Two priority buffered multistage interconnection networks

Galia Shabtai [*], Israel Cidon and Moshe Sidi
Department of Electrical Engineering, Technion – Israel Institute of Technology, Haifa 32000, Israel

Abstract. This paper presents a novel architecture of internally two priority buffered Multistage Interconnection Network (MIN). First, we compare by simulation the new architecture against a single priority MIN and demonstrate up to N times higher throughput for the high priority traffic in a hot spot situation, when N is the number of inputs. In addition, under uniform traffic assumption we show an increase in the low priority throughput, without any change in the high priority throughput. Moreover, while in the single priority system the high priority delay and its standard deviation are increased when low priority traffic is present, it is kept constant in the dual priority system. Finally, we introduce a new approach of long Markovian memory performance model to better capture the packets dependency in a single priority MIN under uniform traffic and extend this model for a dual priority MIN. Model results are shown to be very accurate.

Keywords: Interconnection networks, multistage networks, multi-priority networks

1. Introduction

In recent years, there has been much interest devoted to incorporating multimedia applications in packet switching networks. Different types of traffic need different QoS standards, but share the same network resources, such as buffers and bandwidth. For real-time applications, it is important that mean delay and delay-jitter are bounded, while for non real-time applications, such as data transfer, the loss ratio often is the restrictive quantity.

A priority service scheme can be defined in terms of a policy determining: (a) which of the arriving packets are admitted to the buffer(s); and/or (b) which of the admitted packets is served next. The former priority service schemes are typically referred to as space priority (or discarding) schemes and attempt to minimize the packet loss of loss-sensitive traffic, such as data. An overview and classification of some space priority strategies can be found in [1,2]. The latter priority service schemes are typically referred to as time priority (or priority scheduling) schemes and attempt to guarantee acceptable delay boundaries to delay-sensitive traffic, such as voice and video. Several types of time priority schemes, such as Weighted-Round-Robin and Weighted-Fair-Queueing, have been proposed and analyzed, each with their own specific algorithmic and computational complexity, see for example [1] and [3] and the references therein.

There are already several commercial switches which accommodate traffic priority schemes, see for example [4,5]. These switches consist of an internally single priority switch fabric and employ two priority queues for each input port. Packets are queued based on their priority level and packets with higher priority number are allowed to pass first. Chen and Guerin [6] studied an $N \times N$ internally one priority non-blocking packet switch with input queues. They assumed that high priority packets preempt low priority ones at the input and move ahead of all low priority packets waiting for service at their input queue. They also assumed that high priority packets always prevail over low priority packets contending for the same output. Given these assumptions and the fact that the switch is non-blocking, they suggested that the presence of low priority packets is transparent to high priority ones, for which the switch behaves as a single priority switch, and studied the performance of low priority

[*] Galia Shabtai is also with Cisco Systems Inc.

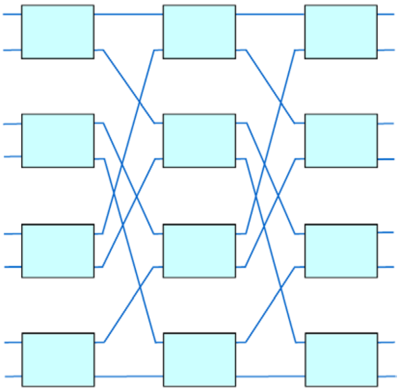

Fig. 1. An 8 × 8 delta–2 network.

packets. They determined the total maximum throughput and established that it can exceed that of an equivalent single priority switch. Ng and Dewar [7] introduced a simple modification to a load sharing replicated banyan networks to guarantee priority traffic transmission. They considered two switch planes, such that one switch plane is designated as the high priority traffic switch plane, and the other is designated as the low priority traffic switch plane. Their simulation results show that when the high priority traffic constitutes less than 30% of the total traffic, one can guarantee extremely low packet loss for the high priority traffic. In addition, when the high priority to low priority traffic ratio increases, the distinction between high and low priority traffic performance decreases. In general, they observed that the high priority traffic delay and packet loss were significantly lower than those of the low priority traffic.

The internal switch structure used in all the above studies is a single priority fabric with controlled inputs. In contrast to these previous works, our paper considers for the first time an internal two priority switch fabric architecture and focuses on the effect of a two priority input buffered Multistage Interconnection Network (MIN) on the performance of high and low priority traffic. We also suggest a new Markovian model for analyzing the performance of the two priority traffic types, assuming uniform traffic, and present numerical results.

A MIN consists of a number of stages of small switching elements (*SE*), which are interconnected by a permutation function. An $(N \times N)$ delta-a network [8] consists of K switching stages of N/a $(a \times a)$ crossbar switches, where $N = a^K$. A packet movement through the network can be controlled locally at each *SE* by a single base-a digit of the packet's destination address. Therefore, no central controller is needed for global routing. Delta networks are subclass of banyan networks which encompass all the useful unique path MINs. An example of an (8 × 8) delta–2 network is given in Fig. 1. The delta network belongs to the blocking type networks. This means that packets may contend for the same output link in an *SE*, which results in a performance loss. One approach to improve the performance of the network can be achieved through the use of buffers in each *SE*. By using buffers, packets, which will be lost otherwise, can be stored in buffers when a conflict occurs. The location of buffers in an *SE* is crucial in the implementation and performance of the network. Dedicated buffers can be used at the inputs or outputs of an *SE*. Alternatively, a shared buffer can also be used in each *SE*. Cisco has built its new CRS-1 router around such a fabric [9,10].

Switches constructed from input buffered *SE*s, which assume uniform traffic, have been widely studied, see for example [11–19]. We elaborate on the approaches of these studies in 3.1. It is important to note that previous works on the subject have considered a single class of packets and so far no study on two priority MIN has been reported (a paper published after the submission of this paper adds priority inside a different switch architecture [20]).

The contribution of this paper is threefold. First, we present a novel architecture of internally two priority buffered MIN and compare by simulation the new architecture against a single priority MIN. Second, we present a novel approach that better captures the cells dependency under uniform traffic, in a one priority single buffered MIN with 2 × 2 *SE*s. Instead of using the common approach of modelling the *SE* buffer with short Markovian memory (the last clock cycle), we propose to extend the Markovian memory to the last two consecutive clock cycles. Third, we analyze both the high and the low priority traffic in a two priority MIN by using the extended Markovian memory approach.

2. Single vs. dual priority MIN

In this section we introduce our novel architecture of internally two priority MIN and compare its performance to a single priority MIN. Our work concentrates on an $(N \times N)$ delta–2 network, i.e., K stages of $N/2$ (2 × 2) crossbar switches, where $N = 2^K$, as illustrated in Fig. 1 for $K = 3$. First, we present the basic single priority *SE* model and review the MIN architecture assumptions. Second, the basic dual priority *SE* model is presented followed by the revised assumptions needed to support two priority MIN. These two models are also used in later sections for the performance analysis model. Third, we outline the simulation environment for both single and dual priority MINs. Finally, two priority traffic simulation results are shown and compared.

2.1. Single priority MIN

Figure 2 shows the basic model of a 2 × 2 single buffered single priority switching element, which mainly consists of two input and two output ports, a single buffer for each incoming link and a non blocking switching matrix to connect the input buffers to the output ports. We assume that a maximum of one packet can be sent from each output port during one clock cycle and therefore a maximum of one packet can be received at each *SE* input link.

As in [11–14] and [17], we consider the network under a synchronous traffic model with a global flow control mechanism, i.e., the following is assumed:

(1) The network clock cycle consists of two phases. In the first phase, flow control information passes through the network from the last stage to the first stage. In the second phase, packets flow from one stage to the next in accordance with the flow control information.
(2) A switch input is able to accept a packet if it has an empty buffer or if the packet in its buffer will leave during the second phase of the current clock cycle.
(3) There is no blocking at the output links of the network.
(4) The arrival process of each input of the network is a simple Bernoulli process, i.e., the probability that a packet arrives within a clock cycle is constant and the arrivals are independent of each other.
(5) The routing logic at each *SE* is fair, i.e., conflicts are randomly resolved.
(6) Packets are of fixed size.

If a uniform traffic model is considered, then the following assumption is added:

(7) Each input link is offered the same traffic load, and the destination addresses of the packets are distributed uniformly over all output links of the network.

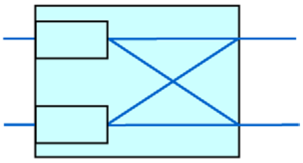

Fig. 2. Basic model of a 2 × 2 single buffered single priority switching element.

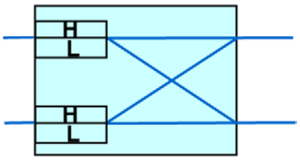

Fig. 3. Basic model of a 2 × 2 single buffered dual priority switching element.

2.2. Dual priority MIN

In the previous section, we assumed that all packets are treated identically, i.e., there is no traffic classification. In this section we extend the model for two traffic classifications: high priority traffic and low priority traffic.

The basic model of a 2×2 single buffered dual priority switching element in shown in Fig. 3. The main difference from the single priority *SE* is that each input buffer is composed of two single *queues*: one for high priority packets and one for low priority packets. The assumption of sending maximum one packet from each *SE* output port during one clock cycle is still valid and therefore each *SE* input link can still receive a maximum of one packet during each clock cycle. On the other hand, we do allow an input buffer to send up to two packets, one high priority and one low priority, during a clock cycle, if each packet is sent to a different output port and the other buffer of the same *SE* does not send any packet during this particular clock cycle.

Following are the revised assumptions for the dual priority model.

(1) The network clock cycle consists of two phases. In the first phase, flow control information passes through the network from the last stage to the first stage. In the second phase, packets flow from one stage to the next in accordance with the flow control information.
(2) A switch input is able to accept a high priority packet if it has an empty high priority queue or if the high priority packet in its high priority queue will leave during the second phase of the current clock cycle.
(3) A switch input is able to accept a low priority packet if it has an empty low priority queue or if the low priority packet in its low priority queue will leave during the second phase of the current clock cycle.
(4) There is no blocking at the output links of the network.
(5) The arrival process of each input of the network is a simple Bernoulli process, i.e., the probability that a packet arrives within a clock cycle is constant and the arrivals are independent of each other. Moreover, there is a fixed probability for each packet to be either high or low priority.
(6) The routing logic within each priority at each *SE* is fair, i.e., same priority conflicts are randomly resolved.
(7) High priority packets have a fixed priority over the low priority packets.
(8) Packets are of fixed size.

If a uniform traffic model is considered, then the following assumption is added:

(9) Each input link is offered the same traffic load and the same high to low priority ratio. In addition, the destination addresses of the packets are distributed uniformly over all output links of the network.

Since the high priority packets have strict priority over the low priority packets, and since we still allow a maximum of one packet into each *SE* input link and out of each *SE* output link, the performance (both throughput and delay) of the high priority traffic in the dual priority MIN is identical to the performance of the single priority traffic in the single priority MIN. Moreover, the low priority traffic is getting served only in those clock cycles in which no high priority traffic is able to move to the desired destination. Therefore, the overall throughput of the dual priority MIN under specific total input load (low priority + high priority) should be at least as high as the single priority MIN throughput under the same total input load and can be even higher.

