

A Stochastic Reward Net Model for Performance Analysis of Prioritized DQDB MAN*

Hairong Sun, Xinyu Zang and Kishor S. Trivedi
{hairong, xzang, kst@ee.duke.edu}
Center for Advanced Computing and Communications
Department of Electrical and Computer Engineering
Duke University
Durham, NC 27708

Abstract

The performance of prioritized DQDB MAN under bursty traffic environment is studied in this paper. The tagged node model is adopted to simplify the analysis. The processes of the packet arrivals to the tagged node, the empty-slot pattern from upstream, and the request pattern from downstream are assumed to be Markov Modulated Poisson Processes, a more general and appropriate model to capture the bursty characteristics existing in the traffic flows in high-speed networks. A stochastic reward nets model for the prioritized DQDB MAN is presented. With the help of the Stochastic Petri Net Package (SPNP) developed in Duke University, the steady state and as well as the transient behavior of packet loss ratio of tagged node are investigated. The influence of the burstiness of the empty-slot pattern, packet arrival and request pattern on the tagged node's performance are examined.

Index Terms: DQDB, Priority, Stochastic Reward Nets, Stochastic Petri Net Package (SPNP)

*This research was supported in part by the National Science Foundation under Grant No. EEC9418765, and by the Department of Defense as an enhancement Project to the Center for Advanced Computing and Communications in Duke University.

1 Introduction

As a standard of Metropolitan Area Network(MAN), IEEE 802.6 DQDB (Distributed Queue Dual Bus) has been studied extensively for the past ten years. A broadband connectionless service for Local Area Networks(LAN) interconnection, called Switched Multimegabit Data Service (SMDS), is already in operation, which adopts DQDB as its Medium Access Control (MAC) protocol. Because of the simplicity of the medium access control[1], high utilization of channel bandwidth independent of data transmission rate and network size, and high compatibility with Asynchronous Transfer Mode (ATM) at cell level and adaptation protocol level, DQDB is a competitive candidate for the access network to Broadband Integrated Service Digital Network (B-ISDN). A recent study has shown that as a MAC protocol for the customers requiring a high degree of quality, DQDB outperforms ATM Passive Optical Network (APON)[2]. Many researchers have studied how to apply DQDB to multichannel Wavelength Division Multiplexing (WDM) networks [3, 4] and use DQDB to interconnect base stations in Personal Communication Networks (PCN) [5, 6]. Thus, DQDB will have an important role in future network infrastructure.

Principal components of DQDB networks consist of two head stations, two high-speed fiber optic unidirectional buses carrying information in opposite directions (i.e., *Bus A* and *Bus B*), and sets of network stations (i.e., nodes) numbered from 1 to N (see Fig. 1). Each head station is attached to one of the buses, and generates slots of fixed duration that travel along the respective bus. The network nodes are attached to both buses via read and write taps. Nodes send slots containing 53-byte packets to the downstream nodes via *Bus A*. They send packets to the upstream nodes via *Bus B*. Access of the buses by the nodes is controlled by the MAC protocol that is responsible for maintaining a distributed queue of all the nodes requesting service on the buses. A node can use an empty slot (i.e., the *Busy*

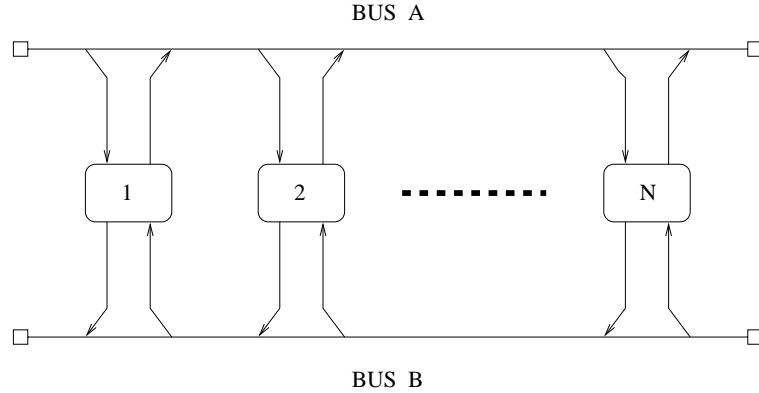


Figure 1: Architecture of DQDB Network

bit field in the slot is 0) only when there is no packet of downstream nodes waiting for transmission, and the node set the *Request* bit field of the slot on the reverse bus to 1, to notify the upstream nodes that there is a packet waiting for transmission.

Although the MAC protocol of DQDB is very simple to be implemented in hardware, the exact analytical solution to the performance of the protocol is rather difficult and even impractical, because of the high degree of interaction among a plethora of stochastic processes and nodes in the DQDB networks. To simplify the analysis, a tagged node model was introduced by C. Bisdikian [7], which partitions the network into an L-NET, an R-NET by aggregating together all the nodes located to the left and the right of the tagged node, respectively (see Fig. 2). Because the tagged node does not distinguish the origin of the slot passing by, this aggregation does not change the behavior of the tagged node. The performance of the tagged node will be mainly determined by the design parameters of the node and three stochastic processes, i.e., the arrival process of the packets to the tagged node, the request pattern from the downstream nodes, and the slot-occupancy (i.e., Busy bit is 1) pattern passing by from upstream nodes. This simplified model is approximate,

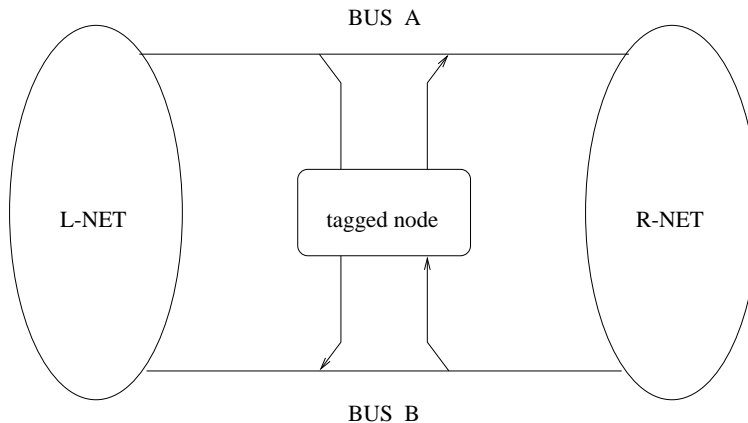


Figure 2: Tagged Node Model of DQDB Network

however, it provides us a window into the behavior of the MAC protocol and interactions among the nodes. The tagged node model can also be extended to analyze network-wide performance [10]. In [7], all these processes are assumed to be Bernoulli, and the tagged node can only buffer one packet. Other researchers adopted this model and assumed the tagged node can buffer one message (containing multiple packets) [8]. In [9], we have analyzed prioritized DQDB with this model and assumed the packet arrivals to the tagged node is bursty, rather than Bernoulli. The above three papers assumed the slot-occupancy pattern is a Bernoulli process.

