval Research, Code 200</td>
<td></td>
</tr>
<tr>
<td>Arlington, Virginia 22217</td>
<td></td>
</tr>
<tr>
<td>Office of Naval Research</td>
<td>2 Copies</td>
</tr>
<tr>
<td>Information Systems Program</td>
<td></td>
</tr>
<tr>
<td>Code 437</td>
<td></td>
</tr>
<tr>
<td>Arlington, Virginia 22217</td>
<td></td>
</tr>
<tr>
<td>Office of Naval Research</td>
<td>1 Copy</td>
</tr>
<tr>
<td>Branch Office, Boston</td>
<td></td>
</tr>
<tr>
<td>495 Summer Street</td>
<td></td>
</tr>
<tr>
<td>Boston, Massachusetts 02210</td>
<td></td>
</tr>
<tr>
<td>Office of Naval Research</td>
<td>1 Copy</td>
</tr>
<tr>
<td>Branch Office, Chicago</td>
<td></td>
</tr>
<tr>
<td>536 South Clark Street</td>
<td></td>
</tr>
<tr>
<td>Chicago, Illinois 60605</td>
<td></td>
</tr>
<tr>
<td>Office of Naval Research</td>
<td>1 Copy</td>
</tr>
<tr>
<td>Branch Office, Pasadena</td>
<td></td>
</tr>
<tr>
<td>1030 East Greet Street</td>
<td></td>
</tr>
<tr>
<td>Pasadena, California 91106</td>
<td></td>
</tr>
<tr>
<td>Naval Research Laboratory</td>
<td>6 Copies</td>
</tr>
<tr>
<td>Technical Information Division, Code 2627</td>
<td></td>
</tr>
<tr>
<td>Washington, D.C. 20375</td>
<td></td>
</tr>
<tr>
<td>Dr. A. L. Slafkosky</td>
<td>1 Copy</td>
</tr>
<tr>
<td>Scientific Advisor</td>
<td></td>
</tr>
<tr>
<td>Commandant of the Marine Corps (Code RD-1)</td>
<td></td>
</tr>
<tr>
<td>Washington, D.C. 20380</td>
<td></td>
</tr>
</tbody>
</table>