2.3. System description

As in most contemporary commercial switches, see for example [4,5], we add two input buffers (FIFOs) in front of each MIN input: one is designated for the low priority packets and the other for the high priority packets. Each

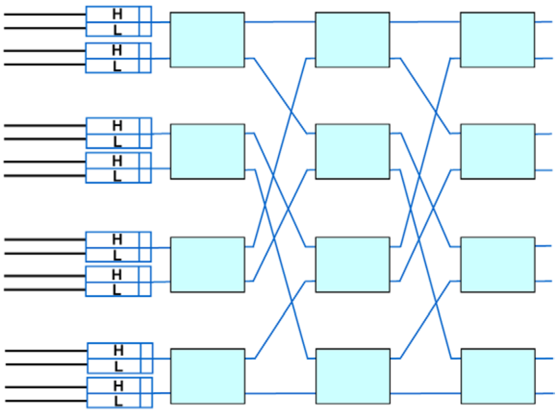

Fig. 4. An 8 × 8 system: delta–2 network with input FIFOs.

low priority packet that arrives to a system input is enqueued to the low priority input FIFO, and each high priority packet that arrives is enqueued to the high priority input FIFO.

In the following we describe single and dual priority systems (the "system" referring to priority aspects of the switch architecture). At times we also refer to single and dual priority traffic (the "traffic" referring to priority aspects of the packets).

An $N \times N$ single priority system comprises of N high priority input FIFOs and N low priority input FIFOs which are connected to an $N \times N$ single priority MIN's inputs, as illustrated in Fig. 4. A high priority packet leaves the high priority input FIFO and enters the MIN input if the corresponding *SE* input is able to accept a packet. On the other hand, a low priority packet can enter the MIN, only if the high priority input FIFO is empty and the corresponding *SE* input is able to accept a packet. This strict priority admission of high priority packets over low priority packets, which is similarly implemented in both [4] and [5], suggests that the throughput of the high priority traffic is not affected by the presence of low priority traffic. In other words, the high priority throughput in the single priority system under dual priority traffic with high priority input load Gh and low priority input load Gl, is equal to the throughput in the single priority system under single priority traffic with input load $G = Gh$, independent of Gl. However, the total delay of the high priority traffic in the single priority system is affected by the presence of low priority traffic, since it increases the congestion probability inside the MIN, and hence increases the delay and its standard deviation.

The dual priority system is obtained by replacing the single priority MIN in the single priority system with a dual priority MIN. In this system, a high priority packet leaves the high priority input FIFO and enters the MIN input if the corresponding *SE* input is able to accept a high priority packet. On the other hand, a low priority packet can enter the MIN, only if there is no high priority packet that can enter and the corresponding *SE* input is able to accept a low priority packet. As in the single priority system, the throughput of the high priority traffic is not affected by the presence of low priority traffic and is equal to the throughput of a single priority traffic in the single priority system under input load that equals to the high priority input load, i.e., $G = Gh$. However, unlike the single priority system, the delay of the high priority traffic in the dual priority system is not affected by the low priority traffic, and hence equals to the delay of a single priority traffic in the single priority system under input load $G = Gh$.

2.4. Simulations results

In order to isolate the input FIFOs size from the system performance, we used infinite input FIFOs in front of each MIN input, so there was actually no packet loss. Nevertheless, it is obvious that a system with low throughput and finite input FIFOs will suffer from higher packet loss than a system that can reach higher throughput with the same input FIFOs size. Therefore, we concentrated on both the delay and the throughput measurements in our simulations.

To emphasize the "immunity" of the high priority traffic over the low priority traffic in the dual priority system vs. the single priority system, we considered an extreme case in which: (a) all inputs send traffic to output link 0, which describes an extreme hot spot situation; (b) all inputs send the same input load; (c) all inputs, except input 0, send low priority traffic, while input 0 sends high priority traffic. While this scenario does not represent a realistic long term steady state, it demonstrates a transient load situation that should be taken into account in the design of contemporary systems. The high priority throughput in both systems is depicted in Fig. 5 for 6 stages networks, with 64 inputs and outputs.

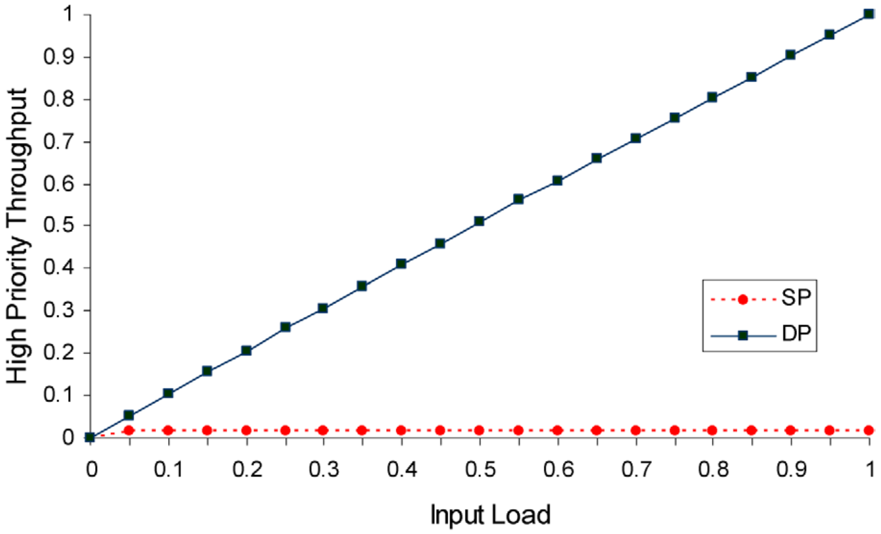

Fig. 5. High priority throughput in both single and dual priority systems with 6 stages under hot spot traffic. SP represents the high priority throughput in the single priority system, while DP represents the high priority throughput in the dual priority system.

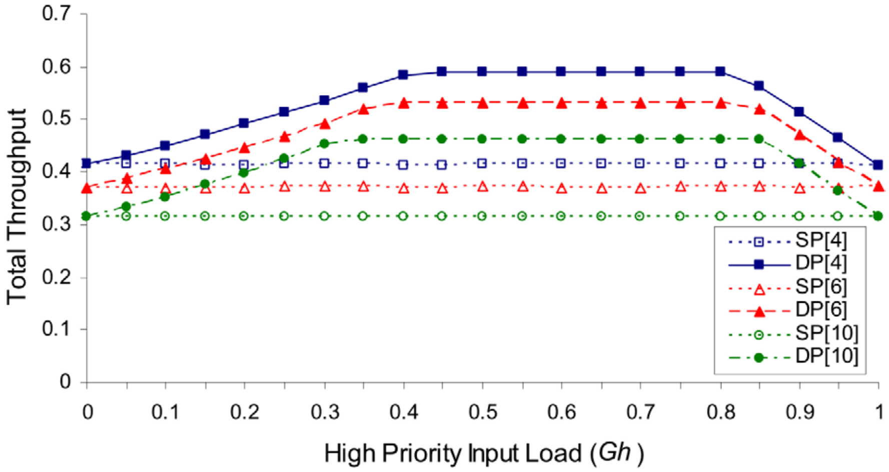

Fig. 6. Maximum total throughput of both single and dual priority systems under dual priority traffic ($G = Gh + Gl = 1$) as a function of the high priority input load (Gh) for various MIN sizes. SP[K] represents a single priority system with K stages MIN. Similarly, DP[K] represents a dual priority system with K stages MIN.

In general, all packets are destined to output 0, which yields throughput of 1 for all inputs together. In the single priority system all packets are treated equally and therefore each input, including input 0 which sends high priority traffic, is able to send throughput of $1/64 = 0.015$. However, in the dual priority system high priority traffic has strict priority over low priority traffic and therefore the high priority throughput equals the high priority input load, while the total low priority throughput equals 1–high priority throughput.

The results in the rest of this section consider a uniform traffic model, as described in Sections 2.1 and 2.2.

The total throughput of both systems under full input load is depicted in Fig. 6 for 6 stages networks, with 64 inputs and outputs. As implied earlier, we can see that the maximum throughput of the dual priority system is higher than that of the single priority system when more than one priority traffic enters the system (up to 47% increase in the 1024 × 1024 system). The source of this extra throughput in the dual priority system is the advance of low priority packets when high priority packets cannot move forward, i.e., this is exactly the low priority throughput difference between the two systems, which is depicted in Fig. 7. We can see that the low priority throughput decreases as long as the low priority input load decreases and high priority throughput increases. The decrease in the low priority throughput stops when high priority throughput arrives to its maximum and is restored when low priority input load further decreases below its throughput value.

Figure 8 shows the low priority throughput in the single priority system under dual priority input load as a function of the high priority input load. We can see that as long as the total input load is below the maximum throughput (∼0.39, as can be seen in Fig. 6), all the low priority input traffic goes out. However, when the total input load goes above the maximum throughput, the low priority throughput is affected, and not all the input low priority load is able to get into (and out of) the MIN. Note that since high priority input load is relatively low in contemporary networks, we considered the range of 0–0.25 for the high priority input load. Since the maximum of the total (high priority and low priority) throughput of the dual priority system is higher than that of the single priority system under dual priority traffic, the low priority throughput in this system starts to be affected when the high priority input load is higher, as can be seen in Fig. 9. As discussed in the previous sub-section, the high priority traffic throughput is identical in both the single and the dual priority systems under the same dual priority input load. Fig. 10 shows the throughput of the high priority traffic in a 64 × 64 single priority system. The graph of the high priority traffic in a 64 × 64 dual priority system is identical.

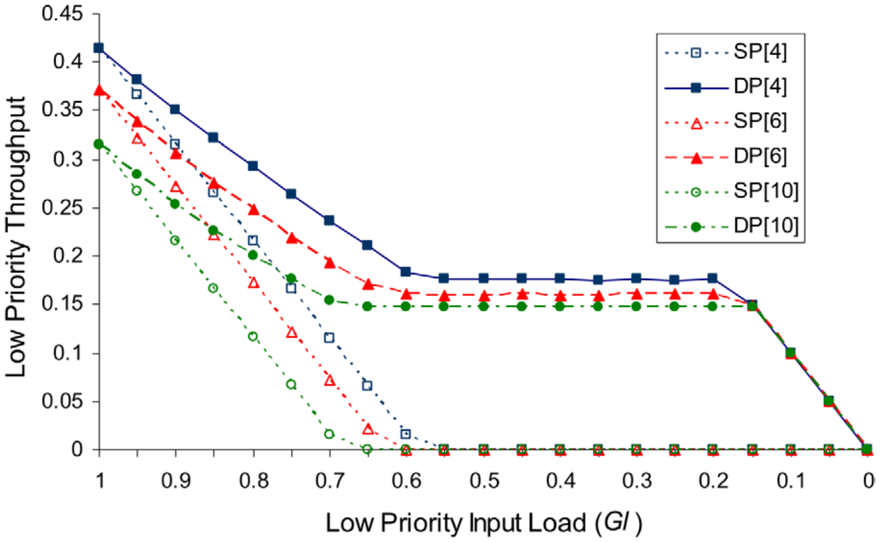

Fig. 7. Low priority throughput of both single and dual priority systems under dual priority traffic ($G = Gh + Gl = 1$) as a function of the low priority input load (Gl) for various MIN sizes. SP[K] represents the low priority throughput in a single priority system with K stages MIN. Similarly, DP[K] represents low priority throughput in a dual priority system with K stages MIN.