Recent studies have shown that the slot-occupancy pattern seen by the tagged node is more complex than Bernoulli, and the interdependence between the adjacent slots is significant, i.e., a slot is more likely to be busy if the previous slot was busy [11, 12]. These researchers also developed an n th-order discrete-time Markov chain to describe the interdependence in the slot-occupancy pattern process. They found that the influence of the slot-occupancy pattern on the access delay experienced by the nodes was significant. However, in their studies, the packet arrival process and request arrival process were assumed

to be Poisson, and only one priority-level was considered.

In this paper, we analyze the performance of prioritized DQDB MAN under a more realistic traffic process than that in [9]. We adopt the tagged node model and present a more general process for the packet arrival to the tagged node, slot-occupancy pattern from the upstream nodes, and request arrival pattern from downstream nodes. All these processes are assumed to be Markov Modulated Poisson Processes (MMPP), which is more accurate and appropriate to describe the burstiness of traffic in high-speed networks. We construct a Stochastic Reward Net (SRN) model to describe the behavior of the tagged node. With the help of the Stochastic Petri Net Package (SPNP) [16, 17], a powerful tool for the automated generation and the numerical solution of Markovian stochastic process, the transient and steady-state behavior of the tagged node is studied.

The organization of the paper is as follows. In Section 2, a brief description of the MAC protocol of prioritized DQDB is given, and the traffic model and the tagged node model are also presented. In Section 3, the SRN model for the tagged node's behavior is presented. Numerical results obtained by using the SPNP are discussed in Section 4. Section 5 concludes the paper.

2 System Description and Assumptions

2.1 Brief Description of Prioritized DQDB

The topology of the DQDB MAN is shown in Fig. 1. Each slot generated by the head stations contains in its header field a *Busy* bit and three *Request* bits. The *Busy* bit indicates whether or not the slot is occupied by a packet. The *Request* bits are used by the nodes to send requests for future packet transmission at the respective priority levels. The DQDB

provides three priority levels to provide the users with different Quality of Service (QoS). Each node has three separate buffers to accommodate the packets belonging to different priority levels and three sets of counters to implement the MAC protocol. In this paper, we only consider two priority levels, and from the latter derivation, it is obvious that the analysis in the paper can be easily extended to the case of three-priority-level DQDB networks. Of course, the number of states of the Markovian chain is larger and more powerful computer is needed to get the numerical results in the three-priority-level case.

If there are packets of priority level I waiting for transmission in a node, the node is called active at priority level I , otherwise it is idle at priority level I ($I = 0, 1, 2$). Higher this number is, higher the priority level is. If the node is idle at priority level I , a request counter $REQ-I-CNTR$ increases by one for each request with priority level equal to or higher than I received on *Bus B* and decreases by one for each empty slot passing by on *Bus A*. (In this paper, we focus our attention to packet transmission on *Bus A*, the procedure and performance of transmission on *Bus B* is similar.) When it becomes active at priority level I , the node transfers the content of the $REQ-I-CNTR$ to another counter named counter-down counter $CD-I-CNTR$, and then resets $REQ-I-CNTR$ to zero. The node then sends a request at priority level I on *Bus B* to upstream by setting the corresponding *Request* bit of the first request-free slot passing by on *Bus B*. Then, the $REQ-I-CNTR$ is increased by one for each request with priority level equal to I received on *Bus B*. The $CD-I-CNTR$ is increased by one for each request with priority level higher than I received on *Bus B*, and is decreased by one for each empty slot passing in front of the node on *Bus A* until it reaches zero. When the $CD-I-CNTR$ is zero, the node transmits the packet of priority level I via the first empty slot passing by on *Bus A*, if there are no higher-priority-level packets waiting to be transmitted at this node.

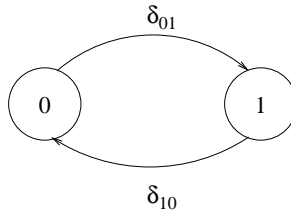


Figure 3: Two-State MMPP

In order to deal with the unfairness introduced by the node location, a bandwidth balancing mechanism is adopted in the DQDB standard. The mechanism ensures that each node in the network only exploits a percentage of its transmission opportunity. In practice, a counter named bandwidth balance counter is used to count the number of packets transmitted by the node. Once it reaches a given threshold, named $BWB-MOD$, it is reset to zero and all the $REQ-I-CNTRs$ and $CD-I-CNTRs$ of the node act as if there is a highest-priority-level request passing by. It means that the tagged node only captures $BWB-MOD/(BWB-MOD+1)$ percentage of its potential bandwidth. Therefore, the bandwidth balancing mechanism could be approximately viewed as that with the $CD-I-CNTR$ being zero and seeing an empty slot passing by, the tagged node does not transmit the packet immediately but transmits with probability p ($p = \frac{BWB-MOD}{BWB-MOD+1}$) [13].

