Markovian Analysis of DQDB MAC Protocol

Falko Bause
Lehrstuhl Informatik IV, Universität Dortmund,
Postfach 50 05 00,
44221 DORTMUND, Germany

Pieter Kritzinger
Data Network Architectures Laboratory,
University of Cape Town,
South Africa

Michael Sczittnick
Lehrstuhl Informatik IV, Universität Dortmund,
Postfach 50 05 00,
44221 DORTMUND, Germany

December 21, 1993

Abstract

The Distributed Queue Dual Bus (DQDB) protocol or IEEE 802.6 has been accepted as the international standard for Metropolitan Area Networks (MAN). This paper describes a combined M/G/1 and Markov model for the steady state performance analysis of a DQDB network. In the analysis each station is considered independently, but not in isolation, thus avoiding the computational complexity which would otherwise be introduced by a large number of stations. Because of this, the proposed model is not restricted to trivially small networks frequently found in other analytical models.

The model takes into account the relative position of a station on the network, the phase difference between the two buses as well as the relative availability of QA-slots. Comparisons with our own simulation studies show that the analytical results lie well within acceptable error limits in all but exceptional cases. The effect of various model parameters on network performance are also reported.

1 Introduction

The Distributed Queue Dual Bus (DQDB) protocol or IEEE 802.6 has been accepted as the international standard for Metropolitan Area Networks (MAN). Since it was first proposed, there have been several performance studies of DQDB networks. Most performance studies are, however, based upon simulation models, e.g., [3, 5, 6, 11] and only few analytical models are reported in the literature, e.g., [4, 9, 14] and then only for either special load situations or trivially small networks.

Potter and Zukerman[10], for instance, assume that there is no delay on the reverse bus and that all segments arrive at the end of a slot interval. The DQDB network is modelled by a multi-queue processor sharing model leading to accurate results under these assumptions.

Tran-Gia in turn[12] proposes a model of nested M/G/1 queues, where each station is represented by such a queue and service time at a station is influenced by the waiting time of the previous station.

Mukherjee and Banerjee[9] propose a Markov model which considers the entire network, with the consequence that only very small networks with 2 or 3 nodes can be analysed.

In this paper we present a Markov model which applies to all load conditions and any number of nodes and captures the relative position dependency amongst network stations in DQDB. The validation of the results against those obtained from simulation show the model to be of adequate accuracy.

2 DQDB Architecture

Figure 1 illustrated the DQDB architecture which consists of a pair of uni-directional buses connecting the stations. We will consider there to be N stations on the network but here is no limit to the number of stations which may be connected to the bus. The Head Station continuously generates frames every 125 μs and transmits these along the *forward bus*. Each frame is subdivided into slots of equal size and each slot in turn has a header containing several fields including three priority request bits. The *End Station* terminates the forward bus and removes all incoming frames and generates slots at the same transmission rate and of the same sort (cf. slot description below) the reverse bus.

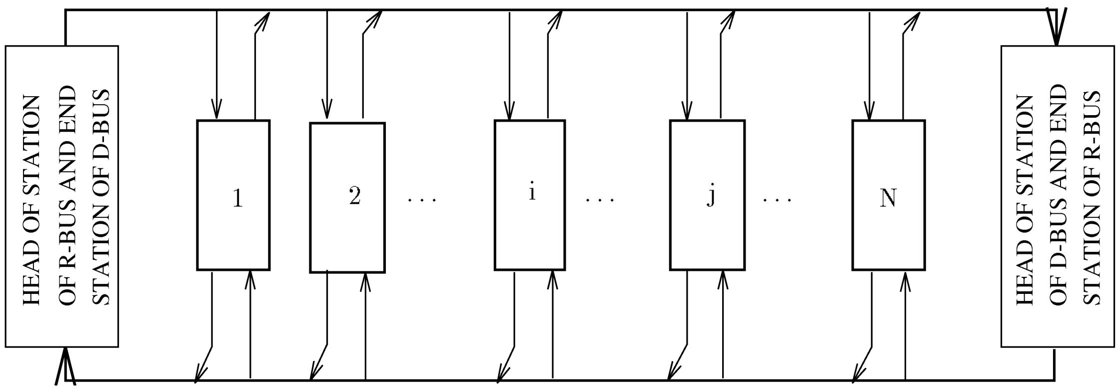

Figure 1: The DQDB architecture.

Referring to Fig. 1, if station $\{j\}$ wants to send data to station $\{i\}$ it would use the reverse bus while if $\{i\}$ wants to transfer data to $\{j\}$ it would use the forward bus.

Each slot on either bus can be allocated either to isochronous traffic (called *pre-arbitrated* slots or PA-slots) or non-isochronous traffic (called *queued arbitrated* slots or

QA-slots). Slots dedicated for isochronous traffic may not be used for non-isochronous data[1]. In the DQDB protocol described in the following, only non-isochronous traffic is considered. We moreover assume that a fraction α of the slots on the forward bus are available for queue arbitrated traffic.

Each header contains, amongst others, a BUSY bit and a REQUEST FIELD. The BUSY bit indicates whether the slot concerned contains information or is empty. The 3-bit REQUEST FIELD is intended for a 3-level priority scheme. The standard currently recognises only one priority level, however, and the model we present is for a single priority class. For completeness, our description of the DQDB protocol in the next section is for the general case.

3 The DQDB Bus Arbitration Algorithm

The DQDB or IEEE 802.6 standard is well-known. In this section we describe only the rules for accessing the bus in order to accurately relate it to our proposed Markov model. Fig. 2 on p. 5 illustrates the bus access process schematically.

In our analyses we consider *data segments* to travel from station $\{i,\ i = 1, \ldots, N\}$ to station $\{j\ j = i, i-1, \ldots, 0\}$ (cf. Figure 1) along the *forward bus*. Similarly we refer to *requests* as moving from station $\{j\}$ to station $\{i\}$ along the *reverse bus*.

We do not include the Bandwidth Balancing Mechanism (BWB) option in IEEE 802.6 in our model, mainly because we wanted to show that our analytical model can model the fact that the bandwidth a station receives is a function of its position on the bus. When BWB is introduced, the same Markovian analysis approach can be applied to analyze the effect of this on network performance. Slot re-use is also not considered.

The times of arrival at station $\{i\}$ of slots on the forward bus and the reverse bus respectively, are not synchronised and the operations described below are performed in the order in which slots arrive on either bus.

Each local queue at any station $\{i\}$ can be in either one of two states: the *idle* state or the *countdown* state. A local queue enters the countdown state each time a request is queued, and the idle state when the corresponding data segment has been transmitted. If a local queue is not empty it is considered to be in the countdown state except for very short time intervals between starting the segment transmission and queueing the next request.

