University of Alberta

Modeling DQDB : An Algorithmic Approach

by

Hazem Nassef ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Master of Science**.

Department of Computing Science

Edmonton, Alberta

Fall 1997

0-612-22645-X

University of Alberta

Library Release Form

Name of Author: Hazem Nassef

Title of Thesis: Modeling DQDB : An Algorithmic Approach

Degree: Master of Science

Year this Degree Granted: 1997

Hazem Nassef

Hazem Nassef

3 Mohy El Deen Abu El Ez

Dokki, Giza

Egypt

Date: 22 August 1997

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Modeling DQDB : An Algorithmic Approach** submitted by **Hazem Nassef** in partial fulfillment of the requirements for the degree of **Master of Science**.

. . . .

Dr. Janelle Harms

. . . .

Dr. Yanhong Wu

. . . .

Dr. Ehab El-Mallah

. . . .

Dr. Ahmed Kamal

Date: 21 August 1997

To the one who saved me and always protects and guides me.

Abstract

Distributed queueing is known to be a complex problem to model because of the high degree of interaction between many stochastic processes. This makes classical analytic techniques inadequate for modeling the problem. An algorithmic model is presented instead of an analytical one. Algorithmic probability is a methodological way of thinking. Instead of following a rigorous mathematical approach until the final solution is achieved, mathematical representation is used until the best point is reached for applying algorithms and data structures. So the modeling problem is transformed to algorithms that can be implemented as a software package. Algorithmic probability is also known as the linear algebraic approach for stochastic models and matrix analytic methods in stochastic models.

We present efficient algorithms in this thesis to compute performance measures for a network of DQDB stations. The algorithms capture details of the protocol that would make a classical solution intractable. A discrete time semi Markov process of the M/G/1 type is defined for a tagged node in the queue. The input processes to the node are assumed to be Poisson, general renewal and Bernoulli processes for the arrival of segments, empty slots and requests respectively. By computing the steady state probability distribution of the semi Markov process the buffer occupancy distribution is derived. An algorithm to compute the distribution of the renewal process at the output of the node is presented. This algorithm makes a network wide solution possible by computing the buffer occupancy of the first node, computing the output process which is the input to the second node, solving the second node and repeating the process until the last node is computed. Finally

an algorithm for the waiting time distribution is presented. Numerical examples are plotted and compared with simulation results.

Acknowledgements

This thesis is the result of efforts made by many people. I am greatly indebted to my parents, brother and sister for their continuous moral and financial support. The thesis is influenced by an older thesis that was not submitted due to a major car accident. Although I left research for several years, my father encouraged me to recover as he had a similar experience. He did his masters twice with a gap of several years due to problems out of his control. My mother provided me with love. care and huge financial support. There are no words that are enough to thank my brother Ashraf who left his Ph.D. at McMaster university for several months to look after me in the hospital after my accident. His continuous scientific support for me is greatly acknowledged.

I would like to thank Prof. Ahmed Kamal, my first supervisor who taught me stochastic models. for his continuous support. His support continued in the second thesis even after he left the University of Alberta. He provided me with the DQDB simulator and many technical hints. If any deep understanding I have for stochastic models, it is mainly attributed to him. I would like to thank Dr. Janelle Harms. my second thesis supervisor. She made everything possible after several years of absence. Although the second thesis was done while I was at ERI in Egypt, she made me feel that I was in Edmonton by continuous communications using international phone calls and emails. It was a unique experience in doing joint research remotely. Her suggestions are greatly acknowledged. Very special thanks goes to Prof. Randy Goebel, the graduate advisor at the time of my accident for making the Faculty of Graduate Studies waive my tuition fees until I recover and reregister. If not for this, I would not have been able to complete my thesis. I would like to thank Prof. Stan

Cabay, my biggest source of mathematical inspirations for lots of things that the space is not enough to mention. I would like to thank Dr. Ehab el Mallah for providing me with MATLAB and Linux. software and manuals, to prototype my models.

I would like to thank TR labs for the scholarship that they provided for the first thesis. Special thanks goes to my dear friend Mike McGregor, from TR labs, who visited me in the hospital.

I would like to thank Prof. Marcel Neuts of the University of Arizona, the founder of algorithmic probability, for his encouraging emails, the reprints of his work that he sent me and many other things. His humble emails made me feel that he is an old friend.

I would like to thank Prof. Nadia Hegazi the associate chair of ERI for her continuous support during my second thesis. Also I would like to thank Prof. Ehsan the chairman of the Informatics Department and Dr. M. Nour for their help and assistance. Special thanks goes to Prof. Essam El Diwany, the chairman of the microwaves department for his unbelievable support , encouragement and assistance. He provided me with lots of books, software and ideas related to computational mathematics. His students Ahmed, Hani, Khaled and Hesham provided me with a very good scientific atmosphere for discussions that are greatly acknowledged. Special thanks goes to Engineer Nayer Wanas of the Informatics Department for his technical support. I wish him all the luck at the University of Waterloo for his Ph.D. Many thanks goes to the talented Unix wizard Eng. Ibrahim Salama at the Informatics Department for help in debugging my code.

Contents

List of Figures 13

List of Tables 15

1 Introduction **1**

1.1 Objectives 2

1.2 Contributions and Contents 2

1.3 History of the Thesis 3

2 The Distributed Queue Dual Bus Network **5**

2.1 Introduction 5

2.2 The Dual Bus Architecture 6

2.3 The Distributed Queueing Protocol 8

2.4 Stochastic Dependence of the Protocol Parameters 9

2.4.1 The Busy Period 10

2.4.2 A FIFO Queue with Zero Inter-nodal Delay 10

2.4.3 The Effect of Inter-nodal Delay 12

2.4.4 The Effect of Correlated Requests 12

2.4.5 The Effect of Correlated Busy Slots 13

2.4.6 The Effect of Position Dependence 13

2.4.7 The coupled nature of the stochastic processes observed by a tagged node . . 14

2.5 Previous Models 14

2.5.1 Analytic Models assuming Bernoulli arrivals on the Forward Bus 15

2.5.2 Analytic Models Assuming Correlated arrivals on the Forward Bus 17

2.6 Thesis Model 20

3 A Markov Renewal Representation of a DQDB node 22

3.1 Introduction 22

3.2 An Overview of Markov Renewal Processes of M/G/1 Type 23

3.2.1 Levels and System states 23

3.2.2 Transition Probabilities 24

3.3 Characterization of the Bus Process 25

3.3.1 The Forward Bus 25

3.3.2 The Reverse Bus 28

3.4 A Markov Renewal model for a DQDB node 29

3.4.1 Transitions from Count-down States. (Level $i \geq 1$) 31

3.4.2 Transitions from Idle States (level 0) 33

3.5 Reduction of the Chain 37

3.5.1 Deletion of States 37

3.5.2 Relabeling States 38

3.6 Summary 39

4 Computation of the Output Process 40

4.1 Introduction 40

4.2 The Output Process 41

4.3 Approximate Algorithm 42

4.4 Exact Algorithm 46

4.5 Summary 48

5 Waiting Time Distribution **49**

5.1 Introduction 49

5.2 Transition probabilities for the residual life 50

5.2.1 Transitions from upper levels $i > 0$ 52

5.2.2 Transitions at the boundary level 52

5.2.3 A Markov chain to model transitions from the idle state 53

5.2.4 The transition matrices $\mathbf{R}'_T(i, j, k, y)$ and $\mathbf{R}'_I(i, j, k, y)$. 55

5.2.5 The matrix $\mathbf{R}_T(0, j, k, x, y)$ 56

5.3 Transition probabilities for the total countdown time 58

5.4 Waiting Time Distribution 60

6 Results and Conclusions **61**

6.1 Introduction 61

6.2 Buffer Distribution and Output Process 61

6.3 Waiting Time Distribution 63

6.4 Conclusions and Future Work 63

A Markov Renewal Processes of M/G/1 Type **72**

A.1 Introduction 72

A.2 The M/G/1 Queue 73

A.3 The M/SM/1 Queue 74

A.4 The Canonical Form 75

A.5 The General Algorithm 77

A.6 First Passage Times 78

A.7 Structural Properties of ${}_z\mathbf{G}$ 80

A.8 Recurrence and the boundary states 81

A.9 Moment formulas for the matrix $\mathbf{G}(k)$ 82

A.10 Moment formulas for the matrix $\mathbf{K}(k)$ 86

A.11 Computation of the steady state vector 88

A.12 Computation of the moments of the steady state vector 88

B Definitions **92**

B.1 Delta Function 92

B.2 Unit Step Function 92

B.3 Computation of the function ϱ 93

C Derivations **94**

C.1 The moment generating function of the output process 94

D Glossary of Notation **96**

List of Tables

3.1 Order of states at level 0 38

3.2 Order of states for maximum requests of 3 at level $i > 0$ 38

3.3 Relabeled order of states for maximum requests of 3 at level $i > 0$ 39

6.1 Examples of load patterns applied to model 64

List of Figures

2.1 The DQDB network 6

2.2 Access Control Field 6

2.3 The DQDB state diagram 9

2.4 A tagged DQDB node 10

5.1 Structure of the renewal period of the tagged segment arrival 49

6.1 PMF of the Buffer Distribution 62

6.2 PMF of the Output Process 62

6.3 PMF of the Waiting time at Node 1 for Pattern 1 64

6.4 PMF of the Waiting time at Node 2 for Pattern 1 65

6.5 PMF of the Waiting time at Node 3 for Pattern 1 65

6.6 PMF of the Waiting time at Node 1 for Pattern 2 66

6.7 PMF of the Waiting time at Node 2 for Pattern 2 66

6.8 PMF of the Waiting time at Node 3 for Pattern 2 67

6.9 PMF of the Waiting time at Node 1 for Pattern 3 67

6.10 PMF of the Waiting time at Node 2 for Pattern 3 68

6.11 PMF of the Waiting time at Node 3 for Pattern 3 68

6.12 PMF of the Waiting time at Node 1 for Pattern 4 69

6.13 PMF of the Waiting time at Node 2 for Pattern 4 69

6.14 PMF of the Waiting time at Node 3 for Pattern 4 70

6.15 PMF of the Waiting time at Node 1 for Pattern 5 70

6.16 PMF of the Waiting time at Node 2 for Pattern 5 71

Chapter 1

Introduction

In this thesis an Algorithmic model is presented to the basic distributed queueing protocol of the IEEE (802.6) DQDB network. Two questions here might arise,

- What is the motivation behind modeling the DQDB network ?
- Why Algorithmic techniques ?

To answer the first question, recall that mathematical modeling is the science of representing physical systems and phenomena using an analytic model. The analytic model is then analyzed to give insight about the behavior of the corresponding physical system or phenomena. Throughout the history of modeling it appeared that there are many abstract mathematical representations that have corresponding realizations in unrelated areas. For example Markov Chains, Semi Markov Processes, Renewal Processes, Random Walks, to name a few, have applications in Telecommunications, Civil Engineering, Management Science, Industrial Engineering, Computer Science etc. So modeling a complex protocol like DQDB may yield techniques and algorithms to solve other distributed protocols and problems in other areas in general.

To answer the second question, note that in the past analytic models used the classical techniques of representing a problem with a transform equation and solving that equation either by inversion or numerically. Classical techniques failed to model many complex problems. So researchers moved to advanced techniques of which algorithmic probability is one. Algorithmic research yielded systematic algorithms to solve a general paradigm of problems. For example problems that can be represented with $G/M/1$ type matrices can be solved using the techniques in [Neu81]. On the other hand problems that can be represented with $M/G/1$ type matrices can be solved using techniques in [Neu89]. It is hoped that this thesis contributes to modeling distributed systems using algorithmic probability.

1.1 Objectives

In modeling a physical system, two models differ from each other in the details of the physical system that they capture. For example a natural picture has infinite refresh rate, as many colors as the eye can see and unlimited resolution. One model for this picture is on a black and white monitor with low refresh rate and limited resolution. We can enhance this model by increasing the resolution or refresh rate or by using a color monitor. In stochastic models assumptions are made to simplify the problem. These assumptions depend on the particular problem to be modeled. By eliminating these assumptions we can get better models that represent the system. Most of the DQDB literature models the busy slots on the forward bus by the memoryless geometric distribution. The only exception for that was [CGLN94] where the authors used an n^{th} order Markov process. So correlations depended on the past n slots. In this thesis a general renewal process is used to preserve all the correlations on the bus after the occurrence of an empty slot.

1.2 Contributions and Contents

This thesis has three main contributions. The first, in chapter 3, is a discrete time semi Markov process of $M/G/1$ type representation for a DQDB node with Poisson arrivals, general renewal process for the forward bus and a Bernoulli process for the backward bus. The state matrices depend on the location of the node in the network. Initially the state matrices have quadratic dimensions.

After some relabeling and reduction iterative algorithms are computed over sub matrices with linear dimensions. Solution of this discrete time semi Markov process using the techniques in [Neu89] yield the buffer distribution for a node. The second, in chapter 4, is the output process. By the output process we mean the distribution of the continuous stream of busy slots at the output of the node on the forward bus. The output process is derived from the semi Markov process defined in chapter 3. By approximating the arrivals on the reverse bus by a Bernoulli process, an overall approximate network analysis is possible. The third, in chapter 5, is the waiting time distribution at each node. The model was implemented in C language and the results are compared with simulation.

1.3 History of the Thesis

This thesis is the outcome and experience of two projects. In the first time I was a student at the university of Alberta. I spent 15 month on the thesis from January 1992 until April 1993. In April 1993 I had a car accident and did not submit the thesis as I had health problems and other things to take care of. During that time at the University of Alberta I learned Algorithmic Probability. It was a step ahead of classical modeling. It needed strong background in stochastic models and algorithms. I made a model for one DQDB node, but unfortunately the model was impractical to implement. The model assumed an n^{th} order Markov process to model the arrival of busy slots on the forward bus. The state matrices grew exponentially with the order of the Markov process and was thus hard to implement. I quit research and worked as a professional engineer to finance my plastic surgeries after the car accident until October 1995. My supervisor sent me a paper [CGLN94] that resembled my old model to a great extent. The difference is that they had an approximate algorithm for the output process so that an overall network analysis was possible. Also, they approximated each node by a FIFO while I used a state matrix to model the actual protocol.

Two things made me decide to do a new model. First I wanted to complete the project for TRlabs who gave me a scholarship for that purpose during my first enrollment at the University of Alberta. So I personally funded this new thesis by working in Electronic Research Institute (ERI) in the morning and working on the new thesis in the afternoons. Second I was motivated by the paper I read and felt that I was on the right track and may be I can make an enhanced model. I spent about 18 month in the new thesis beside my work at ERI.

Chapter 2

The Distributed Queue Dual Bus Network

2.1 Introduction

The Distributed Queue Dual Bus (DQDB) has been adopted by the IEEE Standards Committee as the IEEE 802.6 standard for Metropolitan Area Networks [Sta90]. In this thesis an algorithmic approach is presented for modeling the basic distributed protocol of DQDB. This chapter is intended as an overview of the distributed queueing protocol. In Sections 2.2 and 2.3 the architecture and the basic protocol of DQDB are presented respectively. The architecture is explained independently as it is almost the same for a class of protocols that employ distributed queueing. These protocols employ different "add-on" strategies to improve fairness and throughput of the basic distributed queueing protocol. For a good survey for such protocols the reader is referred to [MC92]. In Section 2.4 the interactions between different parameters are explained to give insight to the reader about the stochastic behaviour of the protocol. In Section 2.5 previous analytic models relevant to the thesis are reviewed.

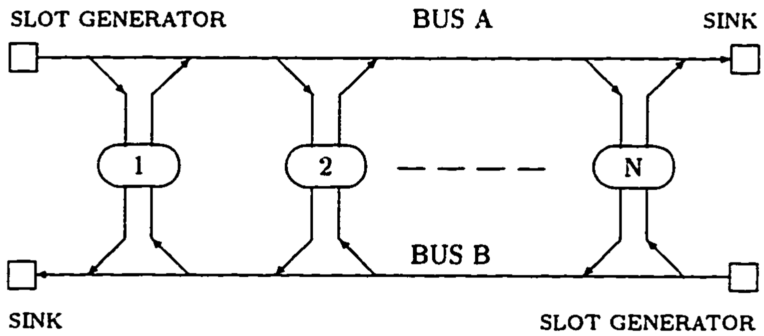

Figure 2.1: The DQDB network

BUSY (1 bit)	SLOT TYPE (1 bit)	PREVIOUS SLOT RCVD (1 bit)	RESERVED (1 bit)	RESERVED (1 bit)	REQUEST (3 bit)

Figure 2.2: Access Control Field

2.2 The Dual Bus Architecture

A DQDB network consists of two unidirectional buses and a series of nodes attached to it as shown in Fig (2.1). Nodes observe the activity on each bus as a sequence of empty and busy slots. Slots traverse the bus starting from the slot generator (Head of the Bus) until they reach the sink. The length of each slot was chosen by the standard committee to be 53 octets in order to be compatible with the ATM standard. Information is transmitted in 48 octets while the other 5 octets are used as a header. The first octet in the slot is the *Access Control Field* (ACF) with the format shown in Fig (2.2). Although the request field is formed from 3 bits, only 1 bit is used by the standard. The busy bit and the request bit of the ACF are used by the access protocol to coordinate the transmission between nodes. Messages arriving at the node may be partitioned into segments equal to slot size before they are transmitted on the bus. The segments are reassembled into the original message at the destination node. In this thesis arrivals of busy slots will be referred to as packet arrivals while internal arrivals will be referred to as segment arrivals. Bus A (the Forward Bus) and Bus B (the Reverse Bus) have opposite directions in order to achieve full duplex communications.

For a node with index i, nodes with index $j > i$ are called downstream nodes with respect to bus A while nodes with index $j < i$ are called upstream nodes. Full duplex communications is achieved between two nodes i and j, $i < j$, by using Bus A for transmission from i to j and Bus B for transmission from j to i.

Each node has two taps that connect it to a bus, one for reading and the other for writing. The node reads the busy bit at the beginning of the slot and then one of the following three scenarios takes place,

- If the bit is set then the slot is carrying a packet. The node reads the packet if it is destined to it, otherwise, it just leaves it to traverse the bus.
- If the bit is unset and the *Medium Access Control* (MAC) protocol decides to use the first empty slot for transmsission, then the node uses the writing tap to set the busy bit and writes the segment in the information field.
- The node leaves an empty slot to pass to downstream nodes in one of two cases. Either it has no segments to transmit, or the (MAC) protocol decides not to transmit the segment and just passes the slot to downstream nodes.

The (MAC) protocol decides whether to leave an empty slot or not according to reservations made by downstream nodes. Downstream nodes reserve empty slots on Bus A by sending requests on Bus B to upstream nodes. Similarly upstream nodes reserve empty slots on Bus B by sending requests on Bus A to downstream nodes. A node sends a request by setting the request bit in the ACF of the first slot in which the request bit is unset.

Several access protocols appeared in the literature that employ this architecture. In the next subsection the basic distributed queueing protocol is presented. It is worth noting that the access protocol described in the next section has many problems such as unfairness in overload conditions and bad throughput because the busy slots cannot be reused once they are read [vAWZ90, Won89]. So many schemes were introduced in the literature such as bandwidth balancing [HCM90] and erasure nodes to solve such problems. The new protocols employ almost the same architecture.

These protocols are not presented in this thesis as the only concern here is to model the basic distributed queueing protocol using algorithmic methods and show its complexities. It is hoped that such techniques can be extended to more complex protocols employing the dual bus architecture and other similar architectures.

2.3 The Distributed Queueing Protocol

In this section, segment transmission is considered on Bus A with the reservations made on Bus B. The procedure is the same for transmission on Bus B with segment reservations made on Bus A, due to the symmetry of the network. A node monitors the transmission on Bus A by keeping track of two counters the request counter and count-down counter. Note that there are two similar counters for Bus B so that we have request counter(A), request counter(B), count-down counter(A) and count-down counter(B). Unless otherwise stated, in this thesis both counters are related to Bus A. A node is either idle "has no segments to transmit" or busy "has segments to transmit" as shown in Fig 2.3. If the node is busy it is in the count-down state and we reserve the term busy to a busy slot on the Forward Bus. The two intermediate states, T1 and T2, show the actions performed by the protocol during transitions from the idle and count-down states. The state diagram is due to [Bis92] with minor modifications.

In the idle state the request counter is incremented by every request on Bus B and decremented by every empty slot on Bus A. In other words its value gives the total reservations by the downstream nodes minus the satisfied reservations which gives the total unsatisfied requests. Note if the request counter is zero it is not decremented by empty slots.

On the arrival of a segment for transmission the node transits to the count-down state passing by the state T1. In the state T1 the request counter is copied to the count-down counter to record the number of reservations to be satisfied, empty slots to be passed to downstream nodes, before transmitting the segment. A request is queued to be transmitted on Bus B in the first slot that has the request bit unset. This request registers a reservation for this segment in all upstream nodes so

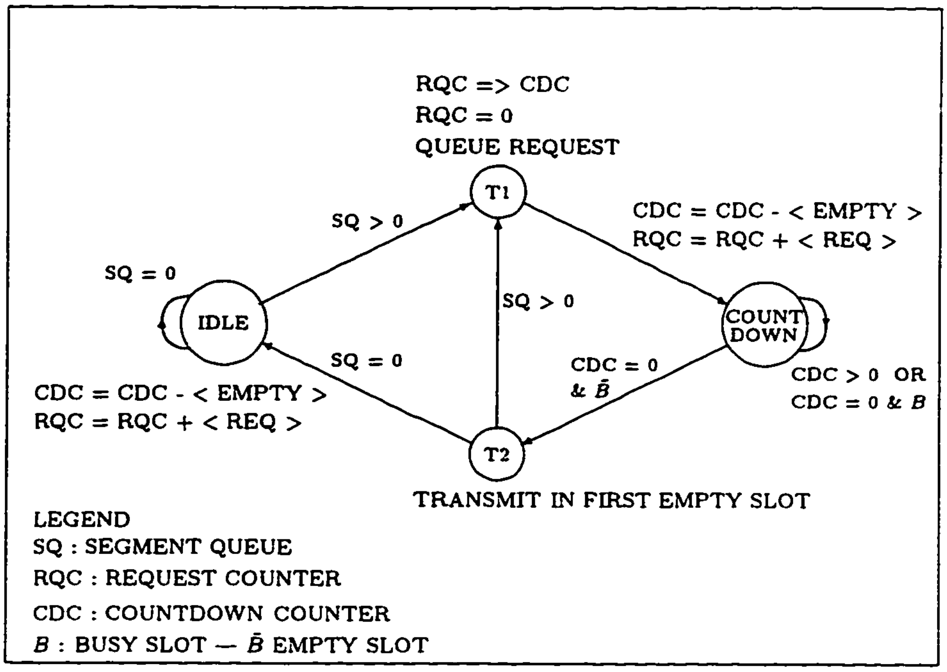

Figure 2.3: The DQDB state diagram

that they pass an empty slot for it.

In the count-down state the count-down counter is decremented by the observation of an empty slot on Bus A and the request counter is incremented by requests registered on Bus B. When the count-down counter reaches zero the node transmits the segment in the next empty slot (state T2 in Fig 2.3). During the count-down time the accumulated requests in the request counter are the reservations made by downstream nodes. These reservations have to be satisfied before the node can transmit a new segment. So in the case when there are more segments to be transmitted, the node must move back to state T1 and the loop is then repeated. The loop through the states T1 , Count Down and T2 is repeated until all segments are transmitted and then the node enters the idle state.

2.4 Stochastic Dependence of the Protocol Parameters

This subsection is intended to show the dependencies of different protocol parameters by intuition. It is meant to help the reader understand the complexity of the DQDB protocol and the shortcomings

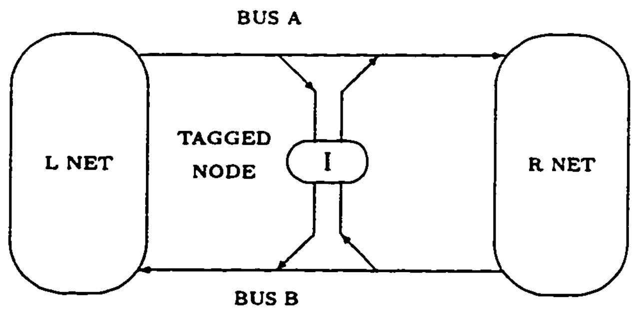

Figure 2.4: A tagged DQDB node

of existing models.

2.4.1 The Busy Period

The total time (in slots) spent in the loop between the states T1 , Count Down and T2 is the busy period from the viewpoint of node i, and is denoted by T_{bp}. Clearly

$$T_{bp} = T_{cd1} + T_{cd2} + \ldots + T_{cdL} \tag{2.1}$$

where T_{cdn} is the count-down time of the n^{th} segment in the busy period. T_{cdL} is the count-down time of the last segment in the busy period. Note that the count-down time is the time of transition from state T1 to state T2 passing through the Count Down state. The duration of T_{cdn} is dependent on the requests accumulated during $T_{cd(n-1)}$ by the request counter from downstream nodes on bus B, and is also dependent on the number of empty slots passed by the upstream nodes on bus A. For the special case of T_{cd1} it is dependent on the unsatisfied requests at the end of the idle period.

2.4.2 A FIFO Queue with Zero Inter-nodal Delay

Let us assume that the inter-nodal delay is zero, although in reality it is a measurable quantity in slots. Consider a tagged node as shown in Figure 2.4. The operation of the node can be thought of as a FIFO queue with a server that has vacations. The server here is an empty slot passed by the L_NET (up stream Nodes). The vacations are the number of busy slots between successive empty slots on bus A. Also it can be modeled by a server that has a general service time. Customers to the FIFO are of two types: internal segments and requests from the R_NET (down stream nodes). Because of the zero inter-nodal delay the requests from nodes down stream are registered in time thus taking the appropriate position in the FIFO queue of the tagged node. This method of approximation was used in the literature in [PGS90, CGL91a, CGL92a, CGLN94]. The first problem with this approximation is that in reality the arrival process is a function of queue parameters. So it would be more appropriate to model it as a queue with feedback or a controlled queue. To demonstrate this intuitively, the following scenario is presented. Assume that a node i in the L_NET has an overload, i.e. it will not pass an empty slot except when it receives a request on the reverse bus. Assume that node j is the rightmost active node in the R_NET. Requests from node j constitute arrivals to the tagged node. From the DQDB protocol node j cannot queue a new request until the segment corresponding to the previous request has been transmitted, (refer to state T1 in Figure 2.3). Therefore, with this scenario node j will not send a new request to the tagged node until the previous one has been served by the tagged node. Hence, the arrival process of requests to the FIFO is dependent on the delay in the FIFO and the service time of the FIFO.

The second problem with this approach is that, even if we ignore this feedback, the arrivals of requests and segments to the FIFO are related. This can be explained by considering the busy period discussed in Subsection 2.4.1. The number of requests between the n^{th} segment S_n and the $n+1^{th}$ segment S_{n+1} in the FIFO is equal to the requests registered during the n^{th} count-down time $T_{cd(n)}$. Hence, the number of requests between two successive segments is a function of the count-down time of the first one. Clearly this phenomenon has to be modeled by an arrival stream to the FIFO that captures the dependence between segment and request arrivals. Therefore, modeling the arrivals of segments and requests to the FIFO by two independent arrival sources is another approximation.

Hence the approximation of the DQDB MAC protocol at the tagged node by a FIFO queue with two independent arrival streams, internal segments and requests from R_NET, ignores the dependence of request arrivals on parameters from L_NET and tagged node arrivals. It would be a better approximation to model the MAC protocol as a FIFO with feedback or a FIFO with two dependent sources.

