DEFINING and ACHIEVING FAIRNESS in COMMUNICATION NETWORKS with APPLICATION to IEEE 802.6

A. A. Fredericks

Monmouth College, West Long Branch, NJ, 07722, USA

In this paper we consider the problem of defining, evaluating and achieving fairness objectives in communications networks using the proposed IEEE 802.6 MAN DQDB protocol as a concrete and important example. For this example, we quantify several aspects of a currently proposed (for a single priority) remedy for throughput fairness including the rate of convergence of the algorithm and the resulting access delays. The latter are shown to necessitate some modifications to the protocol which we suggest. We also consider more general fairness objectives for throughput and provide and evaluate an algorithm for achieving these. We extend our results to multiple priorities and suggest and evaluate an algorithm for achieving fairness in this multiple priority case. Finally, we briefly discuss continuing work on more complex schemes to achieve fairness and on other network architectures.

1. Introduction

In the provisioning and especially in the pricing of data communications network services, it is essential that customers perceive that their costs are reasonable and fair. In particular, it is important that customers believe that the performance characteristics of the services they receive, e.g. network access time and throughput, are commensurate with that received by other customers with the same tariff. However, there are networks whose (proposed) operations are known to have fairness problems. An important example is the currently proposed IEEE 802.6 MAC protocol for the DQDB network architecture. This fairness issue has been recognized for some time (e.g. see [1]-[4]) and several proposals for improving this protocol's fairness have been proposed and some of these suggestions tentatively accepted (e.g. see [4]-[8]). In [4] several definitions of fairness are given one of which was based on giving all active stations an equal share of bandwidth. It was shown there that this objective could be achieved (at least in equilibrium) if each station let a certain fixed fraction of empty slots that they could have used go by idle. (A version of this approach has been proposed and tentatively accepted by the IEEE 802.6 committee.) In Section 2 we study the convergence properties of this algorithm for the single priority case. We expand on some of the fairness definitions given in [4] (again for a single priority) and develop and evaluate the performance of specific algorithms both with respect to throughput and access delays. The analysis of these delays reveals a shortcoming in the current protocol leading us to suggest a change to achieve improved fairness for access delays. Next, in Section 3, we address the multiple priority case, again both from the point of view of throughput and access delays. We find that as currently proposed, the 802.6 protocol does not seem to provide the desired discrimination between priority classes. We propose and evaluate a simple protocol enhancement to solve this problem.

In the above, all of the proposals made and evaluated assume the use of only information that is available under the current IEEE 802.6 proposal, i.e. no modifications or additions to the current cell information is assumed. It is clear that the use of additional information could greatly enhance the capability of the network to achieve fairness in an efficient manner. In Section 4 we briefly discuss some such alternatives that are currently being studied as well as others that seem to have promise. We also comment briefly on the implications of this study on fairness in other communications architectures and on continuing work in this area.

2. Single Priority Case

We begin our study with the case where there is only one priority, i.e., if many stations are transmitting, then they are all using the same priority. We will be concerned with both the sharing of bandwidth for file transfers as well as access delays for short messages.

2.1. Meeting Fairness Performance Objectives in the Single Priority Case

In [4] a general framework for defining fairness is presented. What is clear from [4] is that many factors influence the concept of fairness including the time frame over which fairness is achieved, the difference between fairness for short messages versus large file transfers, the different view of heavy and light users, etc. Shortly we will add to these the need for relative fairness between priorities. It is impossible to come up with a single general definition of fairness although it is clear that any definition and enforcement must
account for the quality of service that the customer has subscribed to - and paid for. Hence we will present an approach that can be used for a wide class of fairness objectives and illustrate its use via concrete meaningful examples. As noted earlier, in this paper we will consider only those algorithms that can be implemented with the information available in the currently proposed protocol. Hence we will have to settle in most cases for approximate fairness objectives.

**Bandwidth sharing for file transfers**

We take as a general objective that for any two stations, i and j, that are a subset of say N active stations on a given bus, the ratio of the relative throughput shall be \( t_i/t_j = \beta_i/\beta_j \). These values would be determined from the class of service that the customer subscribes to. That decision, in turn, will be influenced by the expected traffic for each station, the performance requirements for transfers, etc. For example if every station on a bus expects to be transmitting at a given equal rate to all other stations then a reasonable choice is for station i to subscribe to a \( \beta_i \) proportional to \( N + 1 - i \) if there are \( N + 1 \) stations. (e.g. see [4].)

