PRIORITIZED DEMAND ASSIGNMENT PROTOCOLS
AND THEIR EVALUATION

by

I. Chlamtac, O. Ganz and Z. Koren

Technical Report #415

May, 1986
PRIORITIZED DEMAND ASSIGNMENT PROTOCOLS
AND THEIR EVALUATION

I. Chlamtac, O. Ganz and Z. Koren

Dept. of Computer Science
Technion - Israel Institute of Technology
Haifa 32000, Israel

ABSTRACT

We define a class of message or station based priority protocols for demand assignment based LAN's, such as Token Bus [1], HYPERbus [2], LCN [3] etc. We show how existing priority protocols can be represented within this class and how they can be extended for a more efficient realization with regard to both delay and capacity of prioritized channel access in local area networks.

We introduce an analytic approach for analyzing multiple access systems operating under prioritized demand assignment protocols. The approach permits the modeling of station and priority dependent arrival rates and generally distributed transmission times. The introduced finite population model is especially appropriate for prioritized systems where the number of users per priority class is typically small and users place different service demands on the system. As a special case, it can also be used for analyzing any non-prioritized demand assignment protocol with a general distribution of transmission times which may differ from station to station and traffic intensities which can vary from one station to another. Finally, for modeling systems with large populations of users, we introduce an approximate model which we show to be significantly more computationally efficient than the exact model without imposing additional modeling restrictions.
1. INTRODUCTION

We consider multiaccess communication systems such as local area networks [6]. A communication system may be required to support a variety of applications such as real time voice, file transfer or terminal communication. The various service requirements of the different applications impose a natural priority structure on the message transport facility of the network.

While the need for providing prioritized service is evident, as demonstrated by the introduction of priority mechanism in virtually all commercially successful LAN's [6], the amount of rigorous analytic studies has been surprisingly small. To a large extent this situation can be ascribed to the limited availability of existing analytic models.

For prioritized random access systems, few models have appeared in recent years [10], [11], [13]. For networks based on demand assignment protocols, models incorporating priority functions are even scarcer. In particular, a station based priority mechanism was only evaluated for a CSMA/SD protocol [7], [8], while the approximate evaluation of message based priorities for the same protocol appeared in [9]. Since a CSMA/SD protocol operates in collision free mode only for high loads, even these few models cannot be directly applied to the aforementioned demand assignment controlled networks.

In this paper we define a class of message/station based priority mechanisms consistent with existing demand assignment protocols, exemplified by the Token Bus [1], Token Ring [4], HYPERbus [2], LDDri [5] and other local area networks [6].

We deal with three issues pertinent to implementation of priorities in these systems:

(1) The definition of strict priority order in a multiaccess environment and its realization through a collision free protocol.

(2) The channel utilization efficiency and user response time as emanating from the use of priority mechanisms.

(3) The introduction of flexible and tractable analytic models for evaluating prioritized demand assignment protocols.

We analyze the demand assignment priority protocols and show that the existing realization of priorities via the straightforward replication of the scheduling periods for each priority class may result in prohibitive cost in performance. We therefore also introduce a new "priority reservation" approach. The introduced technique allows the
strict preservation of the required priority order, guarantees fairness among users belonging to the same priority class and maintains the collision free channel access discipline with minimal scheduling cost, thus reducing delay and increasing system capacity.

The analytic approach introduced for analyzing the existing and introduced priority mechanisms in demand assignment protocols, allows us to model the system permitting distinct specification of each priority class in terms of the number of users, user arrival rates and any distribution of message transmission times. As a special case, it also allows us to analyze any non-prioritized demand assignment protocol with a general distribution of transmission times which may differ from station to station, and arrival rates which can vary from one station to another. We use a finite population model, the most appropriate for LAN's where the total number of stations and especially the number of users per priority class are typically small. Finally, to make the analysis tractable even when the number of users increases, we introduce an approximate model which allows time/space efficient system solution without sacrificing too much precision or imposing additional modeling assumptions on the system.

2. PRIORITY SYSTEM AND PROTOCOL MODELS

We consider a network in which the stations are allowed access to the channel by executing one of the demand assignment protocols using implicit (scheduling) or explicit (token passing) control [6], [16].

Without loss of generality, we refer to the channel control mechanism as scheduling. We define a valid message based priority mechanism in such a system as one which satisfies the following functionality requirements:

1. Collision free channel access.
2. No lower priority class message will be transmitted if, at scheduling time, a higher priority message is present in the system.
3. The access is fair among all messages belonging to the same priority class.

(Notice that 2 and 3 are consistent with priority requirements given also for random access protocols [10])

Let us consider a system consisting of \( n \) stations generating messages of \( p \) different priority levels. For specifying the protocol alternatives, we denote a virtual network user by a tuple \((i,j)\) with \(1 \leq i \leq n\) and \(1 \leq j \leq p\). The total number of virtual users in the network (referred to from now on as simply users) is thus bounded by \(n \cdot p\).
2.1. Solution by Replication of Scheduling

A most straightforward approach for implementing message priorities in demand assignment protocols is obtained by a simple replication of the access (scheduling) mechanism [9]. Figure 1.a shows a time diagram of the channel access procedure for \( n \) stations with \( p \) priority classes. It is easy to see why this approach not only preserves the priority associated functional requirements but is easily realized with existing hardware. In this approach however, when the number of users is large or the propagation delays become non-negligible, the scheduling penalty may become prohibitively high. It is important to further notice that unlike in non-prioritized demand assignment protocols, this penalty may become prohibitively high even at high loads. This can happen for instance when, as is in fact typically the case, the dominant channel load contribution comes from the lower priority classes. In this case, even when the total system load is high, the scheduling periods of the higher priority classes will still have to be traversed in order not to violate requirement no. 2, leading in this way to reduced system capacity.

2.2. Solution by Token Bus Priorities

An approach which attempts to reduce this scheduling overhead was introduced in the token bus local area network [1]. In this solution, documented by the time diagram of Figure 1.b, access rights may be passed among all virtual users (of different priorities) in the same station, before the next station is allowed to access the channel. While this solution preserves requirements (1) and (3), it will violate requirement (2) so that the quality of service differentiation expected within a priority system is impaired. In similar circumstances for random access protocol controlled networks, this violation was shown to be rectifiable only by introducing preemption [11]. A prioritized token bus system with preemption has not been proposed or studied so far.