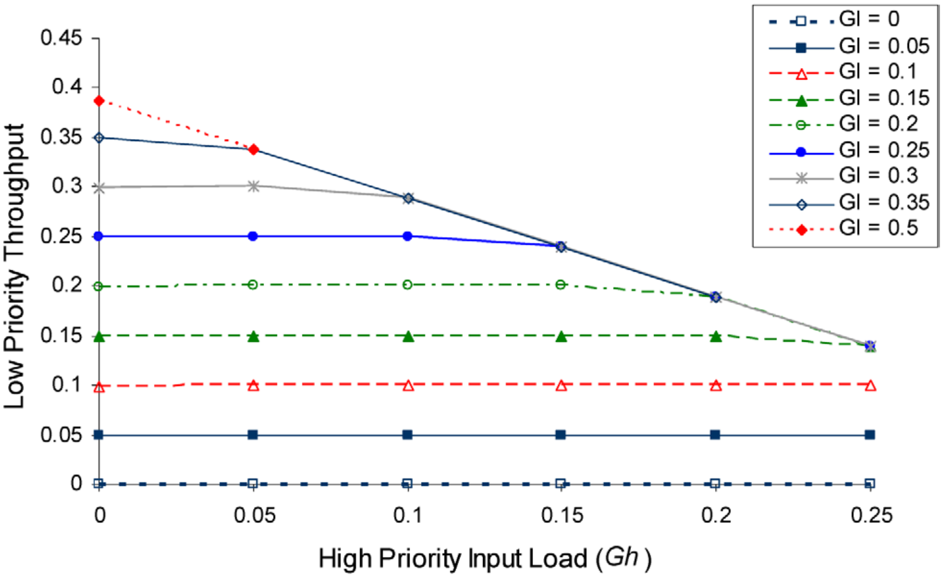

Fig. 8. Low priority throughput in a 64 × 64 single priority system under dual priority traffic. *Gl* represents the low priority input load.

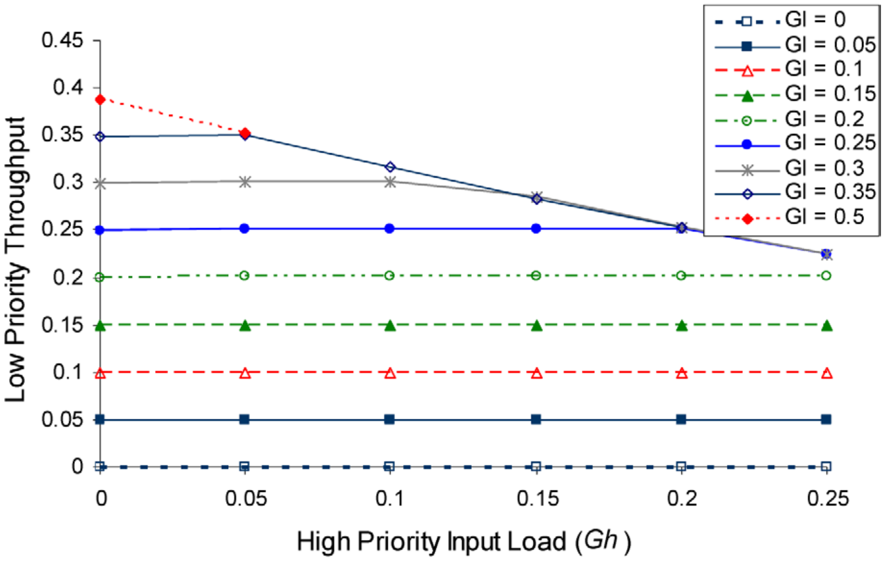

Fig. 9. Low priority throughput in a 64 × 64 dual priority system under dual priority traffic. *Gl* represents the low priority input load.

As opposed to the high priority throughput, the high priority delay is affected by the low priority input load in the single priority system. Figure 11 depicts the average high priority total delay in 64 × 64 single and dual priority systems under dual priority traffic. We can see that in the single priority system the average delay increases with the increase of the low priority input load, but the increase stops when the total input load reaches the maximum throughput of that system. At this point, the low priority load inside the MIN stops increasing and therefore the high priority delay stays constant. As high priority input load increases, the probability that a high priority packet arrives to an empty input FIFO decreases and therefore, the total delay increases. The high priority delay in the dual priority system is not affected by the low priority input load and remains constant. The behavior of the standard

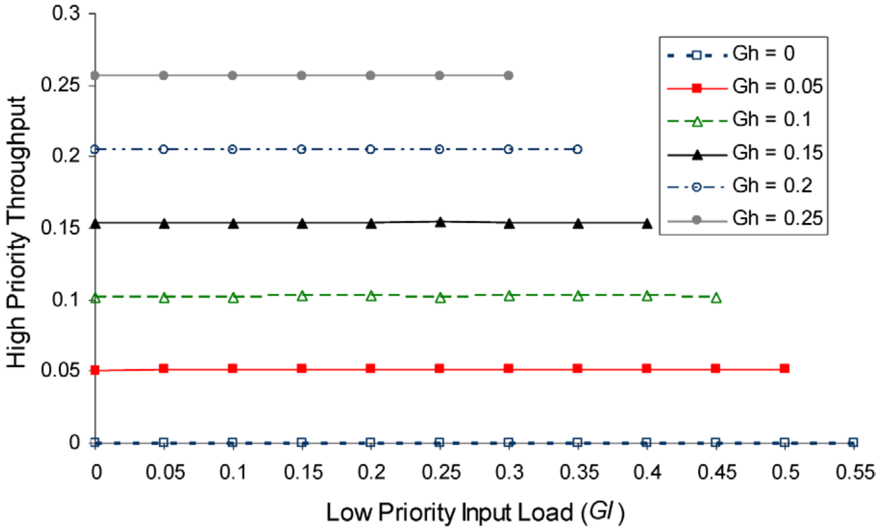

Fig. 10. High priority throughput in a 64 × 64 single priority system under dual priority traffic. Gh represents the high priority input load.

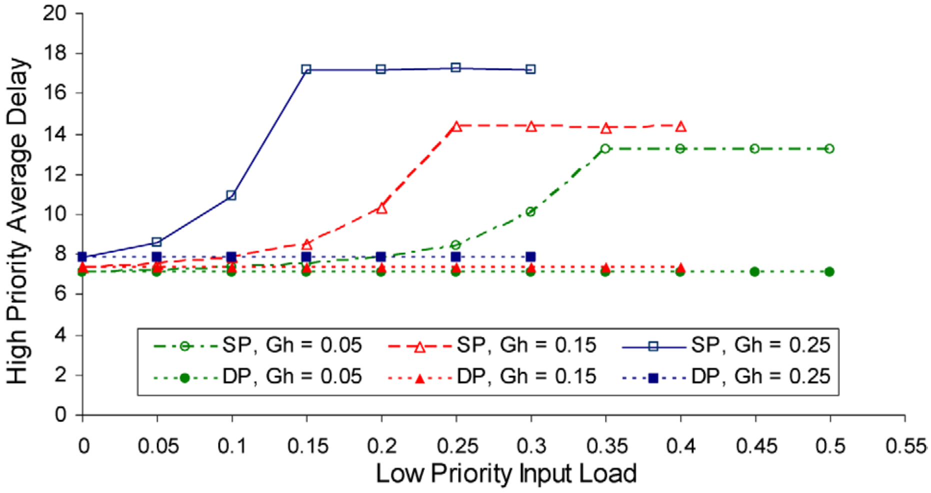

Fig. 11. Average high priority total delay in 64 × 64 single priority (SP) and dual priority (DP) systems under dual priority traffic. Gh represents the high priority input load.

deviation of the high priority delay is similar. Note that the single priority system we used (and the one that is commonly used) has a single buffer vs. two buffers (one for each priority) in the dual priority system. The effect of increasing the number of buffers in a single priority system is explored in [12,13,17].

3. Performance model

As stated before, the performance model is focused on uniform traffic. Previous work for modeling and analyzing MINs under uniform traffic [11–19] used short Markovian memory (the last clock cycle). We propose to extend the Markovian memory to the last two consecutive clock cycles to better capture the packets dependency. First, we briefly review previous models for analyzing MINs. Second, we introduce our novel approach to the single priority model, present the analysis and some numeric results. Finally, we introduce the dual priority model, its analysis and numeric results.

3.1. Previous models for analyzing MINs

To better understand the novelty and significance of the approach presented in this paper, we first review three of the more classical models used for analyzing MINs under uniform traffic, [11] and [16,17].

Jenq [11] was the first to suggest two models for analyzing of single buffered banyan network composed of 2×2 *SE*s. In both models he assumed that packets arriving at each network input link are destined uniformly for all network output links. He also assumed a uniform load for each network input link. In the first model he further assumed that the two buffers in the same *SE* are statistically independent and therefore the state of a stage can be reduced to that of a single buffer, i.e., two states ("empty" and not empty) Markovian model. To verify this assumption he introduced the second model, in which the state of a stage is characterized by that of an *SE*. This model assumes that the two *SE* buffers are dependent and therefore the Markovian model comprises four states. Since both models showed very close results, it was concluded that the assumption of independence between the two *SE* buffers is reasonable. Jenq's models have rather low accuracy when the input load is high, mainly due to the independence assumption between requests in consecutive clock cycles and between states of buffers in adjacent stages.

In a subsequent work, Theimer, Rathgeb and Huber [16] modeled a single buffered banyan networks with 2×2 *SE*s. They assumed that the two *SE* buffers are dependent and added a blocked state for the single buffer. Therefore, their model, which tries to model the *SE*, includes nine states: three states ("empty", "new" and "blocked") for each *SE* buffer. This nine state model captures major part of the correlations of a packet movement between two consecutive clock cycles as well as the states of the buffers in two adjacent network stages. This model demonstrates a significant improvement in accuracy over Jenq model. However, since they derived their model by exhaustively tracing the possible states of input buffers in each *SE*, the generalization of their model to the case of $a \times a$ *SE*s is very difficult.

Later, Mun and Youn [17], developed a model for a multi-buffered MINs with 2×2 *SE*s. They assumed that the two multi-buffers in the same *SE* are statistically independent. They first developed a single buffered model with 2×2 *SE*s, which includes three states: "empty", "new" and "blocked". They later expanded this model to support multi-buffered 2×2 *SE*s. The results of the single buffered model are closer to those of Theimer's model with much less complexity.

There are additional models, such as [15] and [18], that are based on three states model, but the assumptions in [15] do not seem to be realistic and the results of [18] for low loads are very inaccurate.

3.2. Single priority model

In this section we introduce our novel approach, present the analysis and some numeric results.

(1) Model and notations. The basic model of the single priority *SE* and its assumptions are presented in Section 2.1 "Single priority MIN". The analytic model is based on this model and its assumptions, including the uniform traffic assumption.

