

Metropolitan Area Networks (MANs): Protocols, Modeling and Performance Evaluation

M. Conti, E. Gregori, L. Lenzini

CNR - Istituto CNUCE Via S. Maria, 36, 56100 Pisa Italy

Abstract

With the continuing success of Local Area Networks (LANs), there is an increasing demand to extend their capabilities towards higher data rates and wider areas. At high data rates and long distances the packet transmission time may become comparable or significantly less than the network propagation delay. For this reason Medium Access Control (MAC) protocols which were developed for LANs are no longer viable. Together with the progress in fiber optic technology, this has given rise to the so-called Metropolitan Area Networks, or MANs. These can span much greater distances than current LANs, and offer data rates in the region of hundreds of Megabit/sec (Mbps). This survey first sketches the problems encountered in using the MAC protocols defined for LANs on higher-speed and longer-distance networks; and then it focuses on two MAC protocols (FDDI and DQDB) developed for MANs by standardization bodies. These two MAC protocols represents two different approaches to overcome the limits of LAN MAC protocols. FDDI represents the evolution of the token passing class of MAC protocols, while the latter is a new brand of MAC protocol with a completely distributed control. This survey focuses on both MAC protocols with particular attention on their performance analysis.

1 Introduction

The success of MANs is strictly connected to the opportunity they give to develop new networking products capable of providing high-speed communications and interconnectivity between communicating applications at competitive prices which nonetheless give an adequate return on the manufacturers' investments. A major factor in achieving this goal is the availability of appropriate Networking Standards. FDDI and DQDB are the two standard technologies for MANs for which industrial products are already available.

The importance of FDDI and DQDB has meant that their performance has been analyzed by several groups. Most of the existing results have been obtained via simulation as it is extremely difficult to analytically solve detailed models of both MAC protocols. In fact, FDDI has a more complex behavior than a polling system with an exhaustive-limited service discipline [59], while DQDB behaves like a round-robin scheduling algorithm [36] for very short networks, but it deviates significantly when the length of the network increases. Due to the complexity of these protocols, exact models of the FDDI and DQDB MAC protocols have only been solved through simulative analysis, while models with analytical solutions have been developed to approximate each protocol's behavior under specific network configurations and workload conditions.

Research in this field is facing a wide range of performance-related problems, such as: 1) determining the relationship between bandwidth allocation schemes and throughput, delay distribution and packet loss rate 2) dimensioning the key network components (e.g., buffer size), and 3) tuning the network parameters.

The target of this tutorial is to present a structured view of the performance modeling activity related to FDDI and DQDB which has been published. For both MAC protocols we propose a taxonomy of the analytical models. Some relevant models of each class will be discussed by presenting the main simplifying assumptions, the

technique used for solving the model and the performance indices analyzed. In the presentation the original notation is used and this sometimes results in a non uniform use of symbols.

To make reading easier, the most relevant features of the MAC protocols will be summarized before the performance modeling issues of the protocols are discussed.

This survey is divided into three parts. Section 2 discusses the main issues related to the evolution from LAN to MAN. Sections 3 and 4 are devoted to FDDI and DQDB, respectively.

2 From LANs to MANs

Since a LAN/MAN network relies on a common transmission media, MAC protocols have been designed to manage the sharing of the transmission media. The aim of a MAC protocol is to control the interference and competition among users while optimizing overall system performance and yet nevertheless avoiding pitfalls. The target of a MAC protocol is thus to share resources efficiently among several users. This efficiency can be expressed in terms of *fairness* and *capacity* ([37], [1]).

Fairness means that the network does not differentiate between stations in granting them access rights to the transmission bandwidth [28].

Capacity indicates to what extent the protocol utilizes the channel bandwidth. In fact, a MAC protocol is a distributed algorithm among several stations; some information must be, either implicitly or explicitly, exchanged among the stations. Since the transmission media is the only means of communication among the stations, it will not always be fully used to transmit user-messages. In the literature the MAC protocol *capacity* figure (ρ_{\max}) is used to characterize this aspect. This is defined as the fraction of the medium bandwidth used by the nodes when each node tries to seize all the medium capacity.

MAC protocols intended for LANs are not suitable for high transmission speeds and long distances. This applies to all the popular IEEE standards for LANs: Ethernet, Token Ring, and Token Bus. In these protocols, the MAC protocol capacity decreases when the ratio between the length of the network and the length of a packet increases. In literature this ratio is often referred to as a . The ρ_{\max} value depends on the specific MAC protocol. In the following we report the ρ_{\max} value for Ethernet, Token Ring and Token Bus

$$\text{- Ethernet: } \rho_{\max} \leq \frac{1}{1 + \chi \cdot a},$$

where χ depends upon the approximations made during the analysis; for example, $\chi=6.44$ in [55], $\chi=3.44$ [57] and $\chi=7.34$ [30].

$$\text{- Token Ring [57]: } \rho_{\max} \leq \begin{cases} \frac{1}{1 + \frac{a}{N}} & \text{if } a < 1 \\ \frac{1}{\left(1 + \frac{1}{N}\right) \cdot a} & \text{if } a > 1 \end{cases},$$

where N is the number of active stations in the network.

$$\text{Token Bus [30]}^1: \rho_{\max} \leq \frac{1}{1 + \frac{a}{3 \cdot N}}$$

By assuming that there are packets of 1000 bits and a light speed in the fiber of 200,000 km/sec in Table 1 we report the a values for various configurations and transmission speed.

Table 1:
 a values for different network configurations and transmission speeds

Capacity	Coverage		
	1 km	10 km	100 km
10Mbps	0.05	0.50	5
100 Mbps	0.50	5	50

For a network with 50 active stations the maximum network utilization which can be achieved, for various a values, are reported in Table 2.

Table 2:
 ρ_{\max} values

	CSMA/CD	Token Ring	Token Bus
$a=0.05$	76%	100%	100%
$a=0.50$	24%	100%	100%
$a=5$	3%	20%	96%
$a=50$	0.3%	2%	75%

Table 2 clearly shows that the CSMA/CD and the Token Ring perform poorly at speeds and coverages typical of a MAN. On the other hand, the performance of the Token Bus does not drop significantly; in fact, one of the standard for high-speed LAN/MAN, FDDI, inherits most of the Token Bus MAC protocol mechanisms; in [45] it is provided a general proof of the equivalence between Token Bus and FDDI networks.

Fairness and capacity are used to evaluate the MAC protocol algorithms, however from the user standpoint other performance figures are needed to measure the quality of service the user can rely upon. The most widely used performance measure is the delay, which can be defined in several forms, depending on the time instants considered when measuring the delay (access delay, queueing delay, propagation delay, etc.).

In LAN networks, whose main target is the support of EDP data applications, the quality of service is generally expressed in terms of average delay and throughput; while for MANs, which have the potentiality to support beyond EDP data applications time-constrained applications (e.g., voice and video) as well, the delay distribution would also be necessary. In fact, the most important aspect of a time-constrained application is that messages should be delivered to the destination within a given amount of time after their generation. The most relevant performance figure for these applications is therefore the percentage of messages which are delivered with a delay lower than the given constraint.

Extensive analyses of distributed and "centralized" token-passing MAC protocols for MANs have been carried out. For both classes, the pros and cons have been identified. In particular, the most interesting features of the DQDB MAC protocol is its ability to

¹ This bound is obtained by assuming that each node can only transmit a single packet whenever it receives the token.

guarantee a utilization of the medium capacity which does not depend on bus length, medium capacity or packet length, and the capability to provide access delays of only a few microseconds at light loads ([5], [11], [12]). On the other hand, the FDDI MAC protocol [2] behaves more predictably and fairly than DQDB, and this means there is guaranteed bandwidth for at least one class of traffic. Unfortunately, the FDDI MAC protocol capacity still depends on the a value and this makes the FDDI MAC protocol unsuitable for Gigabit MANs.

Below we focus on the relevant aspects, from the performance analysis standpoint, of the FDDI and DQDB MAC protocol. We present an updated survey of some significant analytical models for each protocol.

3 FDDI

Fiber Distributed Data Interface (FDDI), which employs an optical fiber medium, is a 100 Mbit/s LAN based on a Token Ring protocol [2].

The network topology, two contra-directional rings, allows very flexible configurations extending from a few metres of fibre path length supporting a few stations, up to 100 km supporting up to 500 stations. Adjacent ring stations can be up to 2 km apart with standard multimode optical transmission links, but optionally single mode optical links may be used where greater inter-station distances of up to 40 km are required ([51], [44]).

Distributed protocols are used for initialisation and management of the FDDI ring and also for error recovery purposes [2].

The FDDI standard specifies the building blocks of the functional architecture of an FDDI station. The MAC block is mainly responsible for the FDDI performance since it controls the token passing ring protocol which determines the behavior of the FDDI ring. In this tutorial we only focus on the MAC protocol which will be described in the next section.

3.1 Description of the MAC Protocol

The MAC protocol controls the transmissions onto the physical ring media. This control is exercised by means of a Timed Token protocol, in which a unique sequence of data called a token is passed around the ring from station to station. A station wishing to transmit information must first wait for the token to arrive; the token is then "captured" by stripping it from the ring. The station can then transmit its queued frame(s) of data, after which it issues a new token which provides other stations with the opportunity to access the medium.

The frames of data transmitted by the originating station are regenerated and repeated by the other active stations on the ring. While repeating incoming frames, a station also examines the Destination Address for a match with its own address. If a match occurs, the station copies the frame contents into its receive buffers as it transmits the frame onwards.

Each frame of data transmitted eventually comes back to the station that originated it, which is then responsible for stripping the frame from the ring. The originating station recognises the Source Address contained in the frame as its own address, after which stripping takes place. The decision to strip a frame is based on the recognition of the frame's Source Address, which cannot occur until the initial part of the frame has been repeated. This means that the stripping process leaves frame fragments circulating on the medium. Frame fragments are ultimately discarded when they arrive at a station that is transmitting its own frames onto the medium.

Timed Token Protocol. The Timed Token protocol supports two major classes of service: *synchronous* and *asynchronous*.

The synchronous traffic has a preallocated bandwidth while the bandwidth for asynchronous transmissions is instantaneously allocated to a station, when it captures a token, from the unused bandwidth.

Specifically, a Target Token Rotation Time (*TTRT*) is negotiated by a station during ring initialization (the negotiated value of *TTRT* is named *T_Opr*) and a synchronous bandwidth is allocated to each station as an $X\%$ of *TTRT*, i.e., a station can transmit every time it captures the token synchronous data for a time up to its $X\%$ of *TTRT*. In any case the aggregate synchronous bandwidth can never exceed $TTRT - \alpha$ where α is a constant term defined in the standard which takes into account the maximum ring latency, the maximum frame length, and the time it takes to transmit a token.

To compute the maximum time a station can transmit asynchronous data when it captures a token two timers are used: the Token Rotation Timer (*TRT*) and the Token Holding Timer (*THT*). *TRT* measures the time between the receipt of two consecutive tokens while *THT* is used to limit the transmission of a station when a token is captured. If *TRT* reaches *TTRT* before the token returns to the station, a variable, named *Late_Ct*, is set to one and *TRT* is restarted. When the token arrives at a station with *Late_Ct*=1 the token is called a *late token*, only synchronous transmissions are enabled and *Late_Ct* is set to 0. On the other hand if the token arrives before *TRT* reaches *TTRT*, the token is named an *early token*. Whenever an early token is captured, the current value of *TRT* is stored in the *THT*, *TRT* is reset and synchronous transmissions (up to $X\%$ of *TTRT*) are carried out. After synchronous transmission, *THT* is enabled and asynchronous transmissions start. The difference between *TTRT* and *THT* is the maximum time available for asynchronous transmissions in this cycle. A station may initiate a transmission of an asynchronous frame if the timer *THT* has not reached the *TTRT* threshold. This may cause an additional delay in the release of the token (hereafter called *asynchronous overrun*) since the transmission of an asynchronous frame is always completed. The asynchronous overrun is bounded by the time for the transmission of a frame of maximum length. In the FDDI standard asynchronous overrun is 0.361 msec, which corresponds to the time required to transmit a maximum frame length.

Multiple levels of asynchronous priorities may be distinguished by a station. For each priority level n , a threshold value ($T_Pri(n)$) is defined.

$T_Pri(n)$ are an ordered sequence of values in the range $[0, TTRT]$, higher priorities have higher T_Pri values and the highest priority has a threshold which is equal to *TTRT* ([23], [54]).

Asynchronous transmissions start from the highest priority. Asynchronous frames of priority n may only be transmitted if *THT* is less than $T_Pri(n)$. If multiple asynchronous priority levels are not implemented, all asynchronous frames have a threshold value which is equal to *TTRT*.

The protocol behavior was formally studied in ([56], [34]). In these papers it was formally proved that: (1) the average token rotation time does not exceed the *TTRT*, and (2) the maximum token rotation time does not exceed twice the *TTRT*.

3.2 MAC Protocol Modeling

Although many FDDI analytical studies have already been published, the majority contain simplifying assumptions. The main difficulty for the analysis of the FDDI MAC protocol is the high degree of complexity and interdependence of the various processes that describe the operations of the protocol itself. In fact, when a station has seized the token, synchronous frames (if any) are always transmitted, whereas asynchronous frames are only transmitted if the preceding token rotation time does not exceed *TTRT*. This implies that there are interdependencies between the total service time given at one station, the service time required at subsequent stations and the total cycle time. Therefore, exact analytically-tractable solutions for an FDDI network are very difficult to formulate. Simplifying assumptions thus have to be made in order to

obtain analytically-tractable solutions. Furthermore, even when only synchronous traffic is transmitted, and hence the token holding time constraint and the feedback mechanism of cycle time have no influence, only approximate solutions are known for the resulting model (which reduces to a polling system with an *Exhaustive Time-limited discipline*).

Table 3
FDDI Models Taxonomy

		Buffer size	Traffic Type		Performance indices			
Network Wide Models	Deterministic	M=1	Model 1	A	Th	-	-	
		M>1	Model 2	A/S	E[D]	PI	-	
	Stochastic	M>1	Model 3	A	E[D]	M-pdf	PI	
		M=∞	Model 4	S	E[D]	-	-	
	Station in Isolation Models	Stochastic	M>1	Model 5	A	E[D]	M-pdf	PI
			M=∞	Model 6	A	E[D]	M-PGF	-
				Model 7	S	E[D]	M-pdf	D%
			Model 8	S	D-LST	M-PGF	-	

Th: Throughput;

E[D]: Average Delay;

PI: Packet loss rate;

M-pdf: Probability distribution function of the buffer occupancy;

M-PGF: Probability generating function of the buffer occupancy;

D-LST: Laplace-Stieltjes Transform of the delay distribution,

D%: Delay percentiles.

In order to provide a structured overview of the FDDI analytical studies we introduce the taxonomy shown in Table 3, which preliminarily classifies the FDDI models into two categories: the former (or first category) contains *Network-Wide Models* while the latter (or second category) contains *Station-in-Isolation Models*.

The models in the first category characterize the overall FDDI structure in terms of the ring and stations spaced along it. All the models in this category belong to the class of *multiqueue systems with cyclic service* [59]. In literature, these systems have frequently been investigated. They can be distinguished from each other in a number of ways [59] depending on the buffering at each station, the type of service, the polling order, and the assumptions of arrival processes at the various stations.

The analytical models in the second category tag the station under study. With respect to the tagged station, the FDDI network is partitioned into the tagged station itself, and the complementary part of the network (C_NET) which aggregates all the other stations. In this approach the tagged station is modelled as a *single server queueing system with server vacations* [61]. The server vacation time represents the period between the token's departure from the tagged station and its subsequent arrival at this station, i.e., the time it takes the token to cross C_NET.

Models belonging to both classes can be further subdivided (see Table 3) on the basis of buffer size (i.e., M=1, M finite and M infinite), traffic type (i.e., synchronous (S), asynchronous (A) and a mixing of both (A/S)) for which performance indices are derived. For each leaf of the taxonomy, presented in Table 3, we identify a significant model for which we provide the model description, the solution technique and the performance indices obtained.