2.3. Proposed Prioritized Access - The RSP Protocol

To implement a message based priority mechanism adhering to the three requirements specified above but which does not impose the high scheduling penalty, we introduce the following protocol. We assume that all stations are synchronized to multiples of slots of length equal to the maximum propagation delay in the network, denoted by \( a \), and that stations monitor the channel continuously. In the proposed protocol, messages are transmitted following two consecutive reservation/scheduling periods: a priority reservation period which will be called a \( p \)-period, followed by a station reservation period which we shall call an \( n \)-period. At the end of a transmission, up to \( p \) slots are dedicated to finding the highest priority message present in the system. This is achieved by
allowing any station of priority $j$ with a ready message to transmit a burst of unmodulated carrier, or token, in the $j$-th slot, provided the first $j-1$ slots were quiet.

If no reservations are made during the first $p$ reservation slots following a transmission, another period of $p$ slots is started and the $p$-period is repeated until the arrival of one or more messages of a given priority, say $j$. The resulting transmission in slot $j$ serves to reserve the channel for priority $j$ messages and terminate the $p$-period. Since more than one message of priority $j$ may be present, the $n$-period of up to $n$ slots is necessary in order to determine which station of priority $j$ will actually transmit. In this scheduling period, a ready station calculates a scheduling delay and waits for it to pass before attempting to transmit. If following this delay, the channel is idle, the station's transmission begins, thus ending the $n$-period. If on the other end, an earlier priority $j$ transmission has been initiated, the station refrains from transmitting and attempts again in the next reservation/scheduling interval. Since the proposed protocol is based on channel access resolution through consecutive reservation/scheduling periods, we call it the Reservation Scheduling Protocol, or RSP. The associated time diagram of the RSP protocol is demonstrated in Figure 1.c.

3. RSP PROTOCOL INTERPRETATIONS

Note that the proposed RSP protocol does not confine us to any specific assignment of scheduling delays during the $n$ period. Hence, different protocol interpretations can be obtained so that the protocol model described can be viewed as the generalization of two different network operational circumstances as specified next.


Consider a network where each station is capable of generating messages of $p$ different priorities. Clearly the mechanisms of RSP protocol maintain requirements 1 and 2. To provide fairness among stations carrying messages of the same priority, as in requirement 3, a "fair" cyclic order is obeyed in each $n$-period. Specifically, let $c_j$ be the last station to transmit a priority $j$ message, let $i_j$ be a station with a ready priority $j$ message, and let $n_j$ be the number of stations which can generate priority $j$ messages. It is easy to show that when the scheduling delay is calculated separately for each priority class according to the "fair" scheduling function [15], [16] given by

$$[(i_j - c_j + n_j - 1) \mod n_j] \cdot a$$

then fairness among all stations carrying the same priority class messages is maintained.
Note that for the special case $p=1$, we obtain a non-prioritized network operating under a round robin discipline. The analytic model presented in this paper can therefore be used for analyzing demand assignment protocols with a general distribution, possibly station dependent, of message transmission times and arrival rates which may differ from station to station.

3.2. Case B: Station Based Priorities.

Let us consider the case of a network with $N$ stations, each station forming a separate priority class. As in CSMA/SD, we can use a demand assignment protocol with a reservation period of up to $N$ slots followed by the transmission of the highest priority station which at reservation time had a ready message. The RSP protocol can here be used to implement the same station order while significantly reducing the scheduling cost especially when the traffic is low, in the following way. Let us aggregate the $N$ stations into $p$ classes with $n=N/p$ stations in each class. To prevent collisions, following the aggregation of stations, the second scheduling period is used to resolve conflicts that occurred in the first one. Since the stations always retain their original priority, the $n$-reservation period is performed in a fixed order which is the order of priorities within each subgroup. Notice also that this approach can be considered a one-hop implementation of [12] so that an optimal aggregation of stations can always be found. The behavior in each aggregated class depends therefore on the prioritized versus fair channel access which is followed. These differences are reflected in the following analytic models.

4. THE STOCHASTIC MODEL AND ANALYSIS

We consider a finite population of $np$ stations (users), divided into $p$ priority classes of size $n$ each (the terms "station" and "user" will be used interchangeably). A user is identified by the pair $(i,j)$, $j$ indicating its priority while $i$ is its ordinal number among the $j$-th priority stations. Each user has a buffer of size 1 and generates a new message only when this buffer is empty. It does so with rate $\lambda_{ij}$ messages per unit of time. Time is assumed to be divided into slots of size $a$ each, $a$ being the end-to-end propagation delay. The transmission time of a message in station $(i,j)$ is a random variable $\tau_{ij}$ with any general distribution which may depend on $(i,j)$. We denote by $f_{ij}(t)$ the density function of $\tau_{ij}$ (which could be a probability function for a discrete $\tau_{ij}$), by $T_{ij}$ its expected value, and by $\Phi_{ij}(s)$ its moment generating function.

The proposed RSP class of protocols governing the access to the common channel has been described before. It is shown next that both versions of the protocol, i.e., the version with fixed priorities during the $n$-period and the
one with cyclic priorities can be analyzed in a unified manner.

We observe the system at the end of each transmission, which is a regeneration point of the stochastic process. Let us denote the state of the system at the end of the \( k \)-th transmission by the pair \((X(k), C(k))\), where: \(X(k)\) is a binary \(n \times p\) matrix, \(X(k)_{ij}\) assuming the value 1 or 0 according to the existence or non-existence of a ready message in user \((i,j)\), and 

\(C(k)\) is the \(p\)-dimensional vector whose \(j\)-th component is the ordinal number \(i\) within priority \(j\) of the station \((i,j)\) which has transmitted last. We will assume that the system is in a steady state and therefore omit the index \(k\) and describe the state as \((X, C)\).

All the following analysis is performed with respect to the system with the "fair" rotating priorities within each of the \(p\) priority classes, where the information as to which station transmitted last in each priority must be part of the state description, hence the notation of the state as \((X, C)\).

The analysis of the protocol with a fixed order of stations within each priority class (hence within the \(n\)-period) follows the same steps as that of the "fair" version and the resulting equations are exactly the same, except for minor changes which will be specified at the end of this section.

Evaluating the system's performance measures, such as its throughput and the average delay of a message, will be based on the notion of an \((X, C)\) cycle. An \((X, C)\) cycle is defined as the time elapsing between the observation of \((X, C)\) and the next observation of a system state (i.e., the next end of transmission).

To analyze these cycles and find their expected lengths and frequencies, we introduce the following definitions and notations.

