DELAY ANALYSIS OF THE DQDB MAN BASED ON A SIMPLE MODEL

IOANNIS STAVRAKAKIS and RANDALL LANDRY

University of Vermont
CS/EE Dept., Burlington, Vermont 05405 U.S.A.

ABSTRACT In this paper a queueing model is developed for the delay analysis of a DQDB station. This model is based on some of the ideas originally presented in [1] and [2] and is analyzed by incorporating the queueing analysis developed in [3]. Delay results are obtained for a traffic load of 80% and for DQDB networks of up to 50 stations.

I. INTRODUCTION

The Distributed Queue Dual Bus (DQDB) medium access protocol has been adopted as a Metropolitan Area Network (MAN) Standard (IEEE 802.6). It is a high speed (over 100 Mbps) integrated services network designed to accommodate traffic generated by a variety of information sources, as well as provide for the interconnection of LAN's. A literature review for the DQDB network is presented in [4].

Over the last few years, the characteristics of the DQDB protocol have been studied and its performance has been evaluated, mainly through computer simulations. The most significant, probably, outcome from these studies has been the realization of the dependence of the stations performance from their physical location within the network (fairness issue of the DQDB protocol). It should be noted that the unfairness in the bandwidth allocation among the DQDB stations becomes an issue when the network is overloaded. A bandwidth balancing mechanism, incorporated in the final DQDB protocol (IEEE 802.6 Standard), allocates the bandwidth among the stations independently of their location when the system is overloaded.

In this paper the performance of the DQDB protocol is analyzed under underloaded operation. No bandwidth balancing mechanism is assumed to be in effect. Packet delay results are derived analytically by considering an approximate model for the queueing behavior of a DQDB station. The adopted model is based on some ideas which were presented in [1] and it is similar, in part, to the simulator model proposed in [2]. The queueing analysis developed in [3] is adopted for the derivation of the numerical results.

II. THE MODEL FOR A DQDB STATION

[0]Research supported by the National Science Foundation under Grant NCR-9011962

The DQDB network consists of two high speed unidirectional buses carrying information in opposite directions, Fig. 1. The network users (stations) are distributed along the buses and are capable of transmitting information to, or receiving information from, any network station. For the topology shown in Fig. 1, station $i, 1 \leq i \leq N$, uses bus A (forward bus) to transmit information to its downstream stations $i+1, \ldots, N$ and receive information from its upstream stations $1, 2, \ldots, i-1$; it uses bus B (reverse bus) to transmit information to its upstream stations 1,2,…,$i-1$ and receive information from its downstream stations $i+1, \ldots, N$. Each bus is equipped with a terminal station, called the head of the bus, which continuously generates fixed size slots. These slots propagate downstream past each station before being discarded at the end of the bus. Since the media access protocol is identical for both network buses, packet (or segments of information) transmissions only along the forward bus (A) will be considered.

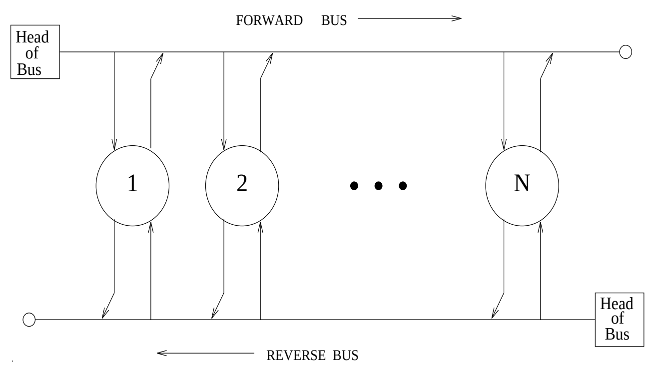

Fig. 1: A simple DQDB network

At any given time a DQDB station may be either busy or idle; it is busy if a packet is queued in its buffer (to be transmitted on the forward bus), and it is idle otherwise. A busy station must notify all upstream stations by registering a request. This is implemented by setting the request bit of the first slot passing by on the reverse bus, whose request bit is reset. Each station is equipped with two counters; the request counter, REQ-CNTR, and the count-down counter, CD-CNTR. The REQ-CNTR of an idle station is incremented by one any time a set request bit is detected. It is decremented by one each time an empty slot is detected on the forward bus, since this slot will be used to satisfy a downstream request.