2.2 Traffic Model and Tagged Node Model

MMPP is used extensively to describe the bursty traffic in high-speed networks [18]. In this paper, we only consider a two-state MMPP, the model we developed can be easily extended to a multi-state MMPP. A two-state MMPP is illustrated in Fig.3, whose packet arrival process is Poisson with rate r_0 while it is at state 0, and is Poisson with rate r_1 while it is at state 1. The transitions between these two states form a homogeneous, continuous time

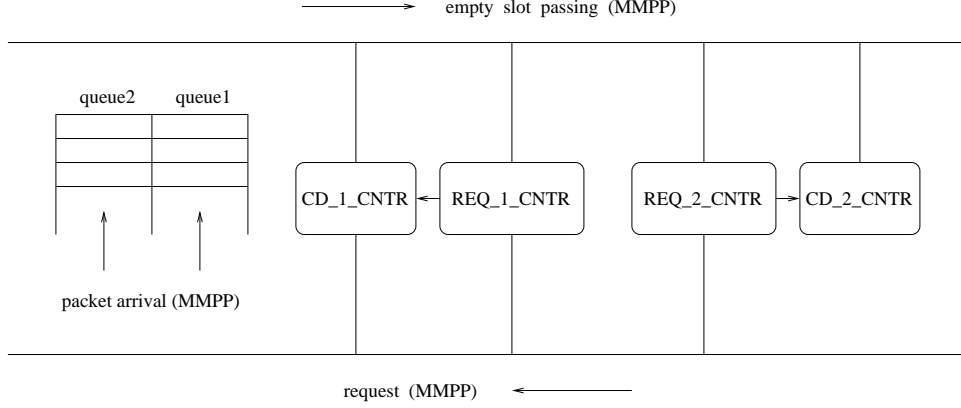


Figure 4: Tagged Node Model of DQDB MAN with Two Priority Levels

Markov chain, with rates δ_{01} and δ_{10} as shown in Fig. 3. The average arrival rate is:

$$\rho = \frac{\delta_{01}r_1 + \delta_{10}r_0}{\delta_{01} + \delta_{10}} \quad (1)$$

Fig. 4 illustrates the tagged node model of DQDB MAN with two priority levels. The packet arrival processes (two priority levels), request arrival processes (two priority levels), and empty-slot pattern (we use empty-slot pattern instead of slot-occupancy pattern for the ease of analysis) are two-state MMPPs. In order to simplify the analysis, we assume the packet arrival patterns of two priority levels are controlled by the same MMPP, and request arrivals of two priority levels are controlled by another MMPP.

Then the state of the tagged node may be characterized by nine stochastic variables:

- $queue1(t)$, $queue2(t)$: the queue length of priority level 1 and 2 at time t ,
- $req1(t)$, $req2(t)$: the value of the $REQ-1-CNTR$ and $REQ-2-CNTR$ at time t ,
- $cd1(t)$, $cd2(t)$: the value of $CD-1-CNTR$ and $CD-2-CNTR$ at time t ,

Table 1: Parameters for the MMPPs

	δ_{01}	δ_{10}	r_0	r_1
Arrival for Higher-priority-level	a_{01}	a_{10}	$\lambda_0(2)$	$\lambda_1(2)$
Arrival for Lower-priority-level	a_{01}	a_{10}	$\lambda_0(1)$	$\lambda_1(1)$
Empty Slot	e_{01}	e_{10}	α_0	α_1
Request for Higher-priority-level	r_{01}	r_{10}	$\beta_0(2)$	$\beta_1(2)$
Request for Lower-priority-level	r_{01}	r_{10}	$\beta_0(1)$	$\beta_1(1)$

- *arrival(t)*: the state where the packet-arrival MMPP of the priority level 1 and 2 is at time t ,
- *request(t)*: the state where the request-arrival MMPP of the priority level 1 and 2 is at time t
- *empty(t)*: the state where the empty-slot MMPP is at time t .

The parameter notations of the MMPPs are listed in Table 1.

Obviously, the average packet arrival rate of the tagged node is:

$$A_{tag} = \sum_{i=1}^2 \frac{a_{01}\lambda_1(i) + a_{10}\lambda_0(i)}{a_{01} + a_{10}} \quad (2)$$

the average arrival rate of requests from downstream nodes is

$$A_{req} = \sum_{i=1}^2 \frac{r_{01}\beta_1(i) + r_{10}\beta_0(i)}{r_{01} + r_{10}} \quad (3)$$

and the average arrival rate of empty slots from upstream nodes is

$$A_{empty} = \frac{e_{01}\alpha_1 + e_{10}\alpha_0}{e_{01} + e_{10}} \quad (4)$$

The system will converge to a steady state if $A_{tag} + A_{req} < A_{empty}$. The approximation made in the paper is that the DQDB network is considered as a continuous-time system,

rather than a discrete-time system. Because of the small packet size and high speed of the DQDB network, considering the DQDB network working at 155Mb/s, the transmission time for one packet is about 2.7 microsecond, the assumption of continuous-time system is reasonable.

The tagged node may be described by a 9-dimensional continuous-time Markov chain (CTMC) where each dimension corresponds to one stochastic variable mentioned above. However, it is a rather tedious and error-prone if not impossible to construct the CTMC by hand. In the next section, we use Stochastic reward nets (SRNs) [15], which can be automatically transformed into CTMCs, to concisely describe the system model. And with the help of the SPNP, the CTMC underlying the system described in the paper is generated and solved automatically.

3 Stochastic Reward Net for the Tagged Node of Prioritized DQDB MAN

Stochastic reward nets are extensions of stochastic Petri nets. We give a brief overview of these constructs before presenting the model of prioritized DQDB MAN.

3.1 Stochastic Reward Nets

A Petri net is a directed graph with two disjoint types of nodes: *places* and *transitions*. A directed arc connecting a place (transition) to a transition (place) is called an input (output) *arc* of the transition. A positive integer called multiplicity can be associated with each arc. Places connected to a transition by input arcs are called the input places of this transition, and those connected by means of output arcs are called the output places. Each

place may contain zero or more tokens. A transition is *enabled* if each of its input places has at least as many tokens as the multiplicity of the corresponding input arc. A transition can *fire* when it is enabled, and upon firing, a number of tokens equal to the multiplicity of the input arc is removed from each of input places, and a number of tokens equal to the multiplicity of the output arc is deposited in each of its output places. The state of a Petri net is characterized by a number of tokens in each place which is called its marking. The initial number of tokens assigned to the places in a Petri net determines the initial state (marking) of the net.