Let \(P\) denote the transition probabilities matrix

\[
P = P((X^{(1)}, C^{(1)}) \rightarrow (X^{(2)}, C^{(2)}))
\]

Let \(\Pi\) denote the steady state probabilities given by the solution of the following linear equations,

\[
\Pi = \Pi \cdot P \quad ; \quad \sum_{(X, C)} \Pi(X, C) = 1
\] (4.1)

Let \(X\) denote the set of all \(n \times p\) binary matrices and \(X\) denote any binary matrix which is a member of \(X\). For convenience, we further define the following special members of \(X\). \(Z\) is the zero matrix (i.e., \(Z_{ij} = 0\) for all \(i,j\)). \(Z\) represents the state where all stations are empty. \(Y^{(i,j)}\) is the matrix whose \((i,j)\)-th element is equal to 1 and all other elements are 0.
Let $L(X,C)$ be the expected length of an $(X,C)$ cycle. As a special case, $L(Z,C)$ is the expected length of a $(Z,C)$ cycle, i.e., the length of the period of time between the moment the system becomes empty and the end of the next transmission.

In addition, let

$$\lambda(j) = \sum_{i=1}^{n} \lambda_{ij}; \quad \lambda = \sum_{i=1}^{n} \sum_{j=1}^{p} \lambda_{ij},$$

represent the arrival rate of priority $j$ and the total arrival rate to the system, respectively.

Two functions which we introduce for simplifying our notations are: $\mu$ - which measures the cyclic distance between two stations of the same priority, and $Ind$ - which is an indicator assuming the values 1 or 0, according to whether a certain statement is true or false, respectively. Their formal definitions are:

$$\mu(c,i) = (i-c+n-1) \mod n \quad (4.3)$$

$$Ind\text{(statement)} = \begin{cases} 1 \text{ if statement is true} \\ 0 \text{ if statement is false} \end{cases} \quad (4.4)$$

We further define operators which will be applied to members of $X$:

$$M(X) = \sum_{i=1}^{n} \sum_{j=1}^{p} X_{ij} \quad (4.5)$$

$0 \leq M(X) \leq np$ is the number of 1's in $X$ (or the number of messages in the network).

The following three operators are defined for $X \neq Z$ ($X$ not the zero matrix) only. $J(X)$ is the highest priority class which has a ready message and will therefore be the one to transmit in an $(X,C)$ cycle, i.e., $1 \leq J(X) \leq p$ is the first column of $X$ which has a non-zero sum. Formally,

$$J(X) = \min \{ j \mid \sum_{i=1}^{n} X_{ij} > 0 \} \quad \text{(for } X \neq Z) \quad (4.6)$$

$I(X,C)$ is the ordinal number of the next station (of priority $J(X)$) to transmit in an $(X,C)$ cycle. for $X \neq Z$, $I(X,C)$ is the row number $i$ of the non-zero element of the column $J(X)$ for which $\mu(C_{J(X)},i)$ is minimal. $1 \leq I(X,C) \leq n$. Notice that, $(I(X,C),J(X))$ is the next station to transmit in state $(X,C)$, for $X \neq Z$.

For mathematical tractability and simplicity of presentation we divide the forthcoming analysis into two parts: In the first part we assume that message transmission time in station $(i,j)$ is a constant $T_{ij}$ (which still allows for different transmission times for the different stations), while in the second part this assumption will be dropped and the necessary changes in the expressions derived in the first part will be demonstrated.
Constant Transmission Times.

We start the analysis by calculating $L(X, C)$, the expected length of an $(X, C)$ cycle. Note that for $X \neq Z$, an $(X, C)$ cycle is composed of the following three components: (1) The $p$-period whose length is $J(X)$ slots, in which the highest priority present in the system is determined. (2) The $k$-period whose length is $\mu(C_{J(X)}(X,C))$ slots and in which the next transmitting station (of the reservation priority) is determined. (3) The message transmission time whose length is $T_{l(X,C)}(X)$. $L(X,C)$ can therefore be expressed as,

$$L(X,C) = J(X) \cdot a + \mu(C_{J(X)}(X,C)) \cdot a + T_{l(X,C)}(X) \quad (4.7)$$

To calculate the system's performance measures, all cycle lengths, including $L(Z,C)$ (which has been defined as the expected length of a cycle starting with an empty system) must be considered. As opposed to the case $X \neq Z$ where $J(X), I(X,C)$ are constant and known in advance, for $X = Z$, $I(Z,C)$ and $J(Z)$ are random variables depending on the arrivals to the system during the $(Z,C)$ cycle. Since $L(Z,C)$ depends on both of them, the way to analyze the stochastic behavior of a $(Z,C)$ cycle is by calculating the joint probability function $P^{(C)}_{b,i,j,X}(b,i,j,x)$, where $B$ is the number of empty $p$-periods, $(i,j)$ is the first station to transmit, and $X$ is the matrix representing the arrivals during the cycle. Clearly these random variables depend on $C$, whose components are the last stations having transmitted in each priority before the system became empty, hence $C$ is used as a parameter of the probability function.

To calculate this joint probability, observe that $P^{(C)}_{b,i,j,X}(b,i,j,x)$ is the probability of the following event:

First we have $b$ idle $p$-periods, then priority $j$ is the first priority to reserve a slot in the $(b+1)$ $p$-period, then station $(i,j)$ is the one to transmit, at the end of its transmission the state of the system is $(x,C^{(i,j)},(x,C^{(i,j)},(x,C^{(i,j)})$ where $C^{(i,j)}$ is the vector $C$ whose $j$-th component has been replaced by $i$). In order for this event to occur, a series of independent events concerning the arrivals of messages to the different stations must occur: No message must arrive during the first $b-p$ slots, station $(i,j)$ must generate a message prior to slot $j$ in the $(b+1)$ $p$-period, all stations $(k,l)$ for which $x_{kl} = 1$ and whose priority is higher than or equal to $j$ must generate a message after this slot, while those stations with a lower priority may generate a message anywhere in the cycle (except the first $b-p$ slots). The stations for which $x_{kl} = 0$ must not generate any message at all during this cycle. Each station is therefore allotted an interval of time in which it must generate a message. Let us denote this interval for station $(k,l)$ by $I_1(C^{(i,j)},l_2(C^{(i,j)},k,l))$. 