2.4.3 The Effect of Inter-nodal Delay

Consider two nodes with indices i and $i+1$ with inter-nodal distance d slots where d is an integer. Assume that the two nodes are the only active nodes in the network and that both nodes are under overload conditions. In other words the segment queues of both nodes are never empty. Since node i is the upstream node it observes only empty slots on the forward bus. It then transmits in all the slots except for the slots it passes to satisfy node $i+1$ requests. The question is what is the rate of empty slots passed to node $i+1$? Clearly it is a function in the inter-nodal delay d. Node $i+1$ sends a request on the reverse bus that arrives at node i after d slots, then node i honors the request by passing an empty slot on the forward bus that arrives at node $i+1$ after d slots. After node $i+1$ transmits the segment it can send another request to node i. In the meantime node i uses all the other slots. So the first effect of inter-nodal delay is on the throughput of the stations and this is the main reason of the unfairness in the DQDB protocol. For a detailed account of this effect the reader is referred to [Won89, vAWZ90].

2.4.4 The Effect of Correlated Requests

Throughout the bibliography of DQDB the arrivals of requests on the reverse bus were modeled by the memoryless Bernoulli process [Bis90, CMM95, CGL89, CGL91b, CGL91a, CGL92b, CGL92a, CGLN94, PGS90]. Also, in this thesis, a Bernoulli process is used to model the arrivals of requests on the reverse bus. In reality arrivals of requests are correlated and modeling them by the memoryless Bernoulli process is an approximation. The approximation was mainly adopted in the literature to make the solution tractable. Little insight into the DQDB protocol shows that requests are correlated. Consider a tagged node as in Figure 2.4, and assume it is in the busy period. The tagged node will not send a request on the reverse bus unless the current segment is transmitted on the forward bus. So that the duration between successive requests placed by the tagged node on the reverse bus depends on the waiting time of segments in the queue. Also this duration depends on the rate of requests arriving from the downstream nodes. By this simple argument. it easy to see that the duration between successive requests on the reverse bus is not geometrically distributed as the Bernoulli process.

2.4.5 The Effect of Correlated Busy Slots

As with requests on the reverse bus, busy slots on the forward bus have been modeled in the literature by a Bernoulli process except in [CGLN94]. In [CGLN94] the authors used an n^{th} order Markov process to model the arrivals of busy slots on the forward bus. The process preserved the correlations up to the past n^{th} slot. Unfortunately the state matrices of the model grew exponentially with n and the model is practical only for small n. The theoretical contribution of the model in [CGLN94] is great. It influenced this thesis, where the arrivals of the busy slots are modeled by a general renewal process and the state matrices are independent of this renewal process.

A little insight into the DQDB protocol can show that the busy slots are correlated. Consider the tagged node in Figure 2.4 and consider two durations ,T_{cd1} and T_{cd2}, between three successive transmissions in the busy period. T_{cd2} consists of a mixture of empty and busy slots passing by the node on the forward bus. The number of empty slots in T_{cd2} depends on the number of requests accumulated in the request counter during T_{cd1} which clearly depends on the length of T_{cd1}. Hence busy slots generated by one node are correlated and this means that busy slots on the forward bus are correlated.

2.4.6 The Effect of Position Dependence

The effect of position dependence is the main consequence of inter-nodal delay. The position of the node in the network affects all of its performance parameters. For example it was stated in [CMM95] that a node will not accumulate requests in it greater than the number of nodes downstream except in rare negligible situations. Although the authors showed this by extensive simulation. one can sense this result intuitively from the mechanics of the DQDB protocol. Clearly the waiting time distribution is affected by the maximum accumulated requests which is dependent on the position of the node in the network.

2.4.7 The coupled nature of the stochastic processes observed by a tagged node

The stochastic dependencies of the DQDB protocol parameters are circular . Consider the tagged node in Figure 2.4, we will show that empty slots on the forward bus coming from L_NET depend on requests coming from R_NET and vice versa.

The dependence of empty slots on the forward bus on the requests from the reverse bus is clear from the DQDB protocol. It is the dependence of the requests on the reverse bus on the empty slots on the forward bus that is less clear. To show the latter, recall that nodes in the R_NET will not schedule a request on the reverse bus until the previous segment is transmitted on the forward bus. This transmission depends on the empty slot stream leaving the tagged node on the forward bus. Hence the dependence is now obvious. It is this coupled nature that makes the protocol analytically intractable. Some authors adopted the approximation in [CGL91a] to reduce the complexity of the problem. The same approximation is used in this thesis see Section 3.3.2.

2.5 Previous Models

In this section analytic models of DQDB are reviewed. Other models of DQDB that depend on simulations and asymptotic analysis are not reviewed. Although these models added to the understanding of DQDB, but they are not relevant to the work in this thesis.

2.5.1 Analytic Models assuming Bernoulli arrivals on the Forward Bus

In [Bis90] an analysis for the waiting time distribution in a DQDB node is presented using generating functions. The author did not provide a network wide model but rather a single node model. The main assumptions are

- Packet arrivals on the forward bus are modeled by a Bernoulli process with parameter α equal to the probability of a busy slot.
- Request arrivals on the reverse bus are modeled by a Bernoulli process with parameter β equal to the probability of a request.
- Node buffer is of size one
- Segment arrivals at the node are geometrically distributed with parameter $e^{-\lambda}$ equal to the probability of no arrivals.

The author defined the concept of the virtual request counter VRQ_CTR where

$$VRQ\_CTR = \begin{cases} CD\_CTR + 1 & \text{when the user is active} \\ RQ\_CTR & \text{when the user is idle} \end{cases}$$

Let the value of the virtual request counter at the instant of the arrival of the n^{th} packet be F_n. The sequence $\{F_n; n \geq 1\}$ forms a Markov chain. After cumbersome algebraic manipulations, the author derived a functional equation for the generating function of the virtual request counter at arrival instants, see equation (34) in [Bis90].

The derivation depended on the generating function of the conditional waiting time $W(i)$ which had an error as pointed out in [CMM95, page 875]. Finally the author obtained an expression for the generating function of the waiting time in equation (37). At the end of the paper some numerical examples were plotted for different values of the parameters α, β and λ.

Although there was an error in the derivation, the model was one of the early models that showed the complexity of the DQDB modeling problem. The author did not show how to determine the parameters α and β, so a network wide model was not possible.

The model was extended in [JP92] by Jing and Paterakis using the same approach to the case where the buffer can queue a message of size l. They modified the definition of the virtual request counter to

$$VRQ\_CTR = \begin{cases} CD\_CTR & \text{when the user is active} \\ RQ\_CTR & \text{when the user is idle} \end{cases}$$

Then a transform equation for the message delay analysis was derived. The single node model in [JP92] was extended to a network wide model in [JP95] by the same authors. In [JP95] an approximate method, based on the analytic results in [JP92], for computing α_i and β_i is given. Where α_i is the probability that the i^{th} node observes a busy slot on the forward bus and β_i is the probability that it observes a request on the reverse bus.

An analytic model by Chen et al. [CMM95] was also based on the work in [Bis90]. The authors did major changes to the assumptions and methods used in the model. Although they had an approximate model. they showed deep insight in the behavior of DQDB. The authors don't follow a rigorous analytic approach using generating functions as in [Bis90] and [JP92]. They used a numerical iterative approach instead. The main insight gained from their work is the upper bound on the number of requests queued in a node. In [Bis90] and [JP92] the state space is assumed to be infinite and this in turn unrealistically over-estimates the waiting time distribution. Also the authors recognized the dependence of the arrivals of busy slots on the forward bus and the requests on the reverse bus on the node state. So they defined α and β as a function of the node state and they called it the modified geometric distribution. They then provided an approximate iterative scheme to compute these parameters for each node in a network-wide model.

In another study by Stavrakakis and Tsakiridou in [ST93, ST94] a matrix analytic approach was used. The occupancy distribution is derived from an M/G/1 type Markov chain embedded at packet departures. They assumed that the number of requests queued in a node is bounded just as in [CMM95]. Based on that the levels of the M/G/1 type chain correspond to packets queued in the node and the phase corresponds to number of requests in the node at the point of departure. They assumed Bernoulli processes for both of the arrivals of empty slots and requests on the forward and reverse buses respectively. They computed the buffer distribution for one node and did not extend it to a network-wide model. A major difference between this model and the model presented in Chapter 3 is the location of the embedding points of the Markov chain of the M/G/1 type. In the paper they used the packet departure instants and in the thesis we used the empty slots arrivals. It is only because of this choice that we could use a process other than Bernoulli to model the arrivals of busy slots. Also in this thesis the state matrices of the M/G/1 type process captures the mechanics of the DQDB protocol in a more detailed way.

2.5.2 Analytic Models Assuming Correlated arrivals on the Forward Bus

In a series of papers [CGL89, CGL91b, CGL91a, CGL92b, CGL92a] Conti et al. analyzed the behaviour of DQDB using simulation and classical analysis. In these papers the authors introduced an approximation to break the circular dependence, see Section 2.4.7, of the stochastic processes that arise in DQDB. The approximation is to model the arrival of requests on the reverse bus as observed by a node using a process with the same average arrival rate [CGL91a]. The authors then used the approximation to solve for a network wide model in [CGL92a]. In [CGL92a] they modeled the slot occupancy on the forward bus by a first order Markov process to capture the interdependence of the slots on the forward bus. The authors together with M.F. Neuts then published a paper that models the busy slots on the forward bus by an n^{th} order Markov process [CGLN94] and solved the model using matrix analytic methods. Their work in studying the interdependence of busy slots is unique. All other models in the literature, to our knowledge, assumed independence and hence modeled busy slots by a Bernoulli process. We will only review the model in [CGLN94] as it is the outcome of the efforts made in the other papers. A simplified DQDB is modeled with the following assumptions,

- A Poisson process is introduced at each node to model the arrivals of the requests from downstream nodes.
- As in the DQDB network, the forward bus is slotted and each slot is either busy or empty.
- Nodes are numbered from 1 to K
- The MAC protocol of each node is modeled by a queue where requests and segments are stored on a FIFO basis. Every node reduces its queue by one whenever it sees an empty slot. The slot remains empty if there is a request at the head of the queue and zero otherwise.

The arrival process to the queue is Poisson with parameter $\lambda(i) = \lambda_S(i) + \lambda_R(i)$, where $\lambda_S(i)$ and $\lambda_R(i)$ are arrival rates of segments and requests, respectively. $\lambda_S(i)$ depends on the workload characterization, whereas

$$\lambda_R(i) = \sum_{j=i+1}^{K} \lambda_S(j)$$

The probability that a queued packet is a request is

$$P_{REQ} = \frac{\lambda_R(i)}{\lambda_R(i) + \lambda_S(i)}$$

and

$$P_{SEG}(i) = 1 - P_{REQ}(i)$$

To preserve the correlations among the busy slots on the Forward Bus the state of n successive slots is considered. The binary random variable S_j^i represents the state of the j^{th} slot at the i^{th} node input which can be either empty (E) or busy (B). So the tuple $\{S_{m+1}^i \cdots S_{m+n}^i\}$represents the state of the input process to the i^{th} node at time m. Increasing n results in good approximations but increases the problem dimensions exponentially. Computing the output process at node i is the process of computing transition probabilities of the states $\{S^{i+1}_{m+1} \dots S^{i+1}_{m+n}\}$. The random variable A^i_n is zero if the i^{th} node does not transmit in the n^{th} slot and one otherwise. It is easy to observe that

$$S^{i+1}_n = \begin{cases} B & if A^i_n = 1 \\ S^i_n & if A^i_n = 0 \end{cases} \tag{2.2}$$

By solving the M/G/1 type Markov chain given by $\{(S_1, A_1), \dots, (S_n, A_n), L_n\}$, where L_n is the queue length at time n, the transition probabilities can be computed for the output process as shown in Section 4 of [CGLN94]. The dimensions of the state matrices of the M/G/1 type Markov chain depends on the order of the Markov process that is used to model the forward bus. So if the Markov process is of the n^{th} order then the matrices will have dimensions equal to 3^n. This increases the computational time dramatically as the steady state probabilities depend on the fundamental period matrix **G** which is computed iteratively see Appendix A.6. This paper influenced the work in this thesis as follows

- The same simplified DQDB is used to approximate the original DQDB and reduce the problem complexity.

- Correlations on the forward bus in the paper is preserved by an n^{th} order Markov Process while in the thesis it is modeled by a general independent renewal process.

- The node operation in the paper is approximated by a FIFO while in the thesis the original protocol is captured in the state matrices of an M/G/1 type semi Markov process.

- The state matrices of the M/G/1 type chain used in the paper has exponentially growing dimensions with respect to the chain. In the thesis they have quadratic dimensions with respect to the maximum numbers of requests that can be accumulated. Iterations are done on linear matrices because they are sparse.

2.6 Thesis Model

In this section a high level description of the algorithm used to compute the buffer occupancy distribution for each node in a DQDB network is outlined. This section serves as a road map to the thesis. Before the analysis of any node in a DQDB network is performed three stochastic processes have to be determined. These are

1. The arrivals of segments to the node.
2. The arrivals of empty slots on the forward bus.
3. The arrivals of requests on the reverse bus.

For each node only the arrivals of segments are known, except for the first node and the last node. For the first node the arrivals of empty slots take place with probability one since no nodes are before it to transmit. For the last node the arrivals of requests take place with probability one. The problem now is to determine the arrivals of empty slots and requests at internal nodes. Notice that the arrivals of requests to node n depends on arrivals of requests to node n+1, also arrivals of empty slots to node n depends on arrivals of empty slots to node n-1. If we consider the first node, arrivals of requests to the first node affect arrivals of empty slots to the second node. Also, if we consider the last node, arrivals of empty slots to the last node affect arrivals of requests to the node before last. So we have here a circular dependence of the arrivals of empty slots and requests. Representing the whole network by one Markov chain to get an exact solution is impossible due to the huge number of states. Any practical model would have to consider modeling nodes individually after determining the arrivals at each node.

In this thesis we used the same overall algorithm used in [CGLN94], but the analysis of each node is completely different. The overall algorithm used in [CGLN94] is to approximate arrivals of requests on the reverse bus by another process that has the same average arrivals. In this thesis we used geometric distribution. After this approximation the only unknowns would be the arrivals of empty slots on the forward bus. Since the arrival of empty slots for node 1 is known, we start the analysis from that node. We compute the buffer distribution for node 1 then we compute the output process which constitutes the arrivals of empty slots to node 2. The computation is repeated successively for all nodes until the last node is analysed. At the end, the algorithm will have computed the buffer distribution for each node at the instants of the occurrence of empty slots. The buffer distribution at these instants is then used to compute the waiting time distribution for segments at each node. In Chapter 3 a Markov renewal model of the M/G/1 type is presented for each node. The steady state probability vector of the embedded chain gives the buffer distribution at renewal epochs. In Chapter 4 an algorithm for the output process is presented. In Chapter 5 an algorithm for the waiting time distribution is presented.

Chapter 3

A Markov Renewal Representation of a DQDB node

3.1 Introduction

In this Chapter a DQDB node is modeled using a discrete time Markov renewal process of the M/G/1 type. In Section 3.2 an overview of Markov renewal processes of M/G/1 type is presented. The intention of the overview is to give the reader an overall perspective of the concept without distracting attention with formulas and proofs. This overview will help as a guide as we proceed to represent the node in Sections 3.3 and 3.4. For a comprehensive reference on Markov renewal process of the M/G/1 type the reader is referred to the book by Neuts [Neu89]. A summary of the algorithms and results in this book is given in Appendix A. In Section 3.3 the arrivals of empty slots on bus A are characterized by a Renewal process. In Section 3.4 a Markov renewal process is defined for the node at the renewal epochs of the process defined in Section 3.3. The main goal of this model is to take into account the correlations of busy slots on the Forward Bus (Bus A). In the model presented in this Chapter the arrivals of packets on the Forward Bus has a general distribution and the arrivals of requests are assumed to be a modified binomial distribution. In Section 3.5 reduction and reordering of the embedded chain is done in order to speed the computation of the first passage times.

3.2 An Overview of Markov Renewal Processes of M/G/1 Type

Markov renewal processes of M/G/1 type are a subset of semi Markov processes. Semi Markov process are a generalization of Markov chains. In Markov chains the next state is completely determined by the current state and hence it is memoryless with respect to the past. As a consequence of this the sojourn time in any state in a discrete time Markov chain is geometrically distributed. In the case of continuous time Markov chains it is exponentially distributed. This is because the exponential distribution is the only memoryless distribution in the case of continuous time see for example [Kle75, pages,45,46] for a proof or [Fel57, page, 413]. Similarly the Geometric distribution is the only memoryless distribution in discrete time see for example [Fel57, pages,304,305]. In semi Markov processes the next state depends on the state at the last transition not the current state. Therefore, the sojourn time is general and the interest is in the embedded chain at transition times. Semi Markov processes have a broad spectrum of applications because of the general distribution of the sojourn times. Markov renewal processes of M/G/1 type are a class of semi Markov processes that have M/G/1 state structure.

3.2.1 Levels and System states

In M/G/1 state structure, states have two dimensions the first is called levels and the second is called system states. The set of levels can be infinitely countable while system states are finitely countable. Note that the division of states to two dimensions is a logical division to help solve the problem in a structured way but as a matter of fact they can be thought of as one dimension. For example the variant of the M/G/1 queue where the server takes a maintenance period after the m^{th} service can be modeled by a Markov renewal process of M/G/1 type [Neu89, pages,64-65]. Here a level I_n. $n \geq 0$ corresponds to n customers being in the queue, and the m system states $\{S_i, 0 \leq i \leq m\}$ correspond to a set of logical states in which the server has different service times. Service in the m^{th} state is different from the first $m-1$ states as the maintenance period is added to it. By merging levels and system states in the tuple (I_n, S_i) we can have the one dimensional perspective in which each state correspond to a tuple.

The importance of the above concept may not be obvious with a simple example as the one given, but this will be clear after presenting a complex model as the one for a DQDB node. It is possible to model the DQDB protocol in many ways using Markov renewal processes of M/G/1 type depending on how the system states are defined. System states can actually have many dimensions as long as when they are reduced to one dimension they are finite.

3.2.2 Transition Probabilities

The transition probabilities for a Markov renewal process of M/G/1 type are given by

$$Q_{iji'j'}(x) = P\{I_n = i', J_n = j', \tau_n \leq x \mid I_{n-1} = i, J_{n-1} = j\} \tag{3.1}$$

where I_n and J_n are random variables denoting level and system states at the n^{th} transition epoch respectively. τ_n is a random variable denoting the time between the n^{th} and $(n-1)^{th}$ transition epochs. In matrix form we have,

$$\mathrm{Q(x)} = \begin{vmatrix} \mathbf{B}_0(x) & \mathbf{B}_1(x) & \mathbf{B}_2(x) & \mathbf{B}_3(x) & \dots \\ \mathbf{C}_0(x) & \mathbf{A}_1(x) & \mathbf{A}_2(x) & \mathbf{A}_3(x) & \dots \\ 0 & \mathbf{A}_0(x) & \mathbf{A}_1(x) & \mathbf{A}_2(x) & \dots \\ 0 & 0 & \mathbf{A}_0(x) & \mathbf{A}_1(x) & \dots \\ \vdots & \vdots & \vdots & \vdots & \end{vmatrix} \tag{3.2}$$

The matrix structure given in Equation 3.2 is called an M/G/1 type structure because it is a generalization of the matrix representing the transition probabilities of the number of customers in the classical M/G/1 queue. In the classical M/G/1 queue the sub matrices $\mathbf{A}_v$, $\mathbf{B}_v$ and $\mathbf{C}_v$ are scalars. In the general M/G/1 type structure these sub matrices are of many dimensions as long as the number of elements are finite. The matrix structure in Equation 3.2 is called left skip-free for levels because the process cannot descend from a level i to a level j, $j < i$, without passing through all intermediate levels. In other words the number of customers in the M/G/1 queue decreases by at most 1 in a single transition. Putting $x = \infty$ in Equation 3.2 gives the embedded Markov chain. Algorithms to solve for the steady state distribution of this chain are given in [Neu89] and summarized in Appendix A. In the remaining sections of this chapter a Markov renewal process of the M/G/1 type representation for a DQDB node is outlined.

3.3 Characterization of the Bus Process

The nodes of DQDB are numbered from 1 to N with node number one at the head-end on the forward bus and node N at the sink. Node i in the network is tagged as shown in Figure 2.4. In this section the busy stream on the Forward Bus and the request stream on the Reverse Bus are characterized.

3.3.1 The Forward Bus

The sequence of empty and busy slots that the tagged node observes is characterized by a renewal process. When the node observes an empty slot on the bus a renewal takes place. The tagged node is either idle, "has no segments to transmit", or in countdown state ,"has segments to transmit". While the node is idle the length of busy stream of slots can extend to ∞. This can happen with the following scenario, there is a node upstream with overload and all other nodes are idle. Let the random variable T_{idle} denote the time between renewals at the input during idle state of the tagged node. Let its probability mass function be $Pr(T_{idle} = n) = f_{idle}(n)$ with $f_{idle}(0) = 0$ and

$$f_{idle} = \sum_{n=1}^{\infty} f_{idle}(n) \ \leq 1 \tag{3.3}$$

If $f_{idle} < 1$ then the renewal process during the idle period is transient. On the other hand if $f_{idle} = 1$ then the renewal process is recurrent.

When a segment arrives the node switches to the countdown state and then the time between renewals becomes bounded by the DQDB protocol. The DQDB protocol implements a distributed queue algorithm and hence the segment will eventually be transmitted. Transmission of the segment means that the node will observe an empty slot at the input, i.e. a renewal. Let the time between renewals during the countdown state be denoted by T_{busy} with an upper bound t_{max}. The probability mass function of T_{busy} is given by $f_{busy}(n) = Pr(T_{busy} = n)$ with $f_{busy}(n) = 0$ when $n > t_{max}$ or $n = 0$. So that

$$\sum_{n=1}^{t_{max}} f_{busy}(n) = 1 \tag{3.4}$$

The distinction between the renewal period distribution when the node is idle or in countdown state is an attempt to take into account the recursive nature of the stochastic process observed by a DQDB node as discussed in Section 2.4. In this chapter a representation for one node is given assuming general distribution for both f_{idle} and f_{busy}. The representation requires that the polynomial representing the generating function of the distribution be finite. If the polynomial is not finite it will be truncated at a sufficiently large index where the remaining probabilities are negligible. The representation in this chapter can be used as an element in an overall network analysis. The algorithms for the overall network analysis should supply f_{idle} and f_{busy} from the output of the L_net (upstream nodes). If the output algorithms for the L_net compute one distribution f_{Lnet} for renewals independent of the state of the tagged node, then, a good approximation that takes into account the recursive nature is to set $f_{idle} = f_{Lnet}$. An estimate is computed for t_{max} for the tagged node and then $f_{busy}(n)$ is computed using conditional probability for $n \leq t_{max}$ as follows

$$Pr(T_{busy} = n) = Pr(T_{idle} = n | T_{idle} \leq t_{max}) = \frac{f_{idle}(n)}{\sum_{i=1}^{t_{max}} f_{idle}(i)} \tag{3.5}$$

The normalization with conditional probabilities eliminates the unrealistic cases where $T_{busy} > t_{max}$. This prevents the model from overestimating the delay that the segment experiences until transmission.

The time until the first renewal takes place, after switching from the idle period, has a different distribution and is denoted by the random variable T_{init}. Clearly T_{init} is bounded in the same way as T_{busy} and its probability mass function is given by $f_{init}(n) = Pr(T_{init} = n)$. With $f_{init}(n) = 0$ when $n > t_{max}$. So that

$$\sum_{n=0}^{t_{max}} f_{init}(n) = 1 \tag{3.6}$$

In terms of classical renewal theory T_{init} is the residual life and the period in which the arrival takes place is the life time. Refer to the "Paradox of Residual Life" in [Kle75, pages 169-174]. In general if the renewal period has a continuous density given by $f(t)$ and a mean m then the life density is given by

$$f_{life}(t) = \frac{tf(t)}{m} \tag{3.7}$$

and the density of the residual life is given by

$$f_{res}(t) = \frac{1 - F(t)}{m} \tag{3.8}$$

where $F(t)$ is the distribution of the renewal period. We want to derive f_{init} from f_{idle}, which have discrete mass functions, using the continuous time Equations 3.7 and 3.8. So we transform the discrete equations to continuous using delta functions, then derive the result in continuous time and finally switch back to discrete time. The density of T_{idle} has the continuous representation

$$f_{cidle}(t) = \sum_{n=1}^{\infty} f_{idle}(n)\delta(t - nT_s) \tag{3.9}$$

where $\delta(t - a)$ is the delta Dirac function, see Appendix B.1, and T_s is the duration of a slot on the bus. The continuous representation of the distribution is given by

$$F_{cidle}(t) = \sum_{n=1}^{\infty} f_{idle}(n)u(t - nT_s) \tag{3.10}$$

where $u(t - a)$ is the unit step function, see Appendix B.2. The density of T_{life} is

$$f_{life}(t) = \frac{\sum_{n=1}^{\infty} nT_s f_{idle}(n)\delta(t - nT_s)}{E(t_{idle})} \tag{3.11}$$

where E(x) is the expectation of x . And the distribution is given by

$$F_{life}(t) = \frac{\sum_{n=1}^{\infty} nT_s f_{idle}(n)u(t - nT_s)}{E(t_{idle})} \tag{3.12}$$

and the discrete representations of the life time density and distribution in terms of slots are respectively given by

$$f_{dlife}(n) = \frac{nT_s f_{idle}(n)}{E(t_{idle})} \tag{3.13}$$

and

$$F_{dlife}(n) = \sum_{i=1}^{n} \frac{iT_s f_{idle}(i)}{E(t_{idle})} \tag{3.14}$$

The density of the residual life is given by

$$f_{res}(t) = \frac{1 - \sum_{n=1}^{\infty} f_{idle}(n)u(t - nT_s)}{E(t_{idle})} \tag{3.15}$$

and hence its distribution is given by

$$F_{res}(t) = \frac{t - \sum_{n=1}^{\infty} f_{idle}(n)u(t - nT_s)(t - nT_s)}{E(t_{idle})} \tag{3.16}$$

Now to return back to the discrete representation where time is measured by the number of slots the mass function f_{dres} is defined by

$$f_{dres}(0) = F_{res}(T_s) \tag{3.17}$$

$$f_{dres}(n) = F_{res}((n+1)T_s) - F_{res}((n)T_s) \tag{3.18}$$

and since T_{res} is bounded by the DQDB protocol in the same way as T_{busy} then

$$f_{init}(n) = \frac{f_{dres}(n)}{\sum_{i=0}^{t_{max}} f_{dres}(i)} = \frac{f_{dres}(n)}{F_{res}((t_{max}n + 1)T_s)} \qquad v \leq t_{max} \tag{3.19}$$

In the calculation of f_{dlife}, F_{res} and f_{init}, the value of T_s is not needed as $E(t_{idle})$ can be calculated in terms of T_s and then T_s cancels from the fraction.

3.3.2 The Reverse Bus

To reduce the problem complexity, the approximation given in [CGL91a] is adopted. In that paper, the reverse bus is not modeled , and a Poisson process is used to account for the request stream of each of the downstream nodes. Let $f_{req}(t,i)$ be the probability of i requests in t slots with $f_{req}(t,i) = 0$ for $i < 0$. Then f_{req} is given by

$$f_{req}(t,i) = \binom{t}{i} p^i q^{t-i} \tag{3.20}$$

where p is the probability of a request and q is the probability of no request. The values of p and q are determined from the total traffic offered by the nodes downstream. It has been shown in [CMM95] that a node cannot accumulate requests greater than the number of nodes downstream except in rare situations that have negligible probabilities. Taking this into account we assume that the sum of CD_n and RQ_n cannot exceed r_{max} at renewal epochs. The value of r_{max} is different for each node on the network and depends on the node position on the bus. If at time T_n the sum of CD_n and RQ_n is r_n, then the number of requests registered on the reverse bus during τ_{n+1} cannot exceed $r_{max} - r_n + \delta_{T_{n+1}}$. If there is no transmission at T_{n+1} on the forward bus then $\delta_{T_{n+1}} = 1$ otherwise $\delta_{T_{n+1}} = 0$. Note that if there is no transmission at T_{n+1} either the countdown counter or the request counter will be decremented depending on the node state. Hence we define $f_{creq}(t.i.j)$ to be the conditional probability of i requests in t slots given that i cannot be greater than j. Hence f_{creq} is given by

$$f_{creq}(t,i.j) = \frac{f_{req}(t,i)}{F_{req}(t,j)} \tag{3.21}$$

where F_{req} is the distribution of f_{req}.