3.3 Network-Wide Models

This section outlines four special cases of the network-wide model. All the models falling in this category are derived from the following abstract model

- The system has N queues (stations) and a single server. The queues can have either one M ($M > 1$) or infinite buffers.
- Synchronous and asynchronous frames arrive at the queues in accordance with Poisson arrival processes.
- The server (token) walks from queue to queue in a fixed order. The time needed to switch the server from queue i to queue $(i + 1)$ is modelled by a delay (*switchover time*) r_i . The switchover time r_i corresponds to the propagation delay of the signals between stations i and $(i + 1)$ plus the latency caused within station i .
- When the server reaches a queue, synchronous frames are always transmitted (if any) whereas asynchronous frames are only transmitted if the preceding token rotation time does not exceed TTRT. Once the server has served a queue it goes to the next queue. The server can take either a constant or a random amount of time to transmit a frame.

The first model (*Model 1* [23]), performs a throughput analysis of the network under the assumption that all nodes are saturated by frames to send, i.e., Asymptotic Analysis. Under this assumption the network behavior is deterministic and the authors derive an exact solution of the network model. The other three models assume Poisson arrivals and thus the system behavior is stochastic. More specifically, under the assumption of a single buffer, *Model 2* [60] exactly represents the network behavior for which an exact solution is provided. *Models 3* and *4* are approximate models for which approximate solutions are derived. The main differences between these models is in the buffer size (finite or infinite) and the solution technique (approximation based on an iterative scheme vs. approximation based on a pseudo-conservation law).

Model 1: Asymptotic Analysis. This model, proposed and analysed by Dikeman and Bux in [23], is used for evaluating the maximum total throughput γ_{\max} for *FDDI* when only one asynchronous priority level is in use.

The expression for γ_{\max} is derived by assuming that each actively transmitting station continuously has asynchronous frames queued for transmission. To simplify the analysis, frame transmission times are assumed to have constant length F . Asynchronous overrun are assumed to be of constant length R . On the basis of the above assumptions, the ring quickly converges to steady-state operation, so that on successive rotations the active stations cycle through a finite number of transmission states. A transmission state is the set of token rotation timer and frame transmission time values for each active station on the ring.

By examining two scenarios in which one and two stations are actively transmitting frames the authors derive, by performing a pure deterministic analysis, the sequence of transmission states for both steady-state cycles. For each scenario, the duration of the steady-state cycle can be easily computed by summing up the duration of each state within the steady-state cycle. Similarly, the total frame transmission time in these states can be derived by summing up the frame transmission times of each state within the steady-state cycle. The total (normalized) throughput is derived by simply forming the ratio between the total frame transmission time and the steady-state cycle time.

By examining additional scenarios (not reported in [23]) the authors obtain a generalized throughput expression for an *FDDI* with an arbitrary number N of active stations.

$$\gamma_{\max} = \frac{(N \cdot \text{tot_tx_time} + N^2 \cdot \text{tx_window}) \cdot v}{N \cdot \text{tot_tx_time} + N^2 \cdot \text{tx_window} + (N^2 + 2N + 1) \cdot r_I}$$

where $\text{tx_window} = T_Opr - r_I$, $\text{tot_tx_time} = \text{CEILING}(\text{tx_window} / F) \cdot F$,

v is the transmission speed, and $r_{-I} = \sum_{i=1}^N r_i$ is the total ring-latency time.

Taking the limit of γ_{\max} as the number of active stations (N) goes to infinity, a bound for the maximum total throughput is obtained

$$\lim_{N \rightarrow \infty} \gamma_{\max} = \frac{tx_window \cdot v}{tx_window + r_{-I}} = \frac{T_Opr - r_{-I}}{T_Opr} \cdot v$$

Equation for γ_{\max} is then generalized to produce the maximum total throughput for FDDI when multiple asynchronous priority levels are being used.

In [23] the lowest priority message that will ever be transmitted it is shown to belong to the class "low", which is the lowest priority level satisfying

$$E = [N - n(low) + 1] \cdot T_Pri(low) - \left[\sum_{i=low+1}^m n(i) \cdot T_Pri(i) \right] > 0$$

where m is the number of asynchronous priority traffic in use in a given scenario ($1 \leq m \leq 8$); $n(i)$ is the number of stations that have priority level i messages ready to transmit during each cycle; and $N = \sum_{i=low}^m n(i)$ is the total number of stations transmitting.

Under the assumption that stations immediately stop transmitting frames when THT has reached $T_Pri(i)$, Dykeman and Bux show that the throughput $t(i)$ of priority i messages is given by

$$t(i) = \frac{r(i)}{(N+1) \cdot r_{-I} + \sum_{j=low}^m r(j)} \cdot v, \quad \text{where}$$

$$r(i) = \begin{cases} n(i)E & \text{if } i = \text{low} \\ n(i) \left[N(T_Pri(i) - T_Pri(low)) + (N - n(low))T_Pri(low) \right. \\ \quad \left. - \sum_{i=low+1}^m n(i)T_Pri(i) + T_Pri(i) - r_{-I} \right] & \text{if } i = \text{low} + 1, \dots, m \end{cases}$$

The overall throughput is given by the sum of the throughputs for each priority level.

Model 2: Single Buffer Model. This model was proposed and analyzed by Takagi in [60]. This model assumes a single-message buffer for each station which can accommodate two classes of messages (frames): the priority message (synchronous frame) and the ordinary message (asynchronous frame). The throughput and the mean waiting times of priority and ordinary messages, the mean token rotation time, and the buffer utilization are computed numerically.

The load is equally shared among the network stations (i.e., symmetric system) and when the buffer is empty a message in the generic station i is generated with an interval which is exponentially distributed with mean $1/\lambda$ for priority messages or $1/\lambda'$ for ordinary messages. Those messages that arrive when the buffer is occupied are lost. The transmission time of both an ordinary and a priority message is assumed to be the same constant b . The TTRT is defined as $R + M \cdot b$ where R is the latency (ring plus stations) M is an integer $0 \leq M \leq N$ where N is the number of stations in the FDDI network.

An exact solution of this model is obtained by applying an embedded Markov chain technique. Before introducing the state of the system, further notation is required.

Stations are indexed from 1 to N ; $\tau_i^{(k)}$ denotes the time at which the token arrives at station i in the k^{th} cycle and a cycle is assumed to start when the token arrives at station 1. The system is observed at the token arrival instants at each network station and the state vector at time $\tau_i^{(k)}$ is $(u_1^{(k)}, \dots, u_{i-1}^{(k)}, u_i^{(k)}, u_{i+1}^{(k-1)}, \dots, u_N^{(k-1)})$ where $u_j^{(k)}$ is the state of station j at time $\tau_j^{(k)}$ and

$$u_j^{(k)} = \begin{cases} 0 & \text{buffer empty} \\ 1 & \text{ordinary message that will not be transmitted} \\ 2 & \text{ordinary message that will be transmitted} \\ 3 & \text{priority message} \end{cases}$$

The cycle time for station i at time $\tau_i^{(k)}$ is $R + Q_i^{(k)} \cdot b$ where $Q_i^{(k)}$ is a function of $u_1^{(k)}, \dots, u_{i-1}^{(k)}, u_i^{(k)}, u_{i+1}^{(k-1)}, \dots,$ and $u_N^{(k-1)}$ given by

$$Q_i^{(k)} = \sum_{j=i+1}^N I(u_j^{(k-1)} \in [2,3]) + \sum_{j=1}^{i-1} I(u_j^{(k)} \in [2,3])$$

where $I(\cdot)$ is the indicator function. We are interested in computing the steady state probabilities of our system

$$P(u_1, \dots, u_{i-1}, u_i, u_{i+1}, \dots, u_N) = \lim_{k \rightarrow \infty} P_i(u_1^{(k)}, \dots, u_{i-1}^{(k)}, u_i^{(k)}, u_{i+1}^{(k-1)}, \dots, u_N^{(k-1)})$$

where $P_i(u_1^{(k)}, \dots, u_{i-1}^{(k)}, u_i^{(k)}, u_{i+1}^{(k-1)}, \dots, u_N^{(k-1)})$ is the probability that the state of the system at time $\tau_i^{(k)}$ is $(u_1^{(k)}, \dots, u_{i-1}^{(k)}, u_i^{(k)}, u_{i+1}^{(k-1)}, \dots, u_N^{(k-1)})$. In steady state we also have that

$$P_{i-1}(u_1, \dots, u_{i-1}, u_i, u_{i+1}, \dots, u_N) = P_i(u_N, u_1, \dots, u_{i-1}, u_i, u_{i+1}, \dots, u_{N-1}).$$

Since all stations are statistically identical we can focus on station 1 omitting the subscript 1. Now $Q = \sum_{j=2}^N I(u_j \in [2,3])$ and a set of 4^N linear equation among the 4^N

unknowns (i.e., $P(u_1, \dots, u_N)$) are presented in the paper.

From these steady state probabilities several performance measures are obtained. The distribution of the cycle length is given by $P(C = R + nb) = \sum_{U(2,3)} P(u_1, \dots, u_N)$ where

$$U(2,3) = \left\{ (u_1, \dots, u_N) \mid \sum_{i=1}^N I(u_i \in [2,3]) = n \right\}$$

The distributions of the number of ordinary and priority messages transmitted in a cycle are $P(Q_{\text{ordinary}} = n) = \sum_{U(2)} P(u_1, \dots, u_N)$ and

$$P(Q_{\text{priority}} = n) = \sum_{U(3)} P(u_1, \dots, u_N) \text{ where } U(j) = \left\{ (u_1, \dots, u_N) \mid \sum_{i=1}^N I(u_i \in [j]) = n \right\}.$$

The throughput of ordinary and priority messages are $\gamma_{\text{ordinary}} = E[Q_{\text{ordinary}}] / E[C]$ and $\gamma_{\text{priority}} = E[Q_{\text{priority}}] / E[C]$ respectively.

The computation of the mean waiting time is more complex. The paper first focuses on the priority messages, it computes this waiting time computing the Laplace-Stieltjes transform of the waiting time for priority messages $W_{\text{priority}}(s)$ from which the average waiting time is derived with routine calculations

$$E[W_{priority}] = \frac{\sum_{u_2=0}^3 \dots \sum_{u_N=0}^3 \left[R + bQ - \frac{1 - e^{-(\lambda+\lambda')(R+bQ)}}{\lambda + \lambda'} \right] \cdot \sum_{u_1 \in \{0,2,3\}} P(u_N, u_1, \dots, u_{N-1})}{\sum_{u_2=0}^3 \dots \sum_{u_N=0}^3 \left[1 - e^{-(\lambda+\lambda')(R+bQ)} \right] \cdot \sum_{u_1 \in \{0,2,3\}} P(u_N, u_1, \dots, u_{N-1})}$$

The mean waiting time for ordinary messages is obtained by applying Little's formula $\bar{N}_{ordinary} = \gamma_{ordinary} (E[W_{ordinary}] + b)$ where $\bar{N}_{ordinary}$ is the mean number of ordinary messages in all buffers at an arbitrary time. $\bar{N}_{ordinary}$ is computed by exploiting the following relationship $U = (\bar{N}_{ordinary} + \bar{N}_{priority})/N = 1 - \gamma_{priority}/\lambda' \cdot N$; and in this formula the only unknown is $\bar{N}_{priority}$ which is easily derived from $\gamma_{priority}$ and $E[W_{priority}]$.

The symmetric traffic assumption in the model proposed by Takagi was removed by Nakumara *et al.* [48] who considered the corresponding asymmetric system.

Model 3: G-limited with Time-Limit Variation Polling Model. This model, proposed by Tangeman and Sauer [62], is used to analyze the performance figures of an FDDI network with N stations which only transmit asynchronous traffic subdivided into multiple-priority levels. Each station transmits traffic of only one priority level, and the priority level of a station i is characterized by its threshold $T_{pri}(i)$. The number of packets which can be buffered in a station i is limited by the value m_i . Packets arrive in the buffer of station i according to a Poisson distribution with rate λ_i . The service time of a packet buffered in a station i ($T_H(i)$) and the switchover time from station i to station $i+1$ ($T_U(i)$) are random variables sampled from general distributions. To represent the timed token protocol behavior the service discipline is of *G-limited with time-limit variation* type (see also Section 3.4). In fact, when a station i captures the token it can use the transmission media for a maximum time of $\max\{0, T_{pri}(i) - T_{TRT}(i)\}$, where $T_{TRT}(i)$ measures the length of the previous cycle (i.e., it is equal to the TRT of the station i when it captures the token). With this discipline, when station i holds the token it can only transmit the packets that were in its buffer at the token arrival (i.e., gated service). The transmission period of a station ends when there are no more packets to transmit or the maximum transmission time for this cycle is reached, whichever occurs first. In the latter case the transmission of the last packet may be completed. In [62] this service discipline is called *cycle-time-dependent timer-controlled gated service discipline*.

The analysis of this model is divided into two main parts. First, for each station i , the statistics of the cycle time observed by the station ($T_C(i)$) are derived, i.e., the time between two consecutive arrivals of the token at station i . Then by analyzing each station in isolation, the packet loss probability and the average waiting time are derived. The analysis of the cycle time is performed by studying, for each station i the steady state statistics of the sequence $\{A_i^{(n)}, T_{TRT}^{(n)}(i)\}$ embedded at the time instants at which the server arrives at the station; where $A_i^{(n)}$ is the number of packets waiting in station i at the n^{th} token arrival, and $T_{TRT}^{(n)}(i)$ is the value of the TRT counter at the same time instant. To simplify the analysis, it is assumed that a station always restarts the TRT timer and therefore $T_{TRT}^{(n)}(i)$ has the same distribution as $T_C^{(n)}(i)$ which measures the

time since the previous token arrival (i.e., the length of the interval between the $(n-1)^{th}$ and n^{th} token arrival instants) at station i .

Due to the complexity of the problem, the authors could not find closed formulas for the steady-state distribution of $\{A_i^{(n)}, T_{TRT}^{(n)}(i)\}$, but by applying an iterative scheme they are able to approximate this distribution. The iterative scheme, applied in this paper, is an extension of the method used by Tran-Gia and Raith to study a finite capacity polling system with limited service discipline [63].

The iterative approach starts by assigning initial values for $\{A_i^{(0)}, T_{TRT}^{(0)}(i)\}$. Through a set of equations the statistics of $\{A_i^{(1)}, T_{TRT}^{(1)}(i)\}$ are derived. This recursive scheme is applied until the system is stable $\sum_{i=0}^N E[A_i^{(n)}] - E[A_i^{(n-1)}] / E[A_i^{(n-1)}] < \varepsilon$, where ε determines the accuracy of the results. When the iteration ends the statistics of $\{A_i^{(n)}, T_{TRT}^{(n)}(i)\}$ approximate the steady state distributions of $\{A_i, T_{TRT}(i)\}$, where $A_i = \lim_{n \rightarrow \infty} A_i^{(n)}$, $T_{TRT}(i) = \lim_{n \rightarrow \infty} T_{TRT}^{(n)}(i)$. Specifically, each iterative step, starting from the statistics of $\{A_i^{(n)}, T_{TRT}^{(n)}(i)\}$, firstly provides the statistics related to $T_{TRT}^{(n+1)}(i)$ and secondly those of $A_i^{(n+1)}$.

The authors also propose to improve the above analysis by taking into consideration the dependencies between consecutive stations. To this end, they propose, in computing the variance of the cycle length, to consider the covariance of each couple $(T_E^{(n)}(i), T_E^{(n)}(i+1))$ of adjacent stations.

In the second part of the analysis each node is analyzed in isolation. Starting from the approximate distribution of the number of packets in the queue of a tagged node at the token arrival instant, first the steady-state distribution of the queue length at arbitrary times is derived. Then from the distribution at arbitrary times the packet loss rate and the average waiting time are obtained.