Assumption 7 in Section 2.1 implies that loads are balanced in the whole switching network and therefore the state of an *SE* at stage k is statistically indistinguishable from that of another *SE* of the same stage. Following Jenq's first model [11], we further assume that the two buffers in the same *SE* are statistically independent and therefore the state of a stage can be reduced to that of a single buffer.

Initial work modelled the single buffer as a two states Markov chain with the following states: "0", buffer empty and "1", buffer not empty (see [11–14]). In order to capture the correlations between consecutive clock cycles as well as between the states of the buffers in the adjacent stages, later work split the "buffer not empty" state into two states: "new", buffer contains a new packet and "blocked", buffer contains a blocked packet (see [15–18]). In this section we introduce a novel model for the single buffer behavior, which considers also the previous state of that buffer. This one clock history consideration increases the dependency capturing by refining the "empty" and "new" states of the three state models. Note that when a network is congested, the *SE*s arrival rate is relatively low. Therefore, in addition to the blocked buffers, there are unblocked buffers which their time dependency and correlation with the next stage should also be captured. Following are the five possible states that we have in our model:

- "00": buffer was empty at the beginning of the previous clock cycle and is empty at the beginning of the current clock cycle as well, i.e., no new packet has been received during the previous clock cycle.
- "01": buffer was empty at the beginning of the previous clock cycle and contains a new packet at the beginning of the current clock cycle, i.e., a new packet has been received during the previous clock cycle.
- "10": buffer had a packet at the beginning of the previous clock cycle but has no packet at the beginning of the current one, i.e., a packet has been sent from this buffer during the previous clock cycle, but no new packet has been received.
- "$11n$": buffer had a packet at the beginning of the previous clock cycle and has a new one at the beginning of the current clock cycle, i.e., a packet has been sent from this buffer during the previous clock cycle, and a new packet has been received.
- "$11b$": buffer had a packet at the beginning of the previous clock cycle and has a blocked one at the beginning of the current clock cycle, i.e., no packet has been sent from this buffer during the previous clock cycle.

The probability of each *SE* buffer in a certain stage to be in each of the above five states are presented below. Following Mun's [17] notations, *SE*(k) denotes an *SE* at stage k. Also, t_b represents the time instance when a clock cycle begins, while t_d represents the duration of a clock cycle.

$P_{00}(k,t)$: Probability that a buffer of *SE*(k) is empty at $(t-1)_b$ and at t_b.

$P_{01}(k,t)$: Probability that a buffer of *SE*(k) is empty at $(t-1)_b$ and has a new packet at t_b.

$P_{10}(k,t)$: Probability that a buffer of *SE*(k) has a packet at $(t-1)_b$ and is empty at t_b.

$P_{11}(k,t)$: Probability that a buffer of *SE*(k) has a packet at $(t-1)_b$ and has a new one at t_b.

$P_{11}(k,t)$: Probability that a buffer of *SE*(k) has a packet at $(t-1)_b$ and has a blocked one at t_b.

The state transition probabilities are shown in Fig. 12 and the notations, which will be used in the sequel, are summarized below.

$q(k,t)$: Probability that a packet is ready to come to a buffer of *SE*(k) at t_d.

$r_{01}(k,t)$: Probability that a packet in a buffer of *SE*(k) is able to move forward at t_d, given that the buffer is in state "01".

$r_{11n}(k,t)$: Probability that a packet in a buffer of *SE*(k) is able to move forward at t_d given that the buffer is in state "$11n$".

$r_{11b}(k,t)$: Probability that a packet in a buffer of *SE*(k) is able to move forward at t_d given that the buffer is in state "$11b$".

(2) Analysis. The evolution of the state probabilities as the clock cycles advance are easily derived from Fig. 12:

$$P_{00}(k,t+1) = [1-q(k,t)][P_{00}(k,t)+P_{10}(k,t)], \tag{1}$$

$$P_{01}(k,t+1) = q(k,t)[P_{00}(k,t)+P_{10}(k,t)], \tag{2}$$

$$P_{10}(k,t+1) = [1-q(k,t)][r_{01}(k,t)P_{01}(k,t)+r_{11n}(k,t)P_{11n}(k,t)+r_{11b}(k,t)P_{11b}(k,t)], \tag{3}$$

$$P_{11n}(k,t+1) = q(k,t)[r_{01}(k,t)P_{01}(k,t)+r_{11n}(k,t)P_{11n}(k,t)+r_{11b}(k,t)P_{11b}(k,t)], \tag{4}$$

$$P_{11b}(k,t+1) = [1-r_{01}(k,t)]P_{01}(k,t)+[1-r_{11n}(k,t)]P_{11n}(k,t)+[1-r_{11b}(k,t)]P_{11b}(k,t). \tag{5}$$

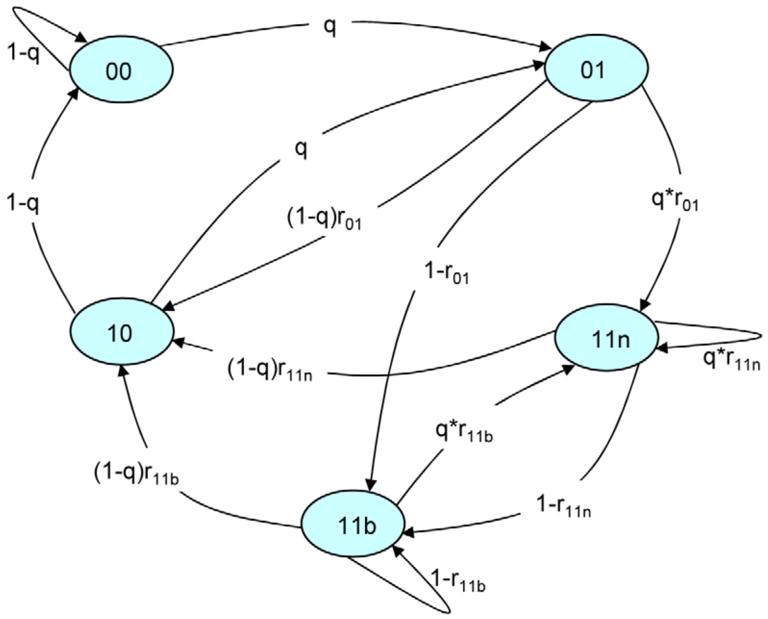

Fig. 12. The state transition diagram of a single priority *SE*(k) buffer.

If the transition probabilities $q(k,t)$, $r_{01}(k,t)$, $r_{11n}(k,t)$ and $r_{11b}(k,t)$ are known for every k ($1 \leqslant k \leqslant K$) and for every $t \geqslant 0$, then the state probabilities of the buffers can be computed iteratively (assuming the probabilities are known for $t = 0$). The derivation of the transition probabilities is explicitly described in Appendix 1 (see Eqs. (14)–(23)).

The throughput of stage $k, S(k,t)$, is the probability that a packet is transmitted from an output port of *SE*(k) at t_d. In other words, it is the probability that a buffer of *SE*($k+1$) receives a packet at t_d and it can be calculated from the state probabilities of stage k and from the transition probabilities as follows:

$$S(k,t) = P_{01}(k,t)r_{01}(k,t) + P_{11n}(k,t)r_{11n}(k,t) + P_{11b}(k,t)r_{11b}(k,t). \tag{6}$$

Finally, the normalized throughput, $S(t)$, is the per output port throughput at t_d of the last switching stage (K), i.e., $S(t) = S(K,t)$, where $K = \log_2 N$ is the number of switching stages.

Now that all quantities have been defined and quantified through explicit equations, we provide in Table 1 the exact procedure for their iterative computation.

(3) Numerical results. Figure 13 shows the normalized throughput of a single buffered single priority delta–2 network with 6 stages as a function of the input load for the three classical models and for our model. All models are very accurate at low loads and accuracy reduces as input load increases. When input load approaches the network maximum throughput, the accuracy of Jenq's model is insufficient. One of the reasons is the fact that many packets are blocked mainly at the network first stages at high traffic rates. Therefore, adding a "blocked" state in Mun's model improved the accuracy. The consideration of the dependencies between the two buffers of an *SE* in Theimer's model leads to further improvement. Taking into account the history of an additional clock cycle in our model, introduces almost the same improvement with much less complication. The influence of network size on the performance is depicted in Fig. 14, which shows the normalized throughput of a single buffered single priority delta–2 network for various network sizes. It can be seen that the model accuracy decreases as network size increases. This is due to the fact that every additional stage introduces further collisions and the inaccuracy of one stage accumulates to the previous stage. Our model seems to be very accurate: the maximum deviation is only 10.9% for a fully loaded 1024 × 1024 network.

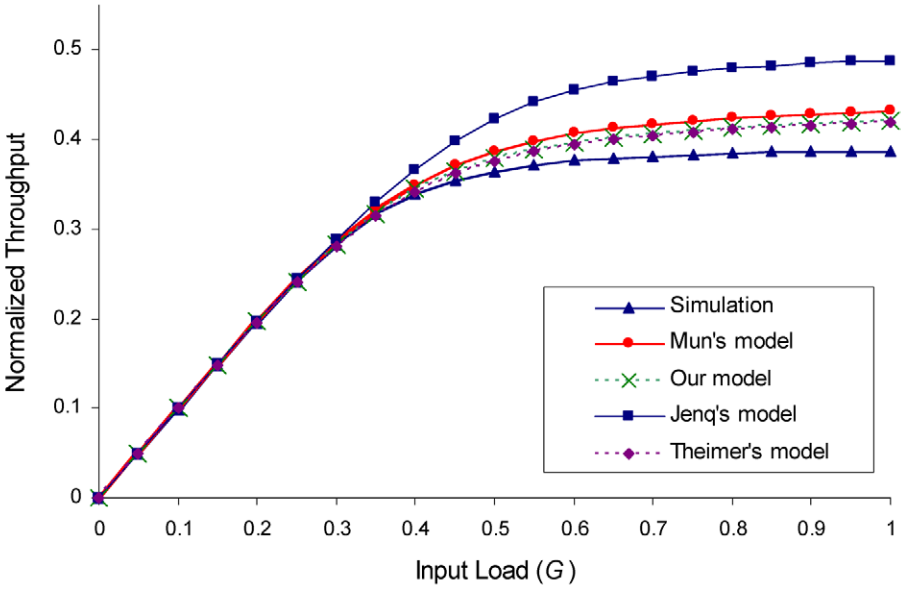

Fig. 13. Normalized throughput of a single buffered single priority delta–2 network with 6 stages.