Stochastic Petri nets are Petri nets where exponentially distributed firing time is attached to each transition. In Generalized Stochastic Petri nets (GSPN) [14], transitions are allowed to be either *timed* (exponentially distributed firing time) or *immediate* (zero firing time). A marking of a GSPN is called *vanishing* if at least one immediate transition is enabled in the marking and *tangible* otherwise. GSPN also introduces *inhibitor arc* connecting a place to a transition. A transition with an inhibitor arc can not fire if the input place of the inhibitor arc contains more tokens than the multiplicity of the arc.

Stochastic reward nets (SRNs) are based on GSPN but extend it further [15]. In SRN, every tangible marking can be associated with a reward rate. It can be shown that an SRN can be mapped into a Markov reward model. Therefore, by using software packages [16, 17], the underlying Markov reward model of an SRN can be automatically generated and solved. Thus a variety of performance measures can be specified and calculated using a very convenient formalism. SRN also allows several other features that makes specification convenient: (1) each transition may have an enabling function (also called a guard) so that a transition is enabled only if its (marking dependent) enabling function is true, (2) marking dependent arc multiplicities are allowed, (3) transitions can also have priorities.

To represent an SRN as a graph, places are represented by circles and tokens are black dots (or integers) inside places. Immediate transitions are drawn as thin bars and timed transitions as white boxes. Inhibitor arcs have a small hollow circle instead of arrows at their terminating ends. The current number of tokens in place p is denoted as $\#p$.

3.2 Stochastic Reward Net for the Tagged Node

In order to build a compact and concise SRN for the tagged node, we do not use the DQDB protocol as described in the IEEE draft, but adopt a version taht is equivalent to the standard. Because the higher-priority-level counters have taken the higher-priority-level requests into account, the lower-priority-level counters need not count them repeatedly. What must be guaranteed is that these higher-priority-level requests should be served before the lower-priority-level packets in the queue and the lower-priority-level requests which are counted by lower-priority-level counters. In our version , the counters of higher priority level act as those in the standard, but the lower-priority-level counters do not count the higher-priority-level requests from downstream nodes, i.e., the $CD-I-CNTR$ and $REQ-I-CNTR$ of the lower priority level do not increase as the higher-priority-level requests passing by, and accordingly they decrease only when the higher-priority-level queue is empty, $CD-I-CNTR$ and $REQ-I-CNTR$ of higher-priority-level are zero and there is an empty slot passing by.

Fig. 5 presents the stochastic reward net of the tagged node's behavior. Table 2 gives the guards for the immediate transitions and the rates of the timed transitions.

The three places on the left hand side, i.e., P_{RON} , P_{EON} , and P_{AON} , and their corresponding transitions in Fig. 5 represent three MMPPs described in Table 1. Six places, i.e., P_{REQ2} , P_{BUF2} , P_{CD2} , P_{REQ1} , P_{BUF1} , P_{CD1} represent the sates of the remaining stochastic variables defined in Section 2.2 The transitions T_{REQ2} and T_{REQ1} respectively, correspond

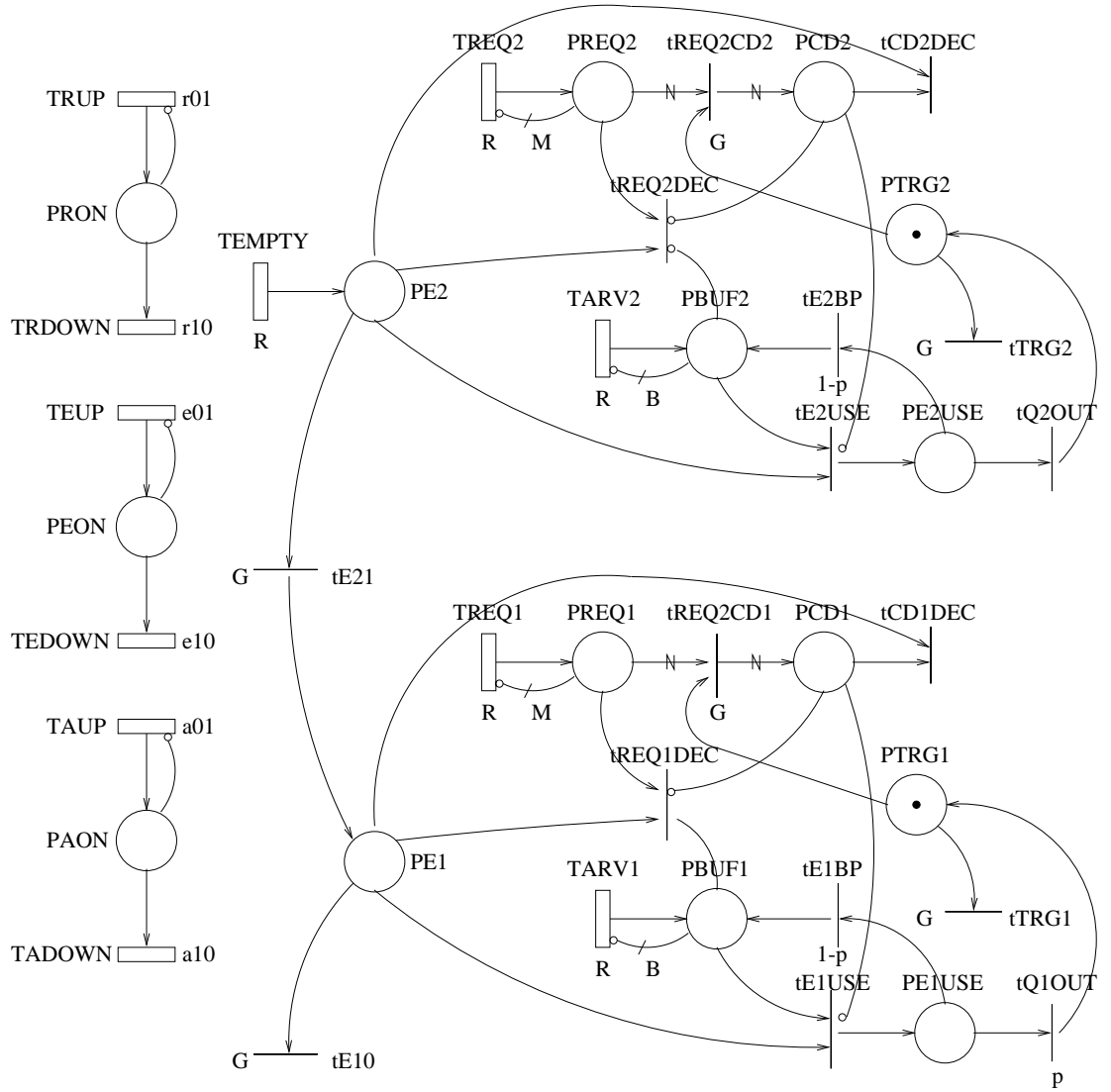


Figure 5: Stochastic Reward Net of the Tagged Node's Behavior

to arrivals of requests of priority level 2 and 1 from downstream nodes, and the places P_{REQ2} and P_{REQ1} , each with capacity M , count the requests.