Model 4: Exhaustive Time-Limited Polling System. This approximate model was proposed by Chang and Sandhu in [7] for the delay analysis of a polling system with a time-limited discipline. The MAC protocol is modelled as a cyclic-service system with an exhaustive time-limited discipline. The time limited policies limit the duration of transmission whenever a station possesses the token; this makes this model suitable for the FDDI and Token Bus delay analysis. The delay analysis is carried out assuming that all the traffic is at the highest priority level i.e., in FDDI only a synchronous service is used. The algorithm can be applied both to FDDI and to the Token Bus with small differences because in FDDI the time a station can hold the server can never exceed the station threshold (H_i) , while in the Token Bus when the H_i threshold is reached no further services are permitted but the server will only leave station i if the ongoing service is completed.

The idea behind the approximate model is to use results obtained in [8] for the polling systems with exhaustive-limited discipline to the exhaustive time-limited discipline. Obviously these two disciplines only coincide when the service time distribution is deterministic.

To derive the mean delay a pseudo-conservation law is used and approximations for the unknowns in the pseudo conservation law formula are proposed.

3.4 Station-in-Isolation Models

This section outlines four special cases of the station in isolation model. All the

models falling in this category are variants of M/G/1 queueing models with vacation times. From the modeling standpoint, the token is a server that provides a service during its visit to the tagged station and is on vacation (i.e., within C_NET) when it is away from it. Thus, with respect to the tagged node, C_NET can also be viewed as a generator of vacation periods. In some of the station in isolation models which will be described in this section, the queue has a finite capacity and this motivates the use of M/G/1/N queue with vacation, where the queue capacity equals N , to represent the behavior of the tagged station. Several service disciplines ([24], [25], [61]) have been proposed and analyzed in the literature. The service disciplines implemented by the models described in this section belong to the following classes

Limited. A fixed limit is placed either on the maximum number of frames that can be served, or on the amount of time the server serves the queue, before going on vacation. The term time-limited is normally used to distinguish the latter discipline from the former. When the "time" prefix is missing, the limit refers to the number of frames. In both cases, the server serves until either (i) the (time-)limit is reached, or (ii) all the frames eligible for service have been served. Both service disciplines can operate in an *exhaustive-limited (E-limited)* or *gated-limited (G-limited)* manner. In the former case, all the frames in the queue are eligible for service, whereas in the latter case, only the frames that were in the queue upon the server's arrival are eligible;

Limited with limit variation. This policy extends the previous one since a dynamic limit (randomly chosen upon server arrival) is placed either on the maximum number of frames that can be served, or on the amount of time the server serves the queue, before going on vacation.

The M/G/1 and M/G/1/N systems with vacation times and *limited* or *limited with limit variation* service discipline are analyzed with the embedded Markov chain technique. The state of the Markov chain includes a random variable to denote the number of frames in the system and other random variables whose semantics vary from model to model. If the tagged station is modelled with a G-limited (or G-limited with limit variation) service discipline the embedding points correspond to the vacation termination instant (i.e., the arrival instants of the server at the tagged node); otherwise, if the service discipline is E-limited (or E-limited with limit variation), apart from the vacation termination instant the service completion instants are also included among the embedding points.

Three techniques have been used to solve these models: z-transform, matrix-analytic and iteration.

z-Transform. According to this technique, formulas for the probability generating functions (PGFs) of the embedded queue size distribution are obtained. These formulas have the following general structure

$$Q(z) = \frac{N(z)}{z^{c \cdot K} - B(z)}, \quad c \in \mathbb{N}^+,$$

where $N(z)$ and $z^{c \cdot K} - B(z)$ are analytic functions in the unit disk. The unknowns in $Q(z)$ are a set of $L = c \cdot K$ boundary probabilities in $N(z)$ which express the probability that the tagged station transmits less than K packets when it receives the token.

To compute the boundary probabilities a standard technique is generally used. In fact, if the system satisfies the stability condition, from Rouché's theorem it is possible to prove that the denominator of $Q(z)$ has L zeros in $|z| \leq 1 + \varepsilon$. Of these L zeros, it is easy to verify that one of them is unity. Let us denote the remaining $L - 1$ zeros, none of which is equal to unity, by z_i , $i = 1, 2, \dots, L - 1$. Lagrange's theorem can be used to find an infinite series representation for each of these zeros. Since $Q(z)$ is analytic in

$|z| \leq 1$, it is possible to conclude that these $L-1$ denominator zeros in $|z| \leq 1$ are also zeros of $N(z)$. By evaluating this numerator at the $L-1$ zeros, it is possible to write $L-1$ linear equations in terms of the L unknown boundary probabilities. The normalization condition, $Q(1) = 1$, is used to find another equation in the L unknowns and this solves the problem of computing the L unknown boundary probabilities.

Matrix-analytic. Chiarawongse *et al.* [9] use the matrix-analytic technique to analyze an M/G/1 queueing system with vacations and timer-controlled service. With this technique, the boundary probabilities are computed by matrix-geometric analysis which involves matrix inversions and successive iterations. The steady state distribution of the embedded Markov chain is then generated recursively from these boundary probabilities to conduct queue size and delay analysis.

Iteration. Finally in Rubin *et al.* [53] an iteration procedure is used to evaluate the steady state probability distribution of the embedded Markov chain. The authors claim that this approach has the following advantages over the z-transform method and matrix-analytic technique, respectively:

- *without tackling the multidimensional boundary value problem in the transform-domain, the potential high-order root-solving problem incurred by the numerous boundary states is avoided;*
- *by iteratively calculating the boundary probabilities based on balance equations, the numerical difficulties involved in inverting huge matrices are eliminated.*

Obviously all the models presented in this section are approximate. The first two models (*Model 5* [53] and *Model 6* [38]) perform a delay analysis of the asynchronous traffic. The performance figures of an FDDI tagged station are studied by solving, either, with an iterative approach, an M/G/1/N queueing system with E-limited and time-limit variation service discipline [53], or, with z-transform, an M/G/1 queueing system with E-limited and limit variation service discipline [38]. *Model 7* ([13], [18]) and *Model 8* [27] are proposed and solved for analyzing the quality of service achieved by the synchronous frames. Since the synchronous service of FDDI is designed to deliver time critical messages, the emphasis in both works is on the development of models which provide estimates of the probability distribution of the number of packets queued in the tagged station, and of the delay experienced by these packets in the buffer. In fact, the analysis of the relationship between the probability distribution figures and the negotiated network parameter values (T_{opr} value, quota of synchronous traffic for the tagged node, etc.) indicate for a given network configuration the likelihood that a given deadline and packet loss rate will be met. In both works, the system is analyzed under the heavy load assumption as it is the most important case for network designers and managers (for the tuning of the network configuration).

Model 5: M/G/1/N with Vacation and E-limited with Time-limit Variation Service Discipline. In [53] the tagged station is modelled as an M/G/1/N single server (token) queuing system with server vacations. The service policy is such that the server can provide contiguous service to the tagged station for a (dynamic) limited period of time (i.e., E-limited with time-limit variation service discipline).

The tagged station queue, which has a finite packet buffer capacity of size N , is fed with packets arriving according to the Poisson distribution with rate λ . The packets belong to a single asynchronous priority class. The timing threshold associated with it

is denoted by T_{pri} . A token holding time limit $\text{THTL} = [T_{pri} - \text{TRT}]^+$ is set, upon the arrival of the token at the station. The station is allowed to initiate transmission of its queued packets (when it has seized the token) at any time, provided at this time $\text{THT} < \text{THTL}$.

Each packet consists of a random number of fixed-size segments, and the transmitting time of a segment is equal to a slot duration, Δ . In addition, the length of the vacation period is assumed to be a multiple of Δ . Therefore the dynamics of the system can be described by means of a discrete time queuing model with time units equal to one slot.

If B_n (slots) denotes the n^{th} packet transmission time, then $\{B_n, n \geq 1\}$ forms a sequence of independently identically distributed random variables. The probability distribution of the packet transmission time is denoted by $b(i) = P\{B_n = i\}$, $i = 1, \dots, b_{\max}$; $b_{\max} < \infty$ (note that for FDDI, the maximum packet length is set to 4500 bytes). The mean packet transmission time is denoted by μ . The n^{th} vacation time relative to the station is denoted as V_n (slots). To simplify the analysis, in [53] it is assumed that the successive vacation times $\{V_n, n \geq 1\}$ form a sequence of independently identically distributed random variables. The probability distribution of the vacation time is thus denoted by $v(i) = P\{V_n = i\}$, $i = 1, \dots, V_{\max}$; $V_{\max} < \infty$ (note that for an FDDI network $V_n \leq 2 \times \text{TTRT}$). The mean vacation time is denoted by \bar{V} . Furthermore, let G_n (slots) represent the n^{th} token dwell time at the station, measured from the instant of the n^{th} token arrival at the station to the instant of the n^{th} token departure from this station. The maximum value of G_n , denoted G_{\max} , is equal to $T_{pri} + b_{\max} - 1$; note that this dwell time may be longer than T_{pri} due to the occurrence of transmission overruns. The mean token dwell time is represented by \bar{G} .

The FDDI network is analyzed by means of an embedded Markov chain, defined at the token arrival and packet departure instants. Those (ordered) time instants are represented as $\{\tau_0, \tau_1, \tau_2, \dots\}$. The state of the station at an embedded point $\tau \in \{\tau_0, \tau_1, \tau_2, \dots\}$, is described by a triplet (S_τ, N_τ, T_τ) , where S_τ is the time since the most recent token arrived, N_τ is the number of packets in the queue (queue size; $N_\tau = N_\tau^-$ if τ is a packet departure point), and T_τ is the token holding time limit.

Note that T_τ is computed at the instant that the token arrives at the station and does not change until the next token arrival instant.

Based on the model assumptions, the process $\{(S_\tau, N_\tau, T_\tau), i \geq 0\}$ is a discrete-time Markov chain over the following finite state space A

$$A = \left\{ (s, n, t) \mid \begin{aligned} & (t = 0, 1, \dots, T_{pri}; s = 0; n = 0, 1, \dots, N) \\ & \cup (t = 1, \dots, T_{pri}; s = 1, \dots, t + b_{\max} - 1; n = 0, 1, \dots, N - 1) \end{aligned} \right\}$$

The Markov chain $\{(S_\tau, N_\tau, T_\tau), i \geq 0\}$ is irreducible and aperiodic. Therefore a unique limiting (steady-state) distribution $p_{s,n,t} = \lim_{i \rightarrow \infty} P\{S_\tau = s, N_\tau = n, T_\tau = t\}$, $(s, n, t) \in A$ exists. From $p_{s,n,t}$ the station's normalized throughput can easily be obtained through the following relation $\rho = \bar{G}/\bar{G} + \bar{V} = (1 - a)\mu/(1 - a)\mu + a\bar{V}$, where

a denotes the probability that an embedding point is a token arrival point which can be expressed as follows $a = \sum_{n=0}^N \sum_{t=0}^{T_{prt}} p_{0,n,t}$.

In [53], an iteration procedure is used to evaluate $p_{s,n,t}$. Once $p_{s,n,t}$ have been evaluated, by using supplementary variables and sample biasing techniques, the queue size distribution at an arbitrary instant of time u , the blocking probability and the mean packet delay are derived.

By using the above techniques the steady-state queue size distribution at an arbitrary instant of time, $P(m) = \lim_{u \rightarrow \infty} P\{X_u = m\}$, are derived. Hence, the blocking probability is given by $P_b = P(N) = 1 - \rho/\lambda\mu$.

Finally, the mean packet delay (i.e., the interval from the instant at which a packet arrives at the tagged station to the instant that this packet departs from the station), denoted as \bar{D} , can be obtained using Little's formula $\bar{D} = \sum_{m=1}^N mP(m) / \lambda(1 - P_b)$.

An empirical procedure is employed to construct an approximation of the vacation time distribution at each station.

Model 6: M/G/1 with Vacation and E-limited with Limit Variation Service Discipline. In [38] the tagged station, which is fed with only one asynchronous traffic class, is modelled as an M/G/1 single server queueing system with server vacations. The service discipline is such that the server provides service until either the system is emptied or a randomly chosen limit of l frames has been served (*E-limited with limit variation or ELV service discipline*). The server then goes on a vacation before returning to service the queue again.

In the analysis performed in [38], the limit l is a bounded random variable for which a mass function, p_l , $l \in \{0, 1, \dots, L\}$, is assumed. The limit l for a service interval is determined at the preceding vacation termination instant. Due to the way FDDI works, the value of l will differ from one token arrival to the next, depending upon the state of the system (i.e., Late_Ct and the TRT) at the time the token is seized by the tagged station. Consequently, a major difficulty in the analysis of FDDI is the fact that this limit depends upon the length of time that the token spends at each of the stations (including the current one) during the previous token cycle. To permit analysis, the following simplifying assumption is made; i.e., the sequence of limits chosen at these vacation termination instants are independent, identically distributed random variables.

It is assumed that the frame arrival process is Poisson with arrival rate λ and that at each station, the incoming frames wait in queues of infinite length. The service times (i.e., the frame transmission times) are assumed to be independent of any process in the system. The probability density function (*pdf*) of the service time and the corresponding Laplace-Stieltjes transform (*LST*) are denoted by $\tilde{b}(t)$ and $B^*(s)$, respectively. The first and second moments of the service time are denoted by b and $b^{(2)}$, respectively. In the analysis, the vacation time is only allowed to be correlated with the limit l that was used for the preceding service interval. The conditional pdf of the vacation time, given that the limit l was used for the preceding service interval, is denoted by $\tilde{v}_l(t)$ with LST $V_l^*(s)$ and first and second moments of v_l and $v_l^{(2)}$, respectively. Similarly, the marginal pdf of the vacation time is denoted by $\tilde{v}(t)$ with LST $V^*(s)$ and the first and second moments of v and $v^{(2)}$, respectively. Estimates of the function $\tilde{v}_l(t)$ and the mass function p_l that are used in the theory to obtain the

mean waiting times are determined from the simulation.

The M/G/1 vacation model described above is analyzed by using a similar approach to the one used by Lee for an E-limited (i.e., with a fixed limit) service [41] (also see Takagi [61]). Specifically, it is defined an embedded Markov chain at the vacation termination and service completion instants. The state of the system at an embedded point $t \in \{t_0, t_1, \dots\}$, is described by a triplet, $\{\tilde{k}_t, \tilde{n}_t, \tilde{l}_t\}$. $k_t = 0$ indicates that an embedded point t is a vacation termination instant while $k_t = k$, ($k = 1, 2, \dots, L$) indicates that it is a service completion instant of the k th frame in the service interval. The random variable \tilde{n}_t , is the number of frames in the system at time instant t . Finally, \tilde{l}_t , is the limit on the number of frames that can be served during the (possibly zero length) visit of the server as determined by a random choice of limit at the vacation termination instant. Thus, the limit \tilde{l}_t , associated with a vacation termination instant t is the limit chosen at this point for the following service interval. The Markov chain that derives from these definitions is irreducible and aperiodic. By assuming that the system is stable, closed formulas for the PGFs of the number of frames in the system at the above embedding points are derived. The following steady-state joint probabilities are first defined

$$f_{n,l} = \lim_{i \rightarrow \infty} P\{\tilde{k}_i = 0, \tilde{n}_i = n, \tilde{l}_i = l\}, \quad n = 0, 1, \dots; \quad l = 0, 1, \dots, L$$

$$\pi_{n,l}^{[k]} = \lim_{i \rightarrow \infty} P\{\tilde{k}_i = k, \tilde{n}_i = n, \tilde{l}_i = l\}, \quad n = 0, 1, \dots; \quad l = 0, 1, \dots, L; \quad k = 1, 2, \dots, l$$

The steady-state probability that an embedded point is a vacation termination instant and there are n frames in the system is denoted by f_n , $n = 0, 1, \dots$. Starting from f_n and $\pi_{n,l}^{[k]}$ the following $F(z) = \sum_{n=0}^{\infty} f_n z^n$ and $Q(z) = \sum_{n=0}^{\infty} \sum_{l=1}^L \sum_{k=1}^l \pi_{n,l}^{[k]} z^n$. PGFs are defined

After extensive algebraic manipulation the following expressions for $F(z)$ and $Q(z)$ are derived.

$$F(z) = \frac{\sum_{l=1}^L p_l V_l^*(\lambda - \lambda z) \left[\left\{ 1 - \left[\frac{B^*(\lambda - \lambda z)}{z} \right]^l \right\} f_0 + R_l(1) - R_l \left(\frac{B^*(\lambda - \lambda z)}{z} \right) \right] z^L}{z^L - \sum_{l=0}^L p_l V_l^*(\lambda - \lambda z) [B^*(\lambda - \lambda z)]^l z^{L-1}}$$

$$Q(z) = \frac{B^*(\lambda - \lambda z) \sum_{l=1}^L p_l \left[\{F(z) - f_0\} \left\{ 1 - \left[\frac{B^*(\lambda - \lambda z)}{z} \right]^l \right\} - R_l(1) + R_l \left(\frac{B^*(\lambda - \lambda z)}{z} \right) \right]}{z - B^*(\lambda - \lambda z)}$$

$$\text{where } R_l(z) = \begin{cases} 0, & l = 1, \\ \sum_{k=1}^{l-1} \pi_0^{[k]} z^{l-k}, & l = 2, 3, \dots, L. \end{cases}$$

What is noteworthy in $F(z)$ and $Q(z)$ is the presence of L unknown boundary probabilities f_0 , and $\pi_0^{[1]}, \pi_0^{[2]}, \dots, \pi_0^{[L-1]}$ which can be found by using Rouché's theorem and the normalization condition. From the $F(z)$ and $Q(z)$ expressions the derivation of the mean queueing time in terms of the L boundary probabilities is straightforward. The mean queue length at the service completion instants is given by the normalized z-transform derivative, $Q'(1)/Q(1)$.