The following events and corresponding actions are possible:

An exogenous priority j segment $s_i(j)$ arrives at station $\{i\}$ for transmission on the forward bus:

1. If $D_i(j)$ is in the *idle* state and no request of an data segment, already dispatched, is waiting: Set
$$CD_i(j) \leftarrow R_i(j), \quad R_i(j) \leftarrow 0$$
and $D_i(j)$ enters the countdown state.
2. If $D_i(j)$ is in the countdown state: $s_i(j)$ has to wait in $D_i(j)$.

[1] The request field of any slot, however, can be used for setting a request

A priority r request arrives on the reverse bus:

3. If $D_i(j)$ is in the *idle* state
   - if r is of the same or a higher priority, increase the priority j request counter for the forward bus at *this* station by 1; i.e., $R_i(j) \leftarrow R_i(j) + 1$;
   - otherwise do nothing;
4. If $D_i(j)$ is in the countdown state:
   - if r is of a higher priority, $CD_i(j) \leftarrow CD_i(j) + 1$;
   - if r is of the same priority, $R_i(j) \leftarrow R_i(j) + 1$;
   - otherwise do nothing.

An empty slot arrives on the forward bus:

5. If $D_i(j)$ is in the *idle* state:
   - If $R_i(j) > 0$, then note the fact that, an outstanding request at priority level $\{j\}$ at some downstream station will be served by this request, by decreasing $R_i(j)$ by 1;
   - otherwise do nothing.
6. If $D_i(j)$ is in the countdown state:
   - if $CD_i(j) > 0$, this station is not allowed to seize the empty slot for $D_i(j)$ and $CD_i(j)$ is decreased by 1 if the empty slot is allowed to pass;
   - if $CD_i(j) = 0$, access the forward bus and transmit segment $s_i(j)$ and enter the idle state.

4 Analytical Model

The schematic diagramme in Fig. 2 illustrates the service process for segments at any station i in the network. Segments arrive at the local queue, a single request and its corresponding segment enters the request queue and distributed queue respectively, and are served by the reverse bus and forward bus respectively in time τ_i. The delay time T_i includes the segment waiting time in the local queue which therefor reflects the delay a user with a packet comprising several segments may experience.

The performance measures we use in our analyses are the bus access time and the segment delay time. These we define as follows:

1. *The Mean Bus Access time, $\overline{\tau_i}$ at station $\{i\}$.* This is the time elapsed from the moment a request is queued at the station until the instant *both* the request and the corresponding segment have left. This time does not include the propagation delay on the bus nor the waiting time in the local queue at $\{i\}$.
2. *The Mean Segment Delay time, $\overline{T_i}$ at station $\{i\}$.* This is the time elapsed from the moment a segment arrives at the local queue until the instant it and its corresponding request have left station $\{i\}$. It therefore includes the queueing time of a segment at a station and therefore is a good representation of the delay a multi-segment packet may typically experience at a station.

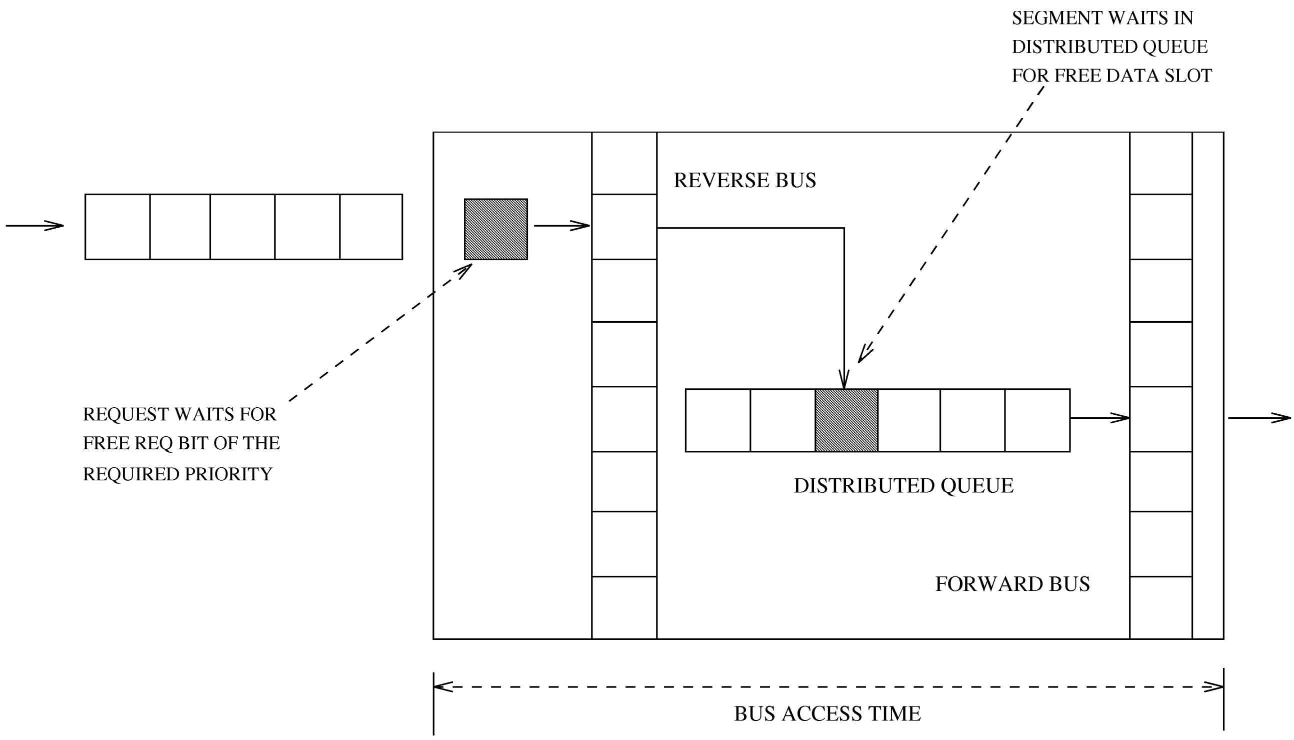

Figure 2: Schematic of segment service process at station i.

For our analysis we will assume that time is discrete and measured in units of δ, the interarrival time of slots at a station and that distance is measured in slot lengths. Since we only consider one priority class the parameter j is dropped in the following discussion. We moreover assume that,

1. the arrival process of segments at the local queue at station $\{i\}$ is Poisson with parameter λ_i, and
2. the slot occupancy pattern is determined by a Bernoulli process, and
3. a station has infinite buffer capacity, and
4. the segmentation and reassembly time of messages is negligible.