T_{ARV2} and T_{ARV1} represent the arrivals of packets of priority level 2 and 1, and the places P_{BUF2} and P_{BUF1} , each with capacity B , accomodate these packets.

T_{EMPTY} represents the arrivals of empty slots. The empty slot serves the higher-priority-level subsystem first. If the node is active at priority level 2 and $P_{CD2} > 0$, then P_{CD2} decreases by one for each empty slot passing by until it reaches zero, which is represented by transition t_{CD2DEC} . If the node is active at priority level 2, and $P_{CD2} = 0$, and sees an empty slot passing by, t_{E2USE} fires. The packet is transmitted with probability p , which corresponds to the firing of transition t_{Q2OUT} , and the empty slot is abandoned with probability $1-p$, which corresponds to t_{E2BP} . $t_{REQ2DEC}$ is enabled when there is no tokens in palce P_{BUF2} and $P_{CD2} = 0$ (i.e., the node is idle at priority level 2), and an empty slot passing by.

In DQDB protocol, once the node transits from idle to active at priority level 2, the tokens in P_{REQ2} are transfered to place P_{CD2} , which is represented by transition $t_{REQ2CD2}$ and two variable cardinality of arcs from P_{REQ2} to $t_{REQ2CD2}$ and from $t_{REQ2CD2}$ to P_{CD2} .

The place P_{TRG2} contains one token initially, which means the node is idle. If there is no token in P_{TRG2} at an instant of time, it means the node is active. The immediate transition t_{TRG2} is used to handle the abnormal case, i.e., when the node becomes active and transition $t_{REQ2CD2}$ does not fire because there is no token in P_{REQ2} . It should be mentioned that if there is more than one packet in the queue, once one packet is transmitted soon as the node becomes idle, it transits to active immediately.

If the empty slot is not used by the higher-priority-level subsystem, i.e., none of t_{CD2DEC} , $t_{REQ2DEC}$, t_{E2USE} fire, then the empty slot will pass to P_{E1} to serve the lower-priority-level

Table 2: Rate and Guard for the Transitions in Fig.5

Transition	T_{REQ2}	T_{ARV2}	T_{REQ1}	T_{ARV1}	T_{EMPTY}	
Rate	if($\#P_{RON} > 0$) $\beta_1(2)$ else $\beta_0(2)$	if($\#P_{AON} > 0$) $\lambda_1(2)$ else $\lambda_0(2)$	if($\#P_{RON} > 0$) $\beta_1(1)$ else $\beta_0(1)$	if($\#P_{AON} > 0$) $\lambda_1(1)$ else $\lambda_0(1)$	if($\#P_{EON} > 0$) α_1 else α_0	
Transition	$t_{REQ2CD2}$	t_{TRG2}	$t_{REQ2CD1}$	t_{TRG1}	t_{E21}	t_{E10}
Guard	$\#P_{BUF2} > 0$	$\#P_{REQ2} = 0$ & $\#P_{BUF2} > 0$	$\#P_{BUF1} > 0$	$\#P_{REQ1} = 0$ & $\#P_{BUF1} > 0$	$\#P_{BUF2} = 0$ & $\#P_{REQ2} = 0$ & $\#P_{CD2} = 0$	$\#P_{BUF1} = 0$ & $\#P_{REQ1} = 0$ & $\#P_{CD1} = 0$

Table 3: Size of the CTMC for the SRN in Fig.5(In this paper, we solved the CTMC labeled with *)

Traffic	B	M	number of states	number of arcs
All MMPPs	8	4	362,952	1,882,064
All MMPPs*	16	4	1,417,928	7,438,224
All MMPPs	32	4	5,604,552	29,572,880
All MMPPs	8	8	3,537,800	18,633,296
All Poisson	8	4	84,050	450,178
All Poisson*	16	4	328,050	1,796,578
All Poisson	32	4	1,296,050	7,177,378
All Poisson	8	8	863,298	4,775,074
All Poisson	16	8	3,406,050	19,256,578

subsystem. The behavior of the lower-priority-level subsystem, which is analogy to that of the higher-priority-level subsystem.

The empty slot will pass by to downstream nodes if it is not used by the tagged node. The immediate transition t_{E10} will fire if none of t_{CD2DEC} , $t_{REQ2DEC}$, t_{E2USE} , t_{CD1DEC} , $t_{REQ1DEC}$, and t_{E1USE} fire.

The rates and guards for the transitions are listed in Table 2.

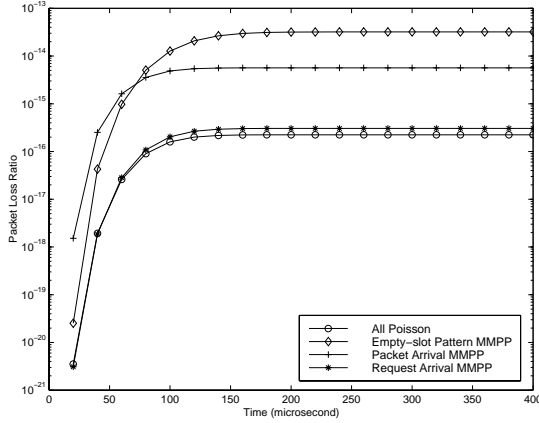
The number of tangible markings and arcs of the reachability graph of the stochastic

reward net in Fig. 5 are listed in Table 3, which represent the number of states and the number of non-zero entries in the infinitesimal generator matrix of the CTMC, respectively. We compare the size of the Markov chain versus the buffer capacity and counter size when all the processes are MMPP or Poisson. The table demonstrates that the size of the Markov chain is rather large and MMPP increases the size sharply. However, with the use of SPNP, we can solve the problem efficiently.