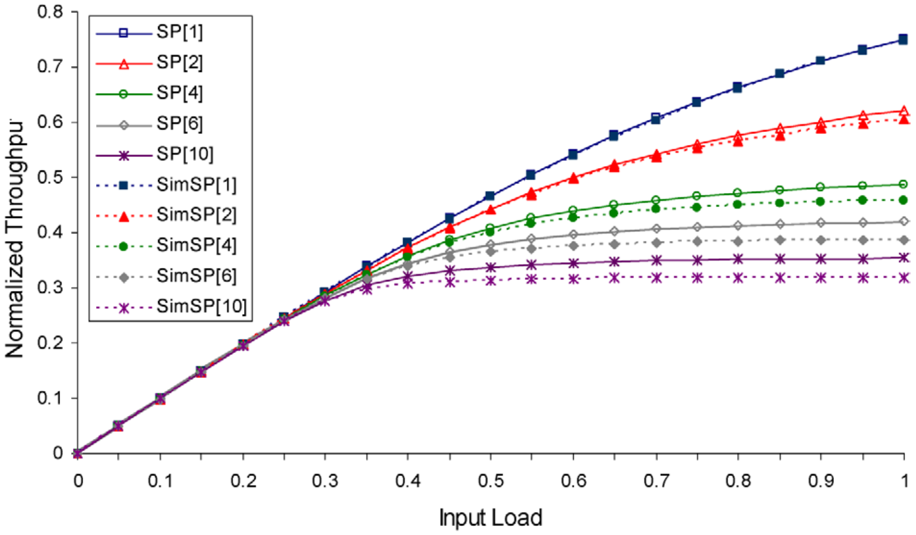

Fig. 14. Normalized throughput of a single buffered single priority delta–2 network for various network sizes. $S[K]$ is the analyzed throughput for a network with K stages. Sim $S[K]$ is the simulated throughput for a network with K stages.

3.3. Dual priority model

In this section we introduce the dual priority model, its analysis and some numerical results.

(1) Model and notations. The basic model of the dual priority *SE* and its assumptions are presented in Section 2 “Dual priority MIN”. The analytic model is based on this model and its assumptions, including the uniform traffic assumption.

Table 1

Iterative computations for the single priority system (Eqs (14)–(23) appear in Appendix 1).

```
ITERATIVE PROCEDURE

INPUT
    $G$ – the probability that a packet arrives within a clock cycle to each input of the network
    $N$ – number of inputs (and outputs)
    $K = \log_2 N$ – number of switching stages

INITIALIZE ($t = 0$)
    For every $k$ ($1 \leqslant k \leqslant K$) choose[1] the probabilities $P_{00}(k,0)$, $P_{01}(k,0)$, $P_{10}(k,0)$
    $P_{11n}(k,0)$, $P_{11b}(k,0)$ so they sum to 1

ITERATE
    For ( $t = 0$ ; $t{+}{+}$ (until convergence[2]) )
    {
        Compute $r_{01}(K,t)$, $r_{11n}(K,t)$ and $r_{11b}(K,t)$ (Eqs (19)–(21))
        Compute $P_0^a(K,t)$ and $r_n(K,t)$ (Eqs (14)–(15))
        Compute $S(K,t)$ (Eq. (6))
        For ( $k = K - 1$; $k \geqslant 1$; $k{-}{-}$ )
            {
                Compute $r_{01}(k,t)$, $r_{11n}(k,t)$ and $r_{11b}(k,t)$ (Eqs (16)–(18))
                Compute $P_0^a(k,t)$ and $r_n(k,t)$ (Eqs (14)–(15))
                Compute $S(k,t)$ (Eq. (6))
            }
        Set $q(1,t) = G$ (Eq. (23))
        For ( $k = 2$; $k \leqslant K$; $k{+}{+}$ )
            Compute $q(k,t)$ (Eq. (22))
        For ( $k = 1$; $k \leqslant K$; $k{+}{+}$ )
            Compute $P_{00}(k,t+1)$, $P_{01}(k,t+1)$, $P_{10}(k,t+1)$, $P_{11n}(k,t+1)$
            and $P_{11b}(k,t+1)$ (Eqs (1)–(5))
    }

OUTPUT
        Normalized throughput $S(K,T)$ where $T$ is the time the iteration stopped
```

[1] For small values of G we recommend to choose for every k ($1 \leqslant k \leqslant K$) $P_{00}(k,0) = 1$ and $P_{01}(k,0) = P_{10}(k,0) = P_{11n}(k,0) = P_{11b}(k,0) = 0$. If one does computations for several values of G, then it is recommended to initialize with the converged values of the probabilities of the previous value of G.

[2] To check convergence we sum the absolute value of the differences between the respective state probabilities in two consecutive cycles. If this sum is smaller than a predefined small quantity, then the iteration is stopped. Note that we do not have a proof that this procedure always converges, but in all our numerical computations, the procedure converged.

As discussed previously, the performance (both throughput and delay) of the high priority traffic in the dual priority MIN is identical to the performance of the single priority traffic in the single priority MIN. Moreover, the low priority traffic is getting served only in those clock cycles in which no high priority traffic is able to move to the desired destination. Therefore, our model includes two separate Markov chains. The first one is a stand alone chain, which represents the high priority traffic queue and is identical to the single priority model, presented in the previous section. However, since the service of the low priority traffic depends on the high priority service, the transitions of the second chain, which represents the low priority traffic queue, depends on the transitions of the first chain. Note that since the dual priority *SE* allows an input buffer to send up to two packets, one high priority and one low priority, during a clock cycle, under the constraint specified in Section 2, the low priority queue and the high priority queue modelled do not necessarily relate to the same buffer, but they do relate to the same *SE*.

The low priority model is an extended version of the single priority model and includes six states. The states "00", "01", "10" and "$11n$" are identical to the states of the single priority model, while state "$11b$" is split into two states as follows.

- "$11hb$": queue had a low priority packet at the beginning of the previous clock cycle and this packet has been blocked by a high priority packet and stayed at least till the beginning of the current clock cycle.
- "$11lb$": queue had a low priority packet at the beginning of the previous clock cycle and this packet has been blocked by a low priority packet and stayed at least till the beginning of the current clock cycle.

The motivation for the blocked state split is the two different possible sources of low priority blocking and their major affect on the inferred destination buffer state and its acceptance probability. A low priority packet, which is blocked by a high priority packet, implies a lower probability of occupancy in the low priority destination queue. On the other hand, a low priority packet, which is blocked by a low priority packet, implies a higher probability of occupancy in the low priority destination queue.

Since the high priority model is identical to the single priority model, we can avoid rewriting all the previous section with an "h" notation simply by saying that each variable with no "l" notation represents the high priority traffic, and each low priority parameter is represented by the "l" notation.

The probability of each low priority *SE* buffer queue in a certain stage to be in each of the six states are presented below.

$Pl_{00}(k,t)$: Probability that a low priority queue of an *SE*(k) buffer is empty at $(t-1)_b$ and at t_b.

$Pl_{01}(k,t)$: Probability that a low priority queue of an *SE*(k) buffer is empty at $(t-1)_b$ and has a new packet at t_b.

$Pl_{10}(k,t)$: Probability that a low priority queue of an *SE*(k) buffer has a packet at $(t-1)_b$ and is empty at t_b.

$Pl_{11n}(k,t)$: Probability that a low priority queue of an *SE*(k) buffer has a packet at $(t-1)_b$ and has a new one at t_b.

$Pl_{11hb}(k,t)$: Probability that a low priority queue of an *SE*(k) buffer has a packet at $(t-1)_b$ and has a blocked one, which is blocked by a high priority packet, at t_b.

$Pl_{11lb}(k,t)$: Probability that a low priority queue of an *SE*(k) buffer has a packet at $(t-1)_b$ and has a blocked one, which is blocked by a low priority packet, at t_b.

The state transition diagram for the low priority traffic queue is shown in Fig. 15 and the notations, which will be used in the sequel, are summarized below.

$ql(k,t)$: Probability that a low priority packet is ready to come to a low priority queue of an *SE*(k) buffer at t_d.

$rlt_{01}(k,t)$: Probability that a packet in the low priority queue of an *SE*(k) buffer is able to move forward at t_d, given that the buffer is state "01".

$rlt_{11n}(k,t)$: Probability that a packet in the low priority queue of an *SE*(k) buffer is able to move forward at t_d, given that the buffer is in state "$11n$".

$rlt_{11hb}(k,t)$: Probability that a packet in the low priority queue of an *SE*(k) buffer is able to move forward at t_d, given that the buffer is in state "$11hb$".

$rlt_{11lb}(k,t)$: Probability that a packet in the low priority queue of an *SE*(k) buffer is able to move forward at t_d, given that the buffer is in state "$11lb$".

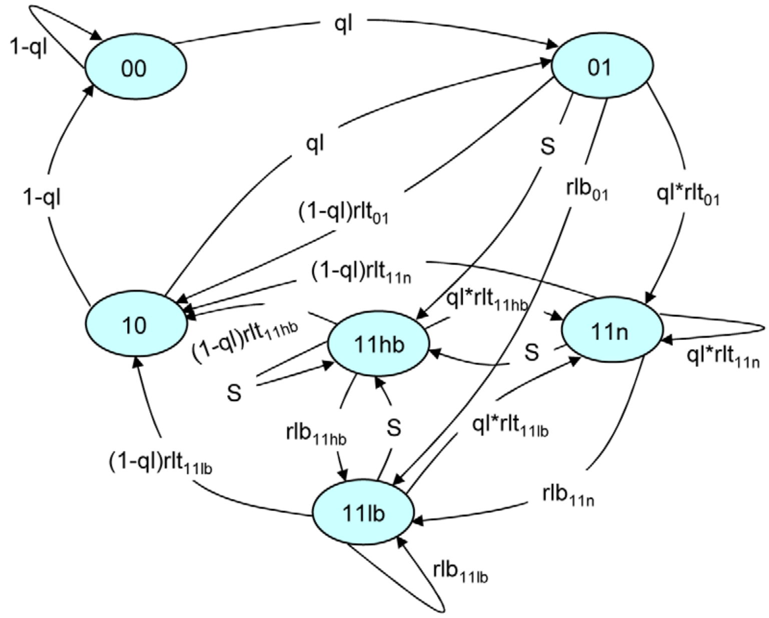

Fig. 15. The state transition diagram of a low priority queue in an *SE*(k) buffer.

$rlb_{01}(k,t)$: Probability that a packet in the low priority queue of an *SE*(k) buffer is not able to move forward at t_d due to a low priority traffic, given that the buffer is in state “01”.

$rlb_{11n}(k,t)$: Probability that a packet in the low priority queue of an *SE*(k) buffer is not able to move forward at t_d due to a low priority traffic, given that the buffer is in state “$11n$”.

$rlb_{11hb}(k,t)$: Probability that a packet in the low priority queue of an *SE*(k) buffer at stage k is not able to move forward at t_d due to a low priority traffic, given that the buffer is in state “$11hb$”.