We will first express the desired probability in terms of these intervals and then find the expression for the appropriate interval for each station. Recall that when $W$ is an exponentially distributed random variable with parameter $\delta$,

$$P(t_1 \leq W \leq t_2) = e^{-\delta t_1} - e^{-\delta t_2}$$

Hence,

$$P^{(C)}_{b,j,j,x}(b,i,j,x) = \prod_{(k,l)} (e^{-\lambda_{kl} t_1^{(C,j,i,k,l)}} - e^{-\lambda_{kl} t_2^{(C,j,i,k,l)}})$$  \hfill (4.8)

For some stations, the end of the interval is at the end of the cycle. We need therefore to find the length of the cycle, denoted by $CT$, conditioned on $(C,b,i,j,x)$,

$$CT(C,b,i,j,x) = b \cdot p \cdot a + j \cdot a + \mu(C,j,i) a + T_{ij}$$  \hfill (4.9)

The table below summarizes the values of $t_1$ and $t_2$ for all $(k,l)$.

<table>
<thead>
<tr>
<th>$x_{kl}$</th>
<th>conditions on $i,j,k,l$</th>
<th>$t_1$</th>
<th>$t_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_{kl}=1$</td>
<td>$l &lt; j$</td>
<td>$a(bp+l)$</td>
<td>$CT$</td>
</tr>
<tr>
<td>$x_{kl}=1$</td>
<td>$l &gt; j$ or $(l=j$ and $\mu(C,j,k) &gt; \mu(C,j,i))$</td>
<td>max{0, a[(b-1)p+l]}</td>
<td>$CT$</td>
</tr>
<tr>
<td>$x_{kl}=1$</td>
<td>$l=j$ and $\mu(C,j,k) &lt; \mu(C,j,i)$</td>
<td>$a(bp+l+\mu(C,j,k))$</td>
<td>$CT$</td>
</tr>
<tr>
<td>$x_{kl}=1$</td>
<td>$l=j$ and $k=i$</td>
<td>max{0, a[(b-1)p+l]}</td>
<td>$a(bp+l)$</td>
</tr>
<tr>
<td>$x_{kl}=0$</td>
<td></td>
<td>$CT$</td>
<td>$\infty$</td>
</tr>
</tbody>
</table>

By summing the joint probability over one or more of the variables, we can obtain any marginal probability we need for the analysis.

To calculate $L(Z,C)$ (the expected cycle length for the empty state), we multiply (4.9) by the joint probability (4.8) and sum over all values of $b$, $i$, $j$ and $x$. Additional uses of the function $P^{(C)}_{b,j,j,x}(b,i,j,x)$ will be demonstrated later.

We will proceed now to calculate the transition probabilities matrix $P$, starting from the case $X \neq Z$. For $X \neq Z$, the only changes in the system between two consecutive ends of transmission can be arrivals to some empty
stations and the deletion of the message that has just been transmitted. Let us denote by \( A^X \) an arrival matrix, i.e., any binary matrix such that \( X_{ij} = 1 \) implies \( A^X_{ij} = 0 \), (only a station which at state \( X \) is empty has a potential for generating a message). For \( X_{ij} = 0 \), \( A^X_{ij} = 1 \) or 0 indicates a message arriving or not arriving, respectively, at the empty station \((i,j)\), at any point of time during the \((X,C)\) cycle. Let \( \eta(X,C) \) be the vector \( C \) whose \( J(X) \)-th component has been replaced by \( I(X,C) \). The only possible transitions are from state \((X,C)\) to states of the form\((X + A^X - Y^X(X,C), \eta(X,C))\), and randomness is introduced through the arrival matrix \( A^X \) only.

Hence, the transition probability can be expressed as the product of the probabilities of arrival to the stations which actually generated a message and the probabilities of non-arrival to the stations which could have generated a message but did not. If we denote by \( F(X,A) \) the set \{ \((i,j) \mid X_{ij} = 0 \) and \( A_{ij} = 1 \) \}, and by \( G(X,A) \) the set \{ \((i,j) \mid X_{ij} = 0 \) and \( A_{ij} = 0 \) \}, we obtain as transition probabilities

\[
P((X,C) \rightarrow (X^*,C^*)) = \prod_{F(X,A^X)} (1 - e^{-\lambda_0 L(X,C)}) \cdot \prod_{G(X,A^X)} e^{-\lambda_0 L(X,C)} \text{ if } (X^*,C^*) = (X + A^X - Y^X(X,C), \eta(X,C))
\]

\[
\text{otherwise}
\]

To calculate the transition probabilities \( P((Z,C) \rightarrow (X^*,C^*)) \) note that \( X^* \) could be any matrix of the form \( A - Y^X(X,C) \) for some \( j \) and some binary matrix \( A \), \( A \neq Z \), and \( C^* \) is the vector \( C \) whose \( j \)-th component has been replaced by some \( i \). We therefore use once more the joint probability function \( P(C)_{b,j,X}(b,i,j,x) \), appearing in (4.8), this time by summing over all values of \( b \) and \( j \),

\[
P((Z,C) \rightarrow (X^*,C^*)) = \sum_{b=0}^{\infty} \sum_{j=1}^{J(X)} P(C)_{b,j,X}(b,C^*,j,X^*)
\]

Having found the expressions for the transition probabilities \( P \), we can now solve the equations (4.1) for the steady state probabilities \( \Pi \).

Using \( \Pi \) and \( L(X,C) \) we now proceed to calculate the system’s performance measures, namely its throughput and delay. Since we are dealing with a prioritized system whose users are differentiated both by priority and by arrival rates and transmission times, it is appropriate to calculate delay and throughput individually for each user \((i,j)\).

We define the throughput for station \((i,j)\) (denoted by \( S_{ij} \)) as the fraction of cycle time devoted by station \((i,j)\) to transmitting messages. Note that since following the reservation-scheduling periods there are no collisions,
all message transmissions are successful. Define \( U_{ij} \) as the amount of cycle time in which station \((i,j)\) is transmitting messages, and \( \bar{L} \) as the average cycle length. Clearly,

\[
S_{ij} = \frac{U_{ij}}{\bar{L}}. \tag{4.12}
\]

We proceed now to calculate \( U_{ij} \) and \( \bar{L} \). Denote by \( P_{ij}^{(C)}(i,j) \) the probability of \((i,j)\) transmitting following the state \((Z,C)\). This probability can be obtained by summing the joint probability in (4.8) over \( b \) and \( x \). Thus,

\[
U_{ij} = \left[ \sum_{(X,C)} \Pi(X,C) \cdot \text{Ind}(X \neq Z \land I(X,C)=i) \land J(X) = j \right] \cdot \sum_{(X,C)} P_{ij}^{(C)}(i,j) \cdot T_{ij} \tag{4.13}
\]

and

\[
\bar{L} = \sum_{(X,C)} \Pi(X,C) \cdot L(X,C) \tag{4.14}
\]

Define the delay at user \((i,j)\) as the average time between the arrival of a message at \((i,j)\) and the end of its transmission. Denote this delay by \( D_{ij} \). To calculate \( D_{ij} \), we apply for each \((i,j)\) the well-known result by Little stating that,

\[
D_{ij} = \frac{Q_{ij}}{S_{ij}}. \tag{4.15}
\]

where \( S_{ij} \) is the throughput given by (4.12) and \( Q_{ij} \) is defined and calculated next.