**Access delay for short messages**

We take as a general objective that the access delays should be approximately the same for all stations, independent of position, or at least below some reasonable delay objective. Moreover, we assume that there are some "short" messages for which access delay is an important consideration. We further assume that the total of all such short messages places an insignificant load on the system.

### 2.2 Analysis for One Priority

#### 2.2.1 Transient Behavior of the \( \alpha \) Policy

In [4] it was shown that if each of N active stations (transferring files) on a given bus let a fixed fraction, \( \alpha \), of otherwise usable slots go by, then, in equilibrium, each station would receive exactly the same fraction, \( t \), of the total bandwidth where \( t \) is given by

\[
t = \frac{1-\alpha}{N-(N-1)\alpha} \quad (1)
\]

Moreover, the total fraction of bandwidth used, \( T \), and the fraction, \( W \), of bandwidth wasted, are given by

\[
T = Nt = \frac{N(1-\alpha)}{N-(N-1)\alpha} \quad (2)
\]

and

\[
W = 1-T = \frac{\alpha}{N-(N-1)\alpha} \quad (3)
\]

In principle, one could minimize the wasted bandwidth, \( W \), by choosing \( \alpha \) very small; however, the rate of convergence to the "fair" state depends inversely on this parameter. To study this convergence, we assume that we have \( N \) stations which, for simplicity, are equally spaced. We discretize time into units corresponding to the number of slots between stations and assume that prior to time \( n_0 \) some subset of the stations are transmitting. At time \( n_0 \) the remaining stations begin to transmit. Let \( t_{i,n} \) be the fraction of bandwidth used by station i during the \( n^{th} \) time period (measured from \( n_0 \)). Then it is easy to see that the \( t_{i,n} \) satisfy

\[
t_{i,n} = (1-\alpha)(1-\sum_{j=1}^{N}(f_{j,n-i-j} - \sum_{j=i+1}^{N} f_{j,n-i})) \quad (4)
\]

Denote the residual error by \( e_{i,n} = t_{i,n} - t_{i,n}^{*} \) where \( t_{i,n}^{*} \) is the appropriate equilibrium value from equation (1). The \( e_{i,n} \) satisfy the vector equation

\[
e_{n} = \sum_{j} A_{i,j} e_{n-j} \quad (5)
\]

where \( e_n = (e_{1,n}, e_{2,n}, \ldots, e_{N,n}) \) and

\[
A_{i,k} = \begin{cases} 
(1-\alpha), & k = i+j \\
0, & \text{otherwise}
\end{cases}
\]

Seeking solutions of the form \( e_n = Z^n e_0 \) leads to the requirement that the determinant of the matrix \( A[N] \) be 0 where

\[
A[N] = \begin{bmatrix}
Z^{N-1}b & Z^{N-2}b & Z^{N-3}b & \ldots & 1 \\
Z^{N-2}b & Z^{N-1}b & Z^{N-2}b & \ldots & Z \\
\ldots & \ldots & \ldots & \ldots & \ldots \\
1 & Z & Z^2 & \ldots & Z^{N-1}b
\end{bmatrix}
\]

and \( b = 1/(1-\alpha) \).

(Note that equation (4) can be readily modified to account for the case where stations only use a fraction, \( f_i \) (possibly \( f_i = 0 \)), of the bandwidth available to it. This also allows us to consider the case of variable station separation.)

For \( N = 2 \), the dominant root (i.e. closest to 1) is given by \( Z = 1/b = (1-\alpha) \). Thus the transient throughput converges to the equilibrium value at the rate of \( (1-\alpha) \) every \( \tau \), the time unit associated with the number of slots between the two stations. For \( N = 3 \) it is easy to see that the dominant root is given by \( Z = 1/b^{1/2} = (1-\alpha)^{1/2} \). Thus convergence is at the rate \( (1-\alpha)^{1/2} \) per \( \tau \) time units or \( (1-\alpha) \) per \( 2\tau \) time units - a not
unexpected result given the convergence rate for \( N = 2 \). Unfortunately, this behavior of the dominant does not appear to hold (at least not precisely) for larger \( N \). In particular, the dominant root for \( N = 4 \) is not \( (1-\alpha)^{1/3} \). However, one can show that for arbitrary \( N \), the dominant root is asymptotic to \( (1-\alpha)^{1/(N-1)} \) at least as \( \alpha \) tends to 0 or 1.