Since the stochastic process of the queue length only makes discontinuous changes of

unit size, the queue length distribution immediately before the arrival instants is the same as the distribution immediately after the service completion instants ([36],[61]). Furthermore, since Poisson arrivals see time averages (PASTA) [68], the steady-state mean queue length (at arbitrary times) is equal to the steady-state mean queue length at the service completion instants. Using this argument and Little's law, the mean waiting (i.e., queueing) time can be found $W = Q'(1)/\lambda Q(1) - b$. After extensive algebraic manipulation it can be shown that

$$W = \frac{v^{(2)}}{2v} + \frac{\lambda v^{(2)} + 2v + \bar{l}\lambda b^{(2)}}{2[\bar{l}(1-\rho) - \lambda v]} - \frac{\bar{l}(1-\rho)(1-\rho + \lambda v)}{\lambda v[\bar{l}(1-\rho) - \lambda v]} \cdot \left[\sum_{i=1}^L l p_i v_i [F(1) - f_0] - \sum_{i=2}^L p_i v_i R_i^*(1) \right] \\ + \frac{(1-\rho)(1-\rho + \lambda v)}{2\lambda[\bar{l}(1-\rho) - \lambda v]} \cdot \left[\sum_{l=2}^L l(l-1) p_l [F(1) - f_0] - \sum_{l=2}^L p_l R_l^*(1) \right]$$

In a parallel paper, Leung [42] analyzed an asymmetric cyclic-service system, with infinite capacity queues, that uses the same service discipline. In [42], a numerical technique based on discrete Fourier transforms is used to obtain the waiting time distributions for the queue.

LaMaire extended this work in [38] to the case of an M/G/1/N queueing system [39] with the same service discipline (E-limited with limit variation). The queue length distribution and the Laplace-Stieltjes transforms of the waiting time, busy period and cycle time distributions are derived in [39]. In addition, an expression for the mean waiting time is developed.

Model 7: Worst Case Model for Synchronous Traffic. In ([13], [18]) an FDDI network with synchronous and asynchronous traffic is analyzed. The authors evaluate whether the quality of service guaranteed by FDDI is adequate to support real time applications when the level of network congestion is very high. The analysis is performed by using a model which is based on the assumption that the interactions among stations generate a "worst" case for the performance indices of a tagged station. The worst case proposed and analysed in [18] assumes that the length of the cycle of the FDDI network, as seen from the tagged station, has its maximum length. Furthermore, to simplify the analysis the tagged station is assumed to transmit synchronous frames of fixed length, and therefore the synchronous quota (maximum number of frames) for the tagged station corresponds to an integer number of synchronous frames, which will be denoted throughout by M .

The resulting stochastic model is an extension of the single server queueing system with vacation and *E-limited service discipline* ([41], [60]) with parameter M . The distribution of the service time is deterministic and is equal to a frame transmission time. The input traffic to the station is generated by the superposition of discrete time Markov processes. Furthermore, to model the behavior of the Timed Token protocol, it is assumed that the cycle length can be either $TTRT$ or $2 \times TTRT$ in order to model the *normal* network behavior (i.e., Late_Ct=0) and the *delayed* network behavior (i.e., Late_Ct=1), respectively. When the Late_Ct=0, with probability P_L the next cycle has a length $TTRT$ and Late_Ct remains equal to zero, while with probability $(1 - P_L)$ the next cycle has a length equal to $2 \times TTRT$ (to model the maximum token rotation time delay) and Late_Ct is set to one. Until Late_Ct=1, cycle lengths are equal to $TTRT$. Late_Ct returns to zero when the queue of the tagged station becomes empty. While in FDDI the memory of a late token is lost when the average token rotation time is again below T_{Opr} , this model loses the memory of a late token whenever its effect on the congestion of the queue of the tagged station is lost (i.e., the queue becomes empty). Therefore, the average length of the cycle is greater than $TTRT$. As the tagged station has a maximum fixed throughput for each cycle this model represents a worst case since in a real FDDI subnetwork the average cycle length is less than or equal to $TTRT$.

The distance between this worst case model and a real system is evaluated in [10].

In [18] two solution methods for this model are proposed. The first one, which was developed to analyze a tagged station with a finite buffer, is based on the periodic Markov chain theory and, due to its computational complexity, it is applicable for buffer sizes less than a given threshold (approximately 250 cells). The second one, which can be applied when the tagged station buffer is infinite is based on an aperiodic positive recurrent embedded Markov chain for which a closed formula for the PGFs of the number of frames in the buffer at the embedding points are derived. There are two groups of embedding points: one related to the normal behavior (Late_Ct=0) and the other related to the delayed behavior (Late_Ct=1).

The state of the system for any cycle i at the embedding points is described by a couple $\{t, N_t, \xi\}_i$ where $\xi = 0$ represents normal states, while $\xi = 1$ identifies delayed states. $t=0$ indicates the vacation termination instant, while $t = m$, $m = 1, 2, \dots, M$ is the embedding point just after the m^{th} transmission. The random variable N_t is the number of frames in the station at time t . The resulting Markov chain is irreducible and aperiodic only if $0 < P_L < 1$.

By assuming that the system is stable, the following steady-state joint probabilities can be defined $\pi_k^{(m)} = \lim_{i \rightarrow \infty} P\{t = m, N_t = k, \xi = 0\}_i$, $m = 1, 2, \dots, M$ and $s_k^{(m)} = \lim_{i \rightarrow \infty} P\{t = m, N_t = k, \xi = 1\}_i$, $m = 0, 1, \dots, M$ where $\pi_k^{(m)}$ ($s_k^{(m)}$) is the steady state probability that the system is in the normal (delayed) state, immediately after the m^{th} packet transmission and there are k packets in the buffer.

By using stochastic arguments $\pi_k^{(1)}$ and $s_k^{(0)}$ are derived. Starting from them, the steady state probabilities, which there are k frames at an embedded point corresponding to the m^{th} service completion instant in the normal state and in the delay state, are computed by applying the following recursive relations

$$\pi_k^{(m)} = \sum_{j=1}^{k+1} \pi_j^{(m-1)} \cdot a_{k-j+1}^{(m)} + I_{\{k=0\}} \cdot s_1^{(m-1)} \cdot a_0^{(m)}, \quad m = 2, 3, \dots, M, \quad k > 0$$

$$s_k^{(m)} = \sum_{j=1}^{k+1} s_j^{(m-1)} \cdot a_{k-j+1}^{(m)}, \quad m = 2, 3, \dots, M, \quad k > 0$$

By defining $\Pi_m(z) = \sum_{k=0}^{\infty} \pi_k^{(m)} z^k$ and $S_m(z) = \sum_{k=0}^{\infty} s_k^{(m)} z^k$ it is possible to show that both sets of z-transforms are functions of $2M$ unknown boundary probabilities $\{\pi_0^{(1)}, \pi_0^{(2)}, \dots, \pi_0^{(M)}, s_1^{(0)}, s_1^{(1)}, \dots, s_1^{(M-1)}\}$. They appear in the following closed formula for $S_0(z)$ which can be derived after some lengthy algebraic manipulations

$$S_0(z) = \frac{\sum_{i=1}^M \pi_0^{(i)} \cdot \psi(z) + s_1^{(0)} \cdot \phi(z) + \sum_{i=1}^{M-1} s_1^{(i)} \cdot \vartheta(z)}{[z^M - C(z)] \cdot [z^M - C(z)(1 - P_L)]},$$

where: $C(z)$ is the PGF of the number of arrivals in a cycle of length $TTRT$, $\psi(z)$, $\phi(z)$ and $\vartheta(z)$ are functions of z which contain known parameters.

Hence, $\{\pi_0^{(1)}, \pi_0^{(2)}, \dots, \pi_0^{(M)}, s_1^{(0)}, s_1^{(1)}, \dots, s_1^{(M-1)}\}$ can be found according to the general methodology described above by considering the normalization condition

$$\Pi(1) + S(1) = 1 \quad \text{where} \quad \Pi(z) = \sum_{m=1}^{\infty} \Pi_m(z), \quad S(z) = \sum_{m=0}^{\infty} S_m(z).$$

For the special case $P_L = 1$ the average and the standard deviation of the average number of frames in the tagged station are directly computed from the closed formula of $S(z)$ while the percentiles are overestimated by applying Chebyshev's inequality, and approximated by using a two-moments approximation [67]. In [19] it is shown how, in this type of model, it is possible to derive the delay statistics from the PGF.

Model 8: Model for Synchronous Traffic Under Heavy Load. The major problem in the analysis of an FDDI system is the correlation between the service time at each station (i.e., the time the station holds the token) and the total cycle time. This difficulty is overcome in [27] by observing that this correlation is almost negligible during heavy load conditions, as in these conditions the transmission of asynchronous traffic is often deferred until traffic moderates. In fact, simulation results have estimated that the TRT value has a very low coefficient of variation under heavy load conditions (e.g., $0.9 \cdot T_{opr} \leq TRT \leq 1.1 \cdot T_{opr}$). Furthermore, simulative experiments indicate that there is a negligible correlation between the time a station holds the token to transmit synchronous traffic and the total cycle time. On the basis of these considerations, in [27] the network behavior observed by a tagged station is approximated by assuming that the value of the TRT of the station is independent of the station behavior (i.e., the length and number of synchronous transmissions and the number of packets queued in the station). With these hypotheses the tagged station is modelled via an M/G/1 queuing system with vacation and G-limited service discipline with parameter M . The service time of a packet, and the vacation time are r.v. with continuous distributions $U(t)$ and $V(t)$, respectively. $U^*(s)$ and $V^*(s)$ are the LSTs of $U(t)$ and $V(t)$, respectively. As the service time is stochastic, M is only an approximation of the synchronous quota of the station. The PGF of the number of packets in the station at the token arrival instants is therefore [61]

$$Q(z) = V^*(\lambda - \lambda z) \frac{\sum_{n=0}^{M-1} \pi_n \{ [U^*(\lambda - \lambda z)]^n z^M - [U^*(\lambda - \lambda z)]^M z^n \}}{z^M - [U^*(\lambda - \lambda z)]^M V^*(\lambda - \lambda z)}$$

where π_n , $n = 0, 1, \dots, M-1$ are the unknown boundary probabilities.

After computing the boundary probabilities by using classical arguments in [27] an expression for the LST of the waiting time is derived. By inverting the LST the probability density function of the waiting time of the synchronous traffic is finally obtained.

4.0 DQDB

In this section, firstly the basic DQDB MAC protocol is described. Then by analyzing the unfairness problems, the Bandwidth Balancing mechanism, which was included in the standard to recover this problem, is introduced. Finally, performance modeling of a DQDB network is discussed. A classification of the analytical models is proposed, and for each class some relevant models are presented.

DQDB has been the subject of intensive research related both to performance modeling issues, and to enhancements and architectural variations of the protocol. In this tutorial we will only focus on the performance modeling related to the standard version of the protocol. A complete overview and bibliography of the DQDB research activities as of 1992 is reported [47].

4.1 DQDB MAC Protocol Description

The basic structure of a DQDB network is shown in Fig. 1. The network consists of

two high speed unidirectional buses carrying information in opposite directions. The network nodes² are distributed along the two buses and they can transmit information to and receive information from both buses, as shown in Fig. 1. The node in the leading edge of each bus is designated as the *head* of its corresponding bus (*HOB*). Each HOB continuously generates slots of fixed length (53 octets) which propagate along their respective buses. The first byte in a slot constitutes the *Access Control Field (ACF)*, which is utilised by the nodes in the network to co-ordinate their transmissions. Each slot accommodates one segment which is 52 octets long; 4 of these octets are designated as the *header* of the segment and the remaining 48 octets are used for information transmission.

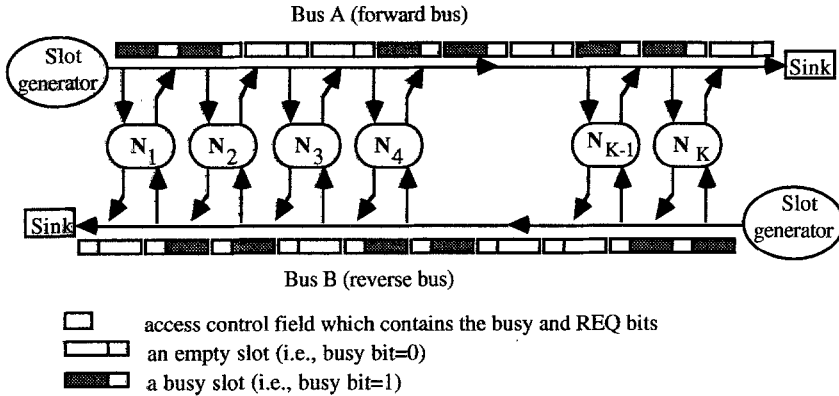


Fig. 1. DQDB Dual Bus Topology

The DQDB MAC protocol provides two modes of access control to the buses: *Queued Arbitrated (QA)* and *Pre-Arbitrated (PA)* which use QA and PA slots, respectively. The distinction between the two types of slots is made through the *SL_TYPE* bit in the ACF of a slot. PA access is reserved for *isochronous* services such as voice and video, while QA access is used typically to provide asynchronous services.

This tutorial will only deal with the QA mode of operation. The PA part of the protocol is analyzed in ([70], [71], [72]).

Without loss of generality, we name bus A the *forward bus*, and bus B the *reverse bus*. In each node, the segments, on arrival, are put in the proper *local node queue (LQ)*, as determined by the destination address (there are two local node queues, one for each bus). Below we will focus on segment transmission by using the QA slots in the forward bus, since the procedure for transmission in the reverse bus is the same.

To manage the QA mode of operation the ACF includes a *busy bit* and 3 *request (REQ) bits*. These four bits are set to "0" by the originating HOB. The busy bit indicates whether or not the corresponding slot has already been used for data transmission. The three REQ bits are provided to implement a three-level priority scheme to access the QA slots. For ease of presentation the tutorial focuses on one priority level.

The procedure for segment transmission on the forward bus utilizes the busy bits in the ACF of the slots of the forward bus and one request bit (the REQ bit below) in the ACF of the slots of the reverse bus.