3.4 A Markov Renewal model for a DQDB node

In this section a Markov renewal model is formulated for a DQDB node based on the processes defined in the previous section. It is assumed that propagation delay on the forward bus is equal to the delay on the reverse bus, internodal delay is always an integer number of slots and slot arrivals on both bus are synchronized. The random variables that constitute the node state are considered at the beginning of each empty slot (renewal epoch) and they are defined as follows

- I_n is the number of packets in the system at the beginning of each empty slot after subtracting the packet transmitted in this slot, if any.

- RQ_n is the number of requests in the request counter at the beginning of the n^{th} empty slot including the one registered on the reverse bus, if any, and after decrementing it if in the idle state.

- CD_n is the number of the requests in the count-down counter at the beginning of the n^{th} empty slot after decrementing it if no transmission took place. If transmission took place then the state is taken after copying the request counter in it.

- T_n is the time at the beginning of the n^{th} empty slot after updating the variables and $\tau_n = T_n - T_{n-1}$.

The pdf of the arrivals of segments at the tagged node is Poisson. and is denoted by $\phi_v(t)$. The Markov renewal matrix for a DQDB node has its elements given by

$$\begin{aligned} Q_{ii'jj'kk'}(x) &= P\{I_n = i', CD_n = j', RQ_n = k', \tau_n \leq x \mid \\ &\quad I_{n-1} = i, CD_{n-1} = j, RQ_{n-1} = k\} \end{aligned} \tag{3.22}$$

and the transition matrix has the structure

$$\mathbf{Q(x)} = \begin{vmatrix} \mathbf{B}_0(x) & \mathbf{B}_1(x) & \mathbf{B}_2(x) & \mathbf{B}_3(x) & \dots \\ \mathbf{C}_0(x) & \mathbf{A}_1(x) & \mathbf{A}_2(x) & \mathbf{A}_3(x) & \dots \\ 0 & \mathbf{A}_0(x) & \mathbf{A}_1(x) & \mathbf{A}_2(x) & \dots \\ 0 & 0 & \mathbf{A}_0(x) & \mathbf{A}_1(x) & \dots \\ \vdots & \vdots & \vdots & \vdots & \end{vmatrix} \tag{3.23}$$

Each row in the matrix $\mathbf{Q}(x)$ is called a level. Levels correspond to number of segments queued in the node. The first row is level zero, where the node has no queued segments (idle). The matrix

$\mathbf{Q}(x)$ is an infinite matrix of the M/G/1 type. The matrices $\mathbf{C}_0(x)$,$\mathbf{A}_v(x)$ and $\mathbf{B}_v(x)$ $v \geq 0$ are square matrices of dimensions r_{max}, and their elements are matrices of dimensions r_{max}. They have the general form of matrix $\mathbf{X}$ shown below along with its sub-matrices.

$$\mathbf{X} = \begin{pmatrix} \mathbf{X}^{00} & \mathbf{X}^{01} & \dots & \mathbf{X}^{0r_{max}} \\ \mathbf{X}^{10} & \mathbf{X}^{11} & \dots & \mathbf{X}^{1r_{max}} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{X}^{r_{max}0} & \mathbf{X}^{r_{max}1} & \dots & \mathbf{X}^{r_{max}r_{max}} \end{pmatrix} \tag{3.24}$$

$\mathbf{X}^{ij}$ is a sub matrix that defines the transition probabilities for the states in which the count-down counter transit from i to j. The elements of each sub matrix define its respective transition probabilities for the request counter.

$$\mathbf{X}^{ij} = \begin{pmatrix} x_{00}^{ij} & x_{01}^{ij} & \dots & x_{0r_{max}}^{ij} \\ x_{10}^{ij} & x_{11}^{ij} & \dots & x_{1r_{max}}^{ij} \\ \vdots & \vdots & \vdots & \vdots \\ x_{r_{max}0}^{ij} & x_{r_{max}1}^{ij} & \dots & x_{r_{max}r_{max}}^{ij} \end{pmatrix} \quad 0 \leq i, j \leq r_{max} \tag{3.25}$$

In the next two subsections $\mathbf{A}_v$, $\mathbf{B}_v$ and $\mathbf{C}_0$ are defined for the DQDB protocol described in Section 2.3.

3.4.1 Transitions from Count-down States. (Level $i \geq 1$)

The matrix $\mathbf{C}_0$ defines transitions from count-down states to idle states (level 0) in which the DQDB node has no segments to transmit. The count-down counter has to be zero at T_{n-1} in order to transmit the last segment at T_n. Since the node enters the idle state then CD_n remains zero at T_n. The requests registered during τ_n on the reverse bus are added to the request counter.

- The matrix $\mathbf{C}_0(x)$ has the following elements
  - $\mathbf{C}_0^{jj'}(x) = \mathbf{0}$ the zero square matrix of dimensions $r_{max} + 1$

    for $j > 0$ and $j' > 0$.

– $\mathbf{C}_0^{00}(x)$ has elements

  * $c_{kk'}(x) = \sum_{l=1}^{x} f_{creq}(l, k' - k, r_{max} - k)\phi_0(l) f_{busy}(l)$

The matrix $\mathbf{A}_0$ defines transitions from states at level $(i+1)$ to states at level i, $i > 0$. A segment is transmitted at T_n, and no arrivals take place during τ_n that is why the process descends one level. Unlike $\mathbf{C}_0$ at T_n the node is still in the count-down state, since the queue is nonempty. Therefore. the request counter is copied into the count-down counter, and this can be easily noticed from the matrix structure. The request counter is zero at T_n. If there is a request registered on the reverse bus just prior to T_n, then it is added to the count-down counter.

- The matrix $\mathbf{A}_0(x)$ has the following elements
  - $\mathbf{A}_0^{jj'}(x) = 0$ the zero square matrix of dimensions $r_{max} + 1$ for $j > 0$.
  - $\mathbf{A}_0^{0j'}(x)$ has elements
    * $a_{k0}(x) = \sum_{l=1}^{x} f_{creq}(l, j' - k, r_{max} - k)\phi_0(l) f_{busy}(l)$
    * $a_{kk'}(x) = 0$ for $k' > 0$

The matrices $\mathbf{A}_v(x)$ for $v > 0$ are split into two parts such that

$$\mathbf{A}_v(x) = \mathbf{\Upsilon}_v(x) + \mathbf{\Omega}_v(x)$$

$\mathbf{\Upsilon}_v(x)$ corresponds to v arrivals and one transmission. $\mathbf{\Omega}_v(x)$ corresponds to $v - 1$ arrivals and no transmission. The matrix structure of $\mathbf{\Upsilon}_v(x)$ is similar to the structure of $\mathbf{A}_0$ except that here v arrivals take place instead of no arrivals.

- $\mathbf{\Upsilon}_v(x)$ has its elements defined by
  - $\mathbf{\Upsilon}_v^{jj'}(x) = 0$ the zero square matrix of dimensions $r_{max} + 1$ for $j > 0$.
  - $\mathbf{\Upsilon}_v^{0j'}(x)$ has elements
    * $v_{k0}(x) = \sum_{l=1}^{x} f_{creq}(l, j' - k, r_{max} - k)\phi_v(l) f_{busy}(l)$
    * $v_{kk'}(x) = 0$ for $k' > 0$

The matrix $\Omega_v(x)$ defines transitions from states at level i to states at level $i+v$ where $i, v \geq 1$ on the condition that no transmission takes place at T_n. At T_{n-1} the count-down cannot be zero otherwise transmission would take place at T_n since the node has segments queued in it. The count-down counter is decremented by one and the request counter is incremented by the requests during τ_n.

- $\Omega_v(x)$ has its elements defined by
  - $\Omega_v^{jj'}(x) = \mathbf{0}$ the zero square matrix of dimensions $r_{max} + 1$ for $j = 0$ or j' not equal $j - 1$.
  - $\Omega_v^{jj'}(x)$ has elements
    - $\omega_{kk'}(x) = \sum_{l=1}^{x} f_{creq}(l, k' - k, r_{max} - j - k + 1)\phi_{v-1}(l) f_{busy}(l)$
      when $j' = j - 1,\ j > 0$.

The matrices $\mathbf{C}_0(x)$, $\mathbf{A}_0(x)$, $\mathbf{A}_v(x)$, $\mathbf{\Upsilon}_v(x)$ and $\Omega_v(x)$ are distribution matrices over x. Their respective mass matrices are ${}_{\mathbf{m}}\mathbf{C}_0(x)$, ${}_{\mathbf{m}}\mathbf{A}_0(x)$, ${}_{\mathbf{m}}\mathbf{A}_v(x)$, ${}_{\mathbf{m}}\mathbf{\Upsilon}_v(x)$ and ${}_{\mathbf{m}}\Omega_v(x)$. Note that $\mathbf{C}_0(x) = \sum_{l=1}^{x} {}_{\mathbf{m}}\mathbf{C}_0(x)$, and similar relations hold for the rest of the matrices. It is better to compute the distribution matrices directly by summing over non zero elements only, and not use mass matrices for summation as they are sparse. This saves computing time.

3.4.2 Transitions from Idle States (level 0)

The renewal period, in which the node switches from the idle state to the count-down state, is divided into two periods. The first period is the time from the last empty slot (renewal epoch) until the slot where the arrival took place. The second period is the time until the next renewal. The matrices $\Psi_v(x)$ and $\Xi_v(x)$ define the transition probabilities for the two periods respectively. A special case happens when the node enters the idle state again after the renewal. This happens only when one arrival takes place between renewals and gets transmitted in the following empty slot. The matrix $\Theta(x)$ defines the transition probabilities for this case, given that an arrival has taken place. The matrices defining the two periods are then convolved to get the transition matrices at level 0.

The conditional probability matrix $\Psi_v(x)$ is defined as the square matrix of dimensions $r_{max} + 1$

and its elements $\Psi_v^{jj'}(x)$ are also square matrices of dimensions $r_{max}+1$. The elements of these matrices are given by

$$\psi_v^{jj'kk'}(x) = P\{I_x = v, CD_x = j', RQ_x = k' \mid I_0 = 0, \ldots, I_{x-1} = 0, CD_0 = j, RQ_0 = k\} \tag{3.26}$$

The slot at time 0 in Equation (3.26) is an empty slot during the idle period. The next renewal takes place after the $(x-1)^{th}$ slot. The probability of this event is expressed by multiplying the elements by $(1 - F_{dlife}(x-1))$. F_{dlife} is used which is the discrete representation of the age distribution instead of F_{idle} as the node acts as an observer to the renewal process. Arrivals take place during the $(x-1)^{th}$ slot and not before that. I_x, CD_x and RQ_x denote the number of packets in the node, the value of the count-down counter and the value of the request counter at the beginning of the x^{th} slot respectively.

- The elements of the matrix $\Psi_v(x)$ are defined by
  - $\psi_v^{0j'k0}(x) = f_{creq}(x, j'-k, r_{max}-k)\varphi_v(x)(1 - F_{dlife}(x-1))$
  - $\psi_v^{0j'kk'}(x) = 0$ for $k' > 0$.
  - $\psi_v^{jj'kk'}(x) = 0$ for $j > 0$

where $\varphi_v(x)$ is the probability that the first arrival is after the beginning of the $(x-1)^{th}$ slot and that there are more $v-1$ arrivals until the beginning of the x^{th} slot. For Poisson arrivals the inter-arrivals are exponentially distributed with parameter λ so $\varphi_v(x)$ is given by

$$\varphi_v(x) = \int_{(x-1)T_s}^{xT_s} \lambda e^{-\lambda t}\, p(xT_s - t; v-1)\, dt \tag{3.27}$$

where $p(t;k) = \frac{(\lambda t)^k e^{-\lambda t}}{k!}$ and hence

$$\varphi_v(x) = \frac{(\lambda T_s)^v e^{-\lambda T_s x}}{v!} \tag{3.28}$$

The matrices $\Theta(x), \Xi_0(x)$ and $\Xi_v(x)$ are almost similar to the matrices $C_0(x)$, $A_0(x)$ and $A_v(x)$. The difference is that f_{init} is used here instead of f_{busy}, and that $\Theta(0), \Xi_0(0)$ and $\Xi_v(0)$ have values instead of the zero matrix. The values at $x = 0$ account for the case when packets arrive during the slot immediately before the next empty slot. In this case the residual life is 0 in terms of the remaining slots, and that is why $f_{init}(0)$ can be positive unlike $f_{busy}(0)$ which is equal to zero. Note that $\phi_0(0) = 1$ and that $\phi_v(0) = 0$ for $v > 1$. Also $f_{creq}(0, 0, j) = 1$ for $j \geq 0$ and $f_{creq}(0, v, j) = 0$ for $v > 1$, $j \geq 0$.

- The matrix $\Theta(x)$ has the following elements
  - $\Theta^{jj'}(x) = \mathbf{0}$ the zero square matrix of dimensions $r_{max} + 1$ for $j > 0$ and $j' > 0$.
  - $\Theta^{00}(x)$ has elements
    - $\theta_{0k'}(x) = \sum_{l=0}^{x} f_{creq}(l, k' - k, r_{max} - k)\phi_0(l) f_{init}(l)$
    - $\theta_{kk'}(x) = 0 \qquad k > 0$
- The matrix $\Xi_0(x)$ has the following elements
  - $\Xi_0^{jj'}(x) = \mathbf{0}$ the zero square matrix of dimensions $r_{max} + 1$ for $j > 0$.
  - $\Xi_0^{0j'}(x)$ has elements
    - $\xi_{k0}(x) = \sum_{l=0}^{x} f_{creq}(l, j' - k, r_{max} - k)\phi_0(l) f_{init}(l)$
    - $\xi_{kk'}(x) = 0 \qquad \text{for} \qquad k' > 0$
- The matrices $\Xi_v(x)$ for $v > 0$ are split into two parts such that $\Xi_v(x) = \dot{\Xi}_v(x) + \ddot{\Xi}_v(x)$. $\dot{\Xi}_v(x)$ corresponds to v arrivals and one transmission. $\ddot{\Xi}_v(x)$ corresponds to $v - 1$ arrivals and no transmission. $\dot{\Xi}_v(x)$ has its elements defined by
  - $\dot{\Xi}_v^{jj'}(x) = \mathbf{0}$ the zero square matrix of dimensions $r_{max} + 1$ for $j > 0$.
  - $\dot{\Xi}_v^{0j'}(x)$ has elements
    - $\dot{\xi}_{k0}(x) = \sum_{l=0}^{x} f_{creq}(l, j' - k, r_{max} - k)\phi_v(l) f_{init}(l)$
    - $\dot{\xi}_{kk'}(x) = 0 \qquad \text{for} \qquad k' > 0$

$\ddot{\Xi}_v(x)$ has its elements defined by

- $\ddot{\Xi}_v^{jj'}(x) = \mathbf{0}$ the zero square matrix of dimensions $r_{max}+1$ for $j=0$ or j' not equal $j-1$.
- $\ddot{\Xi}_v^{jj'}(x)$ has elements
  * $\ddot{\xi}_{kk'}(x) = \sum_{l=0}^{x} f_{creq}(l, k'-k, r_{max}-j-k+1)\phi_{v-1}(l)f_{init}(l)$ when $j' = j-1, j > 1$.

The transition probability matrix $\mathbf{\Gamma}(x)$ defines the transition probabilities at renewal epochs during the idle period. It is a square matrix of dimensions $r_{max}+1$ and its elements $\mathbf{\Gamma}^{jj'}(x)$ are square matrices of dimensions $r_{max}+1$ with scalar entries given by

$$\gamma_v^{jj'kk'}(x) = P\{I_n = 0, CD_n = 0, RQ_n = k', \tau_n \le x \mid I_{n-1} = 0, CD_{n-1} = 0, RQ_{n-1} = k\} \tag{3.29}$$

The matrix $\mathbf{\Gamma}^{jj'}(x) = \mathbf{0}$ for $j > 0$ or $j' > 0$. The elements of the matrix $\mathbf{\Gamma}^{00}(x)$ are given by

- $\gamma_v^{00\,00}(x) = \sum_{l=1}^{x}(f_{creq}(l, 0, r_{max}+1) + f_{creq}(l, 1, r_{max}+1))\phi_0(x)f_{dlife}(x)$
- $\gamma_v^{00kk'}(x) = \sum_{l=1}^{x} f_{creq}(l, k'-k+1, r_{max}-k+1)\phi_0(x)f_{dlife}(x)$

  for $k' \ge 0$, $k \ge 0$ except when $k' = k = 0$.

The mass matrices of $\mathbf{\Theta}(x), \mathbf{\Xi}_0(x)$ and $\mathbf{\Xi}_v(x)$ are denoted by ${}_{\mathbf{m}}\mathbf{\Theta}(x)$, ${}_{\mathbf{m}}\mathbf{\Xi}_0(x)$ and ${}_{\mathbf{m}}\mathbf{\Xi}_v(x)$, respectively. The matrices $\mathbf{B}_v(x)$ are given by

$$\mathbf{B}_0(x) = \mathbf{\Psi}_1(x) \circ \mathbf{\Theta}(x) + \mathbf{\Gamma}(x) \tag{3.30}$$

$$\mathbf{B}_v(x) = \sum_{i=1}^{v+1} \mathbf{\Psi}_i(x) \circ \mathbf{\Xi}_{v-i+1}(x) \qquad v > 0 \tag{3.31}$$

and their respective mass matrices are given by

$${}_{\mathbf{m}}\mathbf{B}_0(x) = \mathbf{\Psi}_1(x) \circ {}_{\mathbf{m}}\mathbf{\Theta}(x) + {}_{\mathbf{m}}\mathbf{\Gamma}(x) \tag{3.32}$$

$${}_{\mathbf{m}}\mathbf{B}_v(x) = \sum_{i=1}^{v+1} \mathbf{\Psi}_i(x) \circ {}_{\mathbf{m}}\mathbf{\Xi}_{v-i+1}(x) \qquad v > 0 \tag{3.33}$$

where the $\circ$ denotes the convolution, with respect to the time x. To compute the transition probability matrix of the embedded Markov chain we need to compute $\mathbf{B}_v(\infty)$. Using transforms we can find an efficient way to do this. For any matrix $\mathbf{M}(x)$ we denote its corresponding z transform (Generating function) by ${}_z\mathbf{M}(z)$, where ${}_z\mathbf{M}(z) = \sum_{i=0}^{\infty} \mathbf{M}(i)z^i$. Hence the transform equations are given by

$$ {}_{\mathbf{zm}}\mathbf{B}_0(z) = {}_{\mathbf{z}}\mathbf{\Psi}_1(z)\ {}_{\mathbf{zm}}\mathbf{\Theta}(z) + {}_{\mathbf{zm}}\mathbf{\Gamma}(z) \tag{3.34} $$

$$ {}_{\mathbf{zm}}\mathbf{B}_v(z) = \sum_{i=1}^{v+1} {}_{\mathbf{z}}\mathbf{\Psi}_i(z)\ {}_{\mathbf{zm}}\mathbf{\Xi}_{v-i+1}(z) \qquad v > 0 \tag{3.35} $$

Note that ${}_{\mathbf{zm}}\mathbf{B}_v(1) = \mathbf{B}_v(\infty)$, ${}_{\mathbf{zm}}\mathbf{\Xi}_v(1) = \mathbf{\Xi}_v(\infty)$, ${}_{\mathbf{zm}}\mathbf{\Theta}(1) = \mathbf{\Theta}(\infty)$ and ${}_{\mathbf{zm}}\mathbf{\Gamma}(1) = \mathbf{\Gamma}(\infty)$. So we finally have

$$ \mathbf{B}_0(\infty) = {}_{\mathbf{z}}\mathbf{\Psi}_1(1)\mathbf{\Theta}(\infty) + \mathbf{\Gamma}(\infty) \tag{3.36} $$

$$ \mathbf{B}_v(\infty) = \sum_{i=1}^{v+1} {}_{\mathbf{z}}\mathbf{\Psi}_i(1)\mathbf{\Xi}_{v-i+1}(\infty) \qquad v > 0 \tag{3.37} $$

3.5 Reduction of the Chain

3.5.1 Deletion of States

The matrices $\mathbf{C}_0(x)$, $\mathbf{A}_v(x)$ and $\mathbf{B}_v(x)$, $v \geq 0$, defines transition probabilities for states at different levels.

Each state defines a different value for the tuple (CD_n, RQ_n). Since RQ can take values from 0 to r_{max}, and similarly for CD, then there are $(r_{max}+1)^2$ states. As stated previously the node will not accumulate requests more than r_{max} , so the states where $(CD_n + RQ_n) > r_{max}$ can be deleted. This is done by removing their corresponding rows and columns from the matrices $\mathbf{C}_0(x)$, $\mathbf{A}_v(x)$ and $\mathbf{B}_v(x)$, $v \geq 0$. The number of remaining states s_r is given by

$$ s_r = \sum_{i=1}^{r_{max}+1} i = \frac{(r_{max}^2 + 3r_{max} + 2)}{2} \tag{3.38} $$

and the number of deleted states s_d is given by

$$ s_d = (r_{max}+1)^2 - s_r = \frac{r_{max}(r_{max}+1)}{2} \tag{3.39} $$

Originally the number of elements in the state matrices are $(r_{max}+1)^4$ and after reduction they are $\frac{(r_{max}^2+3r_{max}+2)^2}{4}$ state. This reduction speeds the computation time of further performance matrices as they use iterative algorithms.

There is a further reduction at the boundary (level 0). The process never visit the states (CD_n, RQ_n) where $CD_n > 0$. This is obvious from the DQDB protocol since the count-down counter can't take positive values during the idle period. Rows with indices greater than $(r_{max}+1)$ are deleted from the $\mathbf{B}_v$ matrices ,$v \geq 0$. Columns with indices greater than $(r_{max}+1)$ are deleted from $\mathbf{B}_0$ and $\mathbf{C}_0$. Finally the dimensions of the matrices are

$\mathbf{B}_0$ $(r_{max}+1 \;\times\; r_{max}+1)$

$\mathbf{C}_0$ $(\frac{(r_{max}^2+3r_{max}+2)}{2} \;\times\; r_{max}+1)$

$\mathbf{B}_v$ $(r_{max}+1 \;\times\; \frac{(r_{max}^2+3r_{max}+2)}{2})$ $\qquad v \geq 0$

$\mathbf{A}_v$ $(\frac{(r_{max}^2+3r_{max}+2)}{2} \;\times\; \frac{(r_{max}^2+3r_{max}+2)}{2})$ $\qquad v \geq 0$

The states at level 0 have the order shown in Table 3.1.

State	1	2	...	$r_{max}+1$
Value of (CD, RQ)	$(0,0)$	$(0,1)$	...	$(0, r_{max})$

Table 3.1: Order of states at level 0

The states at level $i, i \geq 1$ have an ascending order as shown in Table 3.2 for the case where $r_{max} = 3$.

State	1	2	3	4	5
Value of (CD, RQ)	$(0,0)$	$(0,1)$	$(0,2)$	$(0,3)$	$(1,0)$
State	6	7	8	9	10
Value of (CD, RQ)	$(1,1)$	$(1,2)$	$(2,0)$	$(2,1)$	$(3,0)$

Table 3.2: Order of states for maximum requests of 3 at level $i > 0$

3.5.2 Relabeling States

The fundamental period matrix $_z\mathbf{G}$, described in section A.6 , is crucial to the computation of many performance values related to the chain $\mathbf{Q}(\infty)$. A significant reduction in computation can be done if the matrix ${}_zG$ is reducible as discussed in section A.7. The reducibility of ${}_zG$ depends on the matrix A_0. It is proved by Lemma 2.3.6 in [Neu89, page 95] that for every zero column in A_0 there is a zero column in ${}_zG$. From the definition of A_0 it has zero columns whenever $RQ_n > 0$ i.e. $k' > 0$. So the number of nonzero columns in ${}_zG$ is equal to the number of states where $k' = 0$ which is equal to $r_{max} + 1$. The matrices A_v and B_v are relabeled by reversing the tuple (CD_n, RQ_n) to (RQ_n, CD_n) as shown in table [3.3] for the case where $r_{max} = 3$.

State	1	2	3	4	5
Value of (RQ, CD)	$(0,0)$	$(0,1)$	$(0,2)$	$(0,3)$	$(1,0)$
State	6	7	8	9	10
Value of (RQ, CD)	$(1,1)$	$(1,2)$	$(2,0)$	$(2,1)$	$(3,0)$

Table 3.3: Relabeled order of states for maximum requests of 3 at level $i > 0$

Finally the matrix ${}_zG$ can be partitioned as in equation (A.27) and computed as in equations (A.28) and (A.29). The matrices involved in the iterations are of dimensions $r_{max} + 1$ instead of $\frac{(r_{max}^2+3r+2)}{2}$.

The zero rows in the fundamental period matrix expresses the fact that the process will never enter a level on its first passage with $RQ > 0$. This is because the request counter is copied to the count-down counter and then set to zero at the start of a new transmission.

3.6 Summary

In this chapter a Markov renewal model was formulated for a DQDB node. Once the distributions of segments, requests and busy slots arrivals are determined, the algorithms in [Neu89] (summarized in Appendix A) can be used to compute the steady state distribution of the buffer occupancy at the beginning of any empty slot. The buffer distribution at these instants is important as it is used by the output process and the waiting time distribution algorithms presented in the next chapters.

Chapter 4

Computation of the Output Process

4.1 Introduction

In Section 3.3 the occurrence of empty slots on the forward bus at the input of a DQDB node bus was modeled by a renewal process. The renewal period is the number of busy slots between successive empty slots. The distribution of this period was assumed to be general. Since the node may transmit in some of the empty slots, the distribution of the renewal period at the output is different from the input. Successive transmissions by the node concatenates some sequences of busy slots to make the appear as one sequence in the output. The output process is derived for the following three cases

- Assuming successive transmissions are geometrically distributed and independent of busy slot sequences at the input.
- Assuming general distribution for successive transmissions and independence of busy slot sequences at the input.

- Assuming general distribution for successive transmissions and dependence on busy slot sequences at the input.

The three cases are documented here, as this was the initial flow of thought, to make it easier for the reader understand the seemingly complicated matrix recurrence relations of the last exact algorithm.

4.2 The Output Process

Let T_r^i (T_r^o) be the random variable denoting time until the next renewal at the input (output) respectively. Also let N_b^i (N_b^o) be the random variable denoting the number of busy slots between renewals at the input (output) respectively. Clearly $T_r^i = N_b^i + 1$ and at the output we have $T_r^o = N_b^o + 1$. Note that T_r^i and T_r^o are strictly positive random variables, cannot take zero values, while the number of busy slots between renewals can be zero.

Let N_T be the random variable denoting the number of transmissions in successive renewals (empty slots at the input). Then

$$T_r^o = N_b^i(1) + N_b^i(2) + \ldots N_b^i(N_T + 1) + N_T \tag{4.1}$$

where $N_b^i(1), \ldots . N_b^i(N_T + 1)$ are i.i.d, due to our assumption that the arrival of empty slots on the forward bus is a renewal process, with mass function f_{N_i} and transform ${}_zN_i(z)$. Note that f_{N_i} and ${}_zN_i(z)$ are the mass function and transform of N_b^i respectively. Let ${}_zT_o(z)$ and ${}_zN_T(z)$ denote the transforms of T_r^o and N_T. Then ${}_zT_o(z)$ is given by

$${}_zT_o(z) = {}_zN_i(z)\ {}_zN_T(z\ {}_zN_i(z)) \tag{4.2}$$

Note that $z\ {}_zN_i(z)$ is just the transform of f_{idle} of Section 3.3. A simple algorithm to compute equation (4.2) is to compute the polynomial ${}_zN_T(z)$ and substitute the polynomial in the equation (4.2). If we assume that N_t has a geometric distribution with a parameter p equal to the probability of transmission in a slot. In that case ${}_zT_o(z)$ is given by

$${}_zT_o(z) = \frac{p\ {}_zN_i(z)}{1 - (1 - p)z\ {}_zN_i(z)} \tag{4.3}$$

Although this is simple in computation, it destroys the merits of the model which takes into account the correlations on the bus. Even if we compute $_zN_T(z)$ and substitute it in equation (4.2) the result is still approximate since in deriving equation (4.2), see Appendix C.1, we assumed that N_T is independent of N_b^i, $i \geq 1$, which is not true. In Section 4.3 an algorithm to compute the polynomial $_zN_T(z)$ is outlined, which can then be used to compute $_zT_o(z)$ approximately. In Section 4.4 an exact algorithm is outlined to compute $_zT_o(z)$.