The passage of a station from the idle to the busy state triggers the following actions: (a) The value of the REQ-CNTR is downloaded into the CD-CNTR. (b) The REQ-CNTR is reset to zero. (c) A request is placed on the reverse bus as explained above. During the period for which a station is busy the following occur: (a) Its CD-CNTR is decremented by one for each empty slot detected on the forward bus. (b) Its REQ-CNTR is incremented by one any time a set request bit is detected; no decrement of this counter occurs during the busy period.

When the value of the CD-CNTR reaches zero, the station is permitted to transmit into the next empty slot on the forward bus. The zero value of the CD-CNTR indicates that requests for empty slots made by the downstream stations before the tagged station became busy have been satisfied. When the packet is delivered to the forward bus, the station enters the idle state. If another packet has been waiting in the buffer, the station enters the busy state instantaneously. Notice that the above procedure prevents any station from having more than one outstanding request at any given time, which could cause monopolization of the network by a few stations.

It is clear that the implementation of the distributed queueing protocol described above requires that both buses be continuously monitored by all stations. The latter is necessary in order for the request process, $\{r_j^i\}_{j\geq 1}$, and the empty-slot process, $\{e_j^i\}_{j\geq 1}$, associated with station $i, 1 \leq i \leq N$, be observed ; $e_j^i = 1$ ($e_j^i = 0$) if the j^{th} slot on the forward bus passing by station i is empty (occupied); $r_j^i = 1$ ($r_j^i = 0$) if the request bit of the j^{th} slot on the reverse bus passing by station i is set (reset). The empty-slot process $\{e_j^i\}_{j\geq 1}$ is determined by the packet transmission activity of the upstream stations with respect to i. The request process $\{r_j^i\}_{j\geq 1}$ is determined by the requests generated by the downstream stations with respect to station i. In the rest of the paper, station i will be assumed to be the tagged station. All processes associated with a station will be assumed to be associated with the tagged station; the superscript i will be dropped for simplicity in the notation.

The queueing behavior of the tagged DQDB station can be described by considering the three-buffer centralized queueing system shown in Fig. 2. T represents the actual buffer of the tagged station; it is served according to the FIFO policy. The capacity of the forward bus (server) is shared by all network stations according to the DQDB protocol. Whenever an occupied slot passes by the tagged station $\overline{e}_j = 1$, where $\overline{e}_j = 1 - e_j$, and the server is assumed to be at buffer B_{up} containing all packets generated by the upstream users. Whenever an empty slot passes by the tagged station, the server is assumed to be at buffer B_d containing all packets generated by the downstream users. The server is assumed to be at buffer T whenever a packet is transmitted by the tagged station.

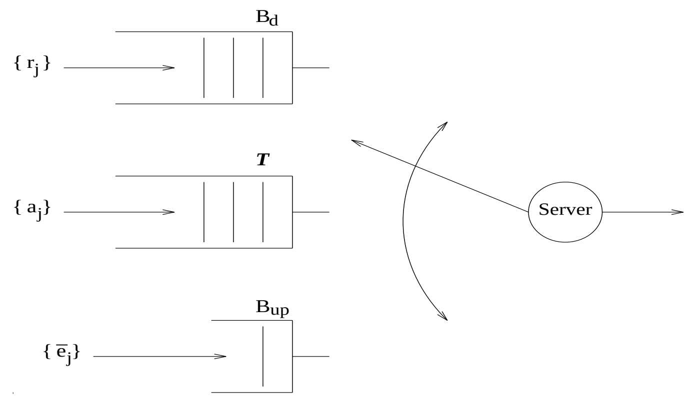

Fig. 2: The 3-buffer queueing model for a DQDB station

The service policy of the three buffer queueing system is very difficult to be exactly described. Since the objective is the development of a queueing model for the tagged station, it is sufficient that the service policy determines the time instants when the server becomes available to T; detailed knowledge regarding the sharing of the server between B_d and B_{up} is not necessary. The service policy for T which results in a buffer behavior similar to that of the tagged DQDB station is established in the sequel.

Initially, suppose that the upstream stations are always idle and, thus, the forward bus capacity is shared by the tagged station and the downstream users. The latter implies that the server never switches to B_{up}.