Each node is either *idle*, when there is nothing to transmit, or *count_down*. When it is idle the node keeps count, via the *request counter (RQ_CTR)*, of the number of

² In this survey the words node and station will be used interchangeably.

outstanding REQs from its downstream nodes. The RQ_CTR increases by one for each REQ received in the reverse bus and decreases by one for each empty slot in the forward bus. When a node in the idle state receives a segment, it enters the $count_down$ state and starts the transmission procedure by taking the following actions: 1) the node transfers the content of the RQ_CTR to a second counter named the *count_down counter* (CD_CTR), 2) resets the RQ_CTR to zero, and 3) generates a request which is inserted into the queue of the pending requests while waiting for transmission on the reverse bus (by setting $REQ=1$ in the first slot with $REQ=0$). In the $count_down$ state the CD_CTR is decreased by one for every empty slot in the forward bus until it reaches zero. Immediately afterwards, the node transmits the segment into the first empty slot of the forward bus. In the meantime, the RQ_CTR increases by one for each new REQ received in the reverse bus from the downstream nodes. After the segment transmission, if the LQ is empty the node returns to the idle state, if not the transmission procedure (1-3) is repeated.

The QA part of the DQDB MAC protocol is an attempt to approximate a Round Robin [36] policy (with the quantum equal to the slot duration) for the sharing of the medium capacity. This is achieved when we neglect the following aspects (due to the physical implementation of the algorithm in a distributed environment):

- 1) the waiting time for setting the $REQ=1$ bit on the reverse bus,
- 2) the transfer time of the $REQ=1$ bit to the upstream nodes, and
- 3) the transfer time of the empty slot from the head node to the node which issued the $REQ=1$ under consideration.

Under these *ideal conditions* the protocol manages the network access by maintaining a global queue named Distributed Queue (DQ) which contains at most one segment for each node, and a set of LQs, one for each node. A segment which arrives in a node which has no representative in the DQ immediately enters in the DQ, otherwise it waits in the node LQ. Segments in the DQ are served on a FIFO basis (each segment gets a service quantum equal to the segment transmission time). After the transmission of one of its segments a node may insert a new segment (if any) at the end of the DQ.

Unfortunately, due to the propagation delays between the nodes, the information that a node receives is usually outdated. For this reason the DQDB MAC protocol deviates from the Round Robin schedule to a degree that depends on several parameters (e.g., the network physical size, the position of the nodes along the buses, the network load etc.) and this gives rise to the so called unfairness in DQDB.

4.2 DQDB Unfairness and the Bandwidth Balancing mechanism

DQDB provides quality of service, either in the form of access delays or bandwidth allocation, which is strongly dependent on the node position in the network ([65], [11], [5]).

The DQDB unfairness in bandwidth allocation was highlighted in [69] by analyzing the behavior of an earlier version of the DQDB MAC protocol (which however, under the configuration presented below, behaves exactly like the IEEE802.6 standard) in a simple network configuration with only two active stations and the propagation delay between the nodes equal to D slots. In [69] it is assumed that initially only the upstream node (*node{1}*) sends data and that this node has enough traffic to fill all the slots travelling on the forward bus. This is allowed by the DQDB protocol as the upstream node, which is the only active node, does not see any request in the reverse bus. Let us now assume that the downstream node (*node{2}*) becomes active at least D slot times after *node{1}* starts its transmission. When *node{2}* becomes active it immediately switches to the countdown state (with CD_CTR equal zero), issues a request on the reverse bus, and it stops to wait for an empty slot to appear on the forward bus. In this scenario, *node{1}* only leaves an empty slot in response to a *node{2}* request. Obviously, it takes D slots before the upstream node can observe a *node{2}* request.

When the upstream node reads the node{2} request, it allows a slot to remain idle. After a D slots propagation delay, the idle slot reaches the node{2} which transmits a segment, a new segment enters the distributed queue, and issues a new request on the reverse bus.

In this configuration the downstream node can only transmit one segment every $2D$ slots. With a separation of 30 Km between nodes, $2D \approx 100$, and consequently the downstream node receives about one slot out of a hundred.

DQDB unfairness in saturated conditions arises because the protocol enables a node to use every *unused slot* (i.e., an empty slot not reserved by downstream nodes) on the bus.

To overcome this problem the Bandwidth Balancing (*BWB*) mechanism [29] has been added to the DQDB standard [31].

The Bandwidth Balancing mechanism follows the basic DQDB protocol, except that a node can only take a fraction of the unused slots. Specifically, the Bandwidth Balancing mechanism limits the throughput of each node to some multiple M of the unused bus capacity (U); this limit is defined as *control rate*. Nodes with less demand than the control rate (not-rate-controlled nodes) get all the bandwidth they want. Therefore, if $\rho(i)$ indicates the bandwidth requirements of node i the throughput $r(i)$ of node i satisfies the following relationship [29]

$$r(i) = \min\left[\rho(i), M \cdot U\right] = \min\left[\rho(i), M \cdot \left(1 - \sum_m r(m)\right)\right].$$

This scheme is fair, in fact if there are N rate-controlled nodes, and S is the utilization due to not rate-controlled nodes, all rate-controlled nodes get the same bandwidth

$$r(i) = M(1 - S)/(1 + M \cdot N).$$

The BWB mechanism achieves a fair bandwidth sharing by sparing a portion of the bus capacity. In fact, while in the basic DQDB (i.e., BWB mechanism disabled) a complete utilization of the transmission media can be always achieved, when the BWB mechanism is enabled, the network utilization depends on the number of active nodes. In [29] it is shown that the bandwidth wastage due to this mechanism is therefore $(1 - S)/1 + M \cdot N$; and the worst case bandwidth wastage is $1/(1 + M)$ which only occurs with one active node.

To implement this mechanism the protocol uses a counter (*BWB_CTR*) to keep track of the number of transmitted segments. Once *BWB_CTR* reaches the *BWB_MOD* value this counter is cleared and the *RQ_CTR* is increased by one. The value of *BWB_MOD* can vary from 0 to 63. The value 0 means that the BWB mechanism is disabled. For further details see [31].

The effectiveness of this mechanism has been extensively studied via simulation ([65], [11], [5]). Simulative analysis has shown that, with the BWB mechanism enabled, DQDB always reaches (after a transient time) a steady state condition where the bandwidth is equally shared among the nodes. During the transient time the network behavior remains unpredictable. The length of this transient interval depends significantly on the *BWB_MOD* value and initial state, while it is not significantly affected by medium capacity and bus length [11]. The relationship between the length of the transient period and the above parameters is still an open issue.

4.3 DQDB Analytical Modelling

Due to the discrete time nature of the DQDB network, it is natural to expect that Markov chains are natural modeling tools for the description of the network behavior. The problem in the use of a Markov chain model is the definition of an appropriate state space. In [46] a discrete time Markov chain which exactly describes a DQDB network with N nodes, constant internode distance of d slots, and K buffers per node is

proposed and the size of its state space is investigated. The state of the Markov chain includes

- the values of the RQ_CTR and CD_CTR in each of the nodes (excluding the most downstream one, since it never receives any requests);
- the number of segments queued at each of the nodes;
- the number of requests queued per node to be transmitted on the reverse bus (excluding the most upstream, since it never sends requests);
- the value of the busy bit for each slot in transit on Bus A;
- the value of the request bit for each slot in transit on Bus B.

However, not all the possible states can be reached. In order to study the relationship between the number of possible states and the number of the valid states (i.e., the reachable states) in [46] a network configuration with only two single buffer stations is analyzed. The relationship between the number of possible states and the number of valid states for this configuration is shown in Table 4.

d	Total possible states	Number of valid states
1	216	61
2	1536	305
4	38400	5642
6	884736	92604

It is easy to observe that the state space explodes quite rapidly and analysis is possible in a few simple cases. Thus a general solution for the DQDB network, i.e., one that encompasses any number of nodes and any internode distance, seems to be highly improbable. Simplifying assumptions therefore have to be made in order to obtain analytically-tractable solutions.

To the best of our knowledge, no exact solution of a general model of DQDB has been presented in literature. Approximate solutions have been proposed for general DQDB model, while exact solutions have been proposed for DQDB networks operating under specific conditions.

Table 5
DQDB Models Taxonomy

			Models	Performance indices	
Network-Wide Models	Deterministic	Node-Spaced Models	1	Th	
	Stochastic	Node-Concentrated Models	2	E[D]	
L_NET			1 st -order	3	output process
Node In Isolation Models	Stochastic	Tagged node Models	n^{th} -order	4	output process
			Single buffer	5-6	E[D]
		Infinite buffer	7-8-9	E[D]	

Th: Throughput; E[D]: Average Delay;

In order to provide a structured overview of the DQDB analytical studies we introduce the taxonomy shown in Table 5. Depending on whether the models consider explicitly all the network nodes or whether they focus on a tagged node, we preliminary identify two main classes of DQDB models: *Network-Wide Models* and *Node-in-Isolation Models* [47].

4.4 Network-Wide Models

Models of a DQDB network which explicitly represent the interdependencies among the network stations can be subdivided into two further classes: models which assume that the network nodes are spaced along the two buses (*Node-Spaced Models*) and models in which the nodes are concentrated in the same place (*Node-Concentrated Models*).

Node-Spaced Models. Models of a DQDB network which represent few active stations spaced along the network buses are often used in literature to study the DQDB asymptotic behavior. By assuming that all network nodes, when active, always have segments to transmit, the resulting models have a deterministic behavior. These models are used to obtain expressions for throughput achieved by every node as a function of the network span and the nodes activation instants. The complexity of the interdependencies among stations make the analysis possible for only few active nodes.

As noted in the previous section, in [69] a model of this type was applied to an earlier version of the DQDB MAC protocol to highlight its unfairness. In [29] the model proposed by Wong was extended to the standard version of DQDB (with the BWB mechanism disabled)³ by taking into consideration all the possible configurations of time instants at which nodes start to transmit.

As in [69], in this model, a simple network configuration with only two active nodes is assumed (hereafter the index 1 will be used to indicate the most upstream node) and the propagation delay between the nodes equal to D slots is analyzed. D (an integer number of slots) is used to indicate the difference in the starting times of the two nodes (i.e., node{2} starting time, t_2 , minus the starting time of node{1}, t_1). When both nodes are active, node{1} leaves to node{2} an empty slot when it observes a request on the reverse bus. The rate at which node{2} can generate its requests is a function of the sum (X) of the number of requests travelling on the reverse bus, the empty slots on the forward bus, and the number of requests queued in the node{1} counters at the time instant at which both nodes are active. After this time, X becomes a constant and it causes the node{2} throughput. The value of X is determined by the following relationship $X = 1 + D - c(D)$, where

$$c(D) = \begin{cases} (t_2 - t_1) & \text{if } -D \leq (t_2 - t_1) \leq D \\ -D & \text{if } -D > (t_2 - t_1) \\ D & \text{if } D < (t_2 - t_1) \end{cases} .$$

Before continuing, the following definitions must be introduced.

- $r(1)$ and $r(2)$ are the throughputs of node{1} and node{2}, respectively;
- $Q(1)$ is the average length of the distributed queue observed by node{1} just after it inserts a segment in the distributed queue (i.e., 1+ the value of the CD_CTR of node{1});
- $Q(2)$ is the average length of the distributed queue observed at node{1} just after the insertion in the queue of a request from the downstream node;
- T is the average delay between the time node{2} issues a request and the time it receives the related empty slot.

The network behavior in steady state can be approximated by the following four equations [29]: $r(1) + r(2) = 1$, $r(1) = 1/Q(1)$, $r(2) = X/T$, and $T = 2D + Q(2)$.

Finally, by solving the system of linear equations, with the approximation

³ In [29] a model to analyze the transient behavior of the BWB mechanism in asymptotic conditions is also proposed. However the model assume that nodes follow the so called *deference scheduling* instead of the DQDB MAC protocol.

$T = 2D + Q(2) \approx 2D + Q(1)$, the nodes throughput are obtained
 $r(1) \approx \frac{2}{2 - D - c(D) + \sqrt{(D - c(D) + 2)^2 + 4Dc(D)}}$ and $r(2) = 1 - r(1)$.

For a short network ($D \approx 0$) the nodes get equal throughput. In a large network, the minimum throughput of node{2} (i.e., $c(D) = D$) is $\approx 1/2D$, while in its most unfavourable scenario ($c(D) = -D$) node{1} is less penalized as it gets $\approx 1/\sqrt{2D}$ of the network capacity.

Deterministic models to analyze the DQDB asymptotic behavior have also been presented in ([66], [21], [26], [43]).

Node-Concentrated Models. In [50] a DQDB network with N stations is analyzed. Packets arriving at a station for transmission on a bus are divided into fixed-length segments. Segments are then queued in the Local Queue, LQ, related to their priority level; four priority levels⁴ are assumed for the transmission of the asynchronous traffic.

To make the analysis possible the following assumptions are made. Propagation and processing delays are zero. The request channel has an infinite capacity. The order in which packets arrive in the system is random, i.e., the order in which segments are transmitted does not depend on the position of the station.

It is then easy to observe that these hypotheses correspond to the *ideal conditions* (see Section 4.1) according to which DQDB provides a Round Robin (RR) sharing type of service to its nodes. Therefore, to analyze the DQDB behavior under these *ideal conditions* the authors propose a discrete time Multi-queue Processor Sharing (MPS) model which extends the classical RR model [36]. The classical RR model is based on a single server queue, in which newly arriving customers join the end of the queue. Customers are served on a FIFO basis. When a customer is served, it receives a quantum of service, and if it requires more service it rejoins the end of the queue. The MPS model extends the RR model by introducing multipriority levels, LQs and discrete time services.

In the MPS, arriving packets are queued in the LQ related to their priority level. There is a separate LQ for each priority class in each station. One representative (if any) for each LQ is inserted in the Processor Sharing (PS) queue. In the PS queue, packets of different priority levels are served on a strict priority discipline (HoL discipline). Packets of the same priority level are served on a RR basis: a packet in the PS queue recycles through the service facility, receiving (each time) a quantum of service equivalent to one segment transmission. The packet only leaves the PS queue after it has recycled enough times to service all the segments in the packet.

By assuming that the number of priority- p packets arriving in a slot time are independent and identically distributed (i.i.d.) and independent of the arrivals at the other LQs; and that the number of segments in the packets are discrete and i.i.d. (the distributions may differ for different priority levels), the authors derive a closed formula for the mean time ($D_p(n)$) that a priority- p packet made up of an n segments spends in the system (i.e., the time from its arrival until it is transmitted).

Hereafter \bar{a}_p and $C_{a,p}^2$ will be used to denote the average and the squared coefficient of variation of the number of priority- p packet arrivals in a slot, respectively; while \bar{b}_p and $C_{b,p}^2$ and $F_{b,p}(\cdot)$ will be used to denote the average, the squared

⁴ This model has been defined to analyze an older version of DQDB in which four priority levels for the asynchronous traffic were designed.

coefficient of variation, and the probability distribution function of the length (in segments) of a priority- p packets, respectively.

In the MPS model $D_p(n)$ includes the time a priority- p packet spends in the LQ (L_p) and the time it spends in the PS queue to serve all its n segments ($S_p(n)$).

As in the RR model, it can be proved that $S_p(n)$ linearly increases with n . In fact, in [50], by indicating with $N_p(n)$ the mean number of priority- p packets in the PS queue which have already obtained exactly n quantum of service (i.e., n segments transmission), it is shown that $S_p(n) = n(N_p(0)/\lambda_p)$, where λ_p is the total priority- p packet arrival rate to the system in a slot time. $N_p(0)$ is derived by measuring the delay of a *long test packet* as proposed by Kleinrock [36]. By considering a long test packet at priority- p made up of x segments, the time it spends in the PS queue tends towards the sum of 1) the service time x of the test packet, 2) the service times required by all the priority- p packets which join the PS queue during the service of the test packet (i.e., the priority- p packets representatives of the LQs except that related to the test packet), and 3) the service times of the packets with priority q higher than p ($q = p + 1, \dots, H$)⁵, which join the PS queue during the service of the test packet

$$S_p(x) \xrightarrow{x \rightarrow \infty} x + S_p(x) \left[\frac{M_p - 1}{M_p} \rho_p + \sum_{q=p+1}^H \rho_q \right],$$

where M_p is the number of priority- p LQs, and ρ_q is the priority- q bus utilization, $\rho_q = \lambda_q \cdot \bar{b}_q$, $q = 0, 1, \dots, H$.