Define the backlog in station \((i,j)\) at time \( t \) as the number of messages present in \((i,j)\) at this time. Define \( Q_{ij} \) as the average backlog (over time) in station \((i,j)\). Observe now a specific station \((i,j)\) and a specific cycle \((X,C)\). If \( X_{ij} = 1 \), then the backlog at \((i,j)\) is \( 1 \) throughout the cycle. If however \( X_{ij} = 0 \), there are no messages in \((i,j)\) at the beginning of the cycle but a message may be generated at some point \( s \) during the cycle, in which case the backlog will be \( 1 \) for the residual cycle time which is \( L(X,C) - s \). The expected value of this residual cycle time is,

\[
\int_0^{L(X,C)} (L(X,C) - s) \lambda_{ij} e^{-\lambda_{ij}s} ds = \frac{e^{-\lambda_{ij}L(X,C)} + \lambda_{ij}L(X,C) - 1}{\lambda_{ij}}
\]

To simplify notation, we define

\[
Res(T,\lambda) = \frac{e^{-\lambda T} + \lambda T - 1}{\lambda} \tag{4.16}
\]

Averaging the backlog in station \((i,j)\) over all states \((X,C)\) and dividing by the average cycle length we obtain,
Generally Distributed Transmission Times.

We show in this section how the expressions developed before are modified when the assumption of constant transmission times is dropped, allowing the analysis of networks with arbitrarily distributed transmission times. Recall that \( t_{ij} \) is the random variable representing the transmission time of station \((i,j)\), and that \( f_{ij}(t) \), \( T_{ij} \), and \( \Phi_{ij}(s) \) denote its density function, expected value, and moment generating function, respectively. All expressions containing \( T_{ij} \) are now modified as follows: assuming that \( t_{ij} = 1 \), each \( T_{ij} \) is replaced by 1 and the resulting expression is averaged over all values of \( t \) with respect to the density function \( f_{ij}(t) \). In equations where \( T_{ij} \) appears in the additive form, it is replaced by its expected value. Since \( T_{ij} \) now denotes the expected transmission time, all such equations, namely (4.7), (4.11), (4.12), (4.13), (4.14), (4.15) remain unchanged when using the above procedure. In equation (4.9) an additional conditioning on \( t_{ij} = t \) is necessary, yielding

\[
CT(C, b, i, j, x, t) = b \cdot p \cdot a + j \cdot a + \mu(C, i) \cdot a + t
\]

(4.9a)

Averaging (4.8) over all values of \( t \) yields,

\[
P(C)_{b, i, j, x, t} = \int_0^{\infty} \left( e^{-\lambda_{ji}(t)} - e^{-\lambda_{ij}(t)} \right) f_{ij}(t) \, dt
\]

(4.8a)

In equation (4.10), the dependence on the transmission time is introduced through \( L(X, C) \). If we denote \( r(X, C) = [J(X) + \mu(C, j) J(X, C)] \cdot a \quad \text{and} \quad l(X, C, t) = r(X, C) + t \), then the product in (4.10) becomes

\[
\int_{t=0}^{\infty} \prod_{i=1}^n \left( 1 - e^{-\lambda_{ij}(X, C, t)} \right) \prod_{i=0}^{\infty} \frac{e^{-\lambda_{ij}(X, C, t)} \mu(C, j) J(X, C)}{e^{-\lambda_{ij}(X, C, t)} f_{ij}(X, C, j)} \, dt
\]

(4.10a)

The last equation to be modified is (4.17). Similarly to the way (4.17) has been obtained, recall that some messages are backlogged for the whole cycle time \( l(X, C, t) \) while others are backlogged for a residual cycle time which is calculated as follows,

\[
\int_0^{l(X, C, t)} (l(X, C, t) - s) \lambda_{ij} e^{-2\lambda_{ij} s} \, ds = \frac{e^{-\lambda_{ij}(X, C, t)} + \lambda_{ij} l(X, C, t) - 1}{\lambda_{ij}}
\]

To calculate \( Q_{ij} \), we calculate the average backlog in station \((i,j)\) over all states \((X, C)\) and then average over \( t \) and divide by the average cycle length. We obtain,
14

\[
Q_{ij} = \frac{1}{L(X,C)} \sum_{X,C} \Pi(X,C) \left[ \text{Ind}(X_{ij}=1) L(X,C) e^{-\lambda_{ij}(X'C)(-\lambda_{ij})} + \text{Ind}(X_{ij}=0) \frac{\Phi_{ij}(X,C) L(X,C) - 1}{\lambda_{ij}} \right]
\] (4.17a)

All the equations appearing in this section refer to the system with cyclic priorities among the stations within each priority class. The following minor changes are necessary to adapt the expressions to the system with a fixed order of stations within each priority class. Since the matrix \( X \) alone is sufficient for describing the state of the system, the vector \( C \) is omitted from the state description and all other references to the vector \( C \) are omitted as well. In addition, the function \( \mu(c,i) \) (defined in (4.3)) is replaced by \( i-1 \).

Although it is theoretically possible to analyze the exact model describing the network’s behavior, it is often not easy to numerically compute the performance measures. The reason for this difficulty is the exponentially increasing number of states in the system. Notice that since \( X \) can assume \( 2^n \) different values and \( C \) has \( n^p \) different values, the number of the system’s states is bounded by \( n^p 2^n \). This number is however slightly smaller, since some states are not reachable. Specifically, at state \((X,C)\), at least one element \( X_{C_{\mu,j}} \) in row \( C_{ij} \) has to equal 0, \( C_{ij} \) representing the station that has just completed its transmission. The actual number of states for the exact model is \( n^p 2^n (n-1) (2^n - 1) \), still a very large number of states, and consequently of linear equations which must be solved, even for relatively small values of \( n \) and \( p \). Moreover, the underlying stochastic process does not appear to be a reversible one, hence a product form solution for the steady state probabilities cannot be claimed to exist.