Let t_k denote the time instant at which the k^{th} packet moves to the Head of the Queue (HoQ) of the tagged station. A time instant t_k is defined as the end of the t_k slot; movements of the packets in the tagged queue are assumed to occur instantaneously at the end of a slot. According to the DQDB protocol, all requests which are pending at time instant t_k will be satisfied by allowing the equivalent number of (empty) slots to pass by the tagged station, before the packet at the HoQ is transmitted. The service policy in the queueing system of Fig. 2, which reflects the network behavior, can be described in the following way. At time instant t_k a gate closes in B_d and all requests arrived by t_k are served. Then the server switches to T and serves the k^{th} packet. Let $\overline{t}_k$ denote the time instant of the completion of the k^{th} packet transmission. If the $(k+1)^{st}$ packet has arrived at the tagged station by that time, it is forwarded to the HoQ at time instant $\overline{t}_k$. Then the server moves as described above. If the $(k+1)^{st}$ packet arrives after time instant $\overline{t}_k$, the server moves to B_d at $\overline{t}_k$ but leaves the gate open until t_{k+1}, the time instant in which the $(k+1)^{st}$ packet enters the HoQ. It serves all requests which arrived by time instant t_{k+1} and then switches to T.

The service policy for T and B_d described above is identical to that introduced for the first time in [1], and which has been adopted for the development of the simulator for a DQDB station in [2]. Note that the two-buffer queueing system models the queueing behavior of the tagged DQDB station in the absence of any traffic generated by the upstream users. When the latter traffic is present, the server must switch to B_{up} to reflect the unavailability of certain slots to either the tagged station or downstream users. The process of the transmitted upstream packets (occupied slots passing by the tagged station) is considered to be the arrival process to B_{up}. The upstream packets are assumed to have priority over any other packets (tagged packets or regular packets) in the system. That is, the server becomes available to T or B_d only if B_{up} is empty. A buffer capacity of one packet is sufficient for B_{up}.

The packet arrival traffic to the tagged station, $\{a_j\}_{j\geq 1}$ is assumed to be described by an independent and identically distributed process. The packet arrival processes to B_d and B_{up} depend on the state of the distributed queue. The exact description of these processes would require a complex probabilistic model. The input process to B_d and B_{up} deliver at most one packet per slot. Upstream packets and downstream requests are not distinguished since they require the same amount of service; both entities will be called packets. One packet arrives at B_{up} at the (end of the) j^{th} slot if $\overline{e}_j = 1$. One packet arrives at B_d at the j^{th} slot if $r_j = 1$, that is, when a request is generated by the downstream users. The simplest discrete-time process which takes the values 0 or 1 is the Bernoulli process. Unless otherwise stated, the arrival processes $\{\overline{e}_j\}_{j\geq 1}$ and $\{r_j\}_{j\geq 1}$ will be assumed to be Bernoulli processes with rates λ_{up} and λ_d, respectively; $\lambda_{up}(\lambda_d)$ denotes the total packet traffic rate generated by the upstream (downstream) stations.

During the service period of the requests accumulated in B_d, the server will be occasionally switched to B_{up}, as explained before. For the purpose of developing a queueing model for the tagged station it may be assumed that the high priority packets (occupied slots in the DQDB) arriving at B_{up} are perceived as additional requests which have to be accommodated as well. Thus, it may be assumed that the arrival process to B_d consists of both the request process $\{r_j\}_{j\geq 1}$ and the slot occupancy process $\{\overline{e}_j\}_{j\geq 1}$ which behaves as an interfering process to the service of the requests.

In view of the above discussion the system consisting of two queues, T_1 and I_1 where $I_1 = B_{up} \bigcup B_d$ with arrival processes $\{a_j\}_{j\geq 1}$ and $\{r_j + \overline{e}_j\}_{j\geq 1}$, respectively, seems to be a meaningful approximate model for the queueing behavior of the tagged DQDB station. The interference on the service of the requests, coming from the upstream traffic, is expected to be accounted for through the introduction of the average Bernoulli interfering process $\{\overline{e}_j\}_{j\geq 1}$. The exact service policy within the buffer I_1 does not affect the queueing behavior of T_1. The total number of upstream packets and downstream requests which are served while the server is at I_1, is of importance and not the ordering in the service. Although the average traffic load due to the upstream packets is accounted for through the consideration of the process $\{\overline{e}_j\}_{j\geq 1}$ at the input of T_1, the interference on the tagged station, caused by the upstream packet flow immediately after the end of the service of I_1, is not sufficiently captured. Notice that according to the DQDB protocol, an empty slot has to pass by after all pending requests have been served, in order for the tagged packet to be transmitted. Referring to the queueing model, the server has to serve all interfering upstream packets arriving over consecutive slots immediately after it leaves I_1, before it becomes available to T_1. To account for this additional delay without superficially overloading the two-buffer system and reflect the Head of the Line (HoL) priority of the interfering packets over the tagged packets, a second stage queueing system is introduced. The complete queueing model for the DQDB station is shown in Fig. 3.