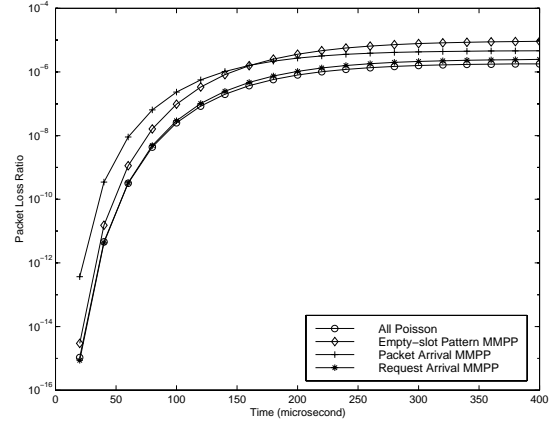
4 Numerical Results

The numerical solution to the transient and steady state of the stochastic reward net illustrated in Fig. 5 is obtained with the help of the SPNP [16, 17]. The performance index we are interested in is the packet loss ratio of the tagged node. Our attention is mainly paid to investigate the influence of the burstiness on this index. In this paper, the burstiness exists in the packet arrival process to the tagged node, empty-slot pattern on *Bus A* and the request from downstream nodes. We assume the buffer capacity for the higher-priority-level packets is 16 packets (i.e., about 800 bytes), and that for the lower-priority-level is 16 packets as well. The bandwidth balancing parameter, *BWB-MOD*, equals to 16. The maximum value of the counters, i.e., *REQ-I-CNTR* and *CD-I-CNTR* is 4. In this paper, we study the behavior of the tagged node under the following load configurations:

- *I*: The empty slot arrival rate of upstream is 0.7, the request arrival rate of downstream is 0.3 (0.2 for the lower priority level, and 0.1 for the higher priority level), and the load of the tagged node is 0.15 (0.1 for the lower priority level, and 0.05 for the higher priority level).
- *II*: The empty slot arrival rate of upstream is 0.7, the request arrival rate of down-



(a) Higher Priority



(b) Lower Priority

Figure 6: Transient Behavior of the Tagged Node under Traffic Configuration I

stream is 0.5 (0.3 for the lower priority level, and 0.2 for the higher priority level), and the load of the tagged node is 0.15 (0.1 for the lower priority level, and 0.05 for the higher priority level).

4.1 Numerical Results: Transient Analysis

The transient analysis is known to be an analytically non-tractable problem even for simple queueing systems. For the 9-dimensional CTMC in this paper, it is almost impossible to get the transient solution with traditional approach, i.e., constructing the Markov chain with hand and then writing a program to solve the ordinary differential equations. The Stochastic Petri Net Package provides a C-like specification language that facilitates the automated generation and solution of the Markov chain, even if the number of the states of the underlying Markov chain is very large.

The transient behavior of the tagged node in this subsection is studied under traffic

configuration *I*. Fig. 6 illustrates the transient state of packet loss ratio in following cases:

- the empty-slot pattern, the packet arrival, and the request arrival are Poisson processes.
- the packet arrival, and the request arrival are Poisson processes, but the empty-slot pattern is an MMPP, with $e_{01} = 0.5$, $e_{10} = 0.1$, $\alpha_0 = 0$, and $\alpha_1 = 0.84$. The MMPP describes the burstiness of the empty-slot pattern, and $\frac{1}{e_{01}}$ represents the average duration of the consecutive busy slots passing by on *Bus A*. Apparently, the burstiness of the empty-slot pattern does have effect on the performance of the tagged node.
- the request arrival, the empty-slot pattern are Poisson processes, but the packet arrival is an MMPP, with $\lambda_0(1) = \lambda_0(2) = 0$, $\lambda_1(1) = 0.2$, $\lambda_1(2) = 0.1$, and $a_{01} = a_{10} = 0.5$.
- the packet arrival and the empty-slot pattern are Poisson processes, but the request arrival is an MMPP, with $\beta_0(1) = \beta_0(2) = 0$, $\beta_1(1) = 0.4$, $\beta_1(2) = 0.2$, and $r_{01} = r_{10} = 0.5$.

Apparently, the burstiness of the empty-slot pattern, the request arrival, and the packet arrival do have effect on the performance of the tagged node, although significance of their impacts are different, while the burstiness of the empty-slot pattern has the most significant effect, and the burstiness of the request arrival has the least significant influence. All the burstiness that exist in these stochastic processes should be considered while we model and analyze the performance of DQDB networks. The previous assumptions on the Bernoulli-type empty-slot pattern, request arrival, or packet arrival underestimated the packet loss ratio of the DQDB networks.

Because $A_{tag} + A_{req} < A_{empty}$, the system approaches a steady-state as time elapses.