Recently, Conti *et al.*[2] used a discrete Markov process to model the slot occupancy pattern on the forward bus rather than assume it to be a Bernouli model as we do. This may well be a better approximation but, our simpler assumption does not detract from the validity of the proposed analytical approach. We show in Sec. 5 that our model already yields adequate results. If one so wishes, a Markov arrival process model can be used with our proposed method without difficulty.

4.1 M/G/1 model of the segment service process

Since segment arrivals (or equivalently, request arrivals[2]) to the local queue are assumed to be Poisson, we view the combination of the local queue and service process in Fig. 2 as an M/G/1 queue. In that case we use the well known Kolmogorov-Chapman equation which relates the service and turnaround time:

$$\overline{T_i} \;=\; \overline{\tau_i} + \frac{\lambda_i \overline{\tau_i^2}}{2(1-\rho_i)} \tag{1}$$

where $\rho_i = \overline{\tau_i}\lambda_i$, as always, and $\overline{\tau_i^2}$ is the second moment of the bus access time.

In order to compute the moments of the bus access time we derive an ergodic Markov chain model of the bus access process. By modifying this model appropriately to create an absorbing Markov chain, and by using an appropriate starting distribution of that chain, we compute the required first and second moments [8].

4.2 Markov model of the Bus Access Time

In order to understand the description of the Markov chain in this section it is important to fully understand all the sequences of events concerning the arrival and transmission of requests and segments. A station cannot transmit a segment prior to queueing (as opposed to transmitting) a request for that segment. Refering to Fig. 2, once a request has been queued, the request (segment) will be sent independent of whether or not the corresponding segment (request) has already been transmitted. No request for a new segment transmission can be queued at a station, before the previous segment and its corresponding request have been transmitted. Hence the definition of the bus access time as the time elapsed from the instant a request arrives until both the request and its corresponding segment have left the station.

All events at a station $\{i\}$ and their corresponding probabilities in our analysis are assumed to be synchronised to the *start of* the slot on the *reverse bus*. There may be a phase difference, however, between the forward and reverse buses which will vary from station to station, depending on the location of the station on the bus. In our analysis we assume that a data slot arrives on the forward bus at station $\{i\}$ a fraction β_i; $0 \leq \beta_i \leq 1$ of a slot length after the arrival of the request slot on the reverse bus. This implies that during one slot interval, it is possible that a segment may arrive, its request be queued and (possibly) transmitted during the same slot interval if a free request bit becomes available. The corresponding segment in turn, may have to wait, or may also be transmitted immediately during the same slot if the data slot arriving a fraction β_i of time later is free. If $\beta_i = 0$ the request and the segment can obviously not leave in the same slot time.

Under heavy load, it is more likely, however, that an arriving segment will find itself at the back of the distributed queue. Let $p_{i,j}; j = 1, \ldots, k_i$ be the probability that a new segment for transmission on the forward bus will enter position j in the distributed queue. The quantity k_i is the maximum length of the distributed queue seen by station $\{i\}$. This value depends upon the following:

- The position of station $\{i\}$ relative to other stations on the reverse bus. This is clearly so since station $\{i\}$ can receive requests from all stations $(1, \ldots, i-1)$ downstream.
- The distance in slot units from the Head Station of the reverse bus to station $\{i\}$. The longer this distance, the more likely it is that downstream stations will generate multiple requests per station due to the effect of unfair data slot use.

Note that β_i and k_i allow us to model this effect of unfair slot usage.

Define the following probabilities for any station $\{i\}$ at steady state:

$$FR_i = \mathcal{P}\{\text{the slot on the reverse bus is free}\}$$
$$FD_i = \mathcal{P}\{\text{the next slot on the forward bus is free}\}$$

Let (u, v) be the state descriptor of the Markov chain which describes the events discussed above, and let

- $u = 1$ if the request r_i is waiting in Q_i and $u = 0$, otherwise.
- v where $v = 1, 2, \ldots, k_i$ is the position of s_i in the distributed queue. $v = 0$ indicates that no segment is queued for transmission.

The resultant state transition diagramme is illustrated in Fig. 3.

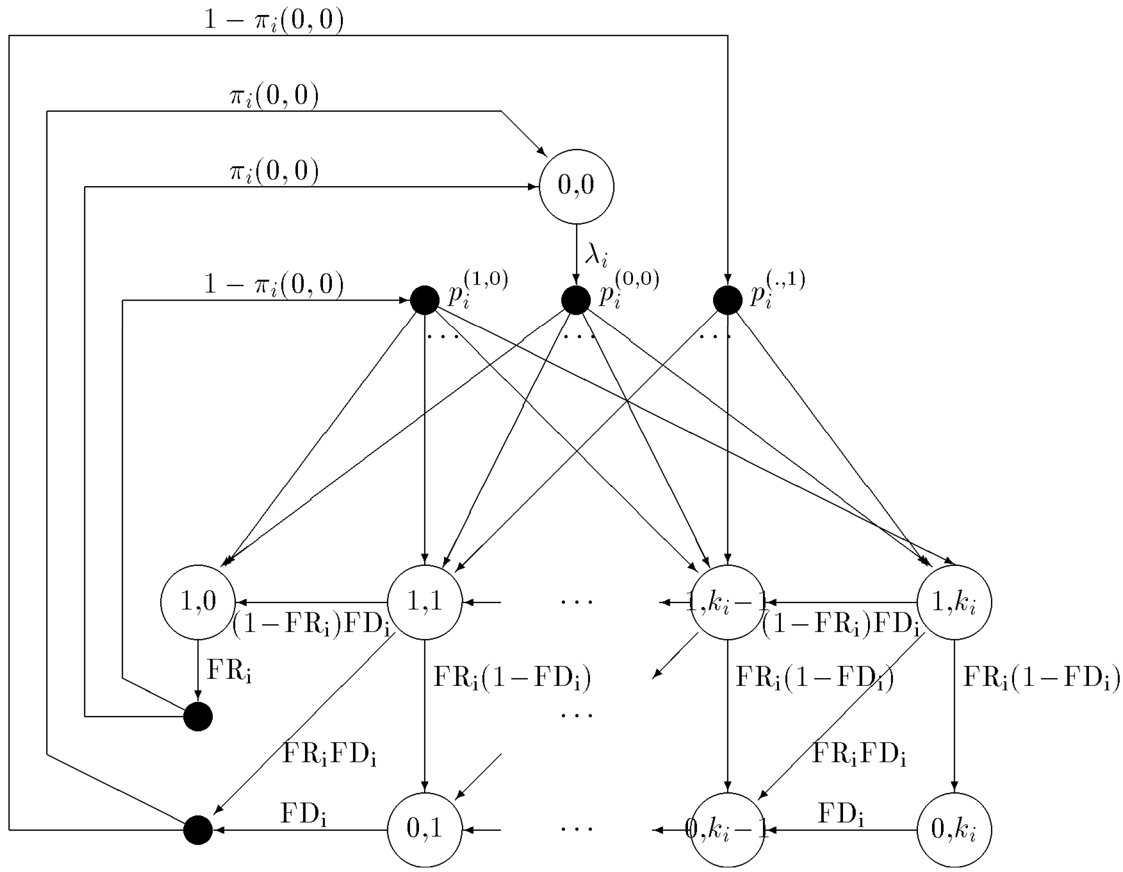

Figure 3: State transition diagramme of the Markov model at station $\{i\}$.