4.3 Approximate Algorithm

The indicator random variable TR_n is defined such that it takes the value one if there is transmission by the tagged node in the empty slot at the end of the n^{th} renewal period and zero otherwise. The Markov renewal process $Q(x)$ of equation 3.23 is then expanded to

$$\begin{aligned} \hat{Q}_{ii'jj'kk'll'}(x) &= P\{I_n = i', CD_n = j', RQ_n = k', TR_n = l', \tau_n \leq x \mid \\ &\qquad I_{n-1} = i, CD_{n-1} = j, RQ_{n-1} = k, TR_{n-1} = l\} \end{aligned} \tag{4.4}$$

with transition matrix

$$\hat{\mathbf{Q}}(\mathrm{x}) = \begin{vmatrix} \hat{\mathbf{B}}_0(x) & \hat{\mathbf{B}}_1(x) & \hat{\mathbf{B}}_2(x) & \hat{\mathbf{B}}_3(x) & \cdots \\ \hat{\mathbf{C}}_0(x) & \hat{\mathbf{A}}_1(x) & \hat{\mathbf{A}}_2(x) & \hat{\mathbf{A}}_3(x) & \cdots \\ 0 & \hat{\mathbf{A}}_0(x) & \hat{\mathbf{A}}_1(x) & \hat{\mathbf{A}}_2(x) & \cdots \\ 0 & 0 & \hat{\mathbf{A}}_0(x) & \hat{\mathbf{A}}_1(x) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \end{vmatrix} \tag{4.5}$$

The submatrices in Equation 4.5 are defined as follows

$$\hat{\mathbf{A}}_0(\mathbf{x}) = \begin{pmatrix} \mathbf{A}_0(x) & 0 \\ \mathbf{A}_0(x) & 0 \end{pmatrix} \tag{4.6}$$

$$\hat{\mathbf{A}}_v(\mathbf{x}) = \begin{pmatrix} \boldsymbol{\Upsilon}_v(x) & \boldsymbol{\Omega}_v(x) \\ \boldsymbol{\Upsilon}_v(x) & \boldsymbol{\Omega}_v(x) \end{pmatrix} \tag{4.7}$$

$$\hat{\mathbf{C}}_0(\mathbf{x}) = \begin{pmatrix} \mathbf{C}_0(x) & \mathbf{0} \\ \mathbf{C}_0(x) & \mathbf{0} \end{pmatrix} \tag{4.8}$$

$$\hat{\mathbf{B}}_0(\mathbf{x}) = \begin{pmatrix} \boldsymbol{\Psi}_1(x) \circ \boldsymbol{\Theta}(x) & \boldsymbol{\Gamma}(x) \\ \boldsymbol{\Psi}_1(x) \circ \boldsymbol{\Theta}(x) & \boldsymbol{\Gamma}(x) \end{pmatrix} \tag{4.9}$$

$$\hat{\mathbf{B}}_v(\mathbf{x}) = \begin{pmatrix} \sum_{i=1}^{v+1} \boldsymbol{\Psi}_i(x) \circ \dot{\boldsymbol{\Xi}}_{v-i+1}(x) & \sum_{i=1}^{v+1} \boldsymbol{\Psi}_i(x) \circ \ddot{\boldsymbol{\Xi}}_{v-i+1}(x) \\ \sum_{i=1}^{v+1} \boldsymbol{\Psi}_i(x) \circ \dot{\boldsymbol{\Xi}}_{v-i+1}(x) & \sum_{i=1}^{v+1} \boldsymbol{\Psi}_i(x) \circ \ddot{\boldsymbol{\Xi}}_{v-i+1}(x) \end{pmatrix} \tag{4.10}$$

The first row (column) in all the submatrices denotes transitions from (to) transmission states while the second denotes transitions from (to) idle states. The Markov renewal matrix $\hat{\mathbf{Q}}(\mathbf{x})$ can be rearranged in a different way to help observe its structural properties as follows

$$\hat{\mathbf{Q}}(\mathbf{x}) = \begin{pmatrix} \hat{\mathbf{Q}}_{11}(\mathbf{x}) & \hat{\mathbf{Q}}_{12}(\mathbf{x}) \\ \hat{\mathbf{Q}}_{21}(\mathbf{x}) & \hat{\mathbf{Q}}_{22}(\mathbf{x}) \end{pmatrix} \tag{4.11}$$

where

$$\hat{\mathbf{Q}}_{11}(x) = \hat{\mathbf{Q}}_{21}(x) = \begin{vmatrix} \hat{\mathbf{Q}}_1^{00}(x) & \hat{\mathbf{Q}}_1^{01}(x) & \hat{\mathbf{Q}}_1^{02}(x) & \hat{\mathbf{Q}}_1^{03}(x) & \cdots \\ \mathbf{C}_0(x) & \boldsymbol{\Upsilon}_1(x) & \boldsymbol{\Upsilon}_2(x) & \boldsymbol{\Upsilon}_3(x) & \cdots \\ \mathbf{0} & \mathbf{A}_0(x) & \boldsymbol{\Upsilon}_1(x) & \boldsymbol{\Upsilon}_2(x) & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_0(x) & \boldsymbol{\Upsilon}_1(x) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \end{vmatrix} \tag{4.12}$$

and $\hat{\mathbf{Q}}_1^{00}(x) = \boldsymbol{\Psi}_1(x) \circ \boldsymbol{\Theta}(x)$ and $\hat{\mathbf{Q}}_1^{0v}(x) = \sum_{i=1}^{v+1} \boldsymbol{\Psi}_i(x) \circ \dot{\boldsymbol{\Xi}}_{v-i+1}(x), v > 1.$

$$\hat{\mathbf{Q}}_{12}(x) = \hat{\mathbf{Q}}_{22}(x) = \begin{vmatrix} \hat{\mathbf{Q}}_2^{00}(x) & \hat{\mathbf{Q}}_2^{01}(x) & \hat{\mathbf{Q}}_2^{02}(x) & \hat{\mathbf{Q}}_2^{03}(x) & \cdots \\ \mathbf{0} & \boldsymbol{\Omega}_1(x) & \boldsymbol{\Omega}_2(x) & \boldsymbol{\Omega}_3(x) & \cdots \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Omega}_1(x) & \boldsymbol{\Omega}_2(x) & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\Omega}_1(x) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \end{vmatrix} \tag{4.13}$$

and $\hat{\mathbf{Q}}_2^{00}(x) = \Gamma(x)$ and $\hat{\mathbf{Q}}_2^{0v}(x) = \sum_{i=1}^{v+1} \Psi_i(x) \circ \bar{\Xi}_{v-i+1}(x), v > 1$.

In order to compute ${}_zN_T(z)$ we consider the embedded chain at ∞. Let S^T be the set of states where $TR = 1$ and S^E be the set of states where $TR = 0$. Then $\hat{\mathbf{Q}}_{11}$ represent transitions from S^T to S^T and $\hat{\mathbf{Q}}_{22}$ represent transitions from S^E to S^E. The steady state probabilities $\mathbf{x}$ of $\hat{\mathbf{Q}}$ are also partitioned such that $\mathbf{x} = [\mathbf{x}(1) \;\; \mathbf{x}(2)]$. $\hat{\mathbf{Q}}_{11}$ can be thought of as a transient chain with absorption taking place whenever it steps in S^E via $\hat{\mathbf{Q}}_{12}$. If $\hat{\mathbf{Q}}_{11}$, $\hat{\mathbf{Q}}_{12}$, $\hat{\mathbf{Q}}_{21}$, and $\hat{\mathbf{Q}}_{22}$ were finite we could have written

$$\begin{aligned} Pr\{N_T = k\} &= \mathbf{x}(2)\hat{\mathbf{Q}}_{21}\hat{\mathbf{Q}}_{11}^{k-1}\hat{\mathbf{Q}}_{12}\mathbf{e} \\ &= \mathbf{x}(2)\hat{\mathbf{Q}}_{11}^{k}\hat{\mathbf{Q}}_{12}\mathbf{e} \end{aligned} \quad (4.14)$$

where $\mathbf{e}$ is a column vector of ones with appropriate dimensions. This is the phase distribution except that the matrices are infinite, with a special case at the boundary

$$Pr\{N_T = 0\} = \mathbf{x}(2)\hat{\mathbf{Q}}_{22}\mathbf{e} = \mathbf{x}(2)\hat{\mathbf{Q}}_{12}\mathbf{e} \quad (4.15)$$

In the following we show an algorithm to compute equation (4.14). The steady state probability vectors are finite in practice. In other words, the probability that the Markov renewal process is at state i decreases as i increases. If we write

$$\mathbf{x}(1) = [\mathbf{x}_0(1), \mathbf{x}_1(1), \ldots, \mathbf{x}_{v_{max}}(1)] \quad (4.16)$$

$$\mathbf{x}(2) = [\mathbf{x}_0(2), \mathbf{x}_1(2), \ldots, \mathbf{x}_{v_{max}}(2)] \quad (4.17)$$

Where we define the max level v_{max} as the maximum level such that

$$\mathbf{x}(1)\mathbf{e} + \mathbf{x}(2)\mathbf{e} > 1 - \epsilon$$

where ϵ is a small value that can be set to the desired precision of computation. The summation matrices are denoted by a bar to simplify notation. So we have $\bar{\Omega}_v = \sum_{n=v}^{\infty} \Omega_n$, $\bar{\Upsilon}_v = \sum_{n=v}^{\infty} \Upsilon_n$, and $\bar{\mathbf{Q}}_i^{0v} = \sum_{n=v}^{\infty} \hat{\mathbf{Q}}_i^{0n}$.

The first step is to compute the conditional probability vectors $\mathbf{P}_t(k)$ where

$$\mathbf{P}_t(0) = \hat{\mathbf{Q}}_{22}\mathbf{e} \tag{4.18}$$

$$\mathbf{P}_t(1) = \hat{\mathbf{Q}}_{21}\hat{\mathbf{Q}}_{12}\mathbf{e} \tag{4.19}$$

$$\mathbf{P}_t(k) = \hat{\mathbf{Q}}_{21}\hat{\mathbf{Q}}_{11}^{k-1}\hat{\mathbf{Q}}_{12}\mathbf{e} \qquad k > 1 \tag{4.20}$$

Each vector can be partitioned to $v_{max}+1$ subvectors starting with the 0^{th} subvector. Subvectors with indices greater than v_{max} are ignored as they will be multiplied by zeros when these vectors are multiplied by $x(2)$. The subvectors of $\mathbf{P}_t(0)$ are given by

$$\mathbf{P}_t(0,0) = \bar{\mathbf{Q}}_2^{00}\mathbf{e} \tag{4.21}$$

$$\mathbf{P}_t(0,i) = \bar{\mathbf{\Omega}}_1\mathbf{e} \qquad i > 0 \tag{4.22}$$

and the subvectors of $\mathbf{P}_t(1)$ are given by

$$\mathbf{P}_t(1,0) = \hat{\mathbf{Q}}_1^{00}\bar{\mathbf{Q}}_2^{00}\mathbf{e} + \bar{\mathbf{Q}}_1^{01}\bar{\mathbf{\Omega}}_1^{01}\mathbf{e} \tag{4.23}$$

$$\mathbf{P}_t(1,1) = \mathbf{C}_0\bar{\mathbf{Q}}_2^{00}\mathbf{e} + \bar{\mathbf{\Upsilon}}_1\bar{\mathbf{\Omega}}_1\mathbf{e} \tag{4.24}$$

$$\mathbf{P}_t(1,i) = (\mathbf{A}_0 + \bar{\mathbf{\Upsilon}}_1)\bar{\mathbf{\Omega}}_1\mathbf{e} \qquad i > 1 \tag{4.25}$$

and in general for $k > 1$

$$\mathbf{P}_t(k,0) = \sum_{i=0}^{v_{max}} \mathbf{Q}_1^{0i}\mathbf{P}_t(k-1,i) \tag{4.26}$$

$$\mathbf{P}_t(k,1) = \mathbf{C}_0\mathbf{P}_t(k-1,0) + \sum_{i=1}^{v_{max}} \mathbf{\Upsilon}_i\mathbf{P}_t(k-1,i) \tag{4.27}$$

$$\mathbf{P}_t(k,2) = \mathbf{A}_0\mathbf{P}_t(k-1,1) + \sum_{i=2}^{v_{max}} \mathbf{\Upsilon}_{i-1}\mathbf{P}_t(k-1,i) \tag{4.28}$$

and for $j \geq 2$

$$\mathbf{P}_t(k,j) = \mathbf{A}_0\mathbf{P}_t(k-1,j-1) + \sum_{i=j}^{v_{max}} \mathbf{\Upsilon}_{i-j+1}\mathbf{P}_t(k-1,i) \tag{4.29}$$

Note that

$$\mathbf{P}_t(k,j) = (\mathbf{A}_0 + \tilde{\mathbf{\Upsilon}}_1)^k \bar{\mathbf{\Omega}}_1 \mathbf{e} \qquad \forall j, k < j \tag{4.30}$$

Therefore, we need only compute the first $k+2$ subvectors of each vector $\mathbf{P}_t(k)$. Hence

$$Pr\{N_T = k\} = \sum_{i=0}^{k} \mathbf{x}_i(2)\mathbf{P}_t(k,i) + (\sum_{i=k+1}^{v_{max}} \mathbf{x}_i(2))\mathbf{P}_t(k,k+1) \tag{4.31}$$

Equation 4.31 constitutes the algorithm and we need only store one vector of the $P_t(k)$ at a time to compute the next one.

4.4 Exact Algorithm

In Section 4.3 the vectors $P_t(k)$ were computed by a recursive scheme and were used to compute $Pr\{N_T = k\}$. $Pr\{N_T = k\}$ can then be fitted to a polynomial to give the transform $_zN_T(z)$. $_zN_T(z)$ can then be used in Equation 4.2 to compute the generating function of the renewals at the output.

In this section a direct algorithm is outlined to compute $Pr\{T_r^o = k\}$. The algorithm takes into account that N_T depends on N_b^i's in Equation 4.1.

A third index is added to the vectors $\mathbf{P}_t(k)$. The new index denotes the number of slots, so the $\mathbf{P}_t(k,i,x)$ is the i^{th} conditional probability subvector of k successive transmissions and renewal period equal to x slots. We can then write

$$\begin{aligned} \mathbf{P}_t(0,0,x) &= {}_m\bar{\mathbf{Q}}_2^{00}(x)\mathbf{e} & (4.32) \\ \mathbf{P}_t(0,i,x) &= {}_m\bar{\mathbf{\Omega}}_1(x)\mathbf{e} \qquad \forall i > 0 & (4.33) \end{aligned}$$

Let the minimum x at which ${}_m\bar{\mathbf{Q}}_1^{00}(x)$, ${}_m\bar{\mathbf{Q}}_2^{00}(x)$, ${}_m\bar{\mathbf{\Omega}}_1(x)$, ${}_m\bar{\mathbf{\Upsilon}}_1(x)$, ${}_m\mathbf{A}_0(x)$ and ${}_m\mathbf{C}_0(x)$ will all vanish be x_{max}.

Summing over x we can write

$$\begin{aligned}
\mathbf{P}_t(0,0) &= \sum_{x=1}^{x_{max}} {}_m\bar{\mathbf{Q}}_2^{00}(x)\mathbf{e} = \bar{\mathbf{Q}}_2^{00}(x)\mathbf{e} && (4.34)\\
\mathbf{P}_t(0,i) &= \sum_{x=1}^{x_{max}} {}_m\bar{\mathbf{\Omega}}_1(x)\mathbf{e} = \bar{\mathbf{\Omega}}_1(x)\mathbf{e}\forall i > 0 && (4.35)
\end{aligned}$$

and for $k = 1$ we have

$$\begin{aligned}
\mathbf{P}_t(1,0,x) &= {}_m\hat{\mathbf{Q}}_1^{00}(x) \circ {}_m\bar{\mathbf{Q}}_2^{00}(x)\mathbf{e} + {}_m\bar{\mathbf{Q}}_1^{01}(x) \circ {}_m\bar{\mathbf{\Omega}}_1(x)\mathbf{e} && (4.36)\\
\mathbf{P}_t(1,1,x) &= {}_m\mathbf{C}_0(x) \circ {}_m\bar{\mathbf{Q}}_2^{00}(x)\mathbf{e} + {}_m\bar{\mathbf{\Upsilon}}_1(x) \circ {}_m\bar{\mathbf{\Omega}}_1(x)\mathbf{e} && (4.37)\\
\mathbf{P}_t(1,i,x) &= ({}_m\mathbf{A}_0(x) + {}_m\bar{\mathbf{\Upsilon}}_1(x)) \circ {}_m\bar{\mathbf{\Omega}}_1(x)\mathbf{e} \qquad i > 1 && (4.38)
\end{aligned}$$

and the general recurrence equations for $k > 1$ are given by

$$\begin{aligned}
\mathbf{P}_t(k,0,x) &= \sum_{i=0}^{v_{max}} {}_m\hat{\mathbf{Q}}_1^{0i}(x) \circ \mathbf{P}_t(k-1,i,x) && (4.39)\\
\mathbf{P}_t(k,1,x) &= {}_m\mathbf{C}_0(x) \circ \mathbf{P}_t(k-1,0,x) + && (4.40)\\
&\quad \sum_{i=1}^{v_{max}} {}_m\mathbf{\Upsilon}_i(x) \circ \mathbf{P}_t(k-1,i,x) && (4.41)\\
\mathbf{P}_t(k,j,x) &= {}_m\mathbf{A}_0(x) \circ \mathbf{P}_t(k-1,j-1,x) + && (4.42)\\
&\quad \sum_{i=j}^{v_{max}} {}_m\mathbf{\Upsilon}_{i-j+1}(x) \circ \mathbf{P}_t(k-1,i,x) \qquad j > 1 && (4.43)
\end{aligned}$$

For $\mathbf{P}_t(k,j,x)$ the maximum x is $x_{max}(k)$ where $x_{max}(k) = (k+1)x_{max}$.

Note that $\mathbf{P}_t(k,j,x) = 0, \forall k > x$. In other words we cannot have k transmissions in $x < k$ slots. So we can write

$$Pr\{T_r^o = t\} = \sum_{i=0}^{v_{max}} \mathbf{x}_i(2) \sum_{k=0}^{t} \mathbf{P}_t(k,i,t) \tag{4.44}$$

4.5 Summary

In this chapter an algorithm to compute the distribution of the output process is presented. This algorithm is important to be able to iteratively solve a network wide model. We start by the most upstream node (node 1). For node 1 the input process is trivial since with probability one all slots are empty on the forward bus. The model of Chapter 3 is used to compute the buffer distribution of node 1 then the output process is computed. The output process of node 1 becomes the input of node 2 and we repeat this sequence until we get the buffer distributions for all of the nodes in the network.

To be able to solve for the buffer occupancy at each node, the probability of a request arriving on the reverse bus at each node has to be computed see Section 3.3.2. We adopt the approximation in [CGL91a], see Section 2.5.2. Hence for a node i in a network of n nodes

$$Pr\{node\ i\ observes\ no\ request\ in\ a\ slot\} = e^{-T_s \sum_{j=i+1}^{n} \lambda_j} \tag{4.45}$$

where T_s is the duration of a slot and λ_i is the rate of arrivals at node i. This is the approximation we used in our implementation to generate network wide results. Note that the model is exact per node given the assumptions of input to the node, but it is approximate for the network wide application.

Chapter 5

Waiting Time Distribution

5.1 Introduction

In this chapter an algorithm for the waiting time distribution is outlined. A segment is tagged and the waiting time until it is transmitted in a slot on the forward bus is studied. Recall that a node observes activity on the forward bus as a discrete renewal process. Renewal takes place when the node observes an empty slot. The tagged segment arrives at a point in time in the n^{th} renewal period as shown in Fig 5.1.

The renewal period in which the tagged segment arrives has a different distribution, see "Paradox of Residual Life" in [Kle75, pages 169-174]. Intuitively, it is more likely that the arriving segment fall in a long renewal period than a short renewal period. The duration of the renewal period

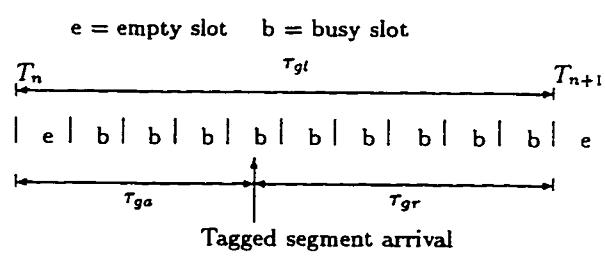

Figure 5.1: Structure of the renewal period of the tagged segment arrival

$\tau_{gl} = T_{n+1} - T_n$ is divided into two parts, as shown in Fig 5.1, such that $\tau_{gl} = \tau_{ga} + \tau_{gr}$. In terms of renewal theory, τ_{gl} is the life time, τ_{ga} is the age and τ_{gr} is the residual life.

Suppose that there were i segments in the queue in front of the tagged segment at T_{n+1}. Then the tagged segment will wait until it observes i transmissions. The tagged segment will further wait until all the requests accumulated during the i^{th} countdown time are satisfied and then get transmitted in the next empty slot. The waiting time of the tagged segment is then the sum of the residual life and the total countdown time, where the total countdown time is the time from T_{n+1} until the segment get transmitted.

5.2 Transition probabilities for the residual life

The transition probabilities of the state variables of the DQDB node from T_n to T_{n+1} are defined by the sequence of matrices $\mathbf{R}_I(i, j, k, x, y)$ if there is no transmission at T_{n+1} and by $\mathbf{R}_T(i, j, k, x, y)$ if there is transmission at T_{n+1}. Where

- i is the number of segments in the node at T_n.
- j is the number of segments in the node at T_{n+1}, not including the one transmitted at T_{n+1} if transmission took place.
- k is the number of segments to be transmitted before the tagged segment. It is set to -1 if the tagged segment is transmitted at T_{n+1}.
- x is the number of slots in residual life.
- y is the number of slots in the age not counting the slot of the arrival of the tagged segment.

So the total renewal period is $x + y + 1$ slots. The function ϱ to compute the probability of arrivals. during this special renewal period is defined by

$$\varrho(i_1, i_2, x, y) = \int_{xT_s}^{(x+1)T_s} \frac{1}{T_s} p(t; i_1) p((x + y + 1)T_s - t; i_2) dt \tag{5.1}$$

where $p(t;k) = \frac{(\lambda t)^k e^{-\lambda t}}{k!}$ and T_s is the slot duration. The function ϱ computes the probability of i_1 and i_2 arrivals before and after the tagged segment arrival respectively, with the condition that the tagged segment will arrive during the $(x+1)^{th}$ slot. This integral can be computed, see Appendix B.3, using

$$\begin{aligned}\varrho(i_1, i_2, x, y) = \quad & e^{-\lambda(x+y+1)T_s}(\lambda T_s)^{(i_1+i_2)} * \\ & \sum_{l=0}^{i_2}\sum_{k=0}^{i_1} \frac{x^k y^l}{k!l!(i_1+i_2-k-l+1)!}\end{aligned} \tag{5.2}$$

The matrix $\mathbf{R}_T(i,j,k,x,y) = 0$ if $j < k+1 < i$, $\forall i > 0, \forall j > 0$. That is, i is decremented by the transmission at T_n and incremented by the tagged segment arrival if there are no other arrivals. hence minimum j is equal to i. Also the tagged segment will not observe k transmissions after residual time if there are $j < k+1$ segments in the node. Finally k cannot be less than $i-1$ otherwise the segment is transmitted before the ones ahead of it in the queue. The tagged segment can be transmitted at T_{n+1} only if the node was empty at T_n, i=0. This is the only case where k is set to -1.

Similarly the matrix $\mathbf{R}_I(i,j,k,x,y) = 0$ if $j-1 < k < i$, $\forall i > 0, \forall j > 0$. Note that for both matrix sequences j can not be equal to 0 because the tagged segment is counted in j. The only special case for this is at the boundary level, where the tagged segment is the only segment in the queue and is transmitted at T_{n+1}. In what follows the matrices elements will be defined and the elements will be referred to by $e(m, m', l, l')$. Where m, l and m', l' are the values of countdown and request counters at T_n and T_{n+1} respectively. The two sequence of matrices are first computed according to the structure given in Equations 3.24 and 3.25 then they are passed to reduction and relabeling algorithms of Section 3.5.1 and Section 3.5.2.

5.2.1 Transitions from upper levels $i > 0$

The matrix $\mathbf{R}_T(i,j,k,x,y)$ has elements $e(m,m',l,l')$, for $i > 0$, defined by

$$e(0,m',l,0) = f_{creq}(y+x+1, m'-l, r_{max}-l)*$$
$$\varrho(k-i+1, j-k-1, x, y) f_{init}(x) f_{dlife}(y+1) \tag{5.3}$$

In this case, at T_{n+1} a segment in the queue is transmitted and the tagged segment waits for a countdown time after the residual time. The countdown counter is equal to zero at T_n and the request counter is zero at T_{n+1}. and for the other cases

$$e(m,m',l,l') = 0 \qquad m > 0 \;\; or \;\; l' > 0. \tag{5.4}$$

The matrix $\mathbf{R}_I(i,j,k,x,y)$, $i > 0$, has elements

$$e(m,m',l,l') = 0 \qquad m = 0 \;\; or \;\; m' \neq m-1. \tag{5.5}$$

For the case where $m > 0$ and $m' = m - 1$

$$e(m,m',l,l') = f_{creq}(y+x+1, l'-l, r_{max}-l-m+1)*$$
$$\varrho(k-i, j-k-1, x, y) f_{init}(x) f_{dlife}(y+1) \tag{5.6}$$

This is the case that an empty slot is left to traverse the bus to the downstream nodes.

5.2.2 Transitions at the boundary level

Transitions from the boundary level are transitions from a set of states where the node has no segments in it. In this set of states one state has the request counter equal zero and the other states have the request counter greater than zero. For the states where the request counter is greater than zero at T_n there won't be any transmission at T_{n+1}. As for the state where the request counter is equal to zero at T_n the process can transit into one of three situations at T_{n+1} as follows

- The node will not transmit any segment at T_{n+1}. This can happen if a request is registered on the reverse bus before the arrival of any segment.

- The node will transmit the tagged segment at T_{n+1}. In this case the total waiting time is equal to the residual time. This can happen if no requests or segments arrive before the tagged segment.

- The node will transmit a segment other than the tagged segment. This can happen if a segment arrives before any request is registered on the reverse bus and before the arrival of the tagged segment.

We define a transient Markov chain to model this behaviour at the boundary until the beginning of the slot in which the tagged segment arrives. Then we define transition probability matrices $\mathbf{R}_T'$ and $\mathbf{R}_I'$ to compute transitions from the beginning of the slot in which the tagged segment arrives until T_{n+1}. Finally we compute $\mathbf{R}_T$ and $\mathbf{R}_I$.