After some algebraic manipulations, it can be shown that

$$S_p(n) = \frac{n}{1 - \sigma_{p+1} - \frac{M_p - 1}{M_p} \rho_p}, \text{ where } \sigma_{p+1} = \sum_{q=p+1}^H \rho_q$$

To obtain L_p , the authors exploit the equivalence, in terms of average segment delay, between the MPS model and the discrete-time M/G/1 with preemptive resume priority queueing model (hereafter referred to as $D[M/G/1]_{PR}$) studied in [52].

Indicating with $R_p(k)$ the probability that a randomly selected priority- p segment is the k^{th} in its own packet, in [50] it is shown that $R_p(k) = (1 - F_{b,p}(k-1))/\bar{b}_p$. Therefore the average segment delay in the MPS queueing system can be expressed as $E[D_{seg(p)}] = \sum_{k=1}^{\infty} [L_p + S_p(k)] \cdot R_p(k)$. Finally, by equating the last expression with the expression of the average priority- p segment delay in the equivalent $D[M/G/1]_{PR}$ system a closed-form expression for L_p can be derived

$$L_p = \frac{v_p + \sum_{q=p}^H \frac{\rho_q v_q}{(1 - \sigma_p)}}{2(1 - \sigma_{p+1})} - \frac{[\bar{b}_p(1 + C_{b,p}^2) + 1]}{2 \left(1 - \sigma_{p+1} - \frac{M_p - 1}{M_p} \rho_p \right)} + \frac{1}{2},$$

⁵ The authors solve the model for a general number H of priority levels. Then they instantiate the model on DQDB by setting $H=4$.

where $v_q = \bar{b}_q (C_{b,q}^2 + \lambda_q C_{a,q}^2 / M_q)$

In [50] it is clearly shown, by using simulative results, that the performance indices obtained with the MPS provide an adequate estimation of the DQDB performance figures, given that the distance between stations is small.

4.5 Node-in-Isolation Models

The modeling and performance analysis of the DQDB network is very difficult problem due to the high degree of interactions among several processes. This implies that exact DQDB models must take into consideration all the details of the protocol (the values of the counters in each node, the status of the slots travelling in the network and the length of the local queues).

Exact models have a very large state space and this means that they can be analyzed in very specific scenarios. However, when DQDB operates in underload conditions the following observations indicate how DQDB modeling complexity can be reduced ([12], [14]):

- 1) in underload conditions the number of empty slots is greater than the number of segments to be transmitted;
- 2) the time it takes a REQ sent by node{j} to affect upstream nodes depends on the physical distance between the nodes. This time interval, in a MAN environment, may have a duration of several slots.

From 1) and 2) it follows that, when DQDB operates in underload conditions, a segment may often be transmitted in a slot ahead of the one corresponding to its REQ. Therefore, the correlation between the transmission of a segment and its REQ is almost negligible, and the only effect of the REQs is to widen the time interval between consecutive transmissions by a given node.

These observations indicate that a significant reduction in complexity is obtained by aggregating the influence of the downstream nodes, on a given node, in a process with the same average arrival rate of downstream REQs. In addition, the influence of the upstream nodes can be modelled via a stochastic process describing the status of the slots travelling on the forward bus observed by node{i}.

Knowledge of these processes reduces the complexity in DQDB modeling by looking at each *node-in-isolation*. Following this approximate approach [4] the node under study is tagged and, with respect to the *tagged node*, the network is partitioned into

- an *L_NET* (for Left Network) which includes all the upstream nodes (from the tagged node);
- the tagged node itself, and
- the *R_NET* (for Right Network) which includes all the downstream nodes (from the tagged node).

With respect to the tagged node, the *L_NET* is a generator of Busy/Empty slots on Bus A (*L_NET process*), while the *R_NET* is a generator of requests on Bus B (*R_NET process*).

In all the existing analytical models *R_NET* generates requests according to a memoryless distribution (Poisson or Bernoulli process), while research on modeling DQDB nodes in isolation has concentrated on solving the following subproblems

- 1) *L_NET* modeling;
- 2) Tagged Node modeling.

L_NET Modeling. Results reported in ([11], [12], [5]) show that the number of consecutive busy slots observed by nodes close to the head node has a nearly geometric distribution, i.e. the number of the busy bit in consecutive slots is independent. On the other hand, while moving towards the end node the correlation among busy bit in consecutive slots sharply increases.

In literature, *L_NET* is frequently modelled by a Bernoulli process ([64], [4]).

Obviously, this model does not take into consideration the correlation between consecutive slots. To overcome this, in ([15], [16], [20]) models are presented which consider the correlation between consecutive slots.

In the previous section we pointed out that a node (e.g. node $\{i\}$) may often transmit a segment in an empty slot positioned ahead of the slot forced empty by the node's own REQ. This event can occur when:

- a) no node upstream of the node $\{i\}$ has a segment to transmit;
- b) the empty slot (somewhere ahead of that requested by node $\{i\}$) has been forced by a REQ which has already been satisfied (henceforth referred to as a *worthless REQ*).⁶

If the L_NET process observed by a node were only due to point a), the L_NET process would correspond exactly to the busy period process of an M/D/1 system [TAKA91] where the input traffic is obtained by the superposition of the input streams of all the upstream nodes. However, in the light of point b), the busy periods of the M/D/1 system are subdivided into smaller units due to the worthless REQs.

It thus follows that L_NET can be studied by referring to a *Simplified DQDB* network which is characterized by the following assumptions.

The reverse bus is modelled by introducing, for each node, a Poisson process to characterize the arrival of REQs generated by downstream nodes; the forward bus is slotted and the status of the slots is modelled via the *L_NET process*; the MAC protocol is modelled by a queue where REQs and segments are stored on a FIFO basis. Consequently, for each node $\{i\}$, where $1 \leq i \leq K$, the input traffic is made up of segments and REQs. The arrival process is Poisson with $\lambda(i)$ parameters, where $\lambda(i) = \lambda_S(i) + \lambda_R(i)$; $\lambda_S(i)$ and $\lambda_R(i)$ are the segment and the REQ arrival rate, respectively. Obviously, $\lambda_S(i)$ depends on the workload characterization, whereas

$\lambda_R(i) = \sum_{j=i+1}^K \lambda_S(j)$. The transmission time of both segments and REQs is constant and

equal to the slot duration, and both can be transmitted in the first empty slot seen by a node. An empty slot used for a transmission remains empty if there is a REQ at the head of the queue, but becomes busy if there is a segment at the head of the queue. The probability that a queued packet is a REQ (P_{REQ}) or a segment (P_{SEG}) are

$P_{REQ}(i) = \lambda_R(i) / (\lambda_R(i) + \lambda_S(i))$ and $P_{SEG}(i) = 1 - P_{REQ}(i)$, respectively.

Simulative results presented in [14] show that the Simplified DQDB behaves very similarly to DQDB at least from the slot-occupancy-pattern process standpoint.

In ([15], [16], [20]) the L_NET process of the Simplified DQDB is characterized by observing, for each node $\{i\}$, the L_NET process immediately ahead ($(S_{inp}(i) := \{S_j^{(i)}; j \in \mathbb{N}\})$) of and behind ($(S_{out}(i) := \{(S_j^{(i)}, A_j^{(i)}); j \in \mathbb{N}\})$) node $\{i\}$ which are referred to as *input* and *output process* respectively. The random variable $S_j^{(i)}$ takes the value B if the j^{th} slot is busy, and the value E if the j^{th} slot is empty, while $A_j^{(i)}$ represents the action of node $\{i\}$ on the j^{th} slot. Thus the random vector $(S_j^{(i)}, A_j^{(i)})$ takes the following values:

⁶ This can happen in two different scenarios. In the first the REQ has been generated by a node upstream of node $\{i\}$ that, by the time it observes the requested empty slot, no longer has a segment to transmit. In the second, the REQ generated by a node downstream of node $\{i\}$ has already been satisfied by the node $\{i\}$ itself via a previous empty slot.

- (B0): if the slot is already busy in the input process;
- (E1): if the slot was empty and the node uses it for segment transmission;
- (E0): if the slot was empty, and the node queue is either empty or there is a REQ on top of it.

Obviously the states of consecutive slots are not independent and thus processes $S_{inp}(i)$ and $S_{out}(i)$ do not satisfy the Markov property. However, the correlation between the status of two slots separated by n slots tends to weaken as n increases. In [15] $S_{inp}(i)$ and $S_{out}(i)$ are therefore approximated with discrete time Markov processes $S_{inp}^{[n]}(i)$ and $S_{out}^{[n]}(i)$ which only capture the dependencies between n consecutive slots (n^{th} -order discrete-time Markov process).

The state space of $S_{inp}^{[n]}(i)$ is $\{(s_1, s_2, \dots, s_n) | s_j \in \{B, E\}, 1 \leq j \leq n\}$, and its transition probabilities are $P_i\{S_{n+1} = s_{n+1} | (S_1 = s_1, S_2 = s_2, \dots, S_n = s_n)\}$. Each n -tuple (s_1, s_2, \dots, s_n) describes the state of the last n consecutive slots observed on the forward bus by the tagged node. While the state space of $S_{out}^{[n]}(i)$ is $\{(s_1 a_1, s_2 a_2, \dots, s_n a_n) | s_j a_j \in \{(B0), (E0), (E1)\}, 1 \leq j \leq n\}$, and its transition probabilities are $Q_i\{S_{n+1} A_{n+1} = s_{n+1} a_{n+1} | (S_1 A_1 = s_1 a_1, S_2 A_2 = s_2 a_2, \dots, S_n A_n = s_n a_n)\}$.

By noting that $S_{inp}^{[n]}(1)$ is known (all slots observed by node{1} are empty), and that the $S_{inp}^{[n]}(i+1)$ can be easily constructed from $S_{out}^{[n]}(i)$ in ([16], [20]) the L_NET study is performed by defining algorithms to compute the n^{th} -order Markov model of the output process of node{ i } starting from the n^{th} -order Markov model of the input process. All these algorithms first compute the joint probability distribution function of the status of $(n+1)$ consecutive slots in the output process (j^{n+1} - pdf) and then define the transition probabilities in the output process by using the definition of the conditional probability density function and the j^n - pdf.

Following the methodologies used for deriving the output process model from the input process, below we divide the 1^{st} -order models and the models which take into consideration higher orders of correlation.

1^{st} -order Markov Models: for this class of models the computation is based on the result of the following theorem proved in [10].

Theorem: the first-order Markov model of the input process, and of the output process related to node{ i } are regenerative processes with respect to the sequence $(T = \{T_j; j \in \mathbb{N}\})$, where T is a renewal process (T defined by the successive instants at which the queue of node{ i } in the Simplified DQDB becomes empty (renewal times)).

By using the regenerative property of the input and output processes we derive a closed formula for the j^2 - pdf. The paper focuses on the computation of $\Pr\{B, B\}$, as the transition probabilities of the output process can easily be determined from it; and $\Pr\{B, B\}$ is computed by applying renewal theoretical arguments [68]: $\Pr\{B, B\} = E[N_{BB}] / E[\text{Cycle}]$, where $E[N_{BB}]$ is the average number of (B,B) couples in a generic renewal period and $E[\text{Cycle}]$ is the average length of a generic renewal period. The computation of these unknown quantities is based on the fact that the renewal period has the same distribution as the delay period of an M/G/1 system with exceptional first service [61], [36].

Results presented in [15], show that even at light loads the interdependence between slots is significant, and that the Bernoulli hypothesis generally used for modeling the length of busy trains diverges from the real behavior at almost any load condition. On the other hand, by using a simulative analysis it is shown that the first-order Markov model is able to capture almost all the 1st-order dependencies in the output process of a DQDB network for a wide range of offered loads (up to OL=.70).

In [17] this model was extended to include the effect of the BWB mechanism on the L_NET process.

nth-order Markov Models: the methodology developed for the 1st-order Markov Models (described in [15] and [17]) can be used only for approximating the slot-occupancy-pattern process via a 1st-order discrete-time Markov process, as it requires that the number of consecutive busy slots in the L_NET process be represented by i.i.d. random variables. On the other hand, the approach presented in ([16], [20]) can deal with all orders of correlation.

In ([16], [20]) the output process, immediately after a generic node {i}, is computed by studying the following auxiliary discrete-time Markov process $\{(S_1, A_1), (S_2, A_2), \dots, (S_n, A_n), L_n\}$, where A_1, A_2, \dots, A_n are random variables which describe the actions performed by node {i} on the last n slots it observed and L_n ($L_n \in \mathbb{N}$) is the length of the node {i} queue immediately after the sequence of actions A_1, A_2, \dots, A_n .

Once the steady state probabilities of the auxiliary process are known the $j^{n+1} - pdf$ and $j^n - pdf$ of the output process are easily derived.

$$\begin{aligned} \Pr\{(S_1 A_1), \dots, (S_n A_n), (S_{n+1} A_{n+1})\} = \\ \Pr\{(S_{n+1} A_{n+1}) \mid (S_1 A_1), (S_2 A_2), \dots, (S_n A_n)\} \cdot \Pr\{(S_1 A_1), (S_2 A_2), \dots, (S_n A_n)\} = \\ \Pr\{(S_{n+1} A_{n+1}) \mid (S_1 A_1), (S_2 A_2), \dots, (S_n A_n), L_n > 0\} \cdot \Pr\{(S_1 A_1), \dots, (S_n A_n), L_n > 0\} + \\ \Pr\{(S_{n+1} A_{n+1}) \mid (S_1 A_1), (S_2 A_2), \dots, (S_n A_n), L_n = 0\} \cdot \Pr\{(S_1 A_1), \dots, (S_n A_n), L_n = 0\}. \end{aligned}$$

$\Pr\{(S_1 A_1), \dots, (S_n A_n), L_n > 0\}$ and $\Pr\{(S_1 A_1), \dots, (S_n A_n), L_n = 0\}$ are the steady state probabilities of the auxiliary process and

$$\begin{aligned} - \Pr\{(S_{n+1} = s_{n+1}, A_{n+1} = a_{n+1}) \mid (S_1 A_1), (S_2 A_2), \dots, (S_n A_n), L_n = 0\} = \\ P\{S_{n+1} = s_{n+1} \mid S_1, S_2, \dots, S_n\} \cdot I_{\{a_{n+1} = 0\}} \\ - \Pr\{(S_{n+1} = s_{n+1}, A_{n+1} = a_{n+1}) \mid (S_1 A_1), (S_2 A_2), \dots, (S_n A_n), L_n > 0\} = \\ P\{S_{n+1} = s_{n+1} \mid S_1, S_2, \dots, S_n\} \cdot \{I_{\{s_{n+1} = B, a_{n+1} = 0\}} + P_{SEG} \cdot I_{\{s_{n+1} = E, a_{n+1} = 1\}} + P_{REQ} \cdot I_{\{s_{n+1} = E, a_{n+1} = 0\}}\} \end{aligned}$$

where $P\{\cdot\}$ are the transition probabilities of the input process and $I_{\{A\}}$ is the indicator function of the event A .

Two solution methods are proposed for computing the steady state probabilities of the auxiliary process required to compute the output process. In [16] a closed formula is derived for the probability generating functions of the number of users in the system (PGF), while in [20] steady state probabilities are numerically computed by solving an M/G/1-type system. The computation of the PGFs, requires a closed formula for the solution of a linear system of 3^{order of correlation} equations with 3^{order of correlation} unknowns and therefore it can only be applied for small orders of correlation (up to three). On the other hand, a numerical solution of the steady state probabilities is obtained by applying the theory developed by Neuts in [49] for M/G/1 type systems, where the size of the

square blocks in the transition matrix is $3^{\text{order of correlation}}$.

By carrying out a simulative analysis in [20], it is shown that the n^{th} -order Markov models can capture almost all the significant dependencies in the output process of a DQDB network for a wide range of offered loads ($OL \leq 0.60$). Furthermore, the n^{th} -order-Markov-model characterization always outperforms the Bernoulli characterization.