The various probabilities $p_{i,j}^{(.,1)}$, $p_{i,j}^{(0,0)}$ and $p_{i,j}^{(1,0)}$ where $j = 0, \ldots, k_i$ is the position of the segment in the distributed queue, are given by formulas in Fig. 4

For $p^{(1,0)}$ the probability $1 - \pi_i(0,0)$ ($\pi_i(0,0)$ is the probability of an idele station) is used as an approximation for the probability that there is another segment waiting in the local queue, which may use the current data slot. The probabilities FR_i and FD_i are as defined above.

Each $p_{i,j}^{(u,v)}$ specifies the probability of entering state $(1, j)$ from the upper three black printed dots in Fig. 3. The black printed dots are the entry and exit points of an absorbing

$$p_{i,j}^{(\cdot,1)} = \begin{cases} 0 & j=0 \\ p_{i,j} & 0<j\le k_i \end{cases} \quad (2)$$

$$p_{i,j}^{(0,0)} = \begin{cases} \beta_i FD_i p_{i,1} & j=0 \\ (1-\beta_i FD_i)p_{i,j} + \beta_i FD_i p_{i,j+1} & 0<j<k_i-1 \\ (1-\beta_i FD_i)p_{i,k_i} & j=k_i \end{cases} \quad (3)$$

$$p_{i,j}^{(1,0)} = \begin{cases} (1-\pi_i(0,0))FD_i p_{i,1} + \pi_i(0,0)\beta_i FD_i p_{i,1} & j=0 \\ (1-\pi_i(0,0))((1-FD_i)p_{i,j} + FD_i p_{i,j+1}) + \\ \quad \pi_i(0,0)((1-\beta_i FD_i)p_{i,j} + \beta_i FD_i p_{i,j+1}) & 0<j<k_i \\ (1-\pi_i(0,0))(1-FD_i)p_{i,k_i} + \pi_i(0,0)(1-\beta_i FD_i)p_{i,k_i} & j=k_i \end{cases} \quad (4)$$

Figure 4: Transition probabilities of the Markov chain

Markov chain for calculation of the first two moments of the bus access time. The upper index (u,v) describes the states from where state $(1,j)$ is entered.

$p_{i,j}^{(\cdot,1)}$ This is the probability of entering state $(1,j)$ just after leaving state $(1,1)$ or $(0,1)$ and assuming there is another segment waiting for transmission (with probability $1-\pi_i(0,0)$). In this case $p_{i,j}$ describes the probability of entering position j of the distributed queue (calculated in section 4.2.2).

$p_{i,j}^{(0,0)}$ If the local queue is empty, indicated by state $(0,0)$, and a new segment arrives at station $\{i\}$ (with probability λ_i), this segment can be transmitted in the same slot interval provided the data slot is free (probability FD_i) and station $\{i\}$ assumes to be in head of the distributed queue (probability $p_{i,1}$) and has the opportunity to see the beginning of the data slot, which surely depends on the point of arrival with respect to δ. Now, an arriving segment will see this data slot with probability β_i, because arrivals can happen at any point in time. Thus the probability for entering state $(1,0)$ is $\lambda_i\beta_i FD_i p_{i,1}$. Similar arguments hold for determining the probabilities $p_{i,j}^{(0,0)}$ for $j>0$. If a new segment arrives after recognition of the beginning of a free data slot (probability $1-\beta_i FD_i$), it will enter position j given by its request counter. If the segment arrives early, thus seeing the beginning of a data slot, it will decrease the countdown counter provided the data slot is free. This all will happen with probability $\beta_i FD_i$. Thus it will enter position j of the distributed queue, although the request counter contains $j+1$ at the point of queueing the segment.

$p_{i,j}^{(1,0)}$ These probabilities are little bit more tricky. Leaving state $(1,0)$, there is a possibility of a segment arrival in the time interval δ.[2] The probability of this case happening can be approximated by $\pi_i(0,0)$. This newly arriving segment will see a free data slot with probability $\beta_i FD_i$, thus entering position j of the distributed queue provided the value of the request counter was $j+1$. If the request counter was 1 the newly arrived segment is transmitted in the considered interval causing a re-entrance to $(1,0)$. With the complementary probability $(1-\pi_i(0,0))$ an already waiting segment will surely see the data slot and use it with probability FD_i.

If the request counter is greater than 1 ($j>0$) a new arriving segment (probability

[2]Note that all states describe the situation at station $\{i\}$ at the beginning of a slot arrival on the reverse bus.

$\pi_i(0,0)$) moves forward in the distributed queue with probability $\beta_i FD_i$ or stay in its position given by the request counter at the point of arrival with probability $1-\beta_i FD_i$. In the other case (probability $1-\pi_i(0,0)$) the following segment recognizes the data slot, thus moving forward in the distributed queue with probability FD_i or remaining in its position with probability $1 - FD_i$.

The probabilities of leaving a state (u,v) are straightforward. In a state $(1,j), j > 0$, a request can be transmitted and/or the segment advances in the distributed queue. E.g. the probability that these events both occur in the same interval δ is given by $FR_i FD_i$.

The transition probabilities of the complete Markov chain are given by multiplication of the above described probabilities attached to the arc from a state to its successor state.

Note that since the transition probabilities from states $(1,0)$, $(1,1)$ and $(0,1)$ depend upon $\pi_i(0,0)$, we have to use an iterative solution method for the steady state distribution of the Markov chain.

The probabilities FR_i and FD_i as well as the probabilities $p_{i,j};\ j = 1,\ldots,k_i$ at every station $\{i\}$ are parameters of the Markov model. We compute those next.

4.2.1 The probabilities FR_i and FD_i

Since we assume ergodicity the mean number of segments arriving at station $\{i\}$ (per unit time) must equal the mean number of served (transmitted) requests or, equivalently, the mean number of served (transmitted) segments on the forward bus.