5.2.3 A Markov chain to model transitions from the idle state

The chain has states similar to the Markov renewal process in equation (3.23). The Markov chain is used to compute the state probabilities at the beginning of the slot in which the tagged segment arrives. Note that the Markov renewal process in 3.23 defines transition probabilities for successive empty slots while the chain we are defining in this section defines transitions for successive busy slots in the renewal period. Let the states of the chain be denoted by $S(L, CD, RQ)$ where L is the level, CD is the count-down counter and RQ is the request counter. Let the state probability vector at time n be given by $\pi_s(n)$. The initial probability vector, at T_n, of this chain is given by $\pi_s(0) = [1, 0, 0, \ldots]$. In other words the chain is started in state $S(0, 0, 0)$. The state probability vector at time n is partitioned with respect to levels so that

$$\pi_s(n) = [\pi_s(n, 0), \pi_s(n, 1), \pi_s(n, 2), \ldots]$$

where $\pi_s(n,i)$ is the state probability vector at the n^{th} transition and the i^{th} level. Remember that in this model levels denote the number of segments queued in the node. Note that the states $S(0, CD, RQ)$ where $CD > 0$ are never visited. Let

$$q(i) = \begin{cases} q & if \;\; i < r_{max} \\ 1 & if \;\; i = r_{max} \end{cases}$$

and

$$p(i) = \begin{cases} p & if \;\; i < r_{max} \\ 0 & if \;\; i = r_{max} \end{cases}$$

where p is the probability of a request on the reverse bus and q is the probability of no request. The transition probability matrix is sparse, and will be defined by the following equations.

$$Pr\{S(0,0,i_n)|S(0,0,i_{n-1})\} =$$
$$\begin{cases} q(i_{n-1})\phi_0(1) & if \;\; i_n = i_{n-1} \\ p(i_{n-1})\phi_0(1) & if \;\; i_n = i_{n-1}+1 \\ 0 & otherwise \end{cases}$$

and for $k_n > 0$ we have

$$Pr\{S(k_n,j_n,i_n)|S(0,0,i_{n-1})\} =$$
$$\begin{cases} q(i_{n-1})\phi_{k_n}(1) & if \;\; j_n = i_{n-1}, i_n = 0 \\ p(i_{n-1})\phi_{k_n}(1) & if \;\; j_n = i_{n-1}+1, i_n = 0 \\ 0 & otherwise \end{cases}$$

and for $k_{n-1} > 0$ we have

$$Pr\{S(k_n,j_n,i_n)|S(k_{n-1},j_{n-1},i_{n-1})\} =$$

$$\begin{cases} q(i_{n-1})\phi_{k_n-k_{n-1}}(1) & if \quad i_n = i_{n-1}, j_n = j_{n-1} \\ p(i_{n-1})\phi_{k_n-k_{n-1}}(1) & if \quad i_n = i_{n-1}+1, j_n = j_{n-1} \\ 0 & otherwise \end{cases}$$

5.2.4 The transition matrices $\mathbf{R}'_T(i,j,k,y)$ and $\mathbf{R}'_I(i,j,k,y)$.

These two sequences of matrices define transitions from the beginning of the slot in which the tagged segment arrives until T_{n+1} given that the residual time is equal to y slots. The parameter i denotes the number of segments queued in the node at the beginning of the slot in which the tagged segment arrives. The parameters j and k are the numbers of segments queued in the node and the numbers of segments ahead of the tagged segment at T_{n+1} respectively. y is the residual life in slots until T_{n+1}. A library routine can easily compute the matrices given the values of the parameters i, j, k, y based on the following definitions.

The matrix $\mathbf{R}'_T(0,0,-1,y)$ has elements $e(m,m',l,l')$ defined by

$$e(m,m',l,l') = 0 \qquad for \quad m > 0 \ or \ m' > 0 \ or \ l > 0 \tag{5.7}$$

$$e(0,0,0,l') = q\, f_{creq}(y,l',r_{max})\varrho(0,0,0,y) \tag{5.8}$$

The matrix $\mathbf{R}'_T(0,j,-1,y)$ where $j > 0$ has elements $e(m,m',l,l')$ defined by

$$e(m,m',l,l') = 0 \qquad for \quad m > 0 \ or \ l > 0 \ or \ l' > 0 \tag{5.9}$$

$$e(0,m',0,0) = q\, f_{creq}(y,m',r_{max})\varrho(0,j,0,y) \tag{5.10}$$

The matrix $\mathbf{R}'_T(0,j,k,y)$ where $k > 0$ and $j > 0$ has elements $e(m,m',l,l')$ defined by

$$e(m,m',l,l') = 0 \qquad for \quad m > 0 \ or \ l > 0 \ or \ l' > 0 \tag{5.11}$$

$$e(0,m',0,0) = q\, f_{creq}(y,m',r_{max})\varrho(k+1,j-k-1,0,y) \tag{5.12}$$

Note that in this case $k > 0$ and $j = 0$ is an impossible situation. The matrix $\mathbf{R}'_T(i,j,k,y)$ where $i > 0$ has elements $e(m,m',l,l')$ defined by

$$e(m,m',l,l') = 0 \qquad for \quad m > 0 \ or \ l' > 0 \tag{5.13}$$

$$e(0,m',l,0) = f_{creq}(y+1,m'-l,r_{max}-l)\varrho(k-i+1,j-k-1,0,y) \tag{5.14}$$

Note that in this case $k > 0$ and $j = 0$ is an impossible situation. The matrix $\mathbf{R}_f'(0,j,k,y)$ has elements $e(m,m',l,l')$ defined by

$$e(m,m',l,l') = 0 \qquad for \;\; m > 0 \tag{5.15}$$

$$e(0,m',l,l') = \begin{cases} q\, f_{creq}(y,l',r_{max}-l+1)\varrho(k,j-k-1,0,y) \\ \quad if \;\; m' = l-1, l' > 0 \\ p\, f_{creq}(y,l',r_{max}-l+1)\varrho(k,j-k-1,0,y) \\ \quad if \;\; m' = l \\ 0 \quad otherwise \end{cases}$$

The matrix $\mathbf{R}_f'(i,j,k,y)$ has elements $e(m,m',l,l')$ defined by $e(0,m',l,l') = 0$. For $m > 0$ we have

$$e(m,m',l,l') = \begin{cases} f_{creq}(y+1,l'-l,r_{max}-l-m+1)\varrho(k-i,j-k-1,0,y) \\ \quad if \;\; m' = m-1 \\ 0 \quad otherwise \end{cases}$$

5.2.5 The matrix $\mathbf{R}_T(0,j,k,x,y)$

The matrix $\mathbf{R}_T(0,j,-1,x,y)$

In this case $k = -1$. In other words the segment arrives to find itself the first in the queue and gets transmitted at T_{n+1}. So the waiting time is equal to the residual time. The matrix $\mathbf{R}_T(0,j,-1,x,y)$ has zero elements except for the first row. If we denote the first row by the vector $\mathbf{r}_1$ then

$$\mathbf{r}_1 = \pi_s(x,0)\mathbf{R}_T'(0,j,-1,y)f_{init}(x)f_{dlife}(y+1)$$

Note that $\mathbf{r}_1$ is the vector of transition probabilities from the state with both request counter and count-down counter equal to zero.

The matrix $\mathbf{R}_T(0, j, k, x, y)$ where $k \geq 0$

The matrix $\mathbf{R}_T(0, j, k, x, y)$ has zero elements except for the first row. If we denote the first row by the vector $\mathbf{r}_1$ then

$$\mathbf{r}_1 = \sum_{i=0}^{min(i_{max},k+1)} \pi_s(x, i)\mathbf{R}_T^{'}(i, j, k, y) f_{init}(x) f_{dlife}(y + 1)$$

where i_{max} is the max level at which $\pi_s(x, i)$ will vanish. Note that $\mathbf{r}_1$ is the vector of transition probabilities from the state with both request counter and count-down counter equal to zero.

The matrix $\mathbf{R}_I(0, j, k, x, y)$

Denote the first row , elements with n=0 and m=0, by the vector $\mathbf{r}_1$ then

$$\mathbf{r}_1 = \sum_{i=1}^{min(i_{max},k+1)} \pi_s(x, i)\mathbf{R}_I^{'}(i, j, k, y) f_{init}(x) f_{dlife}(y + 1)$$

where i_{max} is the max level at which $\pi_s(x, i)$ will vanish. Note that $\mathbf{r}_1$ is the vector of transition probabilities from the state with both request counter and count-down counter equal to zero.

Elements of the matrix with $n > 0$ and $m = 0$ are given by

$$\begin{aligned} & e(0, m^{'}, n, n^{'}) = \\ & \quad \sum_{x_1=0}^{x} f_{creq}(x_1, m^{'} + 1 - n, r_{max} - n) * \\ & \quad f_{creq}(y + 1 + x - x_1, n^{'}, r_{max} - m^{'}) f_{init}(x) * \\ & \quad f_{dlife}(y + 1) \varrho_1(k, j - k - 1, x_1, x, y) \end{aligned} \tag{5.16}$$

where ϱ_1 is given by

$$\varrho_1(i, j, x_1, x - x_1, y) = \begin{cases} e^{-\lambda x} \varrho(i, j, 0, y) \quad if x = x_1 \\ \\ e^{-\lambda x_1}(\varrho(i, j, x - x_1, y) - \\ e^{-\lambda} \varrho(i, j, x - x_1 - 1, y)) \quad if x = x_1 \end{cases} \tag{5.17}$$

The elements are equal to zero when $m > 0$.

5.3 Transition probabilities for the total countdown time

The probability that the tagged segment is transmitted after x slots from T_{n+1}, given that at T_{n+1} the node was at level j and there were $k < j$ segments in the queue ahead of the tagged segment is given by the column vector $\mathbf{D}(j,k,x)$. Note that $\mathbf{D}(j,k,x)$ is defined for $j > 1$ only since j includes the tagged segment. $\mathbf{D}(j,k,x)$ has dimensions $(\frac{(r_{max}^2+3r_{max}+2)}{2}, 1)$ which is the dimensions of the submatrices after reduction and relabeling, see Sections 3.5.1 and 3.5.2. Note that k represents the number of transitons in S^T before the transmission of the tagged segment. The sojourn time between successive transitions is computed by conditioning on the number of transitions into S^E.

The following reccurrence relations can be easily deduced from the matrices in Equations 4.12 and 4.13.

Computation of the vectors for $k = 0$

We start by computing $\mathbf{D}(j,0,x)$. This is the case where the tagged segment is the head of the queue. A fourth index l, that denotes the number of renewals without transmission observed by the tagged segment from T_{n+1} until the next renewal with transmission, is added to $\mathbf{D}(j,0,x)$. So that

$$\mathbf{D}(j,0,x) = \sum_{l=0}^{x} \mathbf{D}(j,0,x,l) \tag{5.18}$$

Note that l cannot be greater than x. In other words, in a period of x slots we cannot have $l > x$ transitions in S^E. By inspecting the matrix in Equation 4.12 we can write

$$\mathbf{D}(j,0,x,0) = \begin{cases} {}_m\mathbf{C}_0(x)\mathbf{e} + {}_m\bar{\mathbf{\Upsilon}}_1(x)\mathbf{e} & \text{if } j = 1 \\ {}_m\mathbf{A}_0(x)\mathbf{e} + {}_m\bar{\mathbf{\Upsilon}}_1(x)\mathbf{e} & \text{if } j > 1 \end{cases} \tag{5.19}$$

It is generally known that if a Markov renewal process of M/G/1 type is positive recurrent then $\mathbf{C}_0\mathbf{e} = \mathbf{A}_0\mathbf{e}$. The equality ${}_m\mathbf{C}_0(x)\mathbf{e} = {}_m\mathbf{A}_0(x)\mathbf{e}$ may not necessarily hold, but fortunately it holds for our case. So we can write Equation 5.19 as

$$\mathbf{D}(j,0,x,0) = {}_m(\mathbf{A}_0(x) + {}_m\bar{\mathbf{\Upsilon}}_1(x))\mathbf{e} \qquad \forall j \geq 1 \tag{5.20}$$

For $l > 0$ we have the following recursive equation which can be deduced by inspecting the matrix in Equation 4.13.

$$\mathbf{D}(j,0,x,l) = {}_m\bar{\Omega}_1(x) \circ \mathbf{D}(1,0,x,l-1) \tag{5.21}$$

Computation of the vectors for $k = 1$

For $k = 1$, the valid levels are $j > 1$ since the node have the tagged segment and one segment ahead of it in the queue. $\mathbf{D}(j,1,x)$ is given by

$$\mathbf{D}(j,1,x) = \sum_{l=0}^{x} \mathbf{D}(j,1,x,l) \tag{5.22}$$

where

$$\mathbf{D}(j,1,x,0) = ({}_m\mathbf{A}_0(x) + {}_m\bar{\Upsilon}_1(x)) \circ \mathbf{D}(1,0,x) \qquad \forall j \geq 2 \tag{5.23}$$

and for $l > 0$ we have

$$\mathbf{D}(j,1,x,l) = {}_m\bar{\Omega}_1(x) \circ \mathbf{D}(2,1,x,l-1) \tag{5.24}$$

Computation of the vectors for $k > 1$

For any value of k the vectors are defined for $j > k$ only. By induction we can compute $\mathbf{D}(j,k,x)$ for any $k > 1$, where

$$\mathbf{D}(j,k,x) = \sum_{l=0}^{x} \mathbf{D}(j,k,x,l) \tag{5.25}$$

and

$$\mathbf{D}(j,k,x,0) = ({}_m\mathbf{A}_0(x) + {}_m\bar{\Upsilon}_1(x)) \circ \mathbf{D}(k,k-1,x) \qquad \forall j > k \tag{5.26}$$

and for $l > 0$ we have

$$\mathbf{D}(j,k,x,l) = {}_m\bar{\Omega}_1(x) \circ \mathbf{D}(k+1,k,x,l-1) \tag{5.27}$$

5.4 Waiting Time Distribution

After the matrices $\mathbf{R}_T(i,j,k,x,y)$, $\mathbf{R}_I(i,j,k,x,y)$ and $\mathbf{D}(j,k,x)$ are computed as outlined in the previous Sections the waiting time distribution of the tagged packet is computed. Let $W(x) = Pr\{tagged\,segment\,waits\,for\,x\,slots\}$. Then

$$
\begin{aligned}
W(x) = \\
&\sum_{y=0}^{fidle_{max}} (\mathbf{x}_0(\sum_{j=0}^{v_{max}} \mathbf{R}_T(0,j,0,x,y)\mathbf{e} + \sum_{j=1}^{v_{max}} \mathbf{R}_I(0,j,0,x,y) \circ \mathbf{D}(j,0,x)) + \\
&\mathbf{x}_0 \sum_{k=1}^{x-1} \sum_{j=k+1}^{v_{max}} (\mathbf{R}_T(0,j,k,x,y) + \mathbf{R}_I(0,j,k,x,y)) \circ \mathbf{D}(j,k,x) + \\
&\sum_{k=0}^{x-1} \sum_{i=1}^{v_{max}} \sum_{j=k+1}^{v_{max}} \mathbf{x}_i(\mathbf{R}_T(i,j,k,x,y) + \mathbf{R}_I(i,j,k,x,y)) \circ \mathbf{D}(j,k,x))
\end{aligned}
\tag{5.28}
$$

Chapter 6

Results and Conclusions

6.1 Introduction

The algorithms in chapters 3, 4 and 5 were implemented in a program using the C programming language. The program was run with different load patterns to generate performance measures. Results are compared with simulation. An important note to make here is that the simulation simulates the exact DQDB network while the analytic model models a network that closely resembles DQDB. We don't expect to get a perfect match in the results but we expect them to be close.

6.2 Buffer Distribution and Output Process

An example for the buffer distribution and output process is plotted in Figures 6.1 and 6.2. In this example a network of 10 DQDB nodes was studied. Each node was subjected to the same arrival rate for segments, $\lambda = 0.05$. From Figure 6.1 it is easy to see that we tend to have longer queues at the nodes downstream which reflect the unfairness of the protocol reported in the literature. Also as we traverse the bus downstream it is more likely to get longer sequences of busy slots, see Figure 6.2.

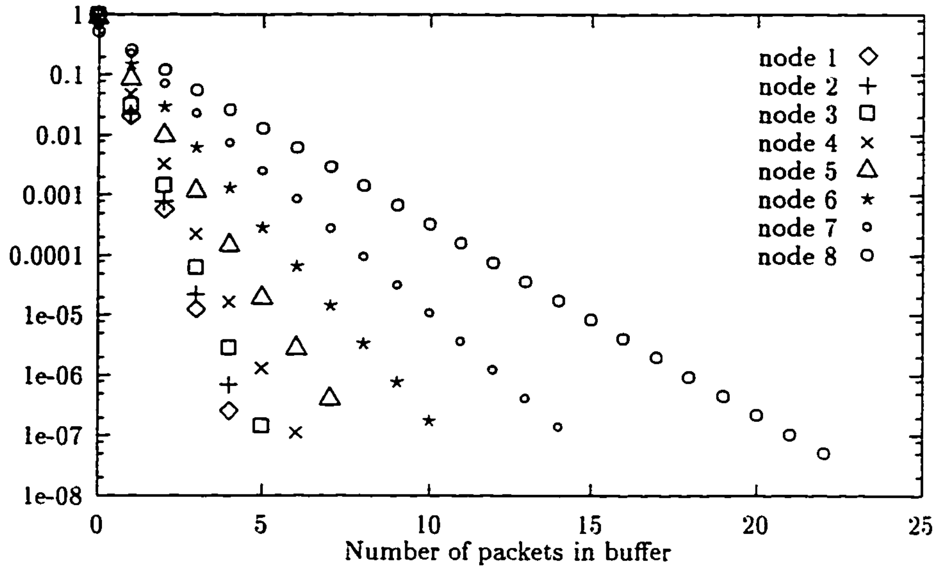

Figure 6.1: PMF of the Buffer Distribution

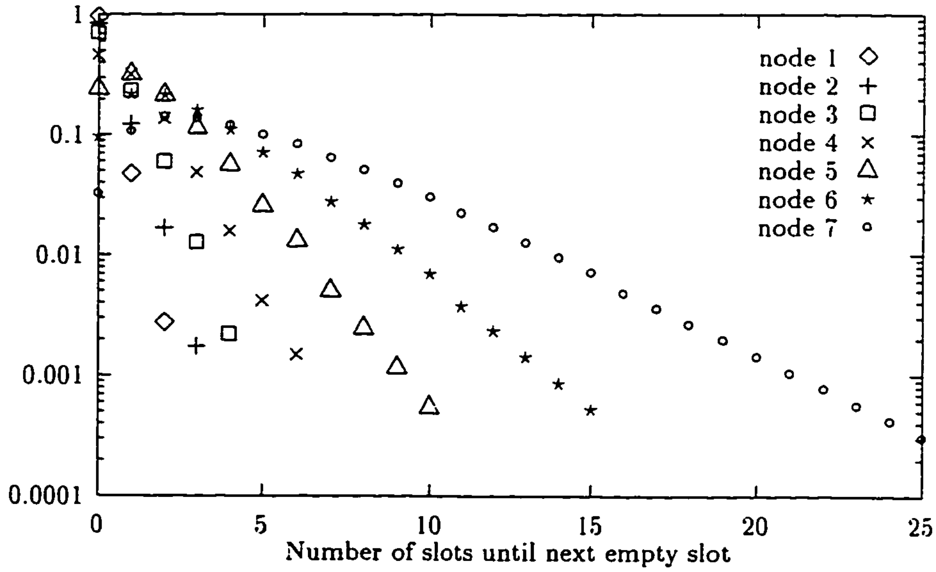

Figure 6.2: PMF of the Output Process

6.3 Waiting Time Distribution

In the experiments for waiting time distribution a DQDB network of six nodes was used with the load patterns shown in Table 6.1. Since the algorithmic model does not capture the internodal delay, load patterns are compared with the same load patterns applied to a DQDB simulator with four internodal delays. Internodal delays of 1,5,10 and 15 slots between successive nodes were used in different simulations for the same load pattern. Results are shown in Figures 6.3, 6.4, 6.5, 6.6, 6.7, 6.8, 6.9, 6.10, 6.11, 6.15 and 6.16. As can be seen from these curves there is a close match between the analytic model and the DQDB simulation for lower values of the waiting time. For higher values of the waiting time a clear mismatch is observed. The reason for this is that in DQDB as mentioned in Section 2.6 the stochastic processes have circular dependence. This dependence is a sort of feedback that regulates the traffic and is meant to control longer queues. This feedback is approximately modeled by the analytic model but for longer waiting times there will be larger mismatches.

6.4 Conclusions and Future Work

In this thesis an algorithmic model for DQDB was presented. The algorithms compute the buffer distribution, the output process and the waiting time distribution for each node. The model captures many of the protocol dependencies but not all. The work in this thesis can be enhanced in many ways and can also open directions for other studies as follows

- The Poisson arrivals at each node can be substituted by the Markovian arrival process.
- A better choice of embedding points for the Markov renewal process can yield better results. For example, modeling the activity on the reverse bus by a renewal process that takes place whenever a request arrives, then considering the superposition of this process with the renewal process of the forward bus.

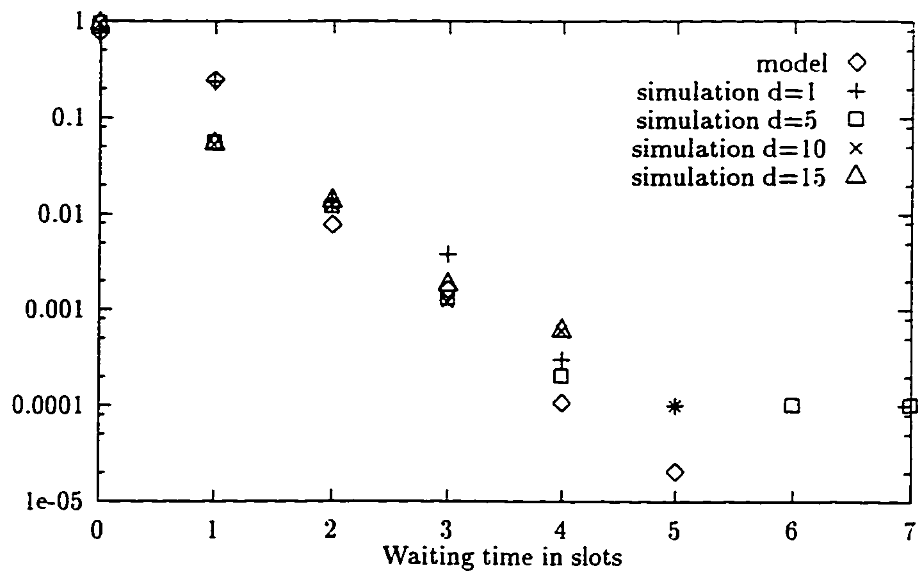

Figure 6.3: PMF of the Waiting time at Node 1 for Pattern 1

- The approximation that was used to break the circular dependence of the stochastic processes can be relaxed. Instead of computing the probabilities of requests initially, an iterative scheme can be developed.

As a general point of research, it would be worth studying the effect of distance in distributed systems. In DQDB the distance between nodes is measured in slots. If the distance between two successive nodes is n slots, the transition in a node state will only affect the other node after time equal to n. It appeared from the simulations that there is a threshold value for n after which any increase in it doesn't affect the waiting time distribution much. Our gesture that need some research is that if the distance increases the correlations between nodes decreases. If this is true, it will help in isolating the correct entities when modeling a distributed system.

Pattern	Node 1	Node 2	Node 3	Node 4	Node 5	Node 6
1	0.05	0.05	0.05	0.05	0.05	0.05
2	0.1	0.05	0.05	0.05	0.05	0.05
3	0.1	0.1	0.05	0.05	0.05	0.05
4	0.1	0.1	0.1	0.05	0.05	0.05
5	0.1	0.1	0.1	0.1	0.05	0.05