$rlb_{11lb}(k,t)$: Probability that a packet in the low priority forward at t_d due to a low priority traffic, given that the buffer is in state “$11lb$”.
98

(2) Analysis. The state probabilities of the high priority traffic at clock cycle $t+1$ are easily derived from Fig. 12, and are identical to those calculated in Eqs (1)–(5). The state probabilities of the low priority traffic at clock cycle $t+1$ are easily derived from Fig. 15 (recall that $S(k,t)$ is the high priority throughput):

$$Pl_{00}(k,t+1) = [1 - ql(k,t)][Pl_{00}(k,t) + Pl_{10}(k,t)], \tag{7}$$

$$Pl_{01}(k,t+1) = ql(k,t)[Pl_{00}(k,t) + Pl_{10}(k,t)], \tag{8}$$

$$\begin{aligned} Pl_{10}(k,t+1) = {} & [1 - ql(k,t)][rlt_{01}(k,t)Pl_{01}(k,t) + rlt_{11n}(k,t)Pl_{11n}(k,t) \\ & + rlt_{11lb}(k,t)Pl_{11lb}(k,t) + rlt_{11hb}(k,t)Pl_{11hb}(k,t)], \end{aligned} \tag{9}$$

$$\begin{aligned} Pl_{11n}(k,t+1) = {} & ql(k,t)[rlt_{01}(k,t)Pl_{01}(k,t) + rlt_{11n}(k,t)Pl_{11n}(k,t) \\ & + rlt_{11lb}(k,t)Pl_{11lb}(k,t) + rlt_{11hb}(k,t)Pl_{11hb}(k,t)], \end{aligned} \tag{10}$$

$$\begin{aligned} Pl_{11lb}(k,t+1) = {} & rlb_{01}(k,t)Pl_{01}(k,t) + rlb_{11n}(k,t)Pl_{11n}(k,t) \\ & + rlb_{11lb}(k,t)Pl_{11lb}(k,t) + rlb_{11hb}(k,t)Pl11_{hb}(k,t), \end{aligned} \tag{11}$$

$$Pl_{11hb}(k,t+1) = S(k,t)[Pl_{01}(k,t) + Pl_{11n}(k,t) + Pl_{11lb}(k,t) + Pl_{11hb}(k,t)]. \tag{12}$$

Table 2

Iterative computations for the dual priority system (Eqs (24)–(45) appear in Appendix 2).

```
ITERATIVE PROCEDURE

INPUT
    $Gl$ – the probability that a low priority packet arrives within a clock cycle to each input
        of the network
    $Gh$ – the probability that a high priority packet arrives within a clock cycle to each input
        of the network
    $N$ – number of inputs (and outputs)
    $K = \log_2 N$ – number of switching stages

INITIALIZE ($t = 0$)
    For every $k$ ($1 \leqslant k \leqslant K$) choose[1] the probabilities $Pl_{00}(k,0)$, $Pl_{01}(k,0)$, $Pl_{10}(k,0)$
    $Pl_{11n}(k,0)$, $Pl_{11lb}(k,0)$, $Pl_{11hb}(k,0)$ so they sum to 1

ITERATE
    For ( $t = 0$ ; $t{+}{+}$ (until convergence[2]) )
    {
        Compute $rl_{01}(K,t)$, $rl_{11n}(K,t)$, $rl_{11lb}(K,t)$ and $rl_{11hb}(K,t)$ (Eqs (39)–(42))
        Compute $rlt_{01}(K,t)$, $rlt_{11n}(K,t)$, $rlt_{11lb}(K,t)$, $rlt_{11hb}(K,t)$, $rlb_{01}(K,t)$,
            $rlb_{11n}(K,t)$, $rlb_{11lb}(K,t)$ and $rlb_{11hb}(K,t)$ (Eqs (24)–(31))[3]
        Compute $Pl_0^a(K,t)$, $rl_n(K,t)$ and $rl_b(K,t)$ (Eqs (32)–(34))
        Compute $Sl(K,t)$ (Eq. (13))
        For ( $k = K - 1$; $k \geqslant 1$; $k{-}{-}$ )
            {
                Compute $rl_{01}(k,t)$, $rl_{11n}(k,t)$, $rl_{11lb}(k,t)$ and $rl_{11hb}(k,t)$ (Eqs (35)–(38))
                Compute $rlt_{01}(k,t)$, $rlt_{11n}(k,t)$, $rlt_{11lb}(k,t)$, $rlt_{11hb}(k,t)$, $rlb_{01}(k,t)$,
                    $rlb_{11n}(k,t)$, $rlb_{11lb}(k,t)$ and $rlb_{11hb}(k,t)$ (Eqs (24)–(31))[3]
                Compute $Pl_0^a(k,t)$, $rl_n(k,t)$ and $rl_b(k,t)$ (Eqs (32)–(34))
                Compute $Sl(k,t)$ (Eq. (13))
            }
        Set $ql(1,t) = Gl$ (Eq. (45))
        For ( $k = 2$; $k \leqslant K$; $k{+}{+}$ )
            Compute $ql(k,t)$ (Eq. (43))
        For ( $k = 1$; $k \leqslant K$; $k{+}{+}$ )
            Compute $Pl_{00}(k,t+1)$, $Pl_{01}(k,t+1)$, $Pl_{10}(k,t+1)$, $Pl_{11n}(k,t+1)$,
            $Pl_{11lb}(k,t+1)$ and $Pl_{11hb}(k,t+1)$ (Eqs (7)–(12))[3]
    }

OUTPUT
        Normalized throughput $Sl(K,T)$ where $T$ is the time the iteration stopped
```

[1] For small values of G we recommend to choose for every k ($1 \leqslant k \leqslant K$) $Pl_{00}(k,0) = 1$ and $Pl_{01}(k,0) = Pl_{10}(k,0) = Pl_{11n}(k,0) = Pl_{11lb}(k,0) = Pl_{11hb}(k,0) = 0$. If one does computations for several values of Gl, then it is recommended to initialize with the converged values of the probabilities of the previous value of Gl.

[2] Convergence is checked in the same way as the procedure described in Table 1.

[3] $S(k,t)$ in these equations is the throughput of the high priority traffic at time t at stage k.

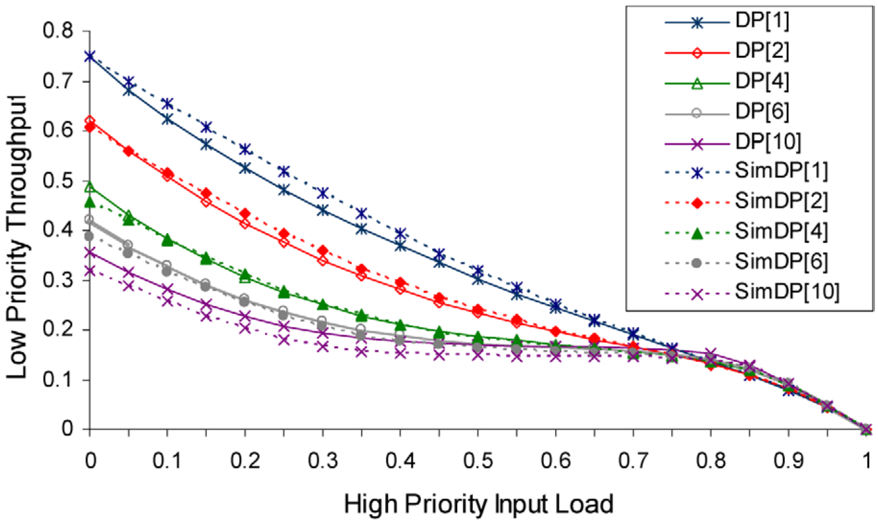

Fig. 16. Low priority normalized throughput of a single buffered dual priority delta–2 network for various network sizes as a function of the high priority input load, G. $Sl[K]$ is the analyzed low priority throughput for a network with K stages. Sim $Sl[K]$ is the simulated low priority throughput for a network with K stages. Low priority input load equals to 1–high priority input load.

As in the single priority model, if the transition probabilities $ql(k,t)$, $rlt_{01}(k,t)$, $rlt_{11n}(k,t)$, $rlt_{11hb}(k,t)$, $rlt_{11lb}(k,t)$, $rlb_{01}(k,t)$, $rlb_{11hn}(k,t)$, $rlb_{11hb}(k,t)$ and $rlb_{11kb}(k,t)$ are known for every k ($1 \leqslant k \leqslant K$) and for every $t \geqslant 0$, then the state probabilities of the buffers can be computed iteratively (assuming the probabilities are know for $t = 0$). The derivation of the transition probabilities is explicitly described in Appendix 2 (see Eqs (24)–(45)).

The low priority throughput of stage k, $Sl(k,t)$, is the probability that a low priority packet is transmitted from an output port of $SE(k)$ at t_d. In other words, it is the probability that a low priority queue of $SE(k+1)$ buffer receives a packet at t_d and it can be calculated from the state probabilities of stage k and from the transition probabilities, as follows,

$$\begin{aligned} Sl(k,t) = {} & Pl_{01}(k,t)rlt_{01}(k,t) + Pl11n(k,t)rlt_{11n}(k,t) \\ & + Pl_{11lb}(k,t)\cdot rlt_{11lb}(k,t) + Pl_{11hb}(k,t)rlt_{11hb}(k,t). \end{aligned} \tag{13}$$

The low priority normalized throughput, $Sl(t)$, which is the per output port throughput at t_d of the last stage, i.e., $Sl(t) = Sl(K,t)$, where $K = \log_2 N$ is the number of switching stages.

Now that all quantities have been defined and quantified through explicit equations, we are ready to describe the exact procedure for their iterative computation.

(3) Numerical results. Figure 16 shows the normalized low priority throughput of a single buffered dual priority delta–2 network for various network sizes as a function of the high priority input load. The offered load is 1, therefore the low priority input load equals to 1–high priority input load. The high priority normalized throughput is identical to the one shown in Fig. 14. There seems to be no specific direction to the model: sometimes optimistic and sometimes pessimistic. Nevertheless, the maximum deviation of our model is only 16.9% for fully loaded delta–2 network with 10 stages, i.e., a 1024 × 1024 delta–2 network.

4. Discussion

This paper presents a novel internally two priority buffered MIN architecture. It compares its performance with a single priority MIN. Simulation results show increase in high priority throughput of up to N times under hot spot traffic. For uniform traffic, we show an increase in low priority throughput, without any change in the high priority throughput. Moreover, while high priority delay and its standard deviation are increased when low priority traffic present in the single priority system, it is kept constant in the dual priority system. Finally, we introduce a new approach of long Markovian memory performance model to better capture the packets dependency in a single priority MIN under uniform traffic and extend this model for a dual priority MIN. Model results seems to be very accurate. Non-homogenous traffic study via simulation and analysis is yet to be studied.

Acknowledgement

The authors would like to thank Michael Laor from Cisco Systems for fruitful discussions on switch fabric architectures.

This research has been supported by the EuroNGI Network of Excellence.

Appendix 1. Explicit calculation of the single priority buffer transition probabilities

The transition probabilities of the single priority model are derived in this appendix and the explicit mathematical expressions are presented.

The probability that a packet is able to move forward depends on the state of the other buffer of the same *SE*, which affects the collision probability, and on the probability that its destination in the next stage is ready to accept the packet. If two packets are contending for the same output of an *SE*, one of the packets is randomly selected for transmission and the other packet has to wait for the next clock cycle.

Considering the probability that a packet can be accepted by its destination buffer at stage $k + 1$, there are three cases to distinguish.