To obtain some insight into the convergence we consider a simple example with 10 stations active with an additional one then coming online.

\[
\beta_i t = (1-\alpha)(1-i\sum_j \beta_j) \tag{7}
\]

from which we find that

\[
\alpha_i = \frac{1-i\sum_j \beta_j}{1-i\sum_j \beta_j} \tag{8}
\]

For a given fixed value of \( \alpha_i \), the first station transmitting we have that

\[
t = \frac{1-\alpha_i}{\sum_j \beta_j - \alpha_i \sum_j \beta_j} \tag{9}
\]

and thus

\[
T = \frac{\sum_j \beta_j (1-\alpha_i)}{\beta_i \alpha_i (1-\alpha_i) \sum_j \beta_j} \tag{10}
\]

and

\[
W = \frac{\beta_i \alpha_i}{\beta_i \alpha_i (1-\alpha_i) \sum_j \beta_j} \tag{11}
\]

Thus we can achieve any desired ratio of throughput - we will later look at one way of using this to meet certain priority objectives.

2.2.3 Access Delays for One Priority

We now turn our attention to access delays. We assume that a station seeking to send a small message (e.g. 1 cell) has not been active. That is, at an arbitrary time point, a previously inactive station needs to send a single cell. Let \( S \) be defined as the time (in slots) from when the countdown counter first equals 0 until the waiting data cell is transmitted. Clearly we have

\[
P_r(S = i) = p_u q_u^{i-1} \tag{12}
\]

where \( P_r() \) denotes probability and

\[
p_u = P_r(slot can be used) = p_r(1-\alpha)
\]

\[
q_u = 1-p_u = (1-p_r) + p_r \alpha
\]

(For any variable \( p \), we denote \( 1-p \) by \( q \),)

and

\[
p_e = P_r(slot is empty)
\]
(Note that $p_e$ and $p_a$ generally depend on station position as might $\alpha$. However, for simplicity of notation we have not explicitly denoted this.)

Hence the generating function for $S$ is given by

$$G_S(z) = E[z^S] = \frac{p_u z}{1 - q_u z}$$

(E{} denotes expectation.)

Also, if $K$ is the value of the countdown counter at an arbitrary time then

$$P_r[K = k] = \rho^k(1 - \rho); \quad \rho = \frac{p_{ae}}{p_{ar}}$$

and

$$G_K(z) = \frac{1 - \rho}{1 - \rho z}$$

where $p_r = P_r\{\text{request occurs}\}$.

Also, if $J$ is the time to reduce the countdown counter by 1, then

$$P_r[J = j] = p_{ae}^{j-1}$$

and

$$G_J(z) = \frac{p_e z}{1 - q_e z}$$

Let $W$ be the waiting time needed to reduce the countdown counter to 0 (from an arbitrary time point) then

$$G_W(z) = E(z^W) = E((E(z^J))^K) = \frac{1(1 - \rho)(1 - q_e z)}{1 - z(q_e + p_{ae})}$$

hence the total access delay, $D$, has generating function

$$G_D(z) = G_w(z)G_S(z) = \frac{1 - \rho(1 - q_e z)p_e z}{(1 - z(q_e + p_{ae}))(1 - q_e z)}$$

By differentiating $G_D(z)$ (or by a direct argument) we find the following for the mean access delay, $E\{D\}$

$$E\{D\} = E\{W\} + E\{S\} = \frac{\rho}{(1 - \rho)p_e} + \frac{1}{(1 - \alpha)p_e}$$

$$E\{D\} = \frac{(1 - p_\alpha)}{(1 - \rho)(1 - \alpha)p_e}$$

Note that although we have suppressed the dependence on $i$, the station number, for simplicity of notation, all of the quantities depend on $i$.