Table 6.1: Examples of load patterns applied to model

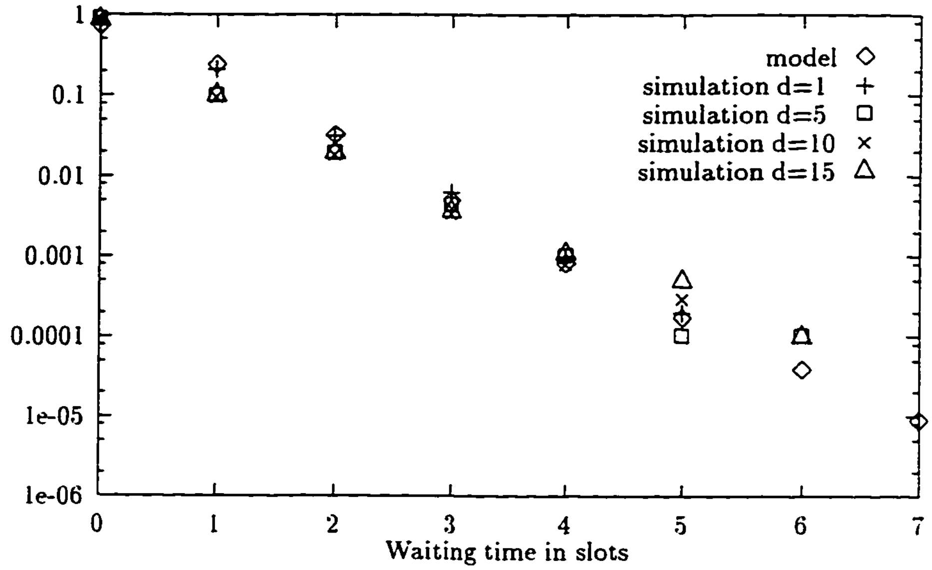

Figure 6.4: PMF of the Waiting time at Node 2 for Pattern 1

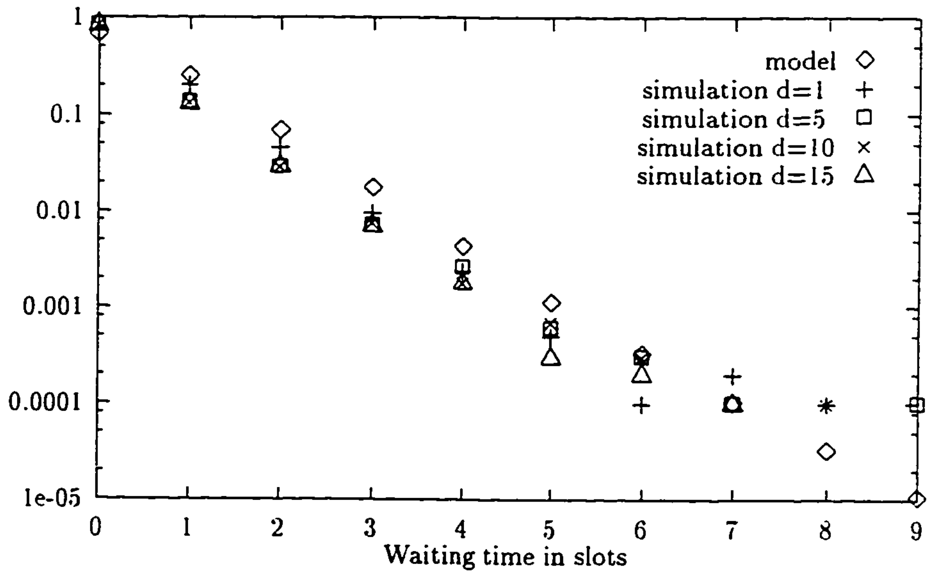

Figure 6.5: PMF of the Waiting time at Node 3 for Pattern 1

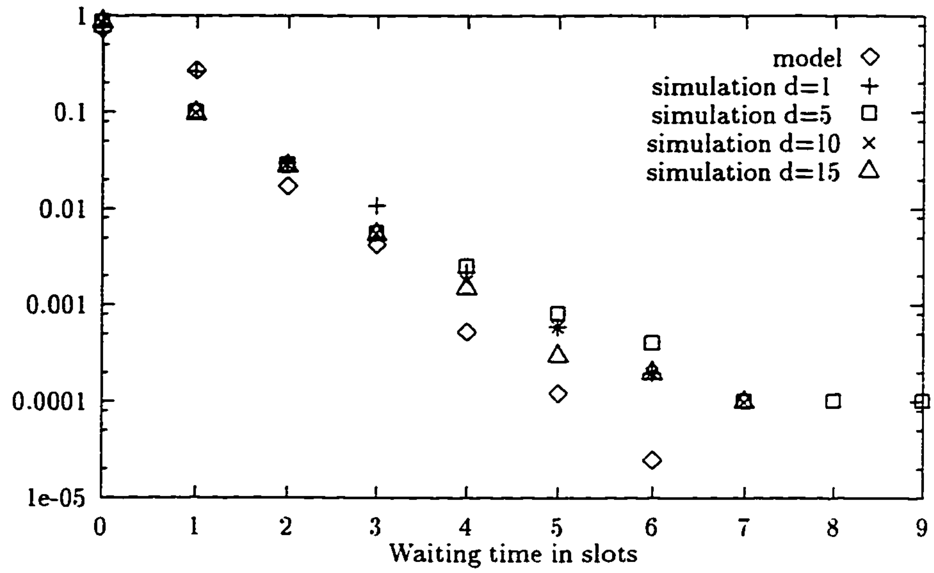

Figure 6.6: PMF of the Waiting time at Node 1 for Pattern 2

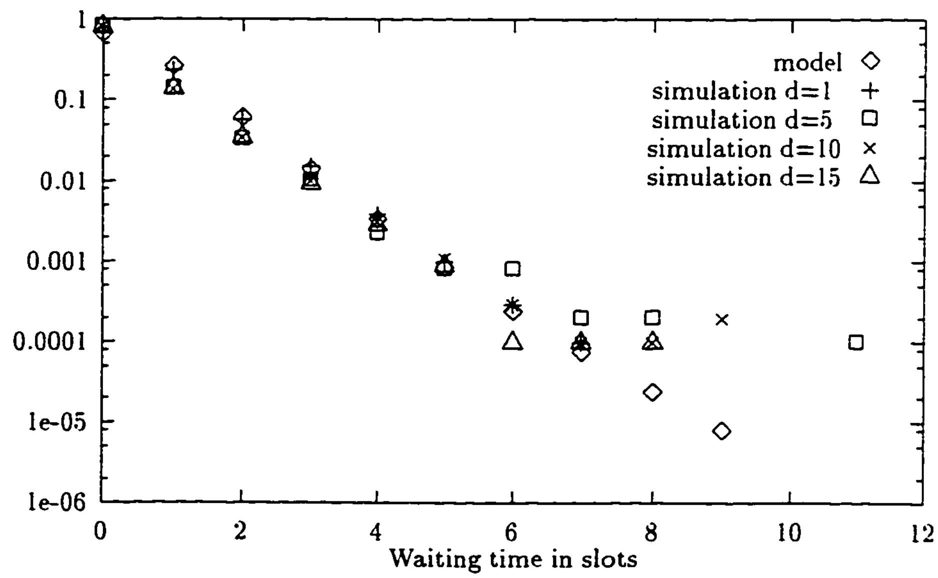

Figure 6.7: PMF of the Waiting time at Node 2 for Pattern 2

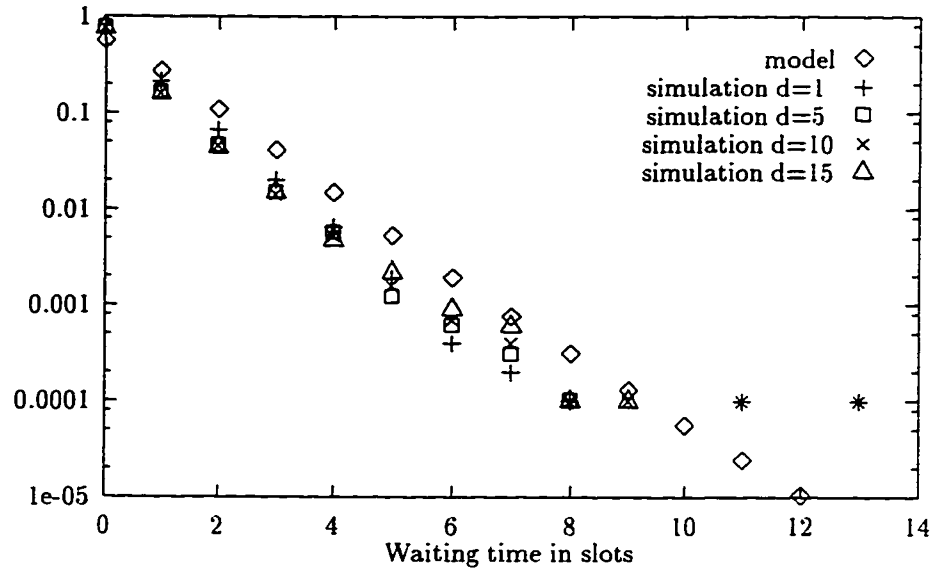

Figure 6.8: PMF of the Waiting time at Node 3 for Pattern 2

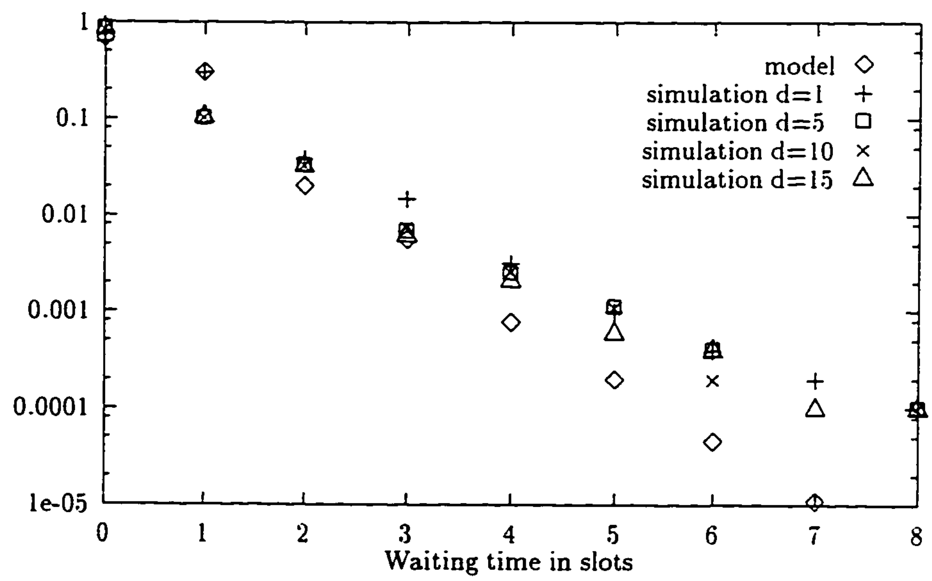

Figure 6.9: PMF of the Waiting time at Node 1 for Pattern 3

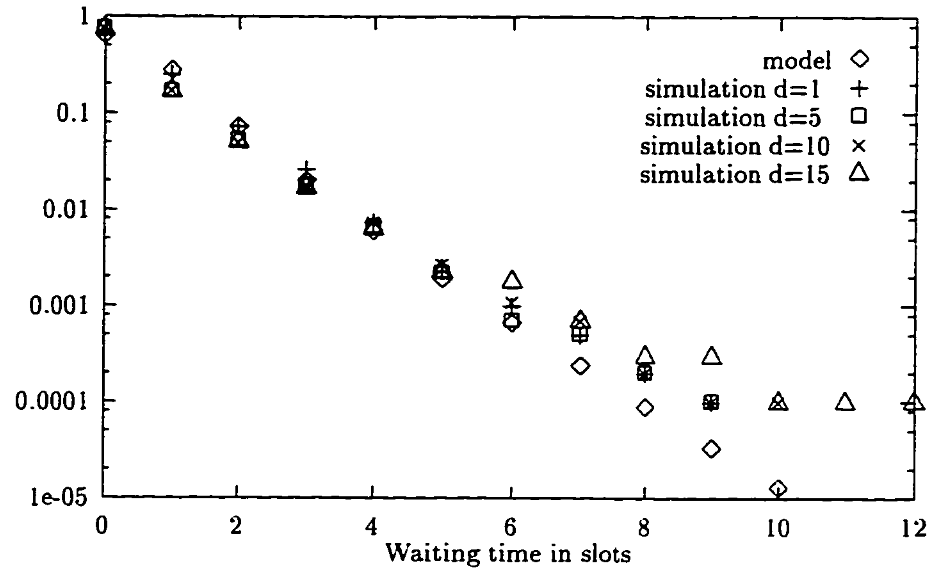

Figure 6.10: PMF of the Waiting time at Node 2 for Pattern 3

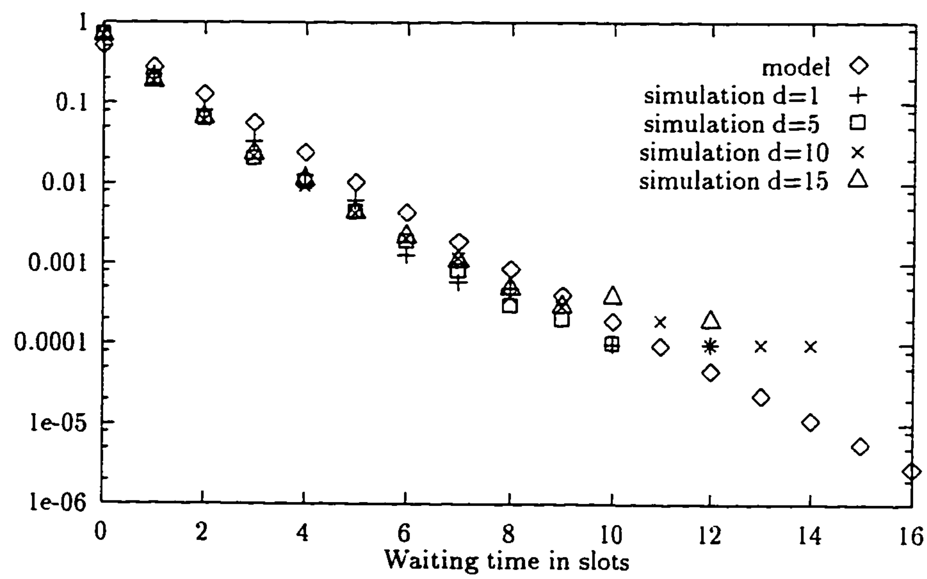

Figure 6.11: PMF of the Waiting time at Node 3 for Pattern 3

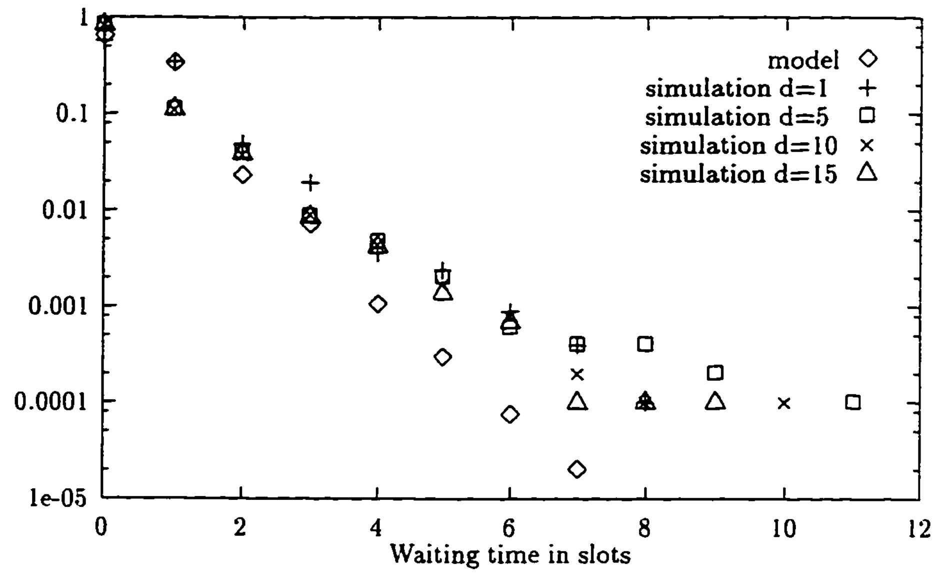

Figure 6.12: PMF of the Waiting time at Node 1 for Pattern 4

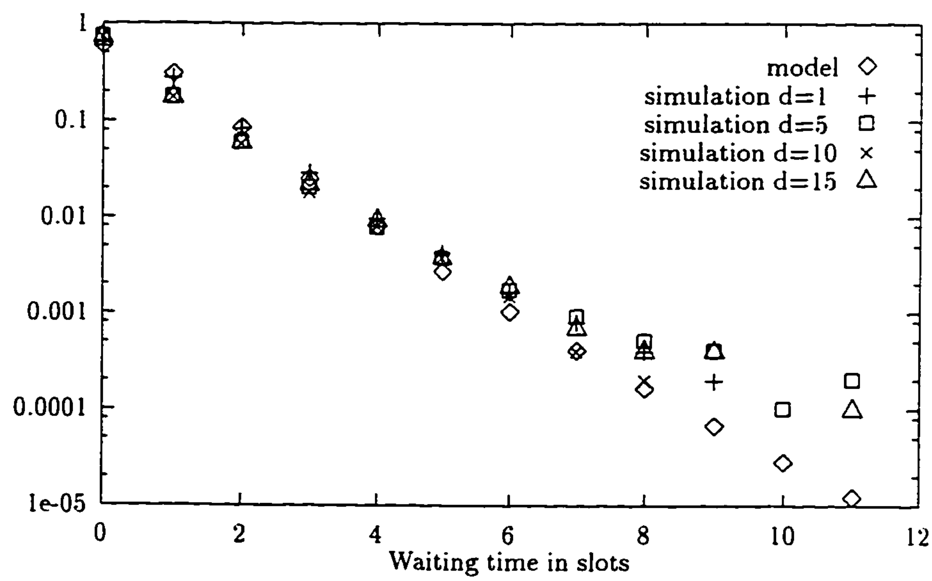

Figure 6.13: PMF of the Waiting time at Node 2 for Pattern 4

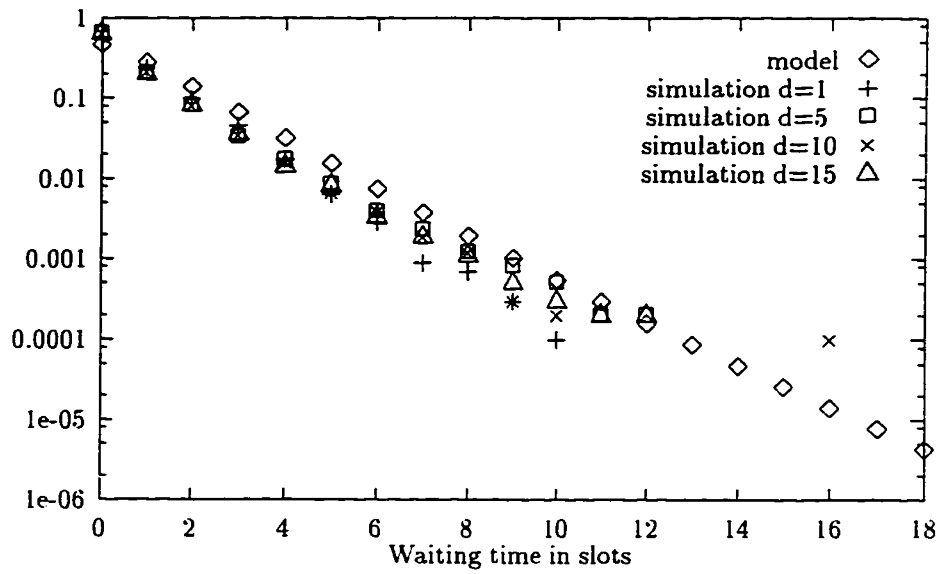

Figure 6.14: PMF of the Waiting time at Node 3 for Pattern 4

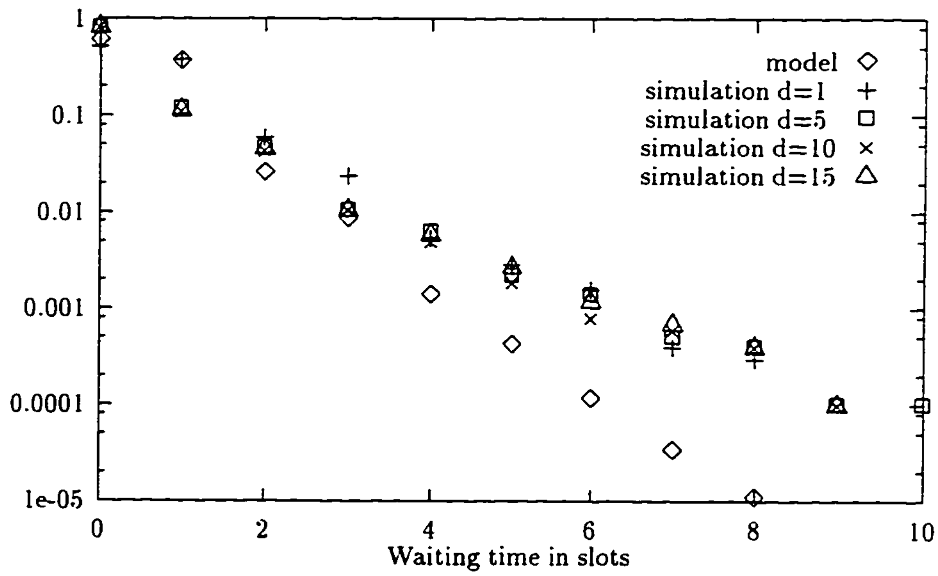

Figure 6.15: PMF of the Waiting time at Node 1 for Pattern 5

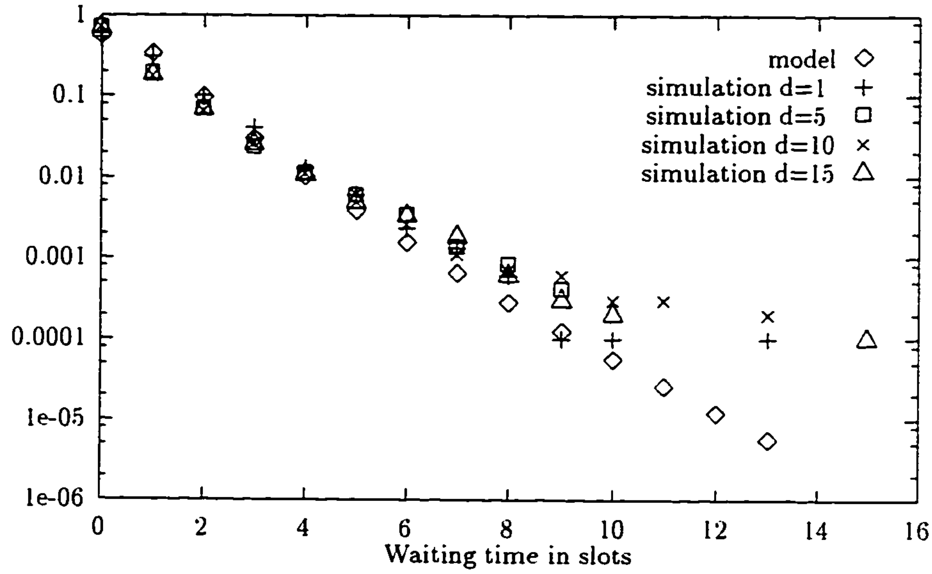

Figure 6.16: PMF of the Waiting time at Node 2 for Pattern 5

Appendix A

Markov Renewal Processes of M/G/1 Type

A.1 Introduction

In this appendix Markov Renewal Processes of M/G/1 type are described. There is a class of problems that can be modeled with the same approach used for Markov Renewal Process of M/G/1 type. This approach uses algorithmic techniques for the solution and not classical techniques. The trick is in the first representation that exploits the structural properties of the problem. Then the general algorithms used for the canonical form can be used directly. In the following sections we show how several problems can be represented and then describe the canonical form representation. Then we proceed by summarizing the results and algorithms for the general solutions from [Neu89]. This chapter is intended as a summary , for a detailed disscussion and proofs the reader is refered to [Neu89].

A.2 The M/G/1 Queue

The M/G/1 queue is modeled here according to the algorithmic approach developed by Neuts [Neu89] and not according to the classical representation. The arrival process is poisson of rate λ. Customers are served singly and the service times are independent, identically distributed nonnegative random variables with common probability distribution H(.). The time origin is taken to correspond to a service completion. Let T_n , $n \geq 0$ denote the times of successive service completions with $T_0 = 0$. The number of customers in the system after the n^{th} service completion is denoted by I_n. The n^{th} service time is denoted by τ_n where $\tau_n = T_n - T_{n-1}$ for $n \geq 1$. The sequence $\{(I_n, \tau_n), n \geq 0\}$ forms a Markov renewal sequence on the state space $\{i \geq 0\} \times \{0, \infty\}$. This follows from the fact that I_{n+1} depends on I_n and the arrivals in τ_n only and is given by

$$I_{n+1} = (I_n - 1)^+ + v_{n+1} \qquad for\ n \geq 0 \tag{A.1}$$

where v_{n+1} is the number of arrivals during the $(n+1)^{st}$ service time. Under the assumption of poisson arrivals the, random variables v_n, $n \geq 1$ are i.i.d.

The transition probability matrix $\mathbf{Q}(.)$ with elements

$$Q_{ii'}(x) = P\{I_n = i', \tau_n \leq x \mid I_{n-1} = i\} \tag{A.2}$$

for $i \geq 0$, $i' \geq 0$ and $x \geq 0$ is given by

$$\begin{aligned} Q_{0i'}(x) &= \int_0^x \lambda e^{-\lambda u} Q_{1i'}(x-u)\, du && for\ i' \geq 0 \\ Q_{ii'}(x) &= \int_0^x e^{-\lambda u} \frac{(\lambda u)^{(i'-i+1)}}{(i'-i+1)!}\, dH(u) && for\ i \geq 1,\ i' \geq i-1 \\ Q_{ii'}(x) &= 0 && for\ i \geq 1,\ i' \leq i-1 \end{aligned} \tag{A.3}$$

Remark:

Integrating the product of the probability that the first arrival happens at u and the probability that i' customers are in the system at x given that one customer was in the system at u from 0 to x gives $Q_{0i'}(x)$ in equation [A.3].

In what follows a mass function is a function that takes values between 0 and 1 but does not necessarily tend to 0 at $-\infty$ or 1 at $+\infty$. If we define the probability mass functions $A_v(x)$ by

$$A_v(x) = \int_0^x e^{-\lambda u} \frac{(\lambda u)^v}{v!} dH(u) \qquad for\ v \geq 0,\ x \geq 0 \tag{A.4}$$

and the probability mass functions $B_v(x)$ by

$$B_v(x) = \int_0^x \lambda e^{-\lambda u} A_v(x-u) du = \int_0^x [1 - e^{-\lambda(x-u)}]\, dA_v(u) \tag{A.5}$$

for $v \geq 0,\ x \geq 0$, then we see that the matrix $\mathbf{Q}(.)$ has the structural form

$$\mathbf{Q}(\mathrm{x}) = \begin{vmatrix} B_0(x) & B_1(x) & B_2(x) & B_3(x) & \dots \\ A_0(x) & A_1(x) & A_2(x) & A_3(x) & \dots \\ 0 & A_0(x) & A_1(x) & A_2(x) & \dots \\ 0 & 0 & A_0(x) & A_1(x) & \dots \\ \vdots & \vdots & \vdots & \vdots & \end{vmatrix} \tag{A.6}$$

A.3 The M/SM/1 Queue

The M/SM/1 Semi Markovian queue is a natural generalization of the M/G/1 queue. The service times of successive customers form a Markov renewal process with a finite number m of states. The sojourn times in successive states has general probability distribution which depend only on the current state and the state to be visited next. The transition probability matrix $\mathbf{H}(.)$ of the markov renewal process which describes the service times of the successive customers is an $m \times m$ matrix of probability mass functions on $[0, \infty)$ Its row sums

$$H_j(x) = \sum_{j'=1}^{m} H_{jj'}(x) \tag{A.7}$$

are proper probability distributions of finite mean α_j , $1 \leq j \leq m$, and the matrix $\mathbf{H} = \mathbf{H}(\infty)$ is an irreducible stochastic matrix.

If we consider the M/SM/1 queue after the successive service completions and form the trivariate sequence (I_n, J_n, X_n) where I_n denotes the queue length, $J_n \epsilon 1, \dots, m$ the state of the Markov renewal process $\mathbf{H}(.)$ and X_n the time between the n^{th} and $(n+1)^{st}$ departures, we obtain . as for the M/G/1 queue the Markov renewal process of the M/SM/1 queue. Its transition probability matrix $\mathbf{Q}(x)$. $x \geq 0$ is given by

$$\mathbf{Q}(\mathrm{x}) = \begin{vmatrix} \mathbf{B}_0(x) & \mathbf{B}_1(x) & \mathbf{B}_2(x) & \mathbf{B}_3(x) & \cdots \\ \mathbf{A}_0(x) & \mathbf{A}_1(x) & \mathbf{A}_2(x) & \mathbf{A}_3(x) & \cdots \\ 0 & \mathbf{A}_0(x) & \mathbf{A}_1(x) & \mathbf{A}_2(x) & \cdots \\ 0 & 0 & \mathbf{A}_0(x) & \mathbf{A}_1(x) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \end{vmatrix} \tag{A.8}$$

where

$$\mathbf{A}_v(x) = \int_0^x e^{-\lambda u} \frac{(\lambda u)^v}{v!} \, d\mathbf{H}(u) \qquad \textit{for } v \geq 0,\ x \geq 0 \tag{A.9}$$

$$\mathbf{B}_v(x) = \int_0^x \lambda e^{-\lambda u} \mathbf{A}_v(x-u) du = \int_0^x [1 - e^{-\lambda(x-u)}] \, d\mathbf{A}_v(u) \tag{A.10}$$

and the elements of the matrix $\mathbf{A}_v(x)$ is given by

$$[A_v(x)]_{jj'} = \int_0^x e^{-\lambda u} \frac{(\lambda u)^v}{v!} \, dH_{jj'}(u) \qquad \textit{for } v \geq 0,\ x \geq 0 \tag{A.11}$$

The irreducibility of the matrix $\mathbf{H}$ is inherited by all the nonnegative matrices $\mathbf{A}_v = \mathbf{A}_v(\infty)$ and $\mathbf{B}_v = \mathbf{B}_v(\infty)$, for $v \geq 0$.

A.4 The Canonical Form

The Markov Renewal process represented by the matrix $\mathbf{Q}(\mathrm{x})$ given by

$$\mathbf{Q}(\mathrm{x}) = \begin{vmatrix} \mathbf{B}_0(x) & \mathbf{B}_1(x) & \mathbf{B}_2(x) & \mathbf{B}_3(x) & \cdots \\ \mathbf{C}_0(x) & \mathbf{A}_1(x) & \mathbf{A}_2(x) & \mathbf{A}_3(x) & \cdots \\ 0 & \mathbf{A}_0(x) & \mathbf{A}_1(x) & \mathbf{A}_2(x) & \cdots \\ 0 & 0 & \mathbf{A}_0(x) & \mathbf{A}_1(x) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \end{vmatrix} \tag{A.12}$$

is a general form that describes many applications. As seen in the M/G/1 queue its elements were scalar while in the case of the M/SM/1 queue the elements were matrices. There are many applications in the literature that fits this general form from a wide variety of areas. The interested reader can find more of them in the examples and problems given in [Neu89].

Markov renewal processes that have a structure of the form given by Q(x) are called Markov renewal processes of the M/G/1 type. Its elements are defined by

$$Q_{iji'j'}(x) = P\{I_n = i', J_n = j', \tau_n \leq x \mid I_{n-1} = i, J_{n-1} = j\} \tag{A.13}$$

Note that J_n is a r.v. that describes the server or system state. In some applications J_n can be a vector. The subset of states defined by { (i , j): 1 $\leq$ j $\leq$ m } are called level i. The matrix $\mathbf{Q}(\infty)$ is stochastic and Markov chains that have a structure of its form are called Markov chains of the M/G/1 type.