Tagged Node Models. The exact queueing model of the DQDB MAC protocol in the tagged node is a discrete time single server queue with two classes of customers: the segments generated by a (tagged) node and the reservations from downstream nodes. The two classes are served according to a *two-state discipline* which reflects the different behaviors of DQDB depending on whether or not the segment queue is empty. When the segment queue is empty, *idle-discipline*, the server attends the reservation queue continuously. When a segment arrives the *busy-discipline* is applied. In this discipline the server serves the reservations queued according to a gated discipline, and then the segment queue according to a one limited discipline. After segment transmission, the busy-discipline is repeated over and over again until the segment queue becomes empty. This two state discipline is referred to as *quasi-gated* discipline in [6] or the consistent gated/limited priority policy with head of line service (*c-G/L/HoL*) in [40].

Several papers in literature have analyzed the tagged node in isolation with the quasi-gated discipline. A general solution has not yet been found. Exact solutions exist only when the tagged node has a single buffer for queueing its messages ([4], [33]). In the case of a tagged node with an infinite buffer, the works in this survey can be divided into two classes: 1) papers which reports solution methodologies which provides bounds on the average performance figures ([3], [40]); 2) papers which provide simple closed formulas which approximate the tagged node performance figures ([64], [14], [22]).

Single Buffer Models: an exact analysis of the node model which provides the generating function of the access delay is reported in [4]. The main simplifying hypothesis is that no more than one segment can be stored in a node (single buffer model); while the cumulative traffic generated by L_NET and R_NET are assumed to be Bernoulli processes with probabilities α , and β to observe a busy and a request, respectively.

The work computes the waiting time (W) experienced by a packet from the moment it enters the buffer until the time instant at which its transmission starts. To this end, the author first derives the probability generating function of the random variable F which is equal to the number of outstanding requests at the time a packet is ready for

transmission ($G_F(z) = \sum_{i=0}^{\infty} \phi_i z^i$, where ϕ_i is the steady state probability that a packet

finds i outstanding requests) and then, he easily computes the statistics of the packet waiting time by conditioning on the number of outstanding requests. In fact, by conditioning the waiting time on the event $\{F = i\}$ its probability generating function

($G_w(z; i)$) satisfies the following equation $G_w(z; i) = \left(\frac{(1-\alpha)z}{1-\alpha z} \right)^i$, $|z| \leq 1$, $i = 0, 1, 2, \dots$,

and then $G_w(z) = \sum_{i=0}^{\infty} G_w(z; i) \cdot \phi_i = G_F \left(\frac{(1-\alpha)z}{1-\alpha z} \right)$, $|z| \leq 1$.

Finally, the statistics of the access delay (D) of a packet, which is equal to the waiting time in the buffer plus a slot time (which corresponds in this analysis to the

time unit) to transmit the packet itself (i.e., $D=W+1$) are easily derived from its probability generating function, $G_D(z)$.

$$G_D(z) = zG_W(z) = zG_F\left(\frac{(1-\alpha)z}{1-\alpha z}\right), |z| \leq 1.$$

From this probability generating function the average access delay is obtained

$$E[D] = \frac{1-\alpha}{1-\alpha-\beta} \left[\beta + \frac{e^{-\lambda}}{1-e^{-\lambda}} \{ \alpha\beta - (1-\alpha)(1-\beta)(1-\theta) \} + 1 - \theta(1-\alpha) \right]$$

where $1 - e^{-\lambda}$ represents the probability of a segment generation in a slot by the tagged node in the hypothesis of exponential segment interarrival times, and θ represents the probability that a segment finds no outstanding requests ahead of it when it is generated by the tagged node; θ can be numerically calculated from the analysis of the Markov chain.

The main part of this work is therefore devoted to deriving the steady state probabilities ϕ_i that an incoming packets finds i outstanding requests. To compute these statistics the author exploits, that, if we denote with F_n the number of outstanding requests at the arrival of the n^{th} -packet, the sequence $\{F_n, n \geq 1\}$ is a Markov chain embedded at the packet's arrival instants. This Markov chain is homogeneous, irreducible, aperiodic and it is ergodic if $\beta < 1 - \alpha$ (i.e., the request arrival rate is less than the rate of the empty slots observed by the tagged node).

The computation of the transition probabilities $\psi_{i,j} = \Pr\{F_{n+1} = j | F_n = i\}$ of the Markov chain is subdivided by applying the Chapman-Kolmogorov equation

$\psi_{i,j} = \sum_{k=0}^{\infty} \sigma_{i,k} \tau_{k,j}$; where $\sigma_{i,k}$ is the conditional probability that there are k outstanding requests after the transmission of a packet given that there were i outstanding requests when the packet arrived; and $\tau_{k,j}$ is the conditional probability that a packet finds j outstanding requests at its arrival given that there were k outstanding requests just after the transmission of the previous packet.

By denoting with $Y(i)$ a random variable such that $\Pr\{Y(i) = k\} = \sigma_{i,k}$, and with $G_Y(z; i)$ its generating function of $Y(i)$ the paper shows that

$$G_Y(z; i) = (1 - \beta + \beta z) \left(\frac{(1-\alpha)(1-\beta+\beta z)}{1-\alpha(1-\beta+\beta z)} \right)^i, |z| \leq 1.$$

The conditional probabilities $\tau_{k,j}$ are studied by investigating the sequence of random variables $\{Z_n(k)\}$ which denote the conditional number of outstanding requests n time units after the last packet transmission, given that there were k outstanding requests just after the transmission. The analysis of the sequence $\{Z_n(k)\}$ is equivalent to studying of the transient behavior of a random walk with the barrier at level 0.

By using the following definition

- $R(k)$ is the value of the sequence $\{Z_n(k)\}$ at a packet arrival instant;
- $G_R(z; k)$ is the generating function of $R(k)$;
- $\pi_j(n; k) = \Pr\{Z_n(k) = j\}$
- $\hat{\pi}_j(k) = \Pr\{R(k) = j\} = (1 - e^{-\lambda}) \sum_{n=0}^{\infty} \pi_j(n; k) e^{-n\lambda}$;

with some elaborate manipulation

$$G_R(z; k) = \frac{(1 - e^{-\lambda})z^k + \varepsilon(1 - z^{-1})e^{-\lambda}\hat{\pi}_0(k)}{1 - (\delta z + 1 - \delta - \varepsilon + \varepsilon z^{-1})e^{-\lambda}}, \text{ where}$$

$\delta = \alpha\beta$, is the probability that there is a slot with Busy=1 on the forward bus and with REQ=1 on the reverse bus;

$\varepsilon = (1 - \alpha)(1 - \beta)$, is the probability that there is a slot with Busy=0 on the forward bus and with REQ=0 on the reverse bus;

$\hat{\pi}_0(k)$ is the probability that the sequence is at the barrier level at the arrival instant; it can be calculated numerically.

From $G_R(z; k)$ and $G_Y(z; i)$ the probability generating function $G_F(z)$ can eventually be derived. In fact, by utilizing the relationship $\phi_j = \sum_{i=0}^{\infty} \phi_i \psi_{i,j} = \sum_{i=0}^{\infty} \phi_i \sum_{k=0}^{\infty} \sigma_{i,k} \tau_{k,j}$ it can be verified that the following functional equation holds.

$$G_F(z) = \sum_{j=0}^{\infty} \phi_j \sum_{i=0}^{\infty} \psi_{j,i} z^i = (\Theta)^{-1} \left[(1 - e^{-\lambda})(1 - \beta + \beta z) \cdot G_F \left(\frac{(1 - \alpha)(1 - \beta + \beta z)}{1 - \alpha(1 - \beta + \beta z)} \right) + (1 - \alpha)(1 - \beta)(1 - z^{-1})e^{-\lambda}\theta \right] z + \theta(1 - \alpha)(1 - z)$$

where $\theta = \phi_0/(1 - \alpha)$ is the probability that an arriving packet finds no outstanding requests; and $\Theta = 1 - [\alpha\beta z + \alpha + \beta - 2\alpha\beta + (1 - \alpha)(1 - \beta)z^{-1}]e^{-\lambda}$.

The model in [4] has been extended in [33] to the case where each node can queue one message of fixed length (l) segments. Results obtained from this model show that for sufficiently large l the message delay behaves like a linear function of the message length which was also observed in the simulative study reported in [5].

Bounds on the Performance Figures for Infinite Buffer Models: in [3] a model for a tagged node which by following the *quasi-gated* discipline is analyzed. For Poisson processes with intensity λ_r for requests registered by a node and λ_s for segments generated by a node and general service requirements for each class of arrivals, the analysis of the priority queueing system (assisted by the conservation law for work conserving queues [36]) yields the following for the mean delay \bar{D}_s experienced by the segments [3]

$$\bar{D}_s = \left(\bar{R} - (1 - \rho) \frac{\rho_r}{\rho_s} \zeta \bar{S}_s \right) / \left((1 - \rho_r)(1 - \rho) \right) + \bar{S}_s \text{ where } \bar{R}, \text{ the residual service time}$$

of the customer in service, $\bar{R} = \frac{1}{2} \left[\lambda_r \bar{S}_r^2 + \lambda_s \bar{S}_s^2 \right]$, and \bar{S}_i^2 ($i \in \{1, 2\}$) is the second moment of the service time, \bar{S}_i is the average service time, ρ_i is the offered load to the queue due to class i customers, $\rho = \rho_r + \rho_s$, and ζ is the probability that the segment queue is empty and the server serves the request queue. An analysis of a dominant and dominated queue shows that $\left[1 - (1 - \rho_s)/S_s^*(\lambda_s) \right] \rho_r \leq \zeta \leq \min \left[\bar{R} \rho_s / \bar{S}_s (1 - \rho), \rho_r \right]$.

A simulative analysis in [3] shows that these bounds are tight especially in light or heavy traffic conditions. In [6] the same queueing system is analyzed in the discrete time domain, but the derived bounds are not as tight as in the continuous case.

The discrete time version of the above model has been studied in [40]. Through an analysis of the model based on renewal/regenerative theory and a work conservation

law the authors obtain upper and lower bounds on the average access delay in the tagged DQDB node. In this model three class of traffic are considered which represent the busy slots travelling on the forward bus (upstream traffic), the segment queue in the tagged node and the request traveling on the reverse bus. In this system the time is assumed to be slotted and the service time of all classes is deterministic and equal to one slot. The segment and request arrival processes are assumed to be Bernoulli, while the busy slot arrival process is modelled by 1st-order Markov model [15] with at most one packet per slot. The upstream traffic queue have the highest priority and is served in accordance with the HoL discipline [68] while the segment and request queue are served following to the consistent gated/limited discipline (c-G/L) [40]; where c-G/L discipline is exactly the two states discipline previously described whenever the service time is deterministic and equal to one slot. This discipline is a priority discipline which guarantees that high priority packets will be served before any low-priority packets which arrive on the same slot or at a future time but at the same time is not inconsiderate of low priority traffic such as HoL. The analysis is based on renewal/regenerative theory, a work conservation law and the theory for approximating the solution of infinite systems of equation [35].

Let $\{S_n : n \in N\}$ be the sequence of time instants in which the system is empty. This sequence S_n denotes renewal cycles and the length of the n^{th} renewal cycle is $X_n = S_n - S_{n-1}$. Let C_n^i indicate the cumulative delay of the i^{th} -priority packets that arrived during the n^{th} cycle. $\{C_n^i : n \in N\}$ is a regenerative process with respect to the renewal process $\{S_n : n \in N\}$ and thus by applying classical renewal/regenerative arguments the mean delay of an i^{th} priority packet D^i can be obtained from $D^i = \bar{C}^i / \lambda^i \cdot \bar{X}$, where \bar{C}^i is the expected value of the cumulative delay of the i^{th} -priority packets that arrived during a renewal cycle, λ^i is the arrival rate of the i^{th} class, and \bar{X} is the average length of a renewal cycle.

Since the c-G/L/HoL is a work conserving system a work conservation law can be used to define a relationship between the average access delay of the same system with the FIFO service discipline (D_{FIFO}) and the weighted sum of the average access delay

of each class in the c-G/L/HoL system: $\lambda \cdot D_{FIFO} = \sum_{i=1}^3 \lambda_i \cdot D_i$.

By analyzing the system in a renewal cycle a set of linear equations among cumulative delays is obtained. The structure of the generic equation is

$$C^H(i, j) = a^H(i, j) + \sum_{i'=0}^{\infty} \sum_{j'=0}^{\infty} b^H(i, j, i', j') \cdot C^H(i', j'), \text{ where}$$

- $a^H(i, j)$ and $b^H(i, j, i', j')$ are constants;
- $C^H(i, j)$ is a r.v. describing the cumulative delay of all the H packets which arrived (and were served) over the time it takes the system to move from the state i, j (at time t_n) to empty;
- j is the amount of time that has elapsed since the gate was closed in High priority queue for the n^{th} time;
- i is such that $i+j$ describes the time distance from t_n and the arrival of the packet at the head of the low priority queue.

By applying the theory of infinite dimensional linear equations for each class a lower bound of the average access delay (D_i^{lo}) is computed. From the work conservation law

and the lower bounds (D_i^{lo}) the upper bound for a specific class can easily be

$$\text{derived: } D_j^{up} = \frac{1}{\lambda_j} \left\{ \lambda \cdot D_{FIFO} - \sum_{\substack{i=1 \\ i \neq j}}^3 \lambda_i \cdot D_i^{lo} \right\}.$$

Approximate Solutions for Infinite Buffer Models: in [64], simple closed-formulas for approximating the performance figures of a tagged node in a DQDB network are derived. The analyzed DQDB network has N stations transmitting asynchronous traffic of one priority level⁷. Incoming messages to the network stations are made up of single segments; segments arrive at each node $\{i\}$ according to a Poisson distribution with rate λ_i . The L_NET process of a node $\{i\}$ is modelled by a Bernoulli process with a

probability q_i $\left(q_i = \sum_{j=1}^{i-1} \lambda_j \right)$ to observe a busy slot, while the R_NET process of

node $\{i\}$ is modelled by a Poisson process with rate Λ_i $\left(\Lambda_i = \sum_{j=i+1}^N \lambda_j \right)$. By indicating

with τ the slot duration, it follows that the bus utilization of node $\{i\}$ is $\rho_i = \lambda_i \cdot \tau$, and

the total bus utilization is $\rho = \sum_{i=1}^N \rho_i$.

One of the key concepts in the analysis of this model is the decomposition of a tagged-segment access delay (the time a segment spends within a node) in intervals identified by the following time instants: 1) arrival epoch of the segment; 2) time instant, at which the observed segment is scheduled for transmission on bus A; 3.) the time instant at which the observed segment arrives at the top of the distributed queue; 4) end of the transmission of the tagged segment on the forward bus.

The segment access delay at a given node i is obviously the interval between instant 1 and 4 (T_{14}), and it can be decomposed into the following random variables.

- T_{12} : is the waiting time in the local queue in station i .
- T_{23} : is time that a segment spends in the distributed queue from its insertion in the distributed queue to the time instant at which it arrives at the head of the distributed queue.
- T_{34} : is the virtual transmission time, i.e., the time between successive empty slots (on Bus A) as seen by the (tagged) node i . According to the above hypothesis T_{34} has a geometric distribution and its LST is $\Phi_{34}(s) = [(1 - q_i)z] / (1 - q_i z)$, where $z = e^{-s\tau}$.
- T_{34} is the time to transmit a segment or to satisfy a downstream reservation. Therefore the waiting time T_{23} of segments in the scheduling position is studied with

a standard M/G/1 system with arrival rate Γ_i $\left(\Gamma_i = \sum_{j=i}^N \lambda_j \right)$ and service time T_{34} .

⁷ In the paper it is also proposed an approximation to study the effect of the isochronous traffic on the performance figures of the asynchronous traffic.