Thus station $\{i\}$ will receive the service capacity of station $\{i-1\}$ on the reversebus minus the mean number of served requests at station $\{i-1\}$. In other words,

$$FR_i = FR_{i-1} - \lambda_{i-1},\ i = 2,\ldots,N \tag{5}$$

One can also prove this from the fact that

$$FR_i = FR_{i-1}[1 - \sum_{j=0}^{k_i} \pi_{i-1}(1,j)],\ i = 2,\ldots,N.$$

Note that $FR_1 = 1$.

A similar argument holds for the probability FD_i:

$$FD_i = FD_{i+1} - \lambda_{i+1},\ i = 1,\ldots,N-1 \tag{6}$$

and $FD_N = \alpha$ is the fraction of slots dedicated to QA-traffic.

4.2.2 The probabilities $p_{i,j}$

The probabilities $p_{i,j},\ \ j = 1,2,\ldots,k_i$ can be computed from the Markov chain illustrated in Fig. 5 where the state space in that case is the combined value of the countdown and request counter $CD_i + R_i + 1$ at station $\{i\}$. Again, k_i is the maximum length of the distributed queue seen at station $\{i\}$. In Fig. 5 we have written $P = (1 - FR_i)(1 - FD_i)$ and $Q = FD_i FR_i$.

The steady state probability $p_{i,j}$ of a state j in this Markov chain is an approximation of the probability that a segment at the time queueing its request will be in position j of the distributed queue with respect to its local information.

The states of the Markov chain describe the sum of request and countdown counter, because this sum will change independently of a segment being queued at station $\{i\}$ or

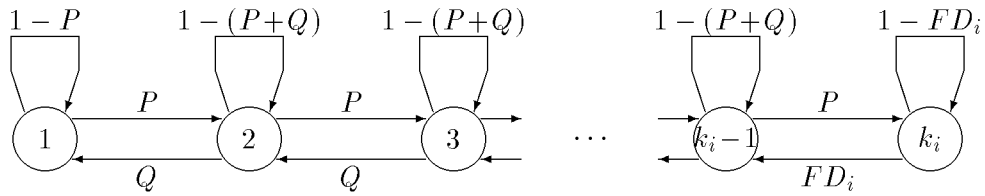

Figure 5: Markov chain model of countdown and request counter at station $\{i\}$.

not. Upon queueing the request of a new segment CD_i is zero and state j indicates the position of this segment in the distributed queue.

The transition probabilities of the Markov chain are determined by the DQDB protocol. If an occupied request slot and no free data slot arrive at station $\{i\}$ (with probability $(1 - FR_i)(1 - FD_i)$) the sum increases and in case of arrival of a free request slot and a free data slot, this sum decreases (with probability $FR_i FD_i$).

The steady state distribution $p_{i,j}$ can be easily calculated and is given by:

if $k_i \geq 3$**:**

$$p_{i,j} = \begin{cases} F_i^{j-1} p_{i,1}, & j = 1, \ldots, k_i - 1 \\ FR_i F_i^{j-1} p_{i,1}, & j = k_i \end{cases} \tag{7}$$

where

$$p_{i,1} = \frac{1 - F_i}{1 + \left(\frac{FR_i - 1}{FD_i}\right) F_i^{k_i}} \tag{8}$$

with

$$F_i = \frac{(1 - FR_i)(1 - FD_i)}{FD_i FR_i}. \tag{9}$$

if $k_i = 2$**:**

$$p_{i,1} = \frac{FD_i}{FD_i + P}, \; p_{i,2} = \frac{P}{FD_i + P}$$

if $k_i = 1$**:** $p_{i,1} = 1$.

The only parameter of the Markov chain still to be determined is k_i. In Sec. 4.2 we have mentioned that this parameter depends on the location of a station, given by its index, and on the length of the bus measured in slot units.

If we consider large networks where the distance between stations is several slots, which is a realistic assumption, we can practically determine k_i by the following heuristic. In Eq. (8) k_i influences the expression $F_i^{k_i}$ and thus the probabilities $p_{i,1}$ and $p_{i,j}$ respectively. If $F_i^{k_i}$ gets very small, which is eventually the case for $F_i < 1$, the probability p_{i,k_i} is negligible and a further increase of k_i will not change the steady state distribution significantly. So k_i can be calculated from $F_i^{k_i} < \epsilon$ for small ϵ, if $F_i < 1$. Now $F_i < 1 \iff FR_i + FD_i > 1$. Eq. (5) implies

$$FR_i = FR_1 - \sum_{k=1}^{i-1} \lambda_k = 1 - \sum_{k=1}^{i-1} \lambda_k$$

and analogously from Eq. (6) we get

$$FD_i = FD_N - \sum_{k=i+1}^{N} \lambda_k = \alpha - \sum_{k=i+1}^{N} \lambda_k$$

Thus

$$FR_i + FD_i = (1+\alpha) - \sum_{\substack{k=1 \\ k \neq i}}^{N} \lambda_k$$

In steady state $FD_1 > 0$ holds, which implies $\alpha > \sum_{k=2}^{N} \lambda_k$ and so

$$\begin{aligned} FR_i + FD_i &> 1 + \sum_{k=2}^{N} \lambda_k - \sum_{\substack{k=1 \\ k \neq i}}^{N} \lambda_k \\ &= \begin{cases} 1 & : \quad i = 1 \\ 1 + (\lambda_i - \lambda_1) & : \quad i \neq 1 \end{cases} \end{aligned}$$

Now $(\lambda_i - \lambda_1) > 0$ is a reasonable assumption, because station i $(i > 1)$ will naturally send more segments on the forward bus than station 1, because it has more possible receivers in that direction.

Therefore k_i can be calculated from $F_i^{k_i} < \epsilon$ giving $k_i > \frac{\log(\epsilon)}{\log(F_i)}$. So choosing

$$k_i = \frac{\log(\epsilon)}{\log(F_i)} + 1$$

will lead to reasonable results for large networks.

5 Results

Unless indicated otherwise, we used a uniform network load in our experiments, given by

$$\lambda_i = \frac{2i\lambda}{N(N+1)}. \tag{10}$$

Note that station $\{0\}$ does not contribute to load on the forward bus, for there is no station to communicate with.

All results are for the performance of a single bus only. Since the two buses are assumed to be identical (which the model does not insist upon) one can easily compute the effect of the combined traffic.

Note, as well, that the advantage of our analytical model is the fact that one can model a DQDB network with a very large number of stations. Results can be computed for any number of stations. In the following we give results for $N = 100$. It should be clear, however, that this choice is arbitrary.

5.1 Model validation

In order to validate our analytical model we developed a simulation of a DQDB network using the modelling tool HIT [1]. Simulation is expensive of computing time, and as is the case with most DQDB simulation models reported in the literature [3, 5, 6, 11], we had to limit the number of stations to a managable number of 39.