The matrices $\mathbf{A}_v(x), v \geq 0$ are square matrices of dimension m. The matrices $\mathbf{B}_v(x), v \geq 0$ at level 0 are of dimensions $m_1 \times m$ and the matrix $\mathbf{C}_0$ is of dimension $m \times m_1$. And since $\mathbf{Q}(\infty)$ is stochastic we have

$$\mathbf{B}_0(\infty)e + \sum_{v=1}^{\infty} \mathbf{B}_v(\infty)e = e \tag{A.14}$$

$$\mathbf{C}_0(\infty)e + \sum_{v=1}^{\infty} \mathbf{A}_v(\infty)e = e \tag{A.15}$$

$$\mathbf{A}(\infty)e = \sum_{v=0}^{\infty} \mathbf{A}_v(\infty)e = e \tag{A.16}$$

where e is a column vector of ones. In the next sections the matrices $\mathbf{A}_v(\infty), \mathbf{B}_v(\infty), \mathbf{C}_0(\infty)$ and $\mathbf{A}(\infty)$ will be denoted by $\mathbf{A}_v, \mathbf{B}_v, \mathbf{C}_0$ and $\mathbf{A}$ respectively. The analysis ,algorithms and results in the following sections for the matrix $\mathbf{Q}(\infty)$ can be used for any markov chain of the M/G/1 type.

The invariant probability vector x of the Markov chain $\mathbf{Q}(\infty)$ may be partitioned as $\mathbf{x} = [\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \ldots]$ where $\mathbf{x}_0$ is of dimensions m_1 and $\mathbf{x}_i$, $i \geq 1$ is of dimensions m. Its generating function is given by $\;_z\mathbf{X}(z) = \sum_0^{\infty} z^i \mathbf{x}_i$.

If the transition matrix $\mathbf{A} = \sum_0^\infty \mathbf{A}_i$ is irreducible then there is a stationary probability vector π such that

$$\pi\mathbf{A} = \pi, \qquad \pi\mathbf{e} = 1$$

The vector β is defined by

$$\beta = \sum_1^\infty i\mathbf{A}_i\mathbf{e}$$

The process is recurrent if and only if $\rho = \pi\beta \leq 1$ see [Neu89] section 2.3.

A.5 The General Algorithm

After representing a problem with a Markov renewal process of the M/G/1 type, we usually want to compute the steady state probability vector at renewals epochs. In this section we will outline this in a high level manner that serves as a road map to the Appendix. The formulation of a Markov renewal process of the M/G/1 type is just the definition of three functions $\mathbf{A}_v(x)$, $\mathbf{B}_v(x)$ and $\mathbf{C}_0(x)$. After this the steps of computation are as follows

1. Compute the embedded chain at $x = \infty$ by summing over all possible values of x. This will yield $\mathbf{A}_v$, $\mathbf{B}_v$ and $\mathbf{C}_0$. The sequence of matrices $\mathbf{A}_v$ and $\mathbf{B}_v$ are truncated at a suitable high index v.

2. Check that the process is recurrent by computing ρ as given in the end of Section A.4. If ρ is less than one proceed.

3. Compute the fundamental period probability matrix $\mathbf{G}$ as outlined in Section A.6.

4. Use $\mathbf{G}$ to compute the matrix $\mathbf{L}$ of the probability of transitions from level 1 to level 0 as outlined in Section A.8.

5. Use $\mathbf{G}$ and $\mathbf{L}$ to compute the level 0 recurrence probability matrix $\mathbf{K}$ as outlined in Section A.8.

6. Compute the steady state probability vector τ of $\mathbf{K}$.

7. Compute the vector τ_1 of mean row sums of the matrix $\mathbf{K}$ as outlined in Section A.10.

8. Using τ and τ_1 compute the steady state probability subvector $\mathbf{x}_0$ as in Equation A.59.

9. Compute the subvectors $\mathbf{x}_i$, $i > 1$ using $\mathbf{x}_0$ and the Ramaswami's algorithm presented in Section A.11.

A.6 First Passage Times

In this section the equations governing the first passage times from one level to another are derived. The first passage times for Markov renewal processes are studied by considering two random variables, the number of state transitions and the time spent in each state. The former is a discrete random variable while the later is either continuous or discrete depending on the model. In the DQDB model presented in section 3.4 time is discrete while in the M/G/1 and M/SM/1 queue presented in sections A.2 and A.3 time is continuous.

In studying first passage times in markov chains of M/G/1 type as $\mathbf{Q}(\infty)$, one random variable is considered which is the number of transitions between states. The random variable $V(i,j;i^{'},j^{'})$ is defined as the time spent travelling from state j in level i to state $j^{'}$ in level $i^{'}$ where $i > i^{'}$. The number of transitions has the probabilities

$$g_{jj'}^{ii'}(k) = P\{V(i,j;i^{'},j^{'}) = k\} \tag{A.17}$$

Due to the homogeneity of the transition matrix in equation (A.8), the above probabilities only depend on the difference $i - i^{'} = r$ for $i > 1$. The probabilities are rewritten as

$$g_{jj'}^{r}(k) = P\{V(i+r,j;i,j^{'}) = k\} \tag{A.18}$$

where $r > 0, i > 0$ and $1 \leq j, j^{'} \leq m$. In matrix form we have $\mathbf{G}^r(k) = \{\ g_{jj'}^{r}(k)\ \}$. The matrix $\mathbf{Q}(\infty)$ implies that the process is skip free to the left for levels. That is any path from level i to level $i^{'}$ with $i > i^{'} \geq 0$ must visit all intermediate levels. So the total time spent during the first passage is the sum of the times spent descending from a level to the level previous to it till the process reaches the desired level. Hence the probability matrix G for $r > 2$ can be written as the convolution of other matrices with $r = 1$ as given by the following lemma.

Lemma A.6.1

$$\mathbf{G}^r(k) = \sum \mathbf{G}^1(k_1) * \mathbf{G}^1(k_2) * \ldots * \mathbf{G}^1(k_r) \tag{A.19}$$

where $r > 1, k \geq r$ and the summation is carried over all r-tuples satisfying $k_1 \geq 1 \ldots k_r \geq 1$, $k_1 + \ldots + k_r = k$

The z transform of $\mathbf{G}^r(k)$ is given by

$$_z\mathbf{G}^r(z) = \sum_{k=0}^{\infty} z^k \mathbf{G}^r(k) \tag{A.20}$$

From now on the matrix $\mathbf{G}^1(k)$ is denoted by $\mathbf{G}(k)$ and its z transform by $_z\mathbf{G}(z)$ and $_z\mathbf{G}(1)$ by $_z\mathbf{G}$ for convenience. The transform of equation (A.19) is given by

$$_z\mathbf{G}^r(z) = [\,_z\mathbf{G}(z)]^r \tag{A.21}$$

Theorem A.6.2

$$\mathbf{G}(1) = \mathbf{A}_0 \quad ; \quad \mathbf{G}(k) = \sum_{v=1}^{\infty} \mathbf{A}_v * \mathbf{G}^v(k-1) \qquad for \;\; k \geq 2 \tag{A.22}$$

and the transform matrix is given by

$$_z\mathbf{G}(z) = z\sum_{v=0}^{\infty} \mathbf{A}_v \; _z\mathbf{G}^v(z) = z\sum_{v=0}^{\infty} \mathbf{A}_v[\,_z\mathbf{G}(z)]^v \tag{A.23}$$

Proof: By conditioning on the first transition and applying the law of total probability the results follow. Setting $z = 1$ in equation A.23 the transform reduces to

$$_z\mathbf{G} = \sum_{v=0}^{\infty} \mathbf{A}_v \; _z\mathbf{G}^v \tag{A.24}$$

The elements of $_z\mathbf{G}$ are theconditional probabilities that the Markov renewal process $\mathbf{Q}(x)$ will eventually hit the set i in the state (i, j') given that it starts in the state $(i + 1, j)$, $i \geq 1$.

Corollary A.6.1 *If the Markov renewal process* $\mathbf{Q}(x)$ *is irreducible , the matrix* ${}_{z}\mathbf{G}$ *does not have zero rows. If the Markov renewal process* $\mathbf{Q}(x)$ *is recurrent the matrix* ${}_{z}\mathbf{G}$ *is stochastic.*

Neuts showed in section 2.2 of [Neu89] that the matrix ${}_{z}\mathbf{G}$ which is of interest is the minimal nonnegative solution to A.24.

The matrix $\mathbf{G}$ is the limit of the sequence $\{ \mathbf{G}_r \}$ given by

$$\mathbf{G}_0 = 0 \quad ; \quad \mathbf{G}_{r+1} = \sum_{v=0}^{\infty} \mathbf{A}_v \mathbf{G}_r^v \qquad for \quad r \geq 0 \tag{A.25}$$

The next theorem states the conditions for $\mathbf{G}$ to be stochastic the proof is omitted but we refer the reader to section 2.3 in [Neu89] for the details.

Theorem A.6.3 *If the matrix* $\mathbf{A}$ *is irreducible, then the matrix* $\mathbf{G}$ *is stochastic iff* $\rho \leq 1$.

A.7 Structural Properties of ${}_{z}\mathbf{G}$

Lemma A.7.1 *To every zero column of the matrix* $\mathbf{A}_0$ *there is a corresponding zero column in the matrix* ${}_{z}\mathbf{G}$

Proof: This follows directly from the fact that $\mathbf{G}$ is the minimal nonnegative solution to equation A.24 and by considering the sequence A.25.

In this case the matrix $\mathbf{A}_0$ can be partitioned ,after relabeling the rows and columns appropriately , as follows

$$\mathbf{A}_0 = \begin{pmatrix} \mathbf{A}_0(1) & \mathbf{0} \\ \mathbf{A}_0(3) & \mathbf{0} \end{pmatrix} \tag{A.26}$$

Similarly ${}_{z}\mathbf{G}$ is partitioned after relabeling such that

$${}_{z}\mathbf{G} = \begin{pmatrix} {}_{z}\mathbf{G}(1) & \mathbf{0} \\ {}_{z}\mathbf{G}(3) & \mathbf{0} \end{pmatrix} \tag{A.27}$$

Equation A.24 then leads to

$${}_{z}\mathbf{G}(1) = \sum_{k=0}^{\infty} \mathbf{A}_k(1)\ {}_{z}\mathbf{G}^k(1) + \sum_{k=1}^{\infty} \mathbf{A}_k(2)\ {}_{z}\mathbf{G}(3)\ {}_{z}\mathbf{G}^{k-1}(1) \tag{A.28}$$

$$ {}_z\mathbf{G}(3) = \sum_{k=0}^{\infty} \mathbf{A}_k(3)\ {}_z\mathbf{G}^k(1) + \sum_{k=1}^{\infty} \mathbf{A}_k(4)\ {}_z\mathbf{G}(3)\ {}_z\mathbf{G}^{k-1}(1) \tag{A.29} $$

where

$$ \mathbf{A}_k = \begin{pmatrix} \mathbf{A}_k(1) & \mathbf{A}_k(2) \\ \mathbf{A}_k(3) & \mathbf{A}_k(4) \end{pmatrix} \qquad k \geq 1 \tag{A.30} $$

A.8 Recurrence and the boundary states

In this section the first passage time from level 1 to level 0 and the return time to level 0 are studied. The following probabilities are similar to those in equation A.18 except that they are from level 1 to level 0.

$$ l_{jj'}(k) = P\{V(1,j;0,j') = k\} \tag{A.31} $$

where $1 \leq j, j' \leq m$. In matrix form we have $\mathbf{L}(k) = \{\ l_{jj'}(k)\ \}$ and the transform matrix is given by ${}_z\mathbf{L}(z) = \sum_{k=1}^{\infty} z^k \mathbf{L}(k)$.

Theorem A.8.1 *The matrix* ${}_z\mathbf{L}(z)$ *satisfies*

$$ {}_z\mathbf{L}(z) = z\mathbf{C}_0 + \sum_{v=1}^{\infty} z\mathbf{A}_v[\ {}_z\mathbf{G}(z)]^{v-1}\ {}_z\mathbf{L}(z) \tag{A.32} $$

$$ {}_z\mathbf{L}(z) = [\mathbf{I} - \sum_{v=1}^{\infty} z\mathbf{A}_v[\ {}_z\mathbf{G}(z)]^{v-1}]^{-1} z\mathbf{C}_0 \tag{A.33} $$

and ${}_z\mathbf{L} = {}_z\mathbf{L}(1)$ *is stochastic if* ${}_z\mathbf{G}$ *is stochastic.*

Proof: Equation A.32 is verified by conditioning on the first transition and applying the law of total probability. The term $z\mathbf{C}_0$ contributes the probability of one transition to level 0. The second term makes the first transition to level $v > 0$ and then the term $[\ {}_z\mathbf{G}(z)]^{v-1}$ accounts for the return back to level 1 and ${}_z\mathbf{L}(z)$ to reach level 0.

In equation A.33 the proof that the inverse exists is given in [Neu89] page 98. Setting $z = 1$ in equation A.32 we get

$$ {}_z\mathbf{L} = \mathbf{C}_0 + \sum_{v=1}^{\infty} \mathbf{A}_v\ {}_z\mathbf{G}^{v-1}\ {}_z\mathbf{L} \tag{A.34} $$

multiplying both sides by e and setting ${}_z\mathbf{L}\ \mathbf{e} = \mathbf{V}$ and noting that $\mathbf{C}_0 e = \mathbf{A}_0 e$

$$\mathbf{V} = \mathbf{A}_0 e + \sum_{v=1}^{\infty} \mathbf{A}_v\ {}_z\mathbf{G}^{v-1}\mathbf{V} \tag{A.35}$$

If ${}_z\mathbf{G}$ is stochastic then the above equation is satisfied by $\mathbf{V} = \mathbf{e}$ and it is the only solution since $[I - \sum_{v=1}^{\infty} \mathbf{A}_v\ {}_z\mathbf{G}^{v-1}]$ is non singular.

For the return time to level 0 there are similar results and proofs. First the distribution of the return time to level 0 is defined by

$$\kappa_{jj'}(k) = P\{V(0,j;0,j') = k\} \tag{A.36}$$

where $1 \leq j, j' \leq m$. In matrix form $\mathbf{K}(k) = \{\kappa_{jj'}(k)\}$ and the transform matrix is given by ${}_z\mathbf{K}(z) = \sum_{k=1}^{\infty} z^k \mathbf{K}(k)$.

Theorem A.8.2 *The matrix* ${}_z\mathbf{K}(z)$ *satisfies*

$${}_z\mathbf{K}(z) = z\mathbf{B}_0 + \sum_{v=1}^{\infty} z\mathbf{B}_v[\ {}_z\mathbf{G}(z)]^{v-1}\ {}_z\mathbf{L}(z) \tag{A.37}$$

and ${}_z\mathbf{K}\mathbf{e} = \mathbf{e}$ *if* ${}_z\mathbf{G}$ *is stochastic.*

The proofs are done in the same way as for the matrix ${}_z\mathbf{L}(z)$.

A.9 Moment formulas for the matrix $\mathbf{G}(k)$

In chapter 3 of [Neu89] moment formulas are given for the fundamental period matrix $\mathbf{G}(k, x)$. Where k is the number of transitions during the period and x is the time spent during the first passage. The formulas stated in the book uses Laplace transforms on one part as time is continous there. The formulas there need little change if the time is discrete as in the DQDB model. In this appendix the results are listed for the first passage time of the chain $Q(\infty)$. So in this case there is no worry about time as the Markov Renewal process is not considered but rather the embedded chain. The matrix $\mathbf{M}$ is defined as the first moment matrix of the matrix $\mathbf{G}(k) = \mathbf{G}(k, \infty)$ and hence is given by

$$\mathbf{M} = [\frac{\partial\ {}_z\mathbf{G}(z)}{\partial z}]_{z=1}$$

differentiating equation (A.23) we get

$$\mathbf{M} = {}_z\mathbf{G} + \sum_{v=1}^{\infty} \mathbf{A}_v \sum_{k=0}^{v-1} {}_z\mathbf{G}^k \mathbf{M}\ {}_z\mathbf{G}^{v-k-1} \tag{A.38}$$

If the matrix ${}_z\mathbf{G}$ is reducible as in equation (A.27) then $\mathbf{M}$ is also partitioned such that

$$\mathbf{M} = \begin{pmatrix} \mathbf{M}(1) & \mathbf{0} \\ \mathbf{M}(3) & \mathbf{0} \end{pmatrix} \tag{A.39}$$

The powers of ${}_z\mathbf{G}$ are then given by

$${}_z\mathbf{G}^k = \begin{pmatrix} {}_z\mathbf{G}^k(1) & \mathbf{0} \\ {}_z\mathbf{G}(3)\ {}_z\mathbf{G}^{k-1}(1) & \mathbf{0} \end{pmatrix} \tag{A.40}$$

and the matrix ${}_z\mathbf{G}^k\mathbf{M}\ {}_z\mathbf{G}^{v-k-1}$ for $k \geq 1$ is given by

$${}_z\mathbf{G}^k\mathbf{M}\ {}_z\mathbf{G}^{v-k-1} = \begin{pmatrix} {}_z\mathbf{G}^k(1)\mathbf{M}(1)\ {}_z\mathbf{G}^{v-k-1}(1) & \mathbf{0} \\ {}_z\mathbf{G}(3)\ {}_z\mathbf{G}^{k-1}(1)\mathbf{M}(1)\ {}_z\mathbf{G}^{v-k-1}(1) & \mathbf{0} \end{pmatrix} \tag{A.41}$$

equation (A.38) can be split into two equations as follows

$$\begin{aligned} \mathbf{M}(1) \quad = \quad & {}_z\mathbf{G}(1) + \sum_{v=1}^{\infty} \mathbf{A}_v(1) \sum_{k=0}^{v-1} {}_z\mathbf{G}^k(1)\mathbf{M}(1)\ {}_z\mathbf{G}^{v-k-1}(1) \\ & + \sum_{v=1}^{\infty} \mathbf{A}_v(2)\mathbf{M}(3)\ {}_z\mathbf{G}^{v-1}(1) \\ & + \sum_{v=2}^{\infty} \mathbf{A}_v(2) \sum_{k=1}^{v-1} {}_z\mathbf{G}(3)\ {}_z\mathbf{G}^{k-1}(1)\mathbf{M}(1)\ {}_z\mathbf{G}^{v-k-1}(1) \end{aligned} \tag{A.42}$$

$$\begin{aligned} \mathbf{M}(3) \quad = \quad & {}_z\mathbf{G}(3) + \sum_{v=1}^{\infty} \mathbf{A}_v(3) \sum_{k=0}^{v-1} {}_z\mathbf{G}^k(1)\mathbf{M}(1)\ {}_z\mathbf{G}^{v-k-1}(1) \\ & + \sum_{v=1}^{\infty} \mathbf{A}_v(4)\mathbf{M}(3)\ {}_z\mathbf{G}^{v-1}(1) \\ & + \sum_{v=2}^{\infty} \mathbf{A}_v(4) \sum_{k=1}^{v-1} {}_z\mathbf{G}(3)\ {}_z\mathbf{G}^{k-1}(1)\mathbf{M}(1)\ {}_z\mathbf{G}^{v-k-1}(1) \end{aligned} \tag{A.43}$$

For the case where the chain is positive recurrent the matrix $\mathbf{M}$ is finite. If the matrix ${}_z\mathbf{G}$ is stochastic the occurrence of infinite elements in $\mathbf{M}$ means that the chain is null recurrent.

In this section the cases where the process is recurrent are discussed. In such cases $\rho < 1$ The vector $\mu = \mathbf{Me}$ is the vector of row sums. Its j_{th} is the expected number of transitions to reach level i from state $(i+1, j)$. Multiplying equation (A.38) by e and noting that ${}_z\mathbf{Ge} = \mathbf{e}$ we get

$$[I - \sum_{v=1}^{\infty} \mathbf{A}_v \sum_{k=0}^{v-1} {}_z\mathbf{G}^k]\mu = \mathbf{e} \tag{A.44}$$

Let $\mathbf{g}$ be the stationary probability vector of ${}_z\mathbf{G}$. Then $\mathbf{g}\ {}_z\mathbf{G} = \mathbf{g}$ and $\mathbf{g}\ \mathbf{e} = \mathbf{e}$. The following two theorems that are stated without proofs , they are used in the computation of $\mathbf{M}$ and μ.

Theorem A.9.1 *When ${}_z\mathbf{G}$ is irreducible*

$$[\mathbf{I} - \sum_{v=1}^{\infty} \mathbf{A}_v \sum_{k=0}^{v-1} {}_z\mathbf{G}^k]^{-1} = (\mathbf{I} - {}_z\mathbf{G} + \mathbf{eg})[\mathbf{I} - \mathbf{A} + (\mathbf{e} - \beta)\mathbf{g}]^{-1} \tag{A.45}$$

where $\rho < 1$.

Using theorem A.9.1 with equation (A.44) we get

$$\mu = (\mathbf{I} - {}_z\mathbf{G} + \mathbf{eg})[\mathbf{I} - \mathbf{A} + (\mathbf{e} - \beta)\mathbf{g}]^{-1}\mathbf{e} \tag{A.46}$$

Using the equation (A.46) and the relation

$$(1 - \rho)\mathbf{g}[\mathbf{I} - \mathbf{A} + (\mathbf{e} - \beta)\mathbf{g}]^{-1} = \pi \tag{A.47}$$

we get

$$\mathbf{g}\mu_1 = (1 - \rho)^{-1} \tag{A.48}$$

A different expression is deduced for μ when ${}_z\mathbf{G}$ is reducible as in equation (A.27). Let $\mathbf{g_1}$ be the stationary vector of ${}_z\mathbf{G}(1)$. The matrix ${}_z\mathbf{G}^\circ$ is defined as

$${}_z\mathbf{G}^\circ = \begin{pmatrix} \mathbf{eg_1} & 0 \\ \mathbf{eg_1} & 0 \end{pmatrix} \tag{A.49}$$

The dimensions of the vectors e agree with those of the blocks in equation (A.27). It is clear that ${}_z\mathbf{G}\,{}_z\mathbf{G}^\circ = {}_z\mathbf{G}^\circ$ and that the matrix $\mathbf{I} - {}_z\mathbf{G} + {}_z\mathbf{G}^\circ$ is non singular. Then μ is given by

$$\mu = (\mathbf{I} - {}_z\mathbf{G} + {}_z\mathbf{G}^\circ)[\mathbf{I} - \mathbf{A} + {}_z\mathbf{G}^\circ - \sum_{v=1}^{\infty} v\mathbf{A}_v\,{}_z\mathbf{G}^\circ]^{-1}\mathbf{e} \tag{A.50}$$

The second theorem concerns the matrix **M**

Theorem A.9.2 *Equation A.38 has unique solution and the sequence*

$$\mathbf{X}_{r+1} = {}_z\mathbf{G} + \sum_{v=1}^{\infty} \mathbf{A}_v \sum_{k=0}^{v-1} {}_z\mathbf{G}^k\mathbf{X}_r\,{}_z\mathbf{G}^{v-k-1} \tag{A.51}$$

starting with $\mathbf{X}_0 = \mathbf{0}$ *converges to this solution.*

The algorithm to compute μ and **M** has the following steps

- Compute μ according to equation (A.46) if ${}_z\mathbf{G}$ is irreducible and according to equation (A.50) if ${}_z\mathbf{G}$ is reducible.
- Compute **M** using successive substitutions as in equation (A.51). If ${}_z\mathbf{G}$ is reducible then equations (A.42) and (A.43) are used as they reduce the computation time. The matrices $\mathbf{D}_v = \sum_{k=0}^{v-1} {}_z\mathbf{G}^k\mathbf{X}\,{}_z\mathbf{G}^{v-k-1}$ for $v \geq 1$ can be computed efficiently by setting $\mathbf{D}_1 = \mathbf{X}$ and then using $\mathbf{D}_v = \mathbf{D}_{v-1}\,{}_z\mathbf{G} + {}_z\mathbf{G}^{v-1}\mathbf{X}$. Also the matrices $\mathbf{E}_v = \sum_{k=1}^{v-1} {}_z\mathbf{G}^{k-1}(1)\mathbf{X}(1)\,{}_z\mathbf{G}^{v-k-1}(1)$ in equations (A.42) and (A.43) can be computed using $\mathbf{E}_v = \mathbf{E}_{v-1}\,{}_z\mathbf{G} + {}_z\mathbf{G}^{v-2}\mathbf{X}$ and setting $\mathbf{E}_2 = \mathbf{X}$. Neuts notes that any real matrix for $\mathbf{X}_0$ can be used as a start but for practical reasons it should satisfy $\mathbf{X}_0 e = \mu$ so that the iterates would have correct row sums. Hence $\mathbf{X}_0$ is set to $\mu\mathbf{g}$.
- Finally equation A.48 is used as numerical check if the fundamental matrix is irreducible. Or generaly the row sums of the computed matrix **M** with **mu**.