5. THE APPROXIMATE MODEL

We introduce an approximate model of the network which preserves the notion of priorities, does not significantly sacrifice accuracy, does not impose any additional modeling restrictions, and simplifies the numerical evaluation of the system’s performance. In the approximate model we continue to observe the system at the end of each transmission. To reduce the system complexity we do not describe the state as \((X,C)\). Instead, we lump all states for which \( M(X)=m \) and \( J(X)=k \) into one state, \((m,k)\), where \( m \) is the total number of messages existing in the system and \( k \) is the highest priority present among these \( m \) messages. We thus reduce the number of states, and the number of linear equations necessary for finding the steady state probabilities, from \( n^p 2^n (n-1) (2^n - 1) \) to less than 

\[
np^2 (npP+1) \quad \text{states to be exact.}
\]
When \( m=0 \), \( k \) is meaningless and will be arbitrarily chosen as 1. Note that when, in the exact model, the state was described as \((X,C)\), the highest priority class present in the system could be determined by observing \( X \) and was therefore not a separate part of the state description. However, with the approximate representation, \( m \) which is the total number of messages does not specify the highest priority present. It therefore becomes necessary to add \( k \) to the state description given as \((m,k)\).

The basis for analyzing the process is the calculation of the matrix of transition probabilities between any two \((m,k)\) states, denoted by \( P^{(a)} \), and the steady state probabilities, denoted by \( \Pi^{(a)}(m,k) \). For this calculation observe that the new state \((m,k)\) actually replaces the set
\[
R(m,k) = \{ (X,C) \mid M(X) = m \wedge J(X) = k \}
\]
(5.1)
To find the size of the set \( R(m,k) \), or the number of original states lumped into the one state \((m,k)\), define \( r(m,k) \) as the number of matrices \( X \) with \( m \) "1"'s and at least one "1" in column \( k \) but not earlier. For \( m>0 \),
\[
r(m,k) = \left[ \frac{n(p-k+1)}{m} \right] - \left[ \frac{n(p-k)}{m} \right].
\]
For \( m=0 \), define \( r(0,1) = 1 \). In addition, note that \( C \) can assume \( n^p \) different values. Hence, the size of the set \( R(m,k) \) is \( n^p \cdot r(m,k) \).

In approximations based on lumping of states, we can calculate the probability of transition from a lumped state as the average of the original probabilities of transition from the states constructing it. To find the probability of transition from the state \((m,k)\), we therefore calculate the average of the probabilities of transition from the states \((X,C) \in R(m,k)\), Using a weighted average rather than a non-weighted one to increase the accuracy of the approximation. The set of weights used for this averaging is based on the assumption that the \( m \) 1's are uniformly distributed within the matrix \( X \) and that the vector of stations \( C \) has a weight proportional to the product of the arrival rates of its components.

Hence, each \((X,C) \in R(m,k)\) is assigned an averaging weight of,
\[
\frac{1}{r(m,k)} \prod_{k=1}^{p} \frac{\lambda_{C,k}}{\lambda^{(k)}}
\]
(5.2)
Averaging the original transition probabilities according to the weights specified above yields,
\[
P^{(a)}((m,k),(m^*,k^*)) = P((m,k) \rightarrow (m^*,k^*)) = \sum_{X,C} \sum_{X',C'} \prod_{k=1}^{p} \frac{\lambda_{C,k}}{\lambda^{(k)}} \cdot \frac{1}{r(m,k)} \cdot P((X,C) \rightarrow (X^*,C^*)) \cdot \text{Ind}(M(X) = m \cap J(X) = k \cap M(X^*) = m^* \cap J(X^*) = k^*)
\]
(5.3)
where \( P((X,C) \rightarrow (X^*,C^*)) \) can be obtained from (4.10) and (4.11).
Note that for large values of \( n \) and \( p \) the complexity of (5.3) could become prohibitively large. For these cases we suggest the reduction of the state \((X, C)\) into \((S, C)\), where \( S \) is a \( p \)-element vector whose \( j \)-th element is equal to the sum of the \( j \)-th column of \( X \), i.e., to the number of messages from priority \( j \) present in the system, for \( 1 \leq j \leq p \). The probabilities \( P\left((S, C) \rightarrow (S^*, C^*)\right) \) can now be calculated using (4.10), (4.11) and the Binomial distribution. To further modify equation (5.3), define

\[
M(\theta)(S) = \sum_{j=1}^{p} S(j) ; \quad J(\theta)(S) = \min j \mid S(j) > 0
\]

to obtain,

\[
P(\theta)(m, k) = P( (m, k) \rightarrow (m^*, k^*) ) = \sum_{S \subseteq S^*} \sum_{C \subseteq C^*} \prod_{k=1}^{p} \frac{\lambda_{C, k} \cdot 1}{r(m, k)}
\]

(5.3a)

(5.3a) has a significantly smaller complexity than (5.3) and is almost as accurate.

The steady-state probabilities \( \Pi(\theta) \) can now be obtained by solving the following set of linear equations.

\[
\Pi(\theta) = \Pi(\theta) \cdot P(\theta) ; \quad \sum_{(m,k)} \Pi(\theta)(m,k) = 1 \tag{5.4}
\]

Having calculated the steady-state probabilities, we suggest two possible approaches to the problem of calculating the performance measures, which we term the \((X, C)\) method and the \((c, i)\) method. In the first approach we stay as close to the exact model as possible, thus increasing the accuracy of the results. The second approach is somewhat less precise, yet it involves much less computation than the first one.

The \((X, C)\) Method.

This approach involves deducing the original state \((X, C)\) out of the lumped state \((m, k)\). It is clear from probabilistic considerations that,

\[
\Pi(\theta)(m, k) = \sum_{(X, C)} \Pi(X, C) \cdot \text{Ind}(M(X) = m \cap J(X) = k) \tag{5.5}
\]

To get \( \Pi(X, C) \) out of \( \Pi(\theta)(m, k) \) we make the following approximating assumptions. As in (5.2), we assume that the 1's are uniformly distributed within the matrix \( X \) and that \( C_k \) has a weight proportional to its arrival rate.