For the same reason we had to restrict the choice of parameter values and load profiles to a manageble number. For instance, we kept β constant at a value of 0.5 in all the experiments. Fig. 6 shows the results for a bus utilisation of 40% and 70% QA-slot avail-

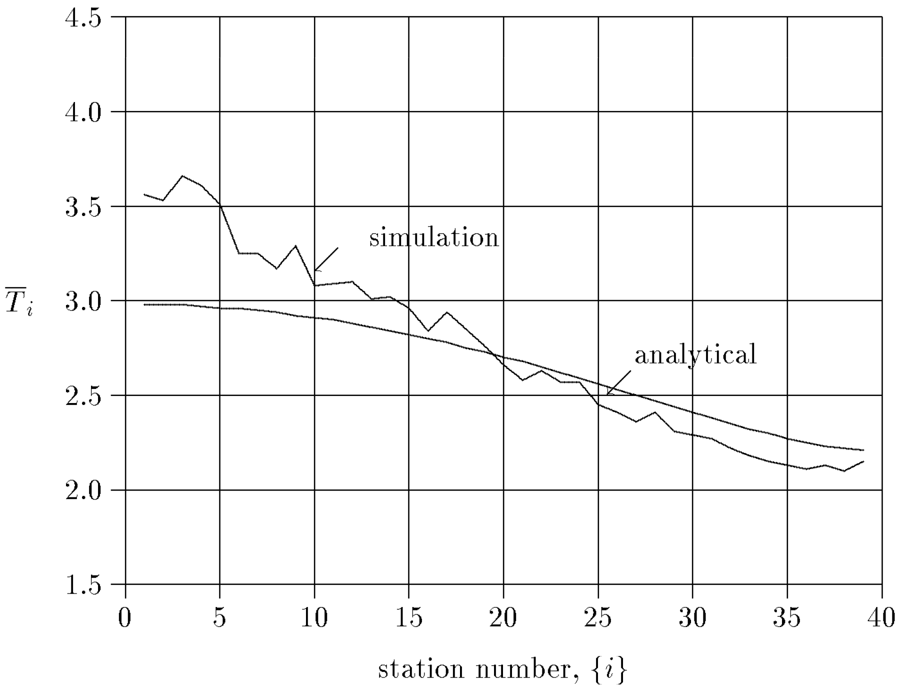

Figure 6: Model validation for bus a utilisation of 40% and 70% QA-slot availability.

ability. The analytical results reflect, at worst, an error of 18.6 percent of the simulated value at all stations. The mean absolute error, however, is 8.8% which, considering the assumptions of the analytical model, is very reasonable. Fig. 7 considers the case of a bus utilisation of 90% and 100% QA-slot availability. It is evident that for stations 1 through 29 the mean error of the simulated value is 6.8% and at worst 22 percent.

In the latter case, and for stations at the end of the bus, however, the analytical results underestimated the segment delay time. In Sec. 5.2 below we offer an explanation for this phenomenon.

In the next four sections we give the results of experiments where we considered various parameter values or load scenarios to illustrate typical uses of the analytical model.

5.2 Delay time as a function of bus utilisation

Fig. 8 illustrates the analytical results for 100 percent QA-slot availability and various bus utilizations. In each case the network traffic was as described by Eq. (10) above.

The segment delay time $\overline{T}_i$ increases with higher bus utilisation as is to be expected. For any one utilisation however, the delay time gradually decreases towards the end of the

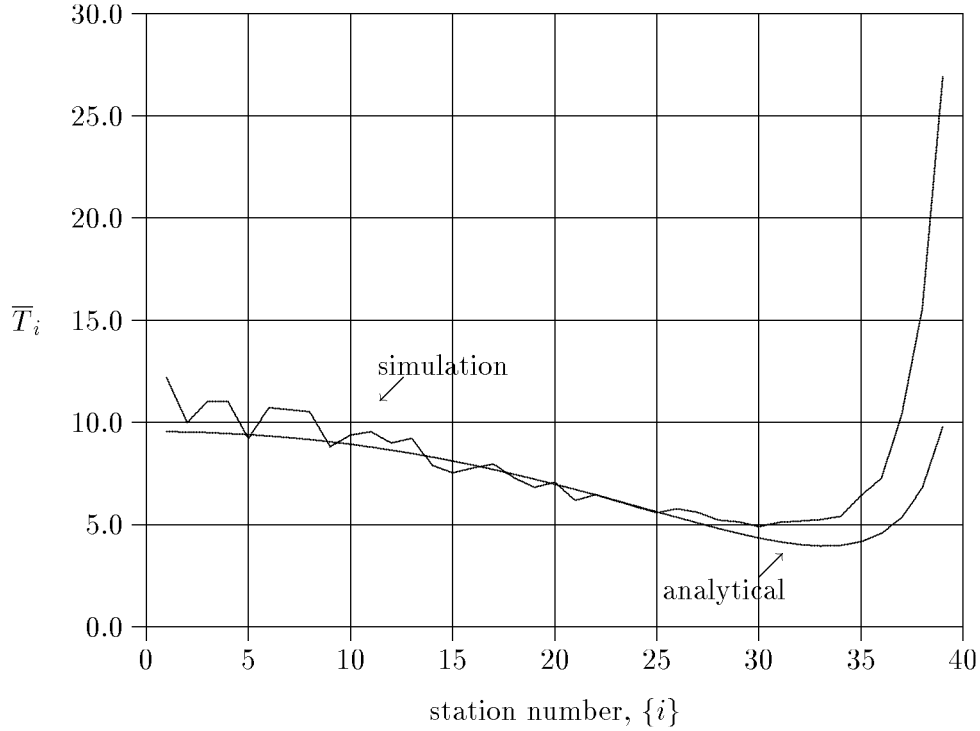

Figure 7: Model validation for bus a utilisation of 90% and 100% QA-slot availability.

bus and then increases sharpely. This phenonenom, confirmed by our various simulation experiments, is identical to that observed by Jacquet[7] for high utilisations.

This particular behaviour is explained by the fact that the probability that a segment which arrives at the distributed queue will discover many segments ahead of itself, is directly proportional to the value F_i [cf. Eq. (9)]. In Fig. 9 we have plotted this value F_i versus the station number $\{i\}$ for a bus utilisation of 90 percent and $\alpha = 1.0$..

A higher value of F_i due to increased λ and hence smaller values of FR_i and FD_i, implies that segments at station $\{i\}$ have longer to wait in the distributed queue. When F_i decreases sharply towards the end of the bus, however, the lower value of F_i is offset by the increased delay experienced by requests waiting for a free request bit, since near the end of the bus, $FR_i \ll 1$. This effect is accentuated by the load scenario used in our experiments, where stations near the end of the bus have the most segments to send.