(i) No packet has been sent to this destination buffer during the previous clock cycle. If none of the buffers of an *SE* at stage k sent a packet during the previous clock cycle to a destination buffer at stage $k + 1$, the possible states of that destination buffer at the beginning of the current clock cycle are: “00”, “10” and “$11b$”. Let $P_0^a(k, t)$ be the probability that a buffer of *SE*(k) is able to receive a packet at t_d, given that it received no packet at $(t - 1)_d$ for $2 \leqslant k \leqslant K$. Then,

$$P_0^a(k,t) = \big[P_{00}(k,t) + P_{10}(k,t) + P_{11b}(k,t)r_{11b}(k,t)\big] / \big[P_{00}(k,t) + P_{10}(k,t) + P_{11b}(k,t)\big]. \quad (14)$$

(ii) A packet has been sent to this destination buffer during the previous clock cycle. If one of the buffers of an *SE* at stage k sent a packet during the previous clock cycle to a destination buffer at stage $k + 1$, the possible states of that destination buffer at the beginning of the current clock cycle are: “01” and “$11n$”. Let $r_n(k, t)$ be the probability that a packet in a buffer of *SE*(k) is able to move forward at t_d, given that the packet is “new”, i.e., the buffer is either in state “01” or in state “$11n$” for $2 \leqslant k \leqslant K$. Then,

$$r_n(k,t) = vl\big[P_{01}(k,t)r_{01}(k,t) + P_{11n}(k,t)r_{11n}(k,t)\big] / \big[P_{01}(k,t) + P_{11n}(k,t)\big]. \quad (15)$$

(iii) The destination buffer is blocked. If one of the buffers of an *SE* at stage k has been blocked during the previous clock cycle, the destination buffer at stage $k + 1$ always contains a packet at the beginning of the current clock cycle. The destination buffer is in state “$11b$” if it did not receive a packet from the other buffer of the *SE* at stage k during the previous clock cycle. The probability of this event is $r_{11b}(k, t)$.

We are now ready to present the explicit mathematical expressions for the transition probabilities $r_{01}(k,t)$, $r_{11n}(k,t)$ and $r_{11b}(k,t)$ for $1 \leqslant k < K$.

$$\begin{aligned}
r_{01}(k,t) = {} & P_{00}(k,t)P_0^a(k+1,t) + P_{01}(k,t)0.75P_0^a(k+1,t) \\
& + P_{10}(k,t)\bigl[0.5P_0^a(k+1,t) + 0.5r_n(k+1,t)\bigr] \\
& + P_{11n}(k,t)\bigl[0.5 \cdot 0.75P_0^a(k+1,t) + 0.5 \cdot 0.75r_n(k+1,t)\bigr] \\
& + P_{11b}(k,t) \cdot \bigl[0.5 \cdot P_0^a(k+1,t) + 0.5 \cdot 0.5r_{11b}(k+1,t)\bigr],
\end{aligned} \tag{16}$$

$$\begin{aligned}
r_{11n}(k,t) = {} & 0.5\bigl\{P_{00}(k,t) \cdot P_0^a(k+1,t) + P_{01}(k,t)0.75P_0^a(k+1,t) + 0.5P_{10}(k,t)r_n(k+1,t) \\
& + 0.5P_{11n}(k,t)0.75r_n(k+1,t) + P_{11b}(k,t)\bigl[0.5 \cdot P_0^a(k+1,t) + 0.5 \cdot 0.5r_{11b}(k+1,t)\bigr]\bigr\} \\
& /\bigl\{P_{00}(k,t) + P_{01}(k,t) + 0.5P_{10}(k,t) + 0.5P_{11n}(k,t) + P_{11b}(k,t)\bigr\} \\
& + 0.5\bigl\{P_{00}(k,t)r_n(k+1,t) + P_{01}(k,t)0.75r_n(k+1,t) + 0.5P_{10}(k,t)r_n(k+1,t) \\
& + 0.5P_{11n}(k,t)0.75r_n(k+1,t) + P_{11b}(k,t)[0.5 \cdot 0.5 \cdot r_n(k+1,t) + 0.5r_n(k+1,t)]\bigr\} \\
& /\bigl\{P_{00}(k,t) + P_{01}(k,t) + 0.5P_{10}(k,t) + 0.5P_{11n}(k,t) + P_{11b}(k,t)\bigr\},
\end{aligned} \tag{17}$$

$$\begin{aligned}
r_{11b}(k,t) = {} & P_{00}(k,t)r_{11b}(k+1,t) + P_{01}(k,t)0.75r_{11b}(k+1,t) \\
& + P_{10}(k,t)\bigl[0.5r_{11b}(k+1,t) + 0.5r_n(k+1,t)\bigr] + P_{11n}(k,t)\bigl[0.5 \cdot 0.75r_{11b}(k+1,t) \\
& + 0.5 \cdot 0.75r_n(k+1,t)\bigr] + P_{11b}(k,t)0.75r_{11b}(k+1,t).
\end{aligned} \tag{18}$$

We now provide explanation for Eq. (17). The explanation of the simpler Eqs (16) and (18) is similar.

If a buffer of *SE*(k) is in state "$11n$", it means that a packet has been sent from it at $(t-1)_d$ and a new packet has been received at t_b. For convenience, let's refer to this buffer as the *relevant buffer*, the new packet as the *relevant packet* and the relevant packet's destination as the *relevant destination* or the *relevant destination buffer*. Similarly, the other buffer in the same *SE*, will be referred as the *neighbor buffer*, the neighbor buffer packet as the *neighbor packet* and its destination as the *neighbor destination*. To calculate the transition probability of the relevant packet, $r_{11n}(k,t)$, we need to consider two cases regarding the destination of these two packets: both packets are heading for the same *SE* output link, or each packet is heading for a different *SE* output link. Since the model assumes uniform distribution of destination addresses, each of the above two cases occurs with probability 0.5. Given one of the above two cases, we next need to consider the state of the neighbor buffer, the collision probability and the relevant destination buffer acceptance probability, which considers its inferred state. Let's, for example, assume that the two packets' destinations are different. This means that no packet has been sent by the relevant buffer to the relevant destination at $(t-1)_d$. The neighbor buffer can be in one of the following five states:

(1) "00", with probability $P_{00}(k,t)$. In this case, since the neighbor buffer is empty at t_b, there is no collision at t_d. Moreover, the neighbor buffer has also been empty at $(t-1)_b$, so no packet has been sent from it at $(t-1)_d$. Therefore, we can infer that no packet has been sent to the relevant destination at $(t-1)_d$ by either buffer, and the acceptance probability of the relevant destination buffer at stage $k+1$, is $P_0^a(k+1,t)$.

(2) "01", with probability $P_{01}(k,t)$. This case differs from the "00" case by the fact that the neighbor buffer has a packet at t_b. Therefore, a collision with the relevant packet can happen with probability 0.5, and since the contention resolution is random, the probability that each packet is not blocked by the other packet is 0.75.

(3) "10", with probability $P_{10}(k,t)$. The neighbor buffer sent a packet at $(t-1)_d$. Recall that the relevant buffer sent a packet at $(t-1)_d$ as well, and this packet was not sent to the relevant destination. Since only one packet can be sent to each *SE* output link during a clock cycle, the packet that the neighbor buffer sent at $(t-1)_d$ necessarily headed to the relevant destination. This imply that the relevant destination buffer has a new packet at t_b, and the acceptance probability is $r_n(k+1,t)$. Moreover, the neighbor buffer can only be

in one of the two sub-states of the "10" state, which are specified by the destination of the packet it sent at $(t-1)_d$. The probability to be in each sub-state is 0.5. There is no collision probability in this case.

(4) "$11n$", with probability $P_{11n}(k,t)$. Same as "10" case, with the addition of collision probability.

(5) "$11b$", with probability $P_{11b}(k,t)$. Here we need to consider the neighbor destination. If it is different than the relevant destination (with probability 0.5) then no packet has been sent to the relevant destination at $(t-1)_d$ by either buffer, and the acceptance probability is $P_0^a(k+1,t)$. No collision probability in this case. On the other hand, if the neighbor destination is equal to the relevant destination (with probability 0.5) then the destination buffer is blocked and the acceptance probability is $r_{11b}(k+1,t)$. The collision probability in this case is 0.5.

Since the whole term is conditioned on some sub-states, we need to divide it now by the sum of the probabilities of these sub-states. The other case, in which the two packets of the relevant buffer are heading to the same destination, is similarly calculated.

We still have to provide explicit mathematical expressions for the transition probabilities for the last switching stage K. Since there is no blocking at the output links of the network, the probability that a packet in a last stage *SE* is able to move forward depends only on the state of the other buffer of the same *SE*, which affects the collision probability. Therefore,

$$\begin{aligned} r_{01}(K,t) = {} & \bigl[P_{00}(K,t) + P_{01}(K,t)0.75 + P_{10}(K,t) + P_{11n}(K,t)0.75\bigr] \\ & /\bigl[P_{00}(K,t) + P_{01}(K,t) + P_{10}(K,t) + P_{11n}(K,t)\bigr], \end{aligned} \tag{19}$$

$$\begin{aligned} r_{11n}(K,t) = {} & \bigl[P_{00}(K,t) + P_{01}(K,t)0.75 + 0.5P_{10}(K,t) + 0.5P_{11n}(K,t)0.75 + 0.5P_{11b}(K,t)0.75\bigr] \\ & /\bigl[P_{00}(K,t) + P_{01}(K,t) + 0.5P_{10}(K,t) + 0.5P_{11n}(K,t) + 0.5P_{11b}(K,t)\bigr], \end{aligned} \tag{20}$$

$$r_{11b}(K,t) = \bigl[P_{10}(K,t) + P_{11n}(K,t)0.75\bigr]/\bigl[P_{10}(K,t) + P_{11n}(K,t)\bigr]. \tag{21}$$

Finally, we provide the expressions for the transition probabilities $q(k,t)$ for $1 \leqslant k \leqslant K$. Due to the backpressure mechanism, no packets are lost in the internal links. Consequently, the probability that a packet is entering a buffer at stage k is equal to the probability that a packet is transmitted on an output link of an *SE* at stage $k-1$.

$$\begin{aligned} q(k,t) = {} & S(k-1,t)/\bigl[P_{00}(k,t) + P_{01}(k,t)r_{01}(k,t) + P_{10}(k,t) + P_{11n}(k,t)r_{11n}(k,t) \\ & + P_{11b}(k,t)r_{11b}(k,t)\bigr] \quad (2 \leqslant k \leqslant K). \end{aligned} \tag{22}$$

Since there is no preceding stage to the first stage, $q(1,t)$ is the offered traffic load to the network inputs, i.e.,

$$q(1,t) = G, \tag{23}$$

where G is the probability that a packet arrives within a clock cycle to each input of the network.

Appendix 2. Explicit calculation of the low priority queue transition probabilities in the dual priority model

The low priority traffic transition probabilities of the dual priority model are derived in this appendix and the explicit mathematical expressions are presented.

The probability that a low priority packet is able to move forward depends on the probability of a high priority traffic transmission to the same destination, the probability of a collision with a packet from the other buffer of the same *SE* and on the probability that its destination in the next stage is ready to accept the packet. In other words, the low priority traffic can be blocked by a high priority traffic transmission, in addition to the single priority blocking probability. Therefore, we have added the definition of rlt, the probability that a low priority packet will move forward, and rlb, the probability that a low priority packet will be blocked by a low priority traffic. In order to keep the similarity to the single priority model and to easily calculate the rlt and rlb probabilities, we use the following parameters:

$rl_{01}(k,t)$: Probability that a packet in the low priority queue of an *SE*(k) buffer is able to move forward at t_d, given that it is not blocked by a high priority traffic and that the buffer is in state "01".