Hence, each \((X, C)\) is assigned a conditional probability (given \((m, k)\)) of

\[
\frac{1}{r(m, k)} \prod_{k=1}^{p} \frac{\lambda_{C, k} \cdot \text{Ind}(M(X) = m \cap J(X) = k)}{\lambda(k)}
\]

And an unconditional probability,
\[ \Pi^{(a)}(X,C) = \sum_{(m,k)} \Pi^{(a)}(m,k) \cdot \prod_{k=1}^{r} \frac{\lambda_{c_{k},k}}{\lambda^{(k)}} \cdot \text{Ind}(M(X)=m \cap J(X)=k) \]  

Note that the sum in (5.6) has only one non-zero element and that (5.6) applies to \( X = Z \) as well. The performance equations (4.12) - (4.17) and (4.17a) can now be modified for the approximate model by substituting (5.6) for \( \Pi(X,C) \).

The \((c,l)\) Method.

As opposed to the \((X,C)\) approach in which the notion of describing the state of the system as \((X,C)\) is preserved, and hence no new performance equations are needed, the \((c,l)\) approach is based on defining an \((m,k)\) cycle, analyzing its stochastic behavior and developing new equations based on these cycles.

Define an \((m,k)\) cycle as the period of time between the end of transmission where the state \((m,k)\) has been observed and the consecutive end of transmission, and define \(L^{(a)}(m,k)\) as the expected length of an \((m,k)\) cycle.

We now calculate \(L^{(a)}(m,k)\), starting from the case \( m \neq 0 \). Except for the \((0,1)\) cycle which starts with a period of time in which the station is empty, each \((m,k)\) cycle is composed of three parts, namely the \( p \) reservation period, the \( n \) scheduling period and the message transmission time. Note that a station of priority \( k \) is the one to transmit in an \((m,k)\) cycle, hence the length of the \( p \)-period is known to be \( k \cdot a \). Yet the length of the \( n \)-period and that of the following message-transmission time can not be deduced from the state description since they depend on \((c,k)\) - the last station of priority \( k \) to have transmitted, and on \((l,k)\) - the station transmitting in the present cycle. In the exact model, the three indices \( c,l,k \) could be deduced from the state description \((X,C)\). In the approximate model where the priority \( k \) is the only available information, \( c \) and \( l \) have to be treated probabilistically, hence the name "The \((c,l)\) method".

Since \( c \) and \( l \) are both stations from priority class \( k \), they are assigned weights proportional to their arrival rates within priority \( k \). In addition, \( c \) and \( l \) are assumed independent, though some dependence between \( c \) and \( l \) could be modeled. These assumptions imply the following conditional probability,

\[ P\{ (c,k) \text{ transmitted last within priority } k, (l,k) \text{ transmits next } \} = \frac{\lambda_{c_{k}} \cdot \lambda_{l_{k}}}{\lambda^{(k)} \cdot \lambda^{(k)}} \]  

Once \( c \) and \( l \) are given, \( L^{(a)}(m,k) \) is equal to \( a \cdot (k + \mu(c,l)) + T_{ik} \). Hence,
\[ L^{(a)}(m,k) = \sum_{c,l} \frac{\lambda_{ck}}{\lambda^{(k)}} \frac{\lambda_{lk}}{\lambda^{(k)}} \left[ a(k+\mu(c,l)) + T_{lk} \right] \quad \text{(for } m \neq 0) \quad (5.8) \]

We next calculate \( L^{(a)}(0,1) \). Recall that a (0,1) cycle is composed of a random number (denoted by \( B \)) of empty \( p \)-periods, a \( p \)-period in which a random priority \( k \) reserves the channel, an \( n \)-period in which a random station \( l \) starts transmitting, and a transmission time whose expected value is \( T_{lk} \). Similarly to the case \( m \neq 0 \), we assign \( c, l, \) and \( k \) the probabilities \( \frac{\lambda_{ck}}{\lambda^{(k)}} \), \( \frac{\lambda_{lk}}{\lambda^{(k)}} \) and \( \frac{\lambda^{(k)}}{\lambda} \), respectively. It thus remains to calculate \( E(B) \) - the expected value of \( B \). Denote

\[ \gamma = e^{-\alpha p \lambda} ; \quad \delta = e^{-\alpha \sum_{\tau \lambda^{(k)}}} \quad (5.9) \]

The independence of the arrival processes to the different stations implies the following probability distribution for \( B \),

\[ P(B=0) = 1-\delta ; \quad P(B=b) = \delta (1-\gamma)^{b-1} \quad (b=1,2,3,...) \]

The expectation of \( B \) can be calculated as,

\[ E(B) = \frac{\delta}{1-\gamma} \]

and hence, the expected length of a (0,1) cycle is given by,

\[ L^{(a)}(0,1) = \frac{\delta}{1-\gamma} \left[ a \alpha p + \sum_{c,l,k} \frac{\lambda_{ck}}{\lambda^{(k)}} \frac{\lambda_{lk}}{\lambda^{(k)}} \frac{\lambda^{(k)}}{\lambda} \left[ a(k+\mu(c,l)) + T_{lk} \right] \right] \quad (5.10) \]

The average cycle length \( \bar{L}^{(a)} \) is given by,

\[ \bar{L}^{(a)} = \sum_{m,k} \Pi^{(a)}(m,k) L^{(a)}(m,k) \quad (5.11) \]

Similar reasoning yields the following expressions for \( U_{lk}^{(a)} \) and \( S_{lk}^{(a)} \),

\[ U_{lk}^{(a)} = \left[ \sum_{m=0} \Pi^{(a)}(m,k) \frac{\lambda_{lk}}{\lambda^{(k)}} + \Pi^{(a)}(0,1) \frac{\lambda_{lk}}{\lambda} \right] T_{lk} \quad (5.12) \]

\[ S_{lk}^{(a)} = \frac{U_{lk}^{(a)}}{\bar{L}^{(a)}} \quad (5.13) \]

We conclude the description of the "(c,l)" method by calculating \( Q_{lk}^{(a)} \) and \( D_{lk}^{(a)} \), at which point we divide the discussion to constant transmission times and to generally distributed transmission times.