5.3 Delay time as a function of QA-slot availability

We next tested the model for sensitivity towards the availability of QA-slots. The results for $\alpha = 1.00$, 0.925 and 0.910 are illustrated in Fig. 10.

From Fig. 10 it is clear that the effect of a lower QA-slot rate is to accentuate the characteristic behaviour of segment delay time as a function of station position on the bus. Comparing the curves of F_i in Fig. 9 for $\alpha = 0.910$ and $\alpha = 1.000$ we note that at any station $\{i\}$ for $i = 1, \ldots, 40$ (say), the value of F_i is much higher in the case of $\alpha = 0.910$ and segments are more likely find themselves in a longer distributed queue than would be the case for $\alpha = 1.000$. Towards the end of the bus the effect of a low value of FR_i is again dominant as mentioned above.

The authors have seen only one previous attempt [12], to include the effects of QA-slot

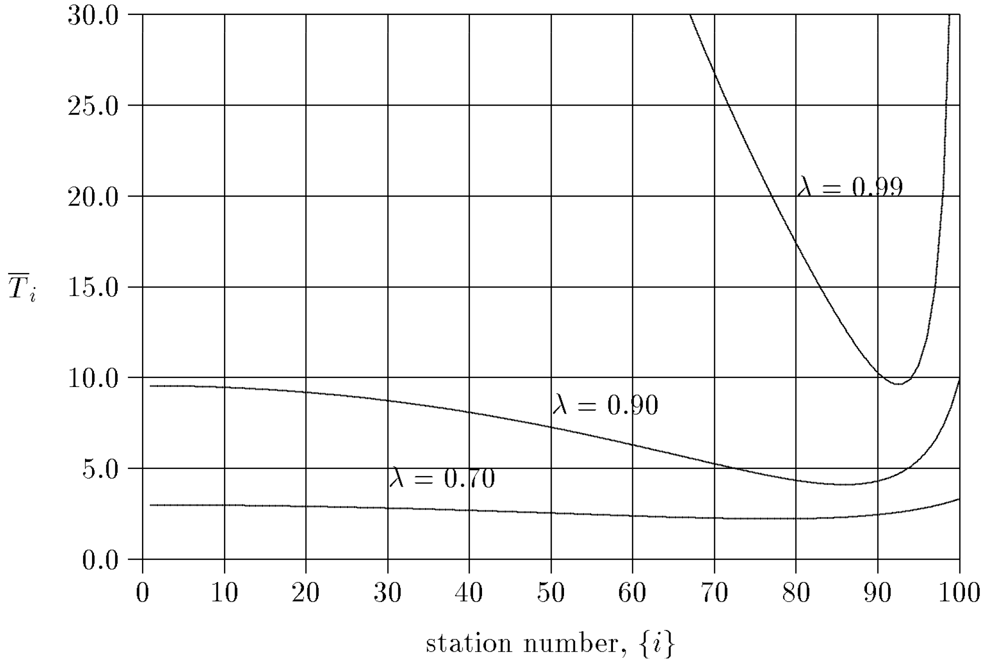

Figure 8: Effect of bus utilisations (λ) on $\overline{T}_i$.

availability on network behaviour in an analytical model.

5.4 Non-uniform network load

In Fig. 11 the effect of high traffic rates at one station (such as a file server) on the performance of the other stations on the network is illustrated. The traffic intensity was kept constant at all stations except at the one arbitrarily chosen station (number 15), where the arrival rate of segments was increased as illustrated. QA-slot availability was 100% and bus utilisation was chosen very high at 90%. The model accurately reflects the inevitable queue which develops downstream from a very busy station. By the time the end of the bus is reached however, the delays experienced by segments at those stations is as bad as that at the stations downstream from the one busy station.

6 Conclusions

The DQDB MAC protocol is deceptively simple and devising an analytical model that is computationally tractible and which can describe the operation of a DQDB network accurately has evaded researchers to date. In this paper, however, we describe and analyse such a model using Markovian analysis techniques. The model is computationally tractable and accounts for such network vagrancies as relative station position, QA-slot rate and relative phase difference between request bus and data bus.

An important discovery we made during the course of our experimentation, is that at high bus utilisations the relative phase shift between the forward and reverse buses are unimportant, even for very large networks. This result we verified by simulation for a 40 station network and analytically for network of up to a 100 stations.

Equally interesting is the effect of the availability rate of QA-slots on the forward

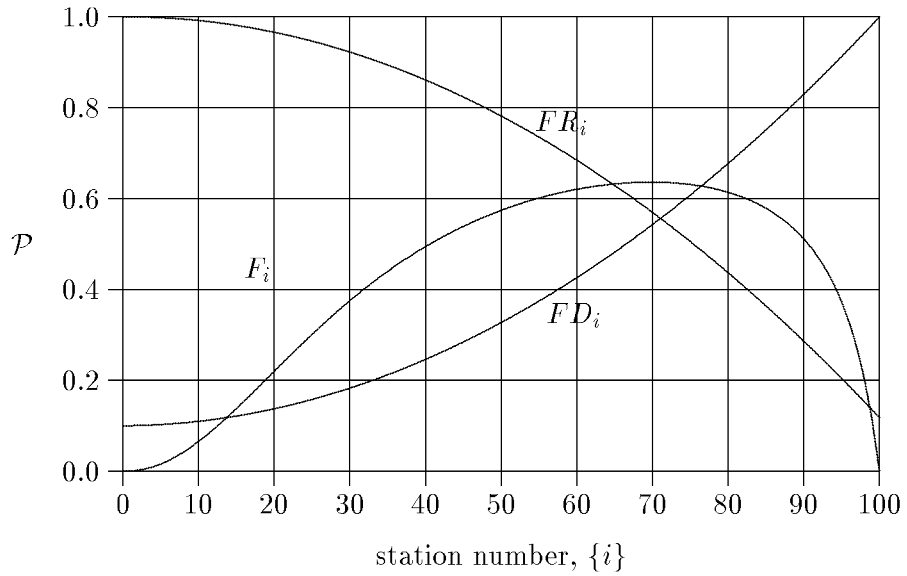

Figure 9: Explaining the observed performance of the DQDB network.

bus. Although it is clear that a lower slot availability will increase the delay times, the performance of individual stations along the bus would seem to be more sensitive to this parameter than to bus utilisation.

A distinct shortcoming of the model is the fact that the absolute position (in say, slot distances) of any one station from the Head Station or from each other are not represented. For large networks and under steady state analysis this omission would appear not to matter that much. Nevertheless, we believe the analysis techniques presented here may be adapted for such a model.

References

[1] H. Beilner et al. Towards a performance modelling environment: News on HIT. In R. Puigjaner, editor, *Proceedings of the 4th International Conference on Modelling Techniques and Tools*, 1988.