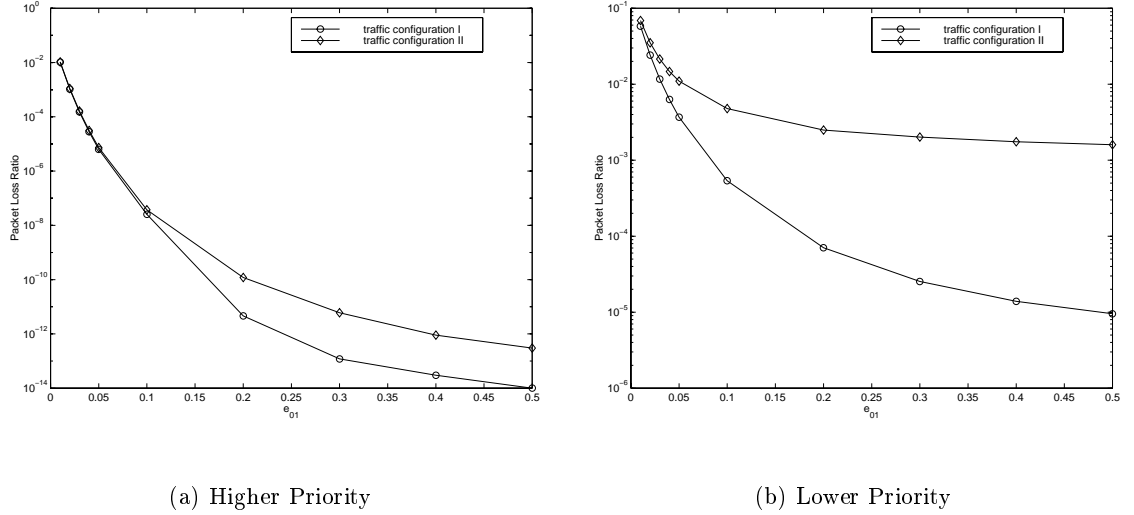


Figure 7: Packet Loss Ratio vs. the Burstiness of the Empty-slot Pattern

4.2 Numerical Results: Steady-State Analysis

4.2.1 The Effect of the Burstiness of the Empty-Slot-Pattern

From the transient analysis, the effect of the burstiness of the empty-slot pattern on the performance of the tagged node are verified. In order to investigate the effect further, Fig. 7 illustrates the packet loss ratio versus the burstiness of the empty-slot pattern. The packet arrival, and the request arrival are Poisson processes, but the empty-slot pattern is MMPP, with $e_{01} = 5e_{10}$, $\alpha_0 = 0$, and $\alpha_1 = 0.84$. Obviously, the more bursty the empty-slot pattern is, the larger the packet loss ratio is.

4.2.2 The Effect of the Burstiness of Packet Arrival Process

Fig. 8 illustrates the effect of the bustiness of the packet arrival process on the performance of the tagged node. The request arrival and the empty-slot pattern are Poisson processes,

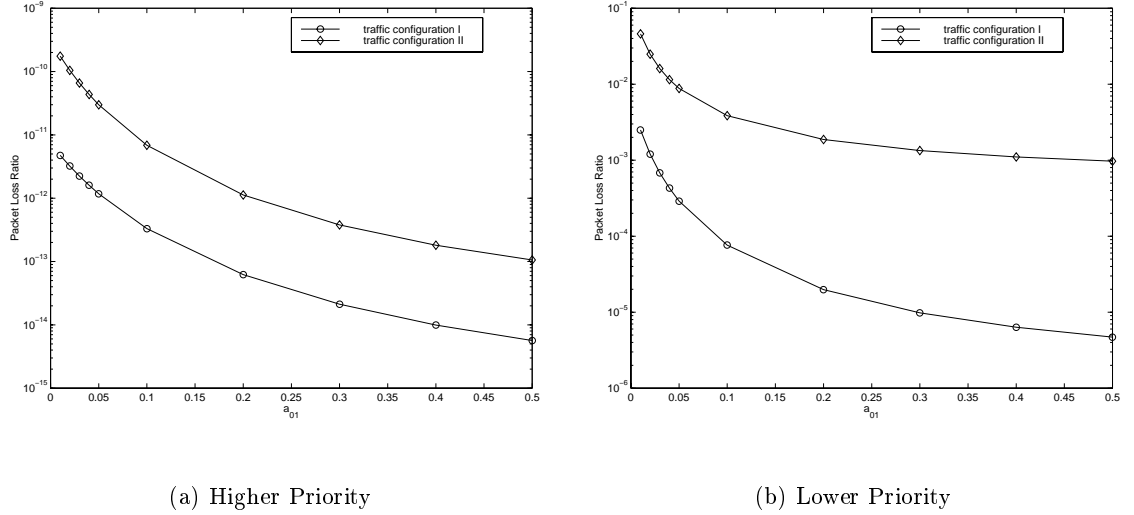


Figure 8: Packet Loss Ratio vs. the Burstiness of the Packet Arrival

but the packet arrival is MMPP, with $\lambda_0(1) = \lambda_0(2) = 0$, $\lambda_1(1) = 0.2, \lambda_1(2) = 0.1$, and $a_{01} = a_{10}$.

4.2.3 The Effect of the Burstiness of the Request Arrival

Fig. 9 illustrates the effect of the bustiness of the request arrival process on the performance of the tagged node. The packet arrival and the empty-slot pattern are Poisson processes, but request arrival is MMPP, with $\beta_0(1) = \beta_0(2) = 0$, $\beta_1(1) = 0.4, \beta_1(2) = 0.2$, and $r_{01} = r_{10}$ for traffic configuration *I*, and with $\beta_0(1) = \beta_0(2) = 0$, $\beta_1(1) = 0.6, \beta_1(2) = 0.4$, and $r_{01} = r_{10}$ for traffic configuration *II*.

From the numerical results, the effect of the burstiness of the request arrival on the tagged node is not so significant compared with Fig.7 and Fig.8. An explanation for this is the request loss in the counters. The counters count the requests from downstream nodes, and then let empty slot pass to downstream, If the counters overflow, then the tagged node will

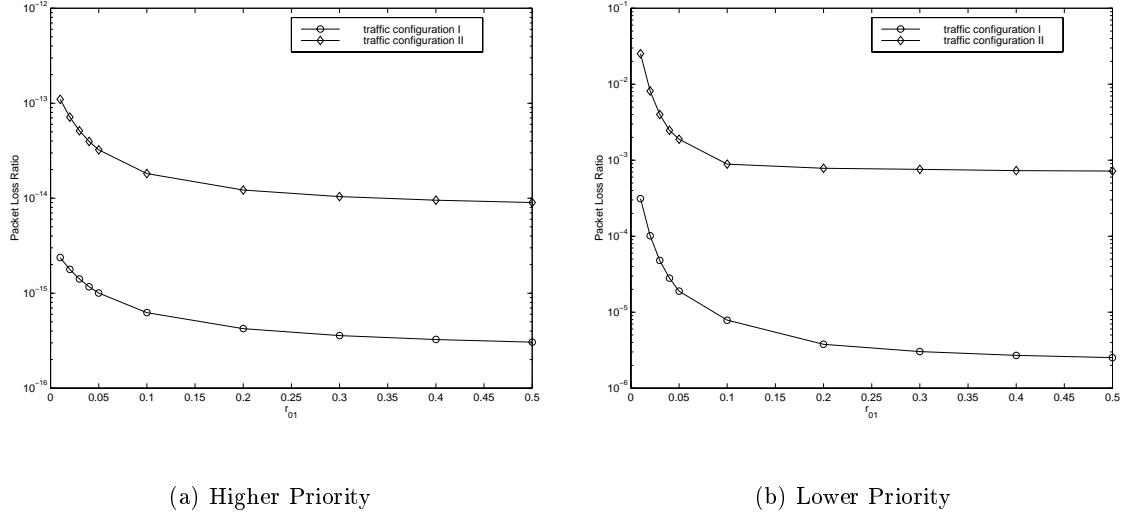


Figure 9: Packet Loss Ratio vs. the Burstiness of the Request Arrival

ignore the requests from downstream.