Thus the LST of T_{23} is $\Phi_{23}(s) = \frac{s(1 - \Gamma_i \cdot E[T_{34}])}{s - \Gamma_i(1 - \Phi_{34}(s))}$, where $E[T_{34}] = \frac{\tau}{1 - q_i}$ is the average of T_{34} . $\Phi_{23}(s) \cdot \Phi_{34}(s)$ can be seen by the segments arriving at node $\{i\}$ as the *virtual service time* to be transmitted on the bus. Therefore T_{12} can be approximated with the waiting time experienced by the segments in an M/G/1 system with arrival rate λ_i with service time equal to the virtual service time. Hence,

$$\Phi_{12}(s) = \left[\frac{s(1 - \lambda_i \cdot E[T_{24}])}{s - \lambda_i(1 - \Phi_{24}(s))} \right].$$

Finally, the LST of the access delay (T_{14}) is approximated by $\Phi_{14}(s) = \Phi_{12}(s) \cdot \Phi_{23}(s) \cdot \Phi_{34}(s)$

Obviously, the main approximations introduced in this approach are related to the computation of the T_{23} distribution. In fact, both the possibility of having more than one segment per node in the distributed queue and the Poisson hypothesis on the REQs arrival are conservative. Despite these approximations, the model is able to capture the dependence of the performance figures either from the node position or from the traffic pattern.

The model proposed in [64] was extended in [14] to take into consideration a non memoryless distribution of the *length of the busy train* (T_{34}), i.e., the number of consecutive busy slots. In that paper the lengths of consecutive busy trains constitutes a renewal process and hence those messages which arrive when the queue is empty are considered separately (i.e., such messages experience an exceptional service). The waiting time of the messages (segments or requests) from the distributed queue (T_{23}) is therefore modelled by the waiting time in an M/G/1 system with an exceptional first service in a busy period [61].

In [22], an attempt to extend the nested M/G/1 queue model to the multi-segment messages was made. Unfortunately, even for small (5-segment) messages and moderate loads (0.60), the proposed approximation significantly deviates from simulative results.

Just recently, [58] reports an M/G/1-type model of the tagged node from which the statistics of the queue-length occupancy distribution are derived. To obtain an M/G/1-type model the authors assume that *a*) the maximum number of outstanding requests in the tagged node is bounded; and *b*) the L_NET process is modelled via a Bernoulli process.

Bibliography

- [1] B. W. Abeyesundara, A. E. Kamal, "High-Speed Local Area Networks and their Performance: a Survey", *ACM Computing Surveys*, 23, (2), pp. 221-264.
- [2] FDDI Token Ring Media Access Control, ANSI X3.139, 1987, ANSI, New York.
- [3] C. Bisdikian, "A Queueing Model with Applications to Bridges and the DQDB MAN", IBM Research Report, RC 15218, December 1989.
- [4] C. Bisdikian, "Waiting time analysis in a single buffer DQDB (802.6) network", *IEEE Journal on Selected Areas in Communications*, Vol. 8, No. 8, October 1990, pp. 1565-1573.
- [5] C. Bisdikian, "A Performance Analysis of the IEEE 802.6 (DQDB) Subnetwork with the Bandwidth Balancing Mechanism", *Computer Networks and ISDN Systems*, Vol. 24 (1992), pp. 367-385.

- [6] C. Bisdikian, "A Queueing Model for a Data Station within the IEEE 802.6 MAN", Proceedings IEEE 17th Local Computer Networks Conference, Minneapolis, MN, USA, September 13-16, 1992.
- [7] K. W. Chang and D. Sandhu, "Delay Analysis of Token-Passing Protocols with Limited Token Holding Times", Proceedings IEEE INFOCOM' 92 Conference, Florence, May 1992, pp. 2299-2305.
- [8] K. W. Chang and D. Sandhu, "Mean Waiting Time Approximations in Cyclic-Service Systems with Exhaustive Limited Service Policy", *Performance Evaluation*, 15 (1992), pp. 21-40.
- [9] J. Chiarawongse, M. M. Srinivasan, T. J. Teorey, "The M/G/1 Queueing System with Vacations and Timer Controlled Service", Technical Report, Center for Information Technology Integration, The University of Michigan, May 1991.
- [10] M. Conti, E. Gregori, L. Lenzini, "On the Approximation of the Slot Occupancy Pattern in a DQDB Network", CNUCE Report C91, December 1991.
- [11] M. Conti, E. Gregori, and L. Lenzini, "A Methodological Approach to an Extensive Analysis of DQDB Performance and Fairness", *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 1, January 1991, pp. 76-87.
- [12] M. Conti, E. Gregori, L. Lenzini, "A Comprehensive Analysis of DQDB", *European Transactions on Telecommunications and Related Technologies* Vol. 2, No. 4, July/August 1991, pp. 403-413.
- [13] M. Conti, E. Gregori, L. Lenzini, "A Vacation Model for Interconnected FDDI Networks", Proceedings ACM 1992 Computer Science Conference, March 3-5, 1992, Kansas City, pp. 9-16.
- [14] M. Conti, E. Gregori, L. Lenzini, "DQDB Modeling: Reduction of the Complexity and a Solution via Markov Chains", *IFIP Transactions on Performance of Distributed Systems and Integrated Communication Networks*, C-5, 1992.
- [15] M. Conti, E. Gregori, L. Lenzini, "On the Approximation of the Slot-occupancy Pattern in a DQDB Network", *Performance Evaluation*, Vol. 16 (1992) No. 1-3, November 1992, pp. 159-176.
- [16] M. Conti, E. Gregori, L. Lenzini, "2nd-order Markov Approximation of the Slot-Occupancy Pattern in a DQDB Network", Proceedings IEEE GLOBECOM'92, December 6-9, 1992, Orlando, Florida, pp. 1647-1651.
- [17] M. Conti, E. Gregori, L. Lenzini, "A Model to Evaluate the Effects of the BWB Mechanism in a DQDB Network in Underload Conditions", Proceedings ACM 1992 Computer Science Conference, March 3-5, 1992, Kansas City, pp. 395-405.
- [18] M. Conti, E. Gregori, L. Lenzini, "Analysis of a Medical Communication System Based on FDDI", to appear on the *International Journal of Microcomputer Applications*. See also Proceedings IC³N, San Diego, CA, 8-10 June, 1992, pp.138-142.
- [19] M. Conti, "Analysis of the Quality of Service in a MAN Environment", Proceedings IFIP WG10.3 International Conference on Decentralized and Distributed Systems, Palma de Mallorca, Spain 15-17 September, 1993.
- [20] M. Conti, E. Gregori, L. Lenzini, M. F. Neuts, "An M/G/1 Type Approach to the Approximation of the Slot-occupancy Pattern in a DQDB Network", submitted for publication.
- [21] P. Davids and P. Maini, "Performance analysis of DQDB," Proceedings IEEE 1990 Phoenix Conference, Phoenix, AZ, March 1990, pp. 548-555.
- [22] L. F. M. de Moraes, "Simple Approximation for Frame Delays in DQDB Networks and Comparison with a Slotted Bus without Reservations", Proceedings 8th Annual EFOC/LAN Conference, Munich, Germany, June 27-29,1990, pp. 240-245.

- [23] D. Dykeman and W. Bux, "Analysis and Tuning of the FDDI Media Access Control Protocol", *IEEE Journal on Selected Areas in Communications*, Vol. 6, No. 6, July 1988, pp. 997-1010.
- [24] B.T. Doshi, "Queueing System with Vacations - a Survey", *Queueing Systems* 1(1986), pp. 29-66.
- [25] B.T. Doshi, "Single Server Queues with Vacations", *Stochastic Analysis of Computer and Communication Systems*, Ed. H. Takagi (Elsevier/North-Holland, 1990), pp. 217-318.
- [26] J. Ferguson, "Towards Formal Structures for IEEE 802.6", NATO Advanced Research Workshop, Sophia Antipolis, France, June 1990.
- [27] W. L. Genter, K. S. Vastola, "Delay Analysis of the FDDI Synchronous Data Class", *Proceedings IEEE INFOCOM'90*, San Francisco, June 1990, pp. 766-773.
- [28] M. Gerla, H. W. Chan, J. R. Boisson de Marca, "Fairness in Computer Networks", *Proceedings ICC'85*, pp. 1384-1389.
- [29] E. L. Hahne, A. K. Choudhury, N. F. Maxemchuck, "DQDB Networks with and without Bandwidth Balancing", *IEEE Transactions on Communications* Vol. 40, N. 7, July 1992, pp. 1192-1204.
- [30] J. L. Hammond, P. J. P. O'Reilly, "Performance Analysis of Local Computer Networks", Addison-Wesley, 1988.
- [31] IEEE802.6 Standard: Distributed Queue Dual Bus (DQDB) Metropolitan Area Network, 1990.
- [32] N. K. Jaiswal. "Priority Queues", Academic Press, 1968.
- [33] W. Jing and M. Paterakis, "Message Delay Analysis of the DQDB (IEEE 802.6) Network", *Proceedings IEEE INFOCOM'92*, Florence, May 6-8, 1992, pp. 527-535.
- [34] M. J. Johnson, "Proof that Timing Requirements of the FDDI Token Ring Protocol are Satisfied", *IEEE Transactions Communications*, Vol. COM-35, June 1987, pp. 620-625.
- [35] L. V. Kantorovich, V. I. Krylov "Approximate Methods of Higher Analysis", P. Noordhoff Ltd, Groningen, The Netherlands, 1958.
- [36] L. Kleinrock *Queueing Systems*, Vol. 1 (1975), Vol. 2 (1976), Wiley, New York.
- [37] J. F. Kurose, M. Schwartz, Y. Yemini, "Multiple Access Protocols and Time-constrained Communication", *ACM Computing Surveys*, 16, (1), pp. 43-70.
- [38] R. O. LaMaire, "An M/G/1 Vacation Model of an FDDI Station", *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 2, February 1991, pp. 257-264.
- [39] R. O. LaMaire, "M/G/1/N Vacation Model with Varying E-limited Service Discipline", *Queueing Systems* 11(1992)357-375.
- [40] R. Landry, I. Stavrakakis, "A Three-Priority Queueing Policy with Applications to Communication Networks", *Proceedings IEEE INFOCOM'93*, San Francisco, 1993, pp. 1067-1074.
- [41] T. T. Lee, "M/G/1/N Queue with Vacation Time and Limited Service Discipline", *Performance Evaluation*, Vol. 9, pp. 181-190, June 1989.
- [42] K.K. Leung, "Cyclic-service Systems with Probabilistically-limited Service Discipline", *IEEE Journal on Selected Areas in Communications*, SAC-9(1991), pp. 185-193.
- [43] P. Martini, "Fairness Issue of the DQDB Protocol", *Proceedings IEEE 4th Conference on Local Computer Networks*, Minneapolis, MN, Oct. 1989, pp. 160-170.
- [44] S. Mirchandani, R. Khanna (editors), "FDDI Technology and Applications", Wiley, New York, 1993.
- [45] P. Montuschi, A. Valenzano, L.Ciminiera, "On the Equivalence of IEEE 802.4

- and FDDI Timed Token Protocols", Proceedings IEEE INFOCOM'91, Miami, 1991, pp. 435-440.
- [46] B. Mukherjee and S. Banerjee, "Alternative Strategies for Improving the Fairness in and an Analytical Model of the DQDB Network", *IEEE Transactions on Computers*, Vol. 42, N. 2, February 1993, pp. 151-167.
- [47] B. Mukherjee, C. Bisdikian, "A Journey Through the DQDB Network Literature", *Performance Evaluation*, Vol. 16, Nos. 1-3, 1992, pp. 159-176.
- [48] K. Nakumara, T. Takine, Y. Takahashi, and T. Hasegawa, "An analysis of an Asymmetric Polling Model with Cycle-time Constraint", NATO Advanced Research Workshop, Sophia Antipolis, France, June 1990.
- [49] M. F. Neuts, "Structured Stochastic Matrices of M/G/1 Type and their Applications", Marcel Dekker, Inc., 1989.
- [50] P. G. Potter and M. Zukerman, "Analysis of a Discrete Multipriority Queuing System Involving a Central Shared Processor Serving Many Local Queues", *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 2, February 1991, pp. 194-202.
- [51] R. Reardon (editor), "FDDI Networking", IBC Technical Services Ltd, 1992.
- [52] I. Rubin, Z. H. Tsai, "Message delay Analysis of multiclass priority TDMA, FDMA, and discrete-time queueing systems", *IEEE Transactions Information Theory*, Vol. 25, No. 3, May 1989.
- [53] I. Rubin, and J. C.-H. Wu, "Analysis of an M/G/1/N Queue with Vacations and its Application to FDDI Asynchronous Timed-Token Service System", Proceedings IEEE GLOBECOM'92, December 6-9, 1992, Orlando, Florida, pp. 1630-1634.
- [54] A. Schill, M. Zieher, "Performance analysis of the FDDI 100 Mbit/s Optical Token Ring", Proceedings IFIPTC6/WG 6.4 High Speed Local Area Networks (HSLAN), Aachen, Germany, February 1987, pp. 53-74.
- [55] M. Schwartz, "Telecommunication Networks", Addison-Wesley, 1987.
- [56] K. C. Sevcik and M. J. Johnson, "Cycle Time Properties of the FDDI Token Ring Protocol", *IEEE Transactions on Software Engineering*, Vol. SE-13, pp. 376-385, March 1987.
- [57] W. Stallings, "Local Networks", Macmillian, 1984.
- [58] I. Stavrakakis, S. Tsakiridou, "Occupancy Distribution for a DQDB Station based on a Queueing System with Markov-Structured Service Requirements", Proceedings IEEE INFOCOM'93, San Francisco, 1993, pp. 1083-1090.
- [59] H. Takagi, "Analysis of Polling Systems", The MIT Press, 1986.
- [60] H. Takagi, "Effects of the Target Token Rotation Time on the Performance of a Timed-Token Protocol", Proceeding Performance 90, Edinburgh, Scotland, September 1990, pp. 363-370.
- [61] H. Takagi, "Queueing Analysis", Volume 1: Vacation and Priority Systems, Part 1, North-Holland, 1991.
- [62] M. Tangemann, and K. Sauer, "Performance Analysis of the Timed Token Protocol of FDDI and FDDI-II", *IEEE Journal of Selected Areas in Communications*, Vol. 9, No. 2, February 1991, pp. 271-278.
- [63] P. Tran-Gia, T. Raith, "Performance Analysis of Finite Capacity Polling Systems with Nonexhaustive Service", *Performance Evaluation* 9 (1988), pp. 1-16.
- [64] P. Tran-Gia, Th. Stock, "Approximate Performance of the DQDB Access Protocol", *Computer Networks and ISDN Systems*, Vol. 20 (1990), No. 1-5, December 1990, pp. 231-240.
- [65] H. R. van As, J. W. Wong, P. Zafiropulo: "Fairness, Priority and Predictability of the DQDB MAC Protocol Under Heavy Load", Proceeding 1990 International Zurich Seminar, Zurich, March 1990, pp. 410-417.
- [66] M. Kabatepe, K. S. Vastola, "Exact and Approximate Analysis of DQDB Under Heavy Load", Proceedings INFOCOM'92, Florence, Italy, May 1992.

- [67] W. Whitt, "The Queueing Network Analyzer", BSTJ, Vol. 62, No. 9, November 1983, pp. 2779-2847.
- [68] R. W. Wolff, "Poisson Arrivals See Time Averages", *Operation Research*, Vol. 30, March 1982, pp. 223-231.
- [69] J. W. Wong, "Throughput of DQDB Networks Under Heavy Load", Proceedings EFOC/LAN'89, Amsterdam, The Netherlands, June 1989.
- [70] M. Zukerman, "QPSX - The Effect of Circuit Allocation on Segment Capacity Under Burst Switching", Proceedings IEEE ICC'88, Philadelphia, PA, June 1988, pp. 599-603.
- [71] M. Zukerman, "Queueing Performance of QPSX," Proceedings 12th ITC, Torino, Italy, June 1988, paper 2.2B.6.
- [72] M. Zukerman, "Overload Control of the Isochronous Traffic in QPSX", Proceedings IEEE GLOBECOM'88, Hollywood, FL, Nov. 1988, pp. 1241-1245.