**Constant Transmission Times.**

Consider a given station \((i,j)\) and a specific \((m,k)\) cycle. If \( j < k \), station \((i,j)\) must be empty at the beginning of the cycle (or priority \( j \) would have transmitted), but it may be empty or full if \( j \geq k \). An empty station may
generate a message at any point during the cycle and remain full for a residual time whose expectation has been calculated in (4.16). The probability of station \((i,j)\) being full at the beginning of an \((m,k)\) cycle \((m\neq 0)\), can be approximated by the combinatorial quantity:

\[
cc(m,k,i,j) = \frac{n(p-k+1)-1}{n(p-k+1)} \frac{n(p-k)-1}{m} \quad (\text{for } j \geq k)
\]  

(5.14)

For \(j < k\), we define \(cc(m,k,i,j) = 0\). Clearly, the average backlog in \((i,j)\) during an \((m,k)\) cycle depends on the two indices \(c,l\) defined before. Denote this average backlog by \(q_{ij}(m,k,c,l)\) and the length of the given cycle by \(len(k,c,l)\).

\[
\text{len}(k,c,l) = r(k,c,l) + T_{ik} \quad \text{where} \quad r(k,c,l) = a(k+\mu(c,l))
\]  

(5.15)

Similarly to the argumentation in section (4), we obtain:

\[
q_{ij}(m,k,c,l) = cc(m,k,i,j) \text{len}(k,c,l)
\]  

(5.16)

At the beginning of a \((0,1)\) cycle, all stations are empty. Hence,

\[
q_{ij}(0,1,c,l) = \sum_{k=1}^{c-l} \frac{\lambda^{(k)}}{\lambda} \text{Res}(\text{len}(k,c,l),\lambda_{ij}) \quad (\text{if } m > 0)
\]  

(5.17)

Averaging over all values of \((c,l)\) with respect to their arrival rates and dividing by \(L^{(a)}\) yields,

\[
Q_{ij}^{(a)} = \frac{1}{L^{(a)}} \sum_{m,k} \Pi^{(a)}(m,k) \sum_{c,j} \frac{\lambda_{ek}}{\lambda^{(k)}} \frac{\lambda_{ik}}{\lambda^{(k)}} q_{ij}(m,k,c,l)
\]  

(5.18)

And finally,

\[
D_{ij}^{(a)} = \frac{Q_{ij}^{(a)}}{S_{ij}^{(a)}}
\]  

(5.19)

Generally Distributed Transmission Times:

The following modifications in equations (5.15) - (5.18) are necessary when transmission times \(\tau_{ij}\) are random variables with expectation, density function and moment generating function \(T_{ij}, f_{ij}, \Phi_{ij}\), respectively. Conditioning on \(\tau_{ij} = t\),

\[
\text{len}^{(a)}(k,c,l,t) = r(k,c,l) + t \quad \text{where} \quad r(k,c,l) = a(k+\mu(c,l))
\]  

(5.15a)

Averaging\(\text{Res}(\text{len}^{(a)}(k,c,l,t),\lambda_{ij})\) with respect to \(f_{ik}(t)\) yields,
and the average backlog in \((i,j)\) during an \((m,k)\) cycle in which station \((l,k)\) transmits, denoted by 
\[ q_{ij}^{(m,k,c,l)} \]
becomes,
\[
q_{ij}^{(m,k,c,l)} = \frac{e^{-\lambda_{ik}(k,c,l)}}{\lambda_{ij}} \Phi_{ik}(-\lambda_{ij}) + \lambda_{ij} \text{len}(k,c,l) - 1
\]

6. PERFORMANCE

Given the proposed RSP protocol, we wish to find the dependence of the system’s performance on its different parameters and compare it to that obtained by the existing protocol using replications of the scheduling period. Since the calculations are based on an approximate model, simulation results are also provided. The exact behavior of the modeled system has been simulated with 20,000 repetitions for each set of parameters. The throughput-delay curves have been calculated for several sets of parameters and showed an excellent fit between the results obtained by simulation and the numerical results obtained by using the approximate solution. Figure (2) depicts both the numerical results and the simulation results for the throughput-delay of a system consisting of three priority classes and two stations in each class. The difference between the two sets of results is shown not to exceed 5%, for three message sizes.

Since the proposed protocol is expected to perform differently for different load compositions of the priority classes, we consider the following two configurations: (a) All three priority classes have equal loads (shown in Figure (3)). (b) A heterogeneous load consisting of 10% messages of the highest priority, 20% of the second priority and 70% of the lowest priority [9] (shown in Figure (4)). We specifically compare the RSP protocol with the “replication of scheduling” approach [9]. We observe a system with three priority classes and four stations in each class. Figures (3) and (4) show the throughput-delay curves of this system for both protocols. As expected, the RSP differentiates well between the service provided to users of different priorities, shown in curves 1, 2 and 3.
see that the delay for the RSP protocol is lower for all 3 priority classes with increased improvement for the lower priorities, for both traffic configurations. Furthermore, the RSP protocol provides even a more significant service improvement when the traffic loads increase with reduced priority, the situation typically found in actual systems. In the case of heterogeneous arrival rates, shown in Figure (4), the suggested RSP protocol with aggregated reservation periods shows a noticeable improvement over the "replication of scheduling" protocol. Not only is there a significant decrease in the delay, but the capacity of the system is increased as well. This is true for all three priority classes and the improvement is larger for higher loads.

Lastly we observe the parameters most affecting the extent of improvement obtained by the RSP protocol: the number of stations and the ratio $a/T$ (the ratio between the propagation delay and message transmission times). The effect of the ratio $a/T$ on the system's performance is shown in Figure 5. This figure also compares the RSP protocol to the "replication of scheduling" protocol. We see that the higher the $a/T$ ratio, the more significant the improvement in the delay compared to the replication approach. Due to the associated decrease in scheduling penalty the influence of this ratio is larger for the lower priorities but exists to some extent in the higher priorities as well.

7. **Conclusions**

A class of demand assignment protocols for message or station prioritized LAN's has been defined. We showed how the proposed protocols correspond to existing demand assignment protocols and how they improve the existing protocols for both message and station based priorities. Two analytic models for analyzing demand-assignment priority-based protocols have been introduced: An exact model based on describing the exact state of the system and an approximate model which simplifies calculations without imposing any additional modeling restrictions and without significantly sacrificing accuracy. Numerical studies have been performed which showed: (1) The approximate solution shows excellent fit to the exact behavior as obtained by simulation. (2) The RSP protocol shows a significant improvement over existing protocols in both average delay and system capacity. (3) Class distinctions are absolute since we preserve priority rule "2" of section (2), hence unlike in random-access based prioritized solutions [11] no preemption is necessary for differentiating among priority classes.
REFERENCES