From [11], the empty-slot pattern of the most upstream nodes have a nearly Bernoulli distribution, and while moving towards downstream the correlation among the state of consecutive slots sharply increases, i.e., the burstiness of MMPP increases. Similarly, it can be deduced that the request arrival observed by the most downstream nodes have a nearly Bernoulli distribution, and while moving towards upstream the correlation among the state of consecutive slots will increase, i.e., the burstiness of MMPP will increase. Because the burstiness of empty-slot pattern have more significant effect on the tagged node than that of the request arrival, it can be concluded that the unfairness still exists in the DQDB MAN even with bandwidth balancing mechanism.

5 Conclusion

In this paper, we studied the performance of the prioritized DQDB MAN under a more accurate and realistic traffic model. The tagged node model is adopted to simplify the analysis. The processes of the packet arrivals to the tagged node, the empty-slot pattern from upstream, and the requests from downstream are assumed to be MMPP. The behavior of the tagged node may be described with a 9-dimensional Markov chain. Instead of constructing the Markov chain by hand, which is a rather tedious and complicated if not impossible mission, we use SPNP, a package developed in Duke University, to construct the Stochastic Reward Net model of the tagged node, generate and solve the underlying 9-dimensional Markov chain automatically. The steady-state and as well as the transient behavior of packet loss ratio of tagged node are investigated. The influence of the burstiness of the empty-slot pattern, packet arrival and request pattern on the tagged node's performance is examined. It is shown that the previous assumptions on the Bernoulli-type empty-slot pattern, request arrival, or packet arrival underestimated the packet loss ratio of the DQDB networks.

References

- [1] ISO/IEC International Standard 88-2-6, "ANSI/IEEE Std.802.6 Distributed Queue Dual Bus(DQDB) Access Method and Physical Layer Specifications", IEEE, 1994.
- [2] E. N. Tsotsios, G. C. Boukis, J. D. Angelopoulos, I. S. Venieris, "Simulation Modelling for Performance Comparison of Cell-based Access Networks", *Simulation*, 70(1), pp.41-56, 1998.

- [3] H. S. Kim, B. C. Shin, "Design of Two Multichannel DQDB Protocols for Single Hop WDM Networks", *IEICE Transactions on Communications*, v E79-B, n.12, Dec. 1996.
- [4] J. H. Lee, J. Y. Lee, S. B. Lee, "Architecture of Multichannel DQDB Network", *Electronics Letters*, 34(1), pp.13-15, 1998.
- [5] D. Pesch, D. Girma, J. A. Dunlop, "Dynamic Channel Assignment in a DQDB MAN Based PCN Access Network", *IEEE Vehicular Technology Conference*, 3,p1674-1678,1997.
- [6] Dong Xuefeng, Lai Tenhwang, "Efficient Protocol for Call Setup and Path Migration in IEEE 802.6 Based Personal Communication Networks", *IEEE Transactions on Computers*, 46(3), pp.326-336, 1997.
- [7] C. C. Bisdikian, "Waiting Time Analysis in a Single Buffer DQDB(802.6) Network", *IEEE J-SAC* 9(8), pp.1562-1573, 1991.
- [8] W. Jing, M. Paterakis, "Message Delay Analysis of the DQDB(802.6) Network", *INFOCOM'92*, pp.527-534, 1992.
- [9] Hairong Sun, Lemin Li, "Sojourn Time Analysis of Prioritised DQDB(IEEE802.6) MAN with Bursty Traffic Input", *Proc. IEE -Commun.*, pp.112-116, 143(2), 1996.
- [10] W. Jing, M. Paterakis, "Extending the Single-node DQDB Analysis model to analyze Network-wide Performance", *Computer Networks and ISDN Systems*, 24, pp.367-385, 1995.
- [11] M. Conti, E. Gregori, L. Lenzini, M. F. Neuts, "M/G/1 Type Approach to the Approximation of the Slot -Occupancy Pattern in an DQDB Network", *Performance Evaluation*, 21(1), pp.59-80, 1994.

- [12] M. Conti, E. Gregori, L. Lenzini, "Influence of the BWB Mechanism on Some Performance Figures of a DQDB Subnetwork", *Computer Networks and ISDN Systems*, 27(7), 1137-1161, 1995.
- [13] Hairong Sun, Ke Huang, Lemin Li, "Performance Analysis of the Constant-Bit-Rate Services in an ATM DQDB MAN", *Computer Communications*, p186-194, Mar. 1998.
- [14] M. Ajmone Marsan, G. Balbo, and G. Conte, "A class of Generalized Stochastic Petri Nets for the performance evaluation of multiprocessor systems," *ACM Transactions on Computer Systems*, vol. 2, pp. 93–122, May 1984.
- [15] G. Ciardo, J. Muppala, K. S. Trivedi, "Analyzing Concurrent and Fault-tolerant Software Using Stochastic Reward Nets", *Journal of Parallel and Distributed Computing*, 15, p255-269, 1992
- [16] G. Ciardo, J. Muppala and K. Trivedi. "SPNP: Stochastic Petri Net Package," *International Conference on Petri Nets and Performance Models*, Kyoto, Japan, December 1989.
- [17] G. Ciardo, K. S. Trivedi, "Manual for SPNP: Stochastic Petri Net Package", version 5.0, CACC, ECE Department, Duke University, 1996.
- [18] W. Fischer, K. Meier-Hellstern, "The Markov-modulated Poisson Process (MMPP) Cookbook", *Performance Evaluation*, 18, p149-171, 1992.