Indeed, as illustrated in Figure 3 where we plot the mean wait, service and total access delays, this dependence is quite strong, and clearly results in "unfair" access delays for stations downstream. Note the total utilization in this case is .99 (equation (2)).

However, as seen in Figure 4, when the total utilization is reduced to .89, the excessive mean delays for the downstream stations are significantly reduced and the distribution tails (90th percentiles) are modest. If however faster access is needed, the following suggestion can virtually guarantee it. First, for such messages, a request is sent at the next highest priority. This message is then placed at the head of the queue to be sent at the same (low) priority. Thus these messages will get essentially immediate access unless the system is highly loaded with higher priority work.

3. Multiple Priorities

We now look at the case of multiple priorities. We will consider here only the case of two priorities; the approach can be readily extended to more priority levels - all be it with increased complexity. The first question to address is an objective or set of objectives.
The original objective for the protocol under study was for strict priorities, i.e. if any station had high priority data to send it should take bandwidth precedence over any and all low priority traffic. We maintain that first, it is unrealistic to attempt to accomplish this in a distributed control environment of the present situation with no globally control. Second, it is not clear that this would be a good objective, even if achievable. An alternative we propose is that if a high priority file transfer becomes active, it should receive considerably more bandwidth than any of the active low priority transfers. Thus low priority traffic would not be shut out by high priority traffic, but potentially significantly slowed. This might be particularly preferable if we consider short low priority messages competing with several large, high priority file transfers. It is also highly desirable to have access delays for short high priority messages be significantly lower than the access delays for the corresponding low priority short messages.

The equilibrium throughput equations now take the form

\[ t_i^h = \frac{(1 - \alpha_i^h)(1 - T - \delta_i^h)}{(1 - \alpha_i^h)(1 - T - \delta_i^h) - \alpha_i^l} \] (21)

\[ t_i^l = \frac{(1 - \alpha_i^h)(1 - T - \delta_i^h)}{(1 - \alpha_i^h)(1 - T - \delta_i^h) - \alpha_i^l} \] (22)

where \( t_i^h \) and \( t_i^l \) denote the high priority and low priority throughput respectively and

\[ L_i^{h0} = \sum_{j=1}^{j-1} t_j^{h0} \]

\[ R_i^{h0} = \sum_{j=1}^{j-1} t_j^{h0} \]

(Note, here also (21), (22) can be modified so that station \( i \) uses a fraction, \( f_i \) of available bandwidth.)

For concreteness we now assume that \( \alpha_i^l = \alpha_i \) which leads to

\[ t_i^l = \frac{(1 - \alpha_i^h)(1 - T)}{\alpha_i^h} \] (23)

and

\[ t_i^h = \frac{(1 - \alpha_i^h)(1 - T)(1 - \frac{\delta_i}{\alpha_i^h})}{\alpha_i^h(1 - T)(1 - \frac{\delta_i}{\alpha_i^h}) + \sum_{j=1}^{N} \delta_j} \] (24)

where \( \delta_i = 1 \) if the \( i^{th} \) station is transmitting a low priority file and again \( T = \) total throughput.

In order to provide the appropriate priority discrimination we choose \( \alpha_i \) to be large relative to the maximum value of the \( \alpha_i^h \), e.g. an order of magnitude. We could use the expected traffic to find values for \( \alpha_i^h \) so that the expected value of the relative throughput would be constant (or to meet any other desired "fairness objective"). If we take as our objective equal throughput for all high priority stations then there is a simpler solution. It is clear that the main problem is that a low priority station upstream of a high priority station will tend to take bandwidth away from that station (e.g. bandwidth that a further up stream high priority station is letting go by for the downstream high priority station). High priority stations upstream of the low priority station do not have a problem. We propose the following simple, but clearly effective solution.