A.10 Moment formulas for the matrix $\mathbf{K}(k)$

For the case where ${}_z\mathbf{G}$ is stochastic ${}_z\mathbf{K}(1) = {}_z\mathbf{K}$ is stochastic. Let its steady state probability vector be τ so that $\tau\mathbf{K} = \tau$ and $\tau\mathbf{e} = \mathbf{e}$. Expanding equation (A.32) and equation (A.37) we get

$$ {}_z\mathbf{K}(z) = z\mathbf{B}_0 + \sum_{v=1}^{\infty} z\mathbf{B}_v[\,{}_z\mathbf{G}(z)]^{v-1}[I - \sum_{v=1}^{\infty} z\mathbf{A}_v[\,{}_z\mathbf{G}(z)]^{v-1}]^{-1} z\mathbf{C}_0 \tag{A.52} $$

The mean vector of row sums τ_1 is given by

$$ \begin{aligned} \tau_1 &= [\frac{\partial\ {}_z\mathbf{K}(z)}{\partial z}]_{z=1}\mathbf{e} \\ &= \mathbf{K}\mathbf{e} + \sum_{v=2}^{\infty}\mathbf{B}_v \sum_{k=0}^{v-2} {}_z\mathbf{G}^k\mathbf{M}\ {}_z\mathbf{G}^{v-k-2}[\mathbf{I} - \sum_{v=1}^{\infty}\mathbf{A}_v\ {}_z\mathbf{G}^{v-1}]^{-1}\mathbf{C}_0\mathbf{e} \\ &\quad + \sum_{v=1}^{\infty}\mathbf{B}_v\ {}_z\mathbf{G}^{v-1}[\mathbf{I} - \sum_{v=1}^{\infty}\mathbf{A}_v\ {}_z\mathbf{G}^{v-1}]^{-1} * \\ &\quad [\sum_{0}^{\infty}\mathbf{A}_v\ {}_z\mathbf{G}^{v-1} + \sum_{v=2}^{\infty}\mathbf{A}_v \sum_{k=0}^{v-2} {}_z\mathbf{G}^k\mathbf{M}_1\ {}_z\mathbf{G}^{v-k-2}] * \\ &\quad [\mathbf{I} - \sum_{v=1}^{\infty}\mathbf{A}_v\ {}_z\mathbf{G}^{v-1}]^{-1}\mathbf{C}_0\mathbf{e} \\ &\quad + \sum_{v=1}^{\infty}\mathbf{B}_v\ {}_z\mathbf{G}^{v-1}[\mathbf{I} - \sum_{v=1}^{\infty}\mathbf{A}_v\ {}_z\mathbf{G}^{v-1}]^{-1}\mathbf{C}_0\mathbf{e} \end{aligned} \tag{A.53} $$

Noting that $\mathbf{C}_0\mathbf{e} = \mathbf{A}_0\mathbf{e}$ and that $[\mathbf{I} - \sum_{v=1}^{\infty}\mathbf{A}_v\ {}_z\mathbf{G}^{v-1}]^{-1}\mathbf{A}_0\mathbf{e} = \mathbf{e}$ then the above experession is simplified to

$$ \begin{aligned} \tau_1 &= \mathbf{e} + \sum_{v=2}^{\infty}\mathbf{B}_v \sum_{k=0}^{v-2} {}_z\mathbf{G}^k\mu \\ &\quad + \sum_{v=1}^{\infty}\mathbf{B}_v\ {}_z\mathbf{G}^{v-1}[\mathbf{I} - \sum_{v=1}^{\infty}\mathbf{A}_v\ {}_z\mathbf{G}^{v-1}]^{-1}[\mathbf{e} + \sum_{v=2}^{\infty}\mathbf{B}_v \sum_{k=0}^{v-2} {}_z\mathbf{G}^k\mu] \\ &= \psi_b + \sum_{v=1}^{\infty}\mathbf{B}_v\ {}_z\mathbf{G}^{v-1}[\mathbf{I} - \sum_{v=1}^{\infty}\mathbf{A}_v\ {}_z\mathbf{G}^{v-1}]^{-1}\psi_a \end{aligned} \tag{A.54} $$

Equation (A.54) is a general equation that does not depend on the irreducibility of ${}_z\mathbf{G}$. If the condition that ${}_z\mathbf{G}$ is imposed then using equations (A.46) and (A.47) ψ_a can be reduced to

$$\begin{aligned}\psi_a &= \mathbf{e} + \sum_{v=2}^{\infty} \mathbf{A}_v \sum_{k=0}^{v-2} {}_z\mathbf{G}^k \mu & \text{(A.55)}\\ &= \mathbf{e} + \sum_{v=2}^{\infty} \mathbf{A}_v [\mathbf{I} - {}_z\mathbf{G}^{v-1} + (v-1)\mathbf{eg}][\mathbf{I} - \mathbf{A} + (\mathbf{e} - \beta)\mathbf{g}]^{-1}\mathbf{e} \\ &= [\mathbf{I} - \mathbf{A} - \sum_{v=1}^{\infty} \mathbf{A}_v \mathbf{G}^{v-1}][\mathbf{I} - \mathbf{A} + (\mathbf{e} - \beta)\mathbf{g}]^{-1}\mathbf{e} + (1-\rho)^{-1}\mathbf{A}_0\mathbf{e}\end{aligned}$$

and ψ_b can be reduced to

$$\begin{aligned}\psi_a &= \mathbf{e} + [\sum_{v=1}^{\infty} \mathbf{B}_v - \sum_{v=1}^{\infty} \mathbf{B}_v \mathbf{G}^{v-1}][\mathbf{I} - \mathbf{A} + (\mathbf{e} - \beta)\mathbf{g}]^{-1}\mathbf{e} & \text{(A.56)}\\ &+ (1-\rho)^{-1} \sum_{v=1}^{\infty} (v-1)\mathbf{B}_v \mathbf{e}\end{aligned}$$

If ${}_z\mathbf{G}$ is reducible as in equation (A.27) then using equation (A.50) ψ_a can be reduced to

$$\begin{aligned}\psi_a &= [\mathbf{I} - \mathbf{A}_0 - \sum_{v=1}^{\infty} \mathbf{A}_v\, {}_z\mathbf{G}^{v-1}][\mathbf{I} - \mathbf{A} + \mathbf{G}^{\circ} - \sum_{v=1}^{\infty} v\mathbf{A}_v \mathbf{G}^{\circ}]^{-1}\mathbf{e} & \text{(A.57)}\\ &+ (1-\rho)^{-1}\mathbf{A}_0\mathbf{e}\end{aligned}$$

and ψ_b can be reduced to

$$\begin{aligned}\psi_b &= \mathbf{e} + [\sum_{v=1}^{\infty} \mathbf{B}_v - \sum_{v=1}^{\infty} \mathbf{B}_v\, {}_z\mathbf{G}^{v-1}][\mathbf{I} - \mathbf{A} + \mathbf{G}^{\circ} - \sum_{v=1}^{\infty} v\mathbf{A}_v \mathbf{G}^{\circ}]^{-1}\mathbf{e} & \text{(A.58)}\\ &+ (1-\rho)^{-1} \sum_{v=1}^{\infty} (v-1)\mathbf{B}_v \mathbf{e}\end{aligned}$$

A.11 Computation of the steady state vector

In this section formulas to compute the vector $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \ldots)$ are summarized. The vector $\mathbf{x}_0$ is equal to the inverse of the mean recurrence time provided that the process is positive recurrent i.e. $\rho < 1$. Hence $\mathbf{x}_0$ is given by

$$\mathbf{x}_0 = (\tau\tau_1)^{-1}\tau \tag{A.59}$$

For the calculation of $\mathbf{x}_i$ when $i \geq 1$ the result given by Ramaswami in [Ram88] is used. It is stated in the following theorem.

Theorem A.11.1

$$\mathbf{x}_i = [\mathbf{x}_0\mathbf{B}_i^{'} + \sum_{j=1}^{i-1} \mathbf{x}_j \mathbf{A}_{i+1-j}^{'}](\mathbf{I} - \mathbf{A}_1^{'})^{-1} \qquad i \geq 1 \tag{A.60}$$

where

$$\mathbf{B}_i^{'} = \sum_{v=i}^{\infty} \mathbf{B}_v \ {}_z\mathbf{G}^{v-i} \qquad and \qquad \mathbf{A}_i^{'} = \sum_{v=i}^{\infty} \mathbf{A}_v \ {}_z\mathbf{G}^{v-i} \tag{A.61}$$

The computation of A.60 can be done efficiently by taking into account that as $i \to \infty$, $\mathbf{A}_i^{'}, \mathbf{B}_i^{'} \to 0$. A large index can be chosen such that $\mathbf{A}_i^{'} = \mathbf{B}_i^{'} = \mathbf{0}$ and compute the other required matrices by the following back recursion.

$$\mathbf{B}_i^{'} = \mathbf{B}_i + \mathbf{B}_{i+1}^{'} \ {}_z\mathbf{G} \qquad and \qquad \mathbf{A}_i^{'} = \mathbf{A}_i + \mathbf{A}_{i+1}^{'} \ {}_z\mathbf{G} \tag{A.62}$$

as a check for the procedure we use the two following equations

$${}_z\mathbf{K} = \mathbf{B}_0 + \mathbf{B}_1^{'} \ {}_z\mathbf{L} \qquad and \qquad {}_z\mathbf{G} = \mathbf{A}_0 + \mathbf{A}_1^{'} \ {}_z\mathbf{G} \tag{A.63}$$

A.12 Computation of the moments of the steady state vector

In this section the algorithm , to calculate the first two moments of the steady state probability vector , is outlined. The details can be found in [Neu89] section 3.3. They are omitted here so as not to lose focus.

By inspecting the transition matrix in equation (A.8) it is clear that for the chain $\mathbf{Q}(\infty)$

$$\mathbf{x}_0 = \mathbf{x}_0\mathbf{B}_0 + \mathbf{x}_1\mathbf{C}_0 \tag{A.64}$$

$$\mathbf{x}_i = \mathbf{x}_0\mathbf{B}_0 + \sum_{j=0}^{i} \mathbf{x}_{i-j+1}\mathbf{A}_j \qquad i \geq 1 \tag{A.65}$$

The generating function $\mathbf{X}(z)$ is redefined in this section as

$$\mathbf{X}(z) = \sum_{i=1}^{\infty} z^i \mathbf{x}_i \tag{A.66}$$

The term x_0 is omitted from the summation to agree with Neuts results. In the differentiation it will be zero and will not affect the final results for the first two moments. The only difference is that the vector $\mathbf{X}(1)$ has different meaning now that is relevant to the following algorithm. Multiplying equation (A.65) by z^i and summing from $i = 1$ *to* ∞ then

$$\mathbf{X}(z)[zI - \ _z\mathbf{A}(z)] = z\mathbf{x}_0 \sum_{k=1}^{\infty} \mathbf{B}_k z^k - z\mathbf{x}_1\mathbf{A}_0 \tag{A.67}$$

By differentiating the above equation and setting $z = 1$ then $\mathbf{X}'(1)$ and $\mathbf{X}''(1)$ can be computed. The derivation is omitted and its details can be found in [Neu89] section 3.3 . The final results and the intermediate quantities needed are listed here in the order of their computation. $\mathbf{B}(z)$ is defined as

$$\sum_{k=1}^{\infty} \mathbf{B}_k z^k \tag{A.68}$$

Then the following vectors are defined as

$$\mathbf{b}_n = \mathbf{B}^{(n)}(1)e = \sum_{k=n}^{\infty} k(k-1)\dots(k-n+1)\mathbf{B}_k\mathbf{e} \qquad n \geq 1 \tag{A.69}$$

and

$$\mathbf{B}(1)\mathbf{e} = \mathbf{b}_0 \tag{A.70}$$

Note that these vectors are easily computed from the input data as $\mathbf{B}_k$ is negligible for large k. $\mathbf{X}(1)$ is given by

$$\mathbf{X}(1) = [\mathbf{x}_0\mathbf{B}(1) - \mathbf{x}_1\mathbf{A}_0](\mathbf{I} - \ _z\mathbf{A} + \mathbf{e}\pi)^{-1} + (1 - \mathbf{x}_0\mathbf{e})\pi \tag{A.71}$$

where ${}_z\mathbf{A} = {}_z\mathbf{A}(1)$ and π is its stationary vector. The matrix $\mathbf{U}$ and its derivatives are defined as

$$\mathbf{U}(z) = z\mathbf{x}_0\mathbf{B}(z) - z\mathbf{x}_1\mathbf{A}_0 \quad (A.72)$$

$$\mathbf{U}(1) = \mathbf{x}_0\mathbf{B}(1) - \mathbf{x}_1\mathbf{A}_0 \quad (A.73)$$

$$\mathbf{U}'(1) = \mathbf{x}_0\mathbf{B}'(1) + \mathbf{x}_0\mathbf{B}(1) - \mathbf{x}_1\mathbf{A}_0 \quad (A.74)$$

$$\mathbf{U}^{(n)}(1) = \mathbf{x}_0\mathbf{B}^{(n)}(1) + n\mathbf{x}_0\mathbf{B}^{(n-1)}(1) \quad n \geq 2 \quad (A.75)$$

and the row sums of these matrices are given by

$$\mathbf{U}(1)\mathbf{e} = \mathbf{0} \quad (A.76)$$

$$\mathbf{U}'(1)\mathbf{e} = \mathbf{x}_0\mathbf{b}_1 \quad (A.77)$$

$$\mathbf{U}^{(n)}(1)\mathbf{e} = \mathbf{x}_0\mathbf{b}_n + n\mathbf{x}_0\mathbf{b}_{n-1} \quad n \geq 2 \quad (A.78)$$

Note that the $\mathbf{U}$ matrix , its derivatives and row sums are all computable from already computed matrices and vectors. We now define $\mathbf{R}_n = \mathbf{X}^{(n)}(1)\mathbf{e}$ and $\beta_v = \mathbf{A}^{(v)}(1)\mathbf{e}$ for $v \geq 1$. The matrix $(\mathbf{I} - {}_z\mathbf{A} + \mathbf{e}\pi)^{-1}$ is denoted by $\mathbf{Z}$. The following equations uses all the above matrices and they are computed in the order shown as each of them uses the ones previous to it.

$$\theta_1 = [\mathbf{U}'(1) + \mathbf{X}(1)\mathbf{A}'(1) - \mathbf{X}(1)]\mathbf{Z}\beta \quad (A.79)$$

$$\mathbf{R}_1 = [2(1-\rho)]^{-1}[2\theta_1 + \mathbf{U}''(1)\mathbf{e} + \mathbf{X}(1)\beta_2] \tag{A.80}$$

$$\mathbf{X}'(1) = \mathbf{R}_1\pi + [\mathbf{U}'(1) + \mathbf{X}(1)\mathbf{A}'(1) - \mathbf{X}(1)]\mathbf{Z} \tag{A.81}$$

$$\theta_2 = [\mathbf{U}''(1) + 2\mathbf{X}(1)'(1)\mathbf{A}'(1) + \mathbf{X}(1)\mathbf{A}''(1) - 2\mathbf{X}'(1)]\mathbf{Z}\beta \tag{A.82}$$

$$\mathbf{R}_2 = [3(1-\rho)]^{-1}[3\theta_2 + \mathbf{U}'''(1)\mathbf{e} + 3\mathbf{X}'(1)\beta_2 + \mathbf{X}(1)\beta_3] \tag{A.83}$$

$$\mathbf{X}''(1) = \mathbf{R}_2\pi + [\mathbf{U}''(1) + 2\mathbf{X}'(1)\mathbf{A}'(1) + \mathbf{X}(1)\mathbf{A}''(1) - 2\mathbf{X}'(1)]\mathbf{Z} \tag{A.84}$$

Appendix B

Definitions

B.1 Delta Function

The delta function, also known as the Dirac delta function, is given by

$$\delta(t) = \begin{cases} \infty & t = 0 \\ 0 & t \neq 0 \end{cases}$$

and

$$\int_{-\infty}^{\infty} \delta(t) dt = 1 \tag{B.1}$$

B.2 Unit Step Function

The unit step function is defined by

$$u(t) = \begin{cases} 1 & t \geq 0 \\ 0 & t < 0 \end{cases}$$

Note that

$$\int_{-\infty}^{t} \delta(s)\, ds = u(t) = \begin{cases} 1 & t \geq 0 \\ 0 & t < 0 \end{cases}$$

These functions are useful in applied probability in modeling problems that have both discrete and continuous nature see [Kle75, pages 372-374].

B.3 Computation of the function ϱ

The function ϱ_1 is given by

$$\varrho(i,j,x,y) = \int_{xT_s}^{(x+1)T_s} \frac{1}{T_s} p(t;i) p((x+y+1)T_s - t; j) dt \tag{B.2}$$

it can be split as follows

$$\varrho(i,j,x,y) = \sum_{l=0}^{j} \sum_{k=0}^{i} p(xT_s;k) \varrho(i-k, j-l, 0, 0) p(yT_s; l) \tag{B.3}$$

where

$$\varrho(i,j,0,0) = \frac{(\lambda T_s)^{i+j} e^{-\lambda T_s}}{i!j!} \int_0^1 t^i (1-t)^j dt \tag{B.4}$$

and

$$\int_0^1 t^i (1-t)^j dt = \frac{j!}{\prod_{n=1}^{j+1} (i+n)} \tag{B.5}$$

so finally after some algebraic manipulation we get

$$\begin{aligned} \varrho(i,j,x,y) = \quad & e^{-\lambda(x+y+1)T_s} (\lambda T_s)^{(i+j)} * \\ & \sum_{l=0}^{i} \sum_{k=0}^{j} \frac{x^k y^l}{k! l! (i+j-k-l+1)!} \end{aligned} \tag{B.6}$$

Appendix C

Derivations

C.1 The moment generating function of the output process

In this section the transform of equation

$$T_r^o = X_1 + X_2 + \ldots X_{N_T+1} + N_T \tag{C.1}$$

is drived. We have

$$\begin{aligned}
f_{T_o}(j) &= Pr\{T_r^o = j\} && \text{(C.2)} \\
&= \sum_{n=0}^{\infty} Pr\{N_T = n\} Pr\{X_1 + X_2 \ldots + X_{N_T+1} + N_T = j\} && \text{(C.3)} \\
&= \sum_{n=0}^{\infty} Pr\{N_T = n\} Pr\{X_1 + X_2 \ldots + X_{n+1} = j - n\} && \text{(C.4)} \\
&= \sum_{n=0}^{\infty} f_{N_T}(n)\{f_{N_i}(j-n)\}^{(n+1)*} && \text{(C.5)}
\end{aligned}$$

where $\{f\}^{y*}$ means that f is convolved with itself y times.

The generating function is given by

$$_zT_o(z) = \sum_{j=0}^{\infty} f_{T_o}(j) z^j \tag{C.6}$$

$$= \sum_{j=0}^{\infty}\sum_{n=0}^{\infty} f_{N_T}(n)\{f_{N_i}(j-n)\}^{(n+1)*} z^j \tag{C.7}$$

$$= \sum_{n=0}^{\infty} f_{N_T}(n)(\sum_{j=0}^{\infty}\{f_{N_i}(j-n)\}^{(n+1)*} z^{j-n}) z^n \tag{C.8}$$

$$= \sum_{n=0}^{\infty} f_{N_T}(n) z^n \;{}_zN_i^{n+1}(z) \tag{C.9}$$

$$= \;{}_zN_i(z)\;{}_zN_T(z\;{}_zN_i(z)) \tag{C.10}$$

Appendix D

Glossary of Notation

- $\mathbf{A}_v$ Transition probability matrix that defines transitions from level i to level $i-1+v$. $i > 1$.
- $\mathbf{B}_v$ Transition probability matrix that defines transitions from level 0 to level v.
- $\mathbf{C}_0$ Transition probability matrix that defines transitions from level 1 to level 0.
- CD_n The value of the countdown counter at the beginning of the n^{th} empty slot after decrementing it if no transmission took place. If transmission took place then the state is taken after copying the request counter in it.
- $\mathbf{D}(j,k,x)$ The conditional probability vector that a tagged segment will be transmitted after x slots from the first renewal it observed given that the node was at level j and there are $k < j$ segments ahead of the tagged segment.
- f_{busy} Probability mass function of the time until next renewal during the busy period.
- f_{cidle} Continuous representation of f_{idle}.
- F_{cidle} Continuous representation of the distribution of f_{idle}.
- $f_{creq}(t,i,j)$ The conditional probability of i requests in t slots given that i cannot be greater than j.

- f_{dlife} discrete representation of f_{life}.
- F_{dlife} discrete representation of F_{life}.
- f_{dres} discrete representation of f_{res}.
- f_{idle} Probability mass function of the time until next renewal during the idle period.
- f_{init} Probability mass function of the time until next renewal after the switch from the idle period to the busy period. Same as f_{dres} except that it is bounded.
- f_{life} Probability density function of the life time in the special renewal period.
- F_{life} Probability distribution function of the life time in the special renewal period.
- $f_{req}(t, i)$ The probability of i requests in t slots.
- f_{res} Probability density function of the residual time in the special renewal period.
- F_{res} Probability distribution function of the residual time in the special renewal period.
- I_n The number of packets in the node at the beginning of each empty slot after subtracting the packet transmitted in this slot, if any.
- N_b^i Random variable denoting number of busy slots between renewals at the input of a node.
- N_b^o Random variable denoting number of busy slots between renewals at the output of a node.
- N_t Random variable denoting the number of transmissions in successive renewals.
- $\mathbf{P}(k)$ Conditional probability vector whose i^{th} element is the probability that the next renewal without transmission is after k renewals with transmission given that at the last renewal without transmission the process was at state i.
- $\mathbf{P}(k, i)$ A partition of the vector $\mathbf{P}(k)$ that represent the set of states at level i.
- $\mathbf{P}(k, i, x)$ The vector $\mathbf{P}(k, i)$ with the condition on the number of slots x between successive renewals without transmission.

- $Q_{ii'jj'kk'}$ Elements of the matrix $\mathbf{Q}(x)$.

- $\mathbf{Q}(x)$ Markov renewal matrix of the M/G/1 type representing a DQDB node.

- $\hat{\mathbf{Q}}(x)$ Markov renewal matrix of the M/G/1 type representing a DQDB node with the inclusion of the indicator random variable TR_n.

- $\hat{\mathbf{Q}}_{11}$ The submatrix of $\hat{\mathbf{Q}}$ that represents transitions from S^T to S^T.

- $\hat{\mathbf{Q}}_{12}$ The submatrix of $\hat{\mathbf{Q}}$ that represents transitions from S^T to S^E.

- $\hat{\mathbf{Q}}_{21}$ The submatrix of $\hat{\mathbf{Q}}$ that represents transitions from S^E to S^T.

- $\hat{\mathbf{Q}}_{22}$ The submatrix of $\hat{\mathbf{Q}}$ that represents transitions from S^E to S^E.

- r_{max} Maximum number of requests the tagged node can accumulate.

- RQ_n The value of the request counter at the beginning of the n^{th} empty slot including the one registered on the reverse bus, if any, and after decrementing it if in the idle state.

- $\mathbf{R}_I$ A matrix that defines transition probabilities for the renewal period in which the tagged segment arrives with the condition that no transmission takes place at the end of the period.

- $\mathbf{R}_T$ A matrix that defines transition probabilities for the renewal period in which the tagged segment arrives with the condition that transmission takes place at the end of the period.

- S^E The set of states of $\hat{\mathbf{Q}}$ in which no transmission takes place.

- S^T The set of states of $\hat{\mathbf{Q}}$ in which transmission takes place.

- T_n The time at the beginning of the n^{th} empty slot after updating the state variables.

- T_r^i Random variable denoting the time until next renewal at the input of the node.

- T_r^o Random variable denoting the time until next renewal at the output of the node.

- TR_n an indicator random variable which is equal one if transmission took place at the n^{th} empty slot and 0 otherwise.

- $\mathbf{x}$ Steady state probability vector of $\hat{\mathbf{Q}}$.

- $\mathbf{x}(1)$ The partition of $\mathbf{x}$ that represent the steady state probabilities of the states in S^T.

- $\mathbf{x}(1)$ The partition of $\mathbf{x}$ that represent the steady state probabilities of the states in S^E.

- $\mathbf{\Gamma}_v$ The matrix component of $\mathbf{B}_0$ that defines transitions during the idle period.

- $\mathbf{\Omega}_v$ The matrix component of the $\mathbf{A}_v$ matrix that corresponds to $v-1$ arrivals and no transmission.

- $\phi_v(t)$ The probability of v segment arrivals at the tagged node in t time units.

- $\mathbf{\Psi}_v$ The matrix component of $\mathbf{B}_v$ that defines transitions from the last renewal epoch until the node switches from idle state to countdown state.

- τ_n The duration between two successive renewals, which is equal to $T_n - T_{n-1}$.

- Θ A component of the matrix $\mathbf{B}_0$ which defines transitions for the arrival of one packet that is transmitted in the next empty slot.

- $\mathbf{\Upsilon}_v$ The matrix component of the $\mathbf{A}_v$ matrix that corresponds to v arrivals and one transmission.

- $\varphi_v(x)$ The probability that the first arrival is after the beginning of the $(x-1)^{th}$ slot and there are more $v-1$ arrivals until the beginning of the x^{th} slot.

- $\varrho(i,j,x,y)$ The probability of i and j arrivals before and after the tagged segment arrival, respectively, with the condition that the tagged segment will arrive during the $(x+1)^{th}$ slot.

- $\mathbf{\Xi}_v$ The matrix component of $\mathbf{B}_v$ that defines transitions from the time the node switched from idle state to countdown state until the next renewal epoch.

- $\dot{\mathbf{\Xi}}_v$ The matrix component of the $\mathbf{\Xi}_v$ matrix that corresponds to v arrivals and one transmission.

- $\ddot{\mathbf{\Xi}}_v$ The matrix component of the $\mathbf{\Xi}_v$ matrix that corresponds to $v-1$ arrivals and no transmission.

- $_{z}.X$ Moment generating function of random variable X.

Bibliography

[Bis90] Chatschik Bisdikian. Waiting time analysis in a single buffer DQDB (802.6) network. *IEEE Journal on Selected Areas in Communications*, 8(8):1565–1573, October 1990.

[Bis92] Chatschik Bisdikian. A performancs analysis of the IEEE 802.6 (DQDB) subnetwork with the bandwidth balancing mechanism. *Computer Networks and ISDN Systems*, pages 367–385, 1992.

[CGL89] Marco Conti, Enrico Gregori, and Luciano Lenzini. Dqdb media access control protocol: Performance evaluation and unfairness analysis. In *3rd IEEE workshop on MANs*, San Diego, March 1989.

[CGL91a] Marco Conti, Enrico Gregori, and Luciano Lenzini. DQDB modeling: Problem complexity reduction and solution via markov chains. In *Proc. Int. Conf. on the Performance of Distributed Systems and Integrated Communication Networks*, pages 21–31, Kyoto Japan, 1991.

[CGL91b] Marco Conti, Enrico Gregori, and Luciano Lenzini. A methodological approach to an extensive analysis of DQDB performance and fairness. *IEEE Journal on Selected Areas in Communications*, 9(1):76–87, January 1991.

[CGL92a] Marco Conti, Enrico Gregori, and Luciano Lenzini. On the approximation of the slot occupancy pattern in a DQDB network. *Performance Evaluation*, 16:159–176, 1992.

[CGL92b] Marco Conti, Enrico Gregori, and Luciano Lenzini. On the approximation of the slot occupancy pattern in a DQDB network. In Infocom92 [Inf92], pages 518–526.

[CGLN94] Marco Conti, Enrico Gregori, Luciano Lenzini, and M.F. Neuts. An M/G/1 type approach to the approximation of the slot-occupancy pattern in a DQDB network. *Performance Evaluation*, 21:59–80, 1994.

[CMM95] Roger Chen, Georges Makhoul, and Dikran Meliksetian. A queueing analysis of the performance of dqdb. *IEEE/ACM Transactions on Networking*, 3(6):1565–1573, December 1995.

[Fel57] William Feller. *An introduction to Probability Theory and its Applications* . volume 1. John Wiley and Sons Inc., 1957.

[HCM90] E.L. Hahne, A Choudhury, and N.F. Maxemchuk. Improving fairness of DQDB networks. In *Proceedings of the 9th Annual joint conference of the IEEE Computer and Communications Societies on Networking*, pages 175–184, San Francisco, CA., 1990.

[Inf92] The Institute of Electrical and Electronics Engineers. *Proceedings of the 11th Annual joint conference of the IEEE Computer and Communications Societies on Networking*, 1992.

[JP92] Wen Jing and Michael Paterakis. Message delay analysis of the DQDB (IEEE 802.6) network. In Infocom92 [Inf92], pages 527–535.

[JP95] Wen Jing and Michael Paterakis. Extending the single-node dqdb analytical model to analyze network-wide performance. *Performance Evaluation*, 27:653–675, 1995.

[Kle75] Leonard Kleinrock. *Queueing Systems* , volume 1. John Wiley and Sons Inc., 1975.

[MC92] B. Mukherjee and C.Bisdikian. A journey through the DQDB network literature. *Performance Evaluation*, 16:129–158, 1992.

[Neu81] Marcel F. Neuts. *Matrix-Geometric Solutions in Stochastic Models An Algorithmic Approach.* Johns Hopkins University Press., 1981.

[Neu89] Marcel F. Neuts. *Structured Stochastic Matrices of M/G/1 Type and Their Applications.* Marcel Dekker , Inc., 1989.

[PGS90] P.Tran-Gia and Thomas Stock. Approximate performance analysis of the DQDB access protocol. *Computer Networks and ISDN Systems*, pages 231–240, 1990.

[Ram88] V. Ramaswami. A stable recursion for the steady state vector in markov chains of the M/G/1 type. *Stochastic Models*, pages 183–188, 1988.

[ST93] Ioannis Stavrakakis and Sophia Tsakiridou. Occupancy distribution for a dqdb station based on a queueing system with markov-structured service requirements. In *Proceedings of the 12th Annual joint conference of the IEEE Computer and Communications Societies on Networking*, pages 1083–1090, San Francisco, CA., 1993.

[ST94] Ioannis Stavrakakis and Sophia Tsakiridou. A markov service policy with application to the queueing study of a dqdb station. *Performance Evaluation*, 26:1503–1522, 1994.

[Sta90] IEEE 802.6 Standard. Distributed queue dual bus (DQDB) metropolitan area network, 1990.

[vAWZ90] H. R. van As, J.W. Wong, and P. Zafiropoulo. Fairness, priority and predictability of the dqdb mac protocol under heavy load. In *Proc. 1990 International Seminar on Digital Communications*, pages 231–239, Zurich Switzerland, March 1990.

[Won89] J.W. Wong. Throughput of dqdb networks under heavy load. In *Proc. European Fibre Optic Communciations and LAN Exposition*, pages 146–151, Amsterdam, June 1989.