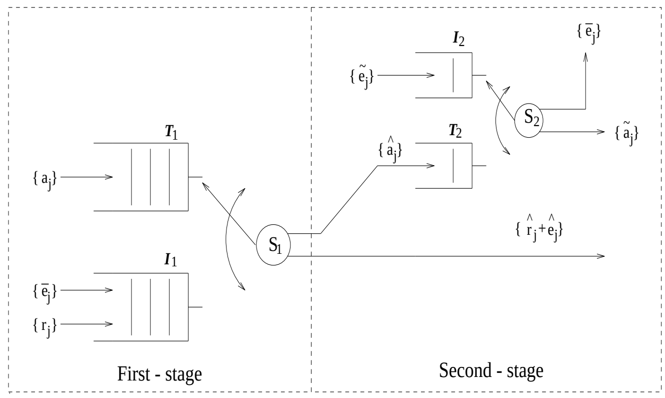

Fig. 3: The 2-stage queueing model for a DQDB station

The server S_1 (Fig. 3) operates on the two queues T_1 and I_1 as described earlier. As soon as a tagged packet receives service from S_1 it is delivered to the network immediately, provided that $\tilde{e}_j = 0$; $\{\tilde{e}_j\}_{j\geq 1}$ denotes the upstream packet process and it is considered to be identical to $\{\overline{e}_j\}_{j\geq 1}$, unless otherwise stated. The service time of the tagged packets at server S_2 is assumed to be zero since the transmission time is already accounted for in the first stage. If $\tilde{e}_j = 1$, that is if an interfering upstream packet arrives at I_2 simultaneously with the tagged packet, then the interfering packet receives service under the HoL priority policy; its service time is equal to one time slot. The additional delay that the tagged packet undergoes is determined by the number of consecutive interfering packets arriving at I_2, where the first of these packets arrives simultaneously with the tagged packet. Notice that no queue is actually formulated in I_2. As a result, a buffer with single packet capacity is sufficient for the queue I_2.

III. NUMERICAL RESULTS

Details on the analysis of the approximate queueing model for the DQDB station presented in the previous section may be found in [5]. The accuracy of the queueing model is illustrated in this section through the comparison of analytical results with computer simulations. Most of the presented results are derived for a network traffic load of $\lambda = .8$, which is considered to be the expected nominal point of operation. All plots present the mean packet delay of a DQDB station as a function of the station's position on the network. Stations 1 and N denote the head of the forward and the reverse buses, respectively; N denotes the network size.

Station N never transmits on the forward bus since no station is downstream with respect to that station. For this reason the packet delay associated with stations 1 to N-1 is plotted only. The solid lines represent the simulation results, while the dotted lines represent the corresponding 95% confidence intervals. The two points associated with each station (position) present the bounds on the mean delay obtained from the analysis of the proposed model. In most cases, the upper and the lower bounds coincide and only one point is shown. The packet arrival process to the i^{th} station is assumed to be Bernoulli; its intensity is a linear function of the number of downstream stations. The unfairness property of the position dependent delay performance of the DQDB station was evident in all of the derived results.

Numerical results for a network of $N = 10$ stations are presented in Fig. 4. The inter-station distance, in signal traveling time, is assumed to be equal to 1 time slot (packet transmission time). For traffic loads of up to 80% the analytical results are found to be very close to the computer simulations. For higher traffic loads larger deviation of the analytical results from the computer simulation was observed. The latter is expected since the impact of the approximation is intensified as the network operation approaches instability.