[2] M. Conti, E. Gregori, and L. Lenzini. On the approximation of the slot occupancy pattern in a DQDB network. *Performance Evaluation*, 16:159–176, 1992.

[3] B.T. Doshi and A.A. Fredericks. Visual modelling and analysis with application to the DQDB protocol. *Computer Networks and ISDN Systems*, 20:241–251, 1990.

[4] M.J. Ferguson. Towards formal structures for IEEE 802.6. In *Proceedings of NATO Advanced Research Workshop, Sophia Antipolis, France*, June 1990.

[5] L. Fratta et al. DQDB performance under multiplexed bursty traffic. In *Proceedings of IEEE Int. Conf. Commun. (ICC) '91, Denver, Colorado*, pages 1056–1061, June 1991.

[6] E.L. Hahne, A.K. Choudhury, and N.F. Maxemchuk. Improving the fairness of DQDB networks. In *IEEE INFOCOM '90*, pages 175–184, June 1990.

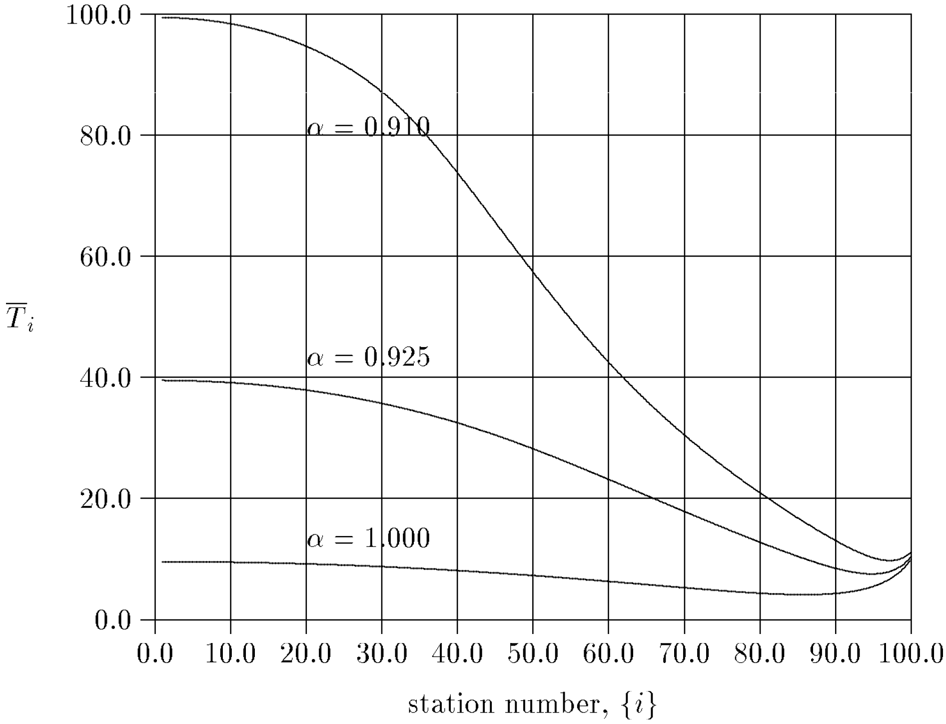

Figure 10: Effect of QA-slot availability (α) on $\overline{T}_i$.

[7] P. Jacquet. An analytical model for the high speed protocol QPSX. Technical Report 1141, Institut National de Recherche en Informatique et en Automatique, France, 1989.

[8] J.G. Kemeny and J.L. Snell. *Finite Markov Chains.* Springer-Verlag, New York, 1960.

[9] B. Mukherjee and S. Banerjee. Alternative strategies for fairness in an analytical model of DQDB networks. In *Proceedings of INFOCOM'91, Bal Harbour, Florida*, pages 879–687, 1991.

[10] P.G. Potter and M. Zukerman. A discrete Shared Processor model for DQDB. *Computer Networks and ISDN Systems*, 20:217–222, 1990.

[11] K. Sauer and W. Schodl. Performance aspects of the DQDB protocol. *Computer Networks and ISDN Systems*, 20:253–260, 1990.

[12] P. Tran-Gia and Th. Stock. Approximate performance analysis of the DQDB access protocol. *Computer Networks and ISDN Systems*, 20:231–240, 1990.

[13] D. Wikarski. Decomposition of stationary continuous-time markov chains - a new approach. *Elektronische Informationsverarbeitung und Kybernetik*, 20:355–373, 1984.

[14] J.W. Wong. Throughput of DQDB networks under heavy load. In *Proceedings of European Fibre Optics Communications and LAN Exposition*, pages 146–151, June 1989.

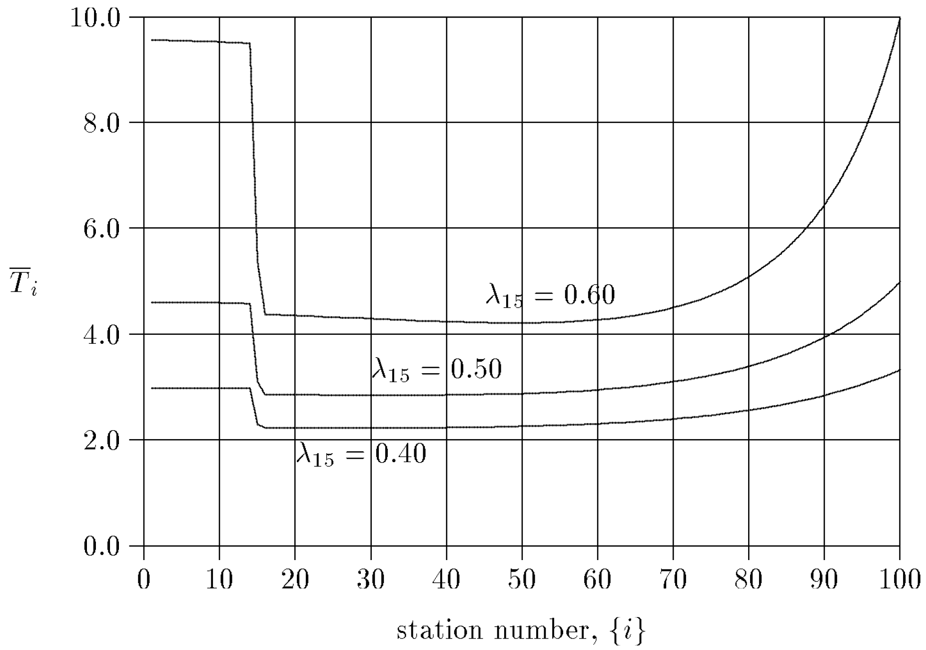

Figure 11: Effect of non-uniform load.