Alternate \( \alpha \) Policy

For \( \alpha_i \) we define two values, \( \alpha_m^l \) and \( \alpha_M^l \). If a given low priority station is not receiving high priority requests, it uses \( \alpha_m^l \), however, whenever it receives a high priority request it switches to using \( \alpha_M^l \). Only after a given time has passed say \( t_d \) without any high priority requests does it again switch to \( \alpha_m^l \). By appropriate choices of \( \alpha_m^l \) and \( \alpha_M^l \) we can effectively ensure that low priority transfers will not take significant bandwidth from high priority stations. On the other hand, when there are no high priority transfers going on, the low priority stations can utilize all of the bandwidth. (Also, the low priority transfers are not completely shut out, they just encounter longer delays.)
Access Delays

Given the implementation of such a policy we see that from the point of view of access delays we can assume that either high priority transfers are present, and hence negligible low priority traffic or high priority traffic is not present at all. In the former case the results already obtained apply for access delays for high priority messages and in the later case for low priority messages. The two remaining cases include low priority access in the presence of high priority traffic and high priority access in the presence of low priority traffic. The later of these is just the time to service the high priority request in the absence of other traffic, hence our previous analysis applies. We now look at the case of low priority access in the presence of high priority transfers.

Similar arguments apply in this case except that we have to use a busy period analysis to account for high priority requests which occur during the "waiting" and "service" time. The results are summarized below.

If $B$ is the "ordinary" busy period associated with "servicing" a request, then

$$G_B(z) = \frac{p_e (1 - q_e + p_e G_B(z))}{1 - q_a + p_a G_B(z)}$$  (25)

If $W'$ is the wait till the countdown counter is first 0 then

$$G_{w'}(z) = \frac{(1 - p) (1 - z (q_a + p_a G_B(z)) q_e)}{1 - z (q_a + p_a G_B(z)) (q_a + p_a)}$$  (26)

If $S'$ is the busy period associated with servicing cell due to arriving high priority packets after the countdown counter $= 0$, then

$$G_{s'}(z) = \frac{p_a (1 - \alpha z (q_a + p_a G_B(z)) z)}{1 - q_a z (q_a + p_a G_B(z))}$$  (27)

And finally, if $D'$ is the access delay, then

$$G_{d'}(z) = G_{w'}(z) G_{s'}(z)$$  (28)

Since the resulting quadratic for $G_{d'}(z)$ can be readily solved, we have all of the needed elements for an explicit representation of $G_{d'}(z)$ in terms of $z$ and hence the inversion methods of [9] can again be applied to obtain the delay distribution for access delays in this case.

4. Concluding Remarks and Continuing Work

More Complex Schemes

The main conclusion to be drawn from these studies as they pertain to IEEE 802.6 is that a straight forward implementation of the proposed protocol may lead to some substantial fairness problems, particularly with respect to access delays and the implementation of priorities. However, there do appear to be rather simple solutions to these problems that are implementable using information which is currently readily available. Two such solutions were suggested here, the "modified" priority treatment for messages with short access time needs and the alternate a policy to achieve appropriate priority discrimination.

In [4], several potential enhancements to the basic 802.6 protocol are suggested, including some that make use of additional information such as the number of stations currently active (by priority). Each station could then readily compute what its fair share of the bandwidth should be and adjust a control parameter (e.g. $\alpha$) to achieve the objective. The problem with this approach is that it requires modification to the protocol and also potentially higher costs for the station interfaces since they would all be required to "read" each cell before relaying it. A compromise (suggested in [4]) is to have certain more sophisticated network management nodes that are strategically placed which would collect, evaluate and take control action based on current traffic. Such nodes could also serve the purpose of destination release nodes. In any event, one can always do better by using more information and control capability, but any improvement must be weighed against the increased costs and complexity.

Other Architectures

Fairness is an important issue in any network architecture and is becoming even more important as new technologies and services evolve. In many cases one has a natural "fairness control" parameter, for example, the window size used in say end to end flow control. Clearly transfers going through parts of the network which are experiencing long delays could have their window size increased relative to other transfers. In order to implement such a control to achieve a given fairness objective we would need to estimate delays for current virtual circuits and then determine the appropriate window sizes. A non trivial problem is to determine and implement an appropriate dynamic policies that will adapt as delays change and virtual circuits are added and deleted. Such problems are also the subject of continuing research.

Acknowledgements

I would like to express my appreciation to B.T. Doshi for many enlightening discussions and helpful suggestions. In particular, the approach to the transient problem and the generalized equilibrium analysis given here were suggested by him.
References