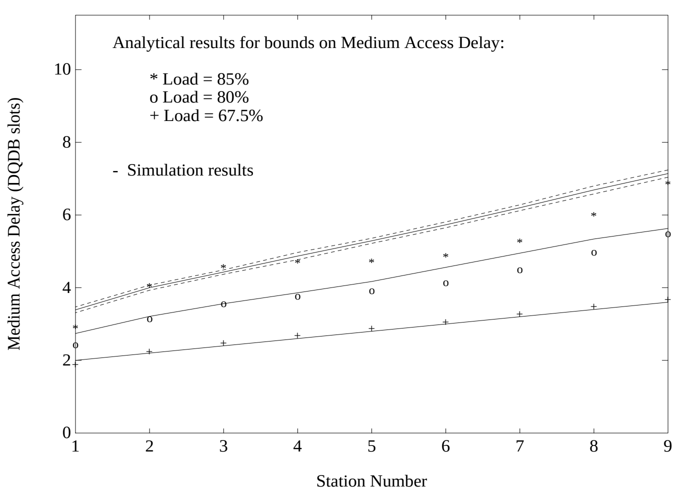

Fig. 4: Mean packet delays for N = 10

For larger networks, in terms of number of supported stations and/or inter-station distances, it is expected that the correlation among consecutive values of the processes $\{r_j\}_{j\geq 1}$ and $\{e_j\}_{j\geq 1}$ be intensified. The participation of a large number of stations in the DQDB protocol tends to generate bursts of occupied slots and requests. Furthermore, the larger the inter-station distance, the greater the number of interfering events due to the greater propagation delay. In view of these remarks, it is expected that the Bernoulli model be inefficient in describing the processes which interfere with the transmission of the tagged packets. The results obtained for networks with inter-station distance of 1 slot with $N = 30$ stations and $N = 50$ stations, under the proposed model, were found to slightly - but consistently - underestimate the mean packet delay. This underestimation may be attributed to the relative inefficiency of the Bernoulli model.

When the inter-station distance increases as well, additional correlations are expected to be introduced to the processes $\{r_j\}_{j\geq 1}$ and $\{e_j\}_{j\geq 1}$, as explained above. As a result, the mean delay results will be, in general, increased compared to those of a network with the same number of supported stations, but smaller inter-station distances. The simulation results for networks with inter-station distance of 5 slots, shown in Fig. 6 (for $N = 50$), illustrate this behavior when compared with those in Fig. 5. Note that the parameters of the proposed model are not affected in any respect by the inter-station distance. As a result, the obtained analytical results remain unchanged for larger inter-station distances, as long as the number of stations and the traffic load distribution remain unchanged. The latter reflects the inefficiency of the proposed model in capturing the impact of large inter-station distances on the queueing behavior of the tagged DQDB station.

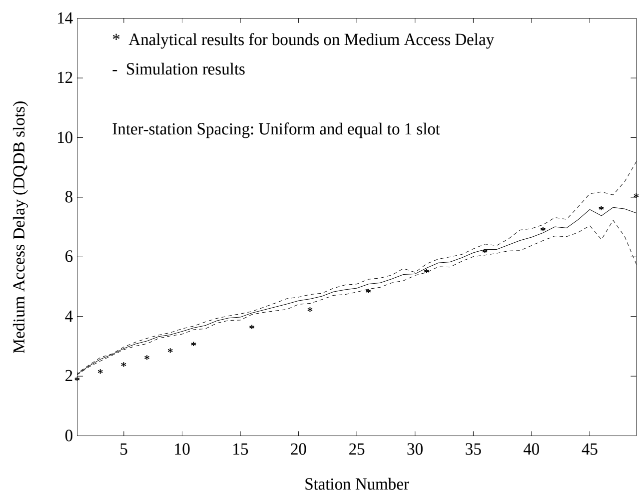

Fig. 5: Mean packet delays for N = 50

To capture the correlations in the true processes $\{r_j\}_{j\geq 1}$ and $\{e_j\}_{j\geq 1}$ introduced by larger networks (in terms of the number of stations and/or inter-station distances), a more complex model is necessary for the approximation of these processes. A two-state Markov model is a fairly simple model which is capable of capturing, to some extent, the correlations present in the interfering processes. Unfortunately, the analysis of the first-stage queueing system under Markov arrival processes is expected to be difficult and computationally involved.

Thus, in this paper only the interfering process $\{\tilde{e}_j\}_{j\geq 1}$ in the second-stage queueing system is modeled as a Markov process, in our effort to increase the accuracy of the proposed model and extend its applicability region.

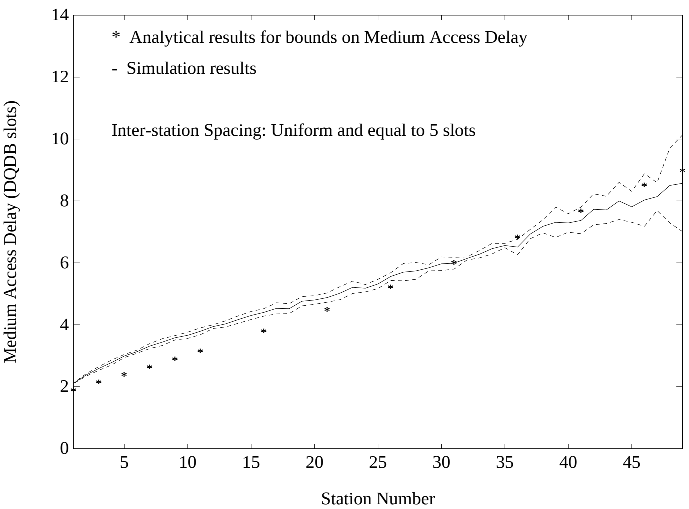

Fig. 6: Mean packet delays for N = 50

The interfering process $\{\tilde{e}_j\}_{j\geq 1}$ is modeled as a two-state Markov chain; a packet is delivered to I_2 when the chain is in state 1; no packet is delivered when in state 0. This process generates blocks of consecutive packet arrivals of geometrically distributed length, followed by idle periods of some other geometrically distributed length. This Markov model is completely determined in terms of the steady-state probability that the chain is in state 1, $\pi(1)$, and the average block length F. The values of these parameters are determined as follows:

$$\pi(1) = \lambda_{up} \quad , \quad F = \frac{N_{up}s}{ks+1}, \quad k = \frac{N}{10} + \frac{10}{N}, \qquad (1)$$

where N_{up} denotes the number of upstream users and s is a positive integer which denotes the inter-station distance (in multiples of slots).

The selection for $\pi(1)$ is obvious; it is desired that the packet rate generated by the true process $\{\tilde{e}_j\}_{j\geq 1}$ and the Markov model be identical. The selection for F is less obvious and, to some extent, arbitrary. The relationship between the size of the block F and N_{up} and s reflects intuitively expected dependencies. The value of k given in (1) was found to generate quite accurate results for a wide range of network parameters and reflects the intention to adopt the Markov model for networks of greater than 10 stations. The mean delay results under the Markov model for $N = 50$ stations are shown in Fig. 5; the inter-station distance s is equal to 1 (slot). Similar plots, when the inter-station distance s is equal to 5 (slots) are shown in Fig. 6. It turns out that the analytical results are in good agreement with the computer simulations.

The useful insight and the results obtained through this work have already inspired extension of the study in a number of directions. Such directions include the study of the first-stage queueing system under Markov arrivals and/or multiple per slot arrivals, and the extension (with possible improvement) of the Markov model for the description of $\{r_j\}_{j\geq 1}$.

REFERENCES

(1) C. Bisdikian, "A Queueing Model with Application to Bridges and the DQDB (IEEE 802.6) MAN", Research Report RC 15218, IBM Thomas J. Watson Research Center, Yorktown Heights, Dec. 1989. Accepted, Computer Networks and ISDN Systems Journal.

(2) C. Bisdikian, "A Quasi Gated Service Discipline Model for a DQDB Data Station", 2^{nd} ORSA Telecommunications Workshop, March 1992, Boca Raton, FL; also, Research Report RC 15587, IBM Thomas J. Watson Research Center, Yorktown Heights, March 1990.

(3) I. Stavrakakis, "A Considerate Priority Queueing System with Guaranteed Policy Fairness", IEEE INFOCOM'92 Conf., May 1992, Florence, Italy.

(4) B. Mukherjee, C. Bisdikian, "A Journey Through the DQDB Literature", Research Report RC 17016, IBM Thomas J. Watson Research Center, Yorktown Heights, Nov. 1991. To appear, Perform. Eval. J., Special Issue on High Speed Telecommun. Systems, 1992.

(5) I. Stavrakakis, R. Landry, "Delay Analysis for a DQDB Station Based on a Two-Stage Queueing Model", Report, CSEE Dept., University of Vermont, 1991.