$rl_{11n}(k,t)$: Probability that a packet in the low priority queue of an *SE*(k) buffer is able to move forward at t_d, given that it is not blocked by a high priority traffic and that the buffer is in state "$11n$".

$rl_{11hb}(k,t)$: Probability that a packet in the low priority queue of an *SE*(k) buffer is able to move forward at t_d, given that it is not blocked by a high priority traffic and that the buffer is in state "$11hb$".

$rl_{11lb}(k,t)$: Probability that a packet in the low priority queue of an *SE*(k) buffer is able to move forward at t_d, given that it is not blocked by a high priority traffic and that the buffer is in state "$11lb$".

The mathematical expressions of rlt and rlb are described below.

$$rlt_{01}(k,t) = [1 - S(k,t)] \cdot rl_{01}(k,t+1), \tag{24}$$

$$rlt_{11n}(k,t) = [1 - S(k,t)] \cdot rl_{11n}(k,t+1), \tag{25}$$

$$rlt_{11lb}(k,t) = [1 - S(k,t)] \cdot rl_{11lb}(k,t+1), \tag{26}$$

$$rlt_{11hb}(k,t) = [1 - S(k,t)] \cdot rl_{11hb}(k,t+1), \tag{27}$$

$$rlb_{01}(k,t) = [1 - S(k,t)] \cdot [1 - rl_{01}(k,t+1)], \tag{28}$$

$$rlb_{11n}(k,t) = [1 - S(k,t)] \cdot [1 - rl_{11n}(k,t+1)], \tag{29}$$

$$rlb_{11lb}(k,t) = [1 - S(k,t)] \cdot [1 - rl_{11lb}(k,t+1)], \tag{30}$$

$$rlb_{11hb}(k,t) = [1 - S(k,t)] \cdot [1 - rl_{11hb}(k,t+1)]. \tag{31}$$

Considering the probability that a low priority packet can be accepted by its destination buffer at stage $k+1$, there are three cases to distinguish.

- No low priority packet has been sent to this destination buffer during the previous clock cycle. Let $Pl_0^a(k,t)$ be the probability that the low priority queue in an *SE*(k) buffer is able to receive a packet at t_d, given that it received no packet at $(t-1)_d$. Then,

$$\begin{aligned} Pl_0^a(k,t) = {} & \bigl[Pl_{00}(k,t) + Pl_{10}(k,t) + Pl_{11lb}(k,t)rlt_{11lb}(k,t) + Pl_{11hb}(k,t)rlt_{11hb}(k,t)\bigr] \\ & /\bigl[Pl_{00}(k,t) + Pl_{10}(k,t) + Pl_{11lb}(k,t) + Pl_{11hb}(k,t)\bigr]. \end{aligned} \tag{32}$$

- A low priority packet has been sent to this destination buffer during the previous clock cycle. Let $rl_n(k,t)$ be the probability that a packet in the low priority queue of an *SE*(k) buffer is able to move forward at t_d, given that the packet is "new", i.e., the queue is either in state "01" or in state "$11n$". Then,

$$rl_n(k,t) = \bigl[Pl_{01}(k,t)rlt_{01}(k,t) + Pl_{11n}(k,t)rlt_{11n}(k,t)\bigr]/\bigl[Pl_{01}(k,t) + Pl_{11n}(k,t)\bigr]. \tag{33}$$

- The low priority destination queue is blocked. Let $rlb(k,t)$ be the probability that a packet in the low priority queue of an *SE*(k) buffer is able to move forward at t_d, given that the packet is "blocked", i.e., the queue is either in state "$11lb$" or in state "$11hb$". Then,

$$rl_b(k,t) = \bigl[Pl_{11lb}(k,t)rlt_{11lb}(k,t) + Pl_{11hb}(k,t)rlt_{11hb}(k,t)\bigr]/\bigl[Pl_{11lb}(k,t) + Pl_{11hb}(k,t)\bigr]. \tag{34}$$

The following explicit mathematical expressions for the low priority traffic transition probabilities for $1 \leqslant k < K$ are calculated similarly to the single priority transition probabilities.

$$\begin{aligned} rl_{01}(k,t) = {} & Pl_{00}(k,t)Pl_0^a(k+1,t) + Pl_{01}(k,t)0.75Pl_0^a(k+1,t) \\ & + Pl_{10}(k,t)\big[0.5Pl_0^a(k+1,t) + 0.5rl_n(k+1,t)\big] \\ & + Pl_{11n}(k,t)\big[0.5 \cdot 0.75Pl_0^a(k+1,t) + 0.5 \cdot 0.75rl_n(k+1,t)\big] \\ & + Pl_{11lb}(k,t)\big[0.5Pl_0^a(k+1,t) + 0.5 \cdot 0.5rl_b(k+1,t)\big] \\ & + Pl_{11hb}(k,t)\big[0.5Pl_0^a(k+1,t) + 0.5 \cdot 0.5 \cdot Pl_0^a(k+1,t)\big], \end{aligned} \tag{35}$$

$$\begin{aligned} rl_{11n}(k,t) = {} & 0.5\big\{Pl_{00}(k,t)Pl_0^a(k+1,t) + Pl_{01}(k,t)0.75Pl_0^a(k+1,t) + 0.5Pl_{10}(k,t)rl_n(k+1,t) \\ & + 0.5Pl_{11n}(k,t)0.75rl_n(k+1,t) + Pl_{11lb}(k,t)\big[0.5Pl_0^a(k+1,t) + 0.5 \cdot 0.5rl_b(k+1,t)\big] \\ & + Pl_{11hb}(k,t)\big[0.5Pl_0^a(k+1,t) + 0.5 \cdot 0.5Pl_0^a(k+1,t)\big]\big\} \\ & /\big\{Pl_{00}(k,t) + Pl_{01}(k,t) + 0.5Pl_{10}(k,t) + 0.5Pl_{11n}(k,t) + Pl_{11lb}(k,t) + Pl_{11hb}(k,t)\big\} \\ & + 0.5\big\{Pl_{00}(k,t)rl_n(k+1,t) + Pl_{01}(k,t)0.75rl_n(k+1,t) + 0.5Pl_{10}(k,t)rl_n(k+1,t) \\ & + 0.5Pl_{11n}(k,t)0.75rl_n(k+1,t) + Pl_{11lb}(k,t) \cdot \big[0.5 \cdot 0.5rl_n(k+1,t) + 0.5rl_n(k+1,t)\big] \\ & + Pl_{11hb}(k,t)[0.5 \cdot 0.5rl_n(k+1,t) + 0.5rl_n(k+1,t)]\big\} \\ & /\big\{Pl_{00}(k,t) + Pl_{01}(k,t) \\ & + 0.5Pl_{10}(k,t) + 0.5 \cdot Pl_{11n}(k,t) + Pl_{11lb}(k,t) + Pl_{11hb}(k,t)\big\}, \end{aligned} \tag{36}$$

$$\begin{aligned} rl_{11lb}(k,t) = {} & Pl_{00}(k,t)rl_b(k+1,t) + Pl_{01}(k,t)0.75rl_b(k+1,t) \\ & + Pl_{10}(k,t) \cdot [0.5 \cdot rl_b(k+1,t) + 0.5rl_n(k+1,t)] \\ & + Pl_{11n}(k,t)[0.5 \cdot 0.75 \cdot rl_b(k+1,t) + 0.5 \cdot 0.75rl_n(k+1,t)] \\ & + Pl_{11lb}(k,t)0.75rl_b(k+1,t) + Pl_{11hb}(k,t)[0.5rl_b(k+1,t) \\ & + 0.5 \cdot 0.5 \cdot rl_b(k+1,t)], \end{aligned} \tag{37}$$

$$\begin{aligned} rl_{11hb}(k,t) = {} & Pl_{00}(k,t)Pl_0^a(k+1,t) + Pl_{01}(k,t)0.75Pl_0^a(k+1,t) \\ & + Pl_{10}(k,t) \cdot \big[0.5 \cdot Pl_0^a(k+1,t) + 0.5rl_n(k+1,t)\big] \\ & + Pl_{11n}(k,t)\big[0.5 \cdot 0.75Pl_0^a(k+1,t) + 0.5 \cdot 0.75rl_n(k+1,t)\big] \\ & + Pl_{11lb}(k,t)\big[0.5Pl_0^a(k+1,t) + 0.5 \cdot 0.5 \cdot rl_b(k+1,t)\big] \\ & + Pl_{11hb}(k,t)0.75Pl_0^a(k+1,t). \end{aligned} \tag{38}$$

We still have to provide explicit mathematical expressions for the transition probabilities for the last switching stage K. Since there is no blocking at the output links of the network, the probability that a low priority packet in a last stage *SE* is able to move forward depends only on the high priority traffic and on the state of the other low priority queue of the same *SE*, which affects the collision probability. The explicit mathematical expressions for the low priority traffic transition probabilities of the last stage *SE* are as follows.

$$rl_{01}(K,t) = \big[Pl_{00}(K,t) + Pl_{01}(K,t)0.75 + Pl10(K,t) + Pl_{11n}(K,t)0.75 + Pl_{11hb}(K,t)0.75\big] \\ /\big[Pl_{00}(K,t) + Pl_{01}(K,t) + Pl_{10}(K,t) + Pl_{11n}(K,t) + Pl_{11hb}(K,t)\big], \quad (39)$$

$$\begin{aligned} rl_{11n}(K,t) = {}& \big[Pl_{00}(K,t) + Pl_{01}(K,t)0.75 + 0.5Pl_{10}(K,t) \\ &+ 0.5Pl_{11n}(K,t)0.75 + 0.5Pl_{11lb}(K,t)0.75 + Pl_{11hb}(K,t)0.75\big] \\ &/\big[Pl_{00}(K,t) + Pl_{01}(K,t) + 0.5Pl_{10}(K,t) + 0.5Pl_{11n}(K,t) \\ &+ 0.5Pl_{11lb}(K,t) + Pl_{11hb}(K,t)\big], \qquad (40) \\ rl_{11lb}(K,t) = {}& \big[Pl_{10}(K,t) + Pl_{11n}(K,t)0.75\big]/\big[Pl_{10}(K,t) + Pl_{11n}(K,t)\big], \qquad (41) \\ rl_{11hb}(K,t) = {}& \big[Pl_{00}(K,t) + Pl_{01}(K,t)0.75 + Pl_{10}(K,t) + Pl_{11n}(K,t)0.75 + Pl_{11hb}(K,t)0.75\big] \\ &/\big[Pl_{00}(K,t) + Pl_{01}(K,t) + Pl_{10}(K,t) + Pl_{11n}(K,t) + Pl_{11hb}(K,t)\big]. \qquad (42) \end{aligned}$$

Finally, the quantities $ql(k,t)$ are calculated in a similar manner to the single priority model, i.e.,

$$ql(k,t) = Sl(k-1,t)/\big[Pl00(k,t) + Pl01(k,t)rlt_{01}(k,t) + Pl_{10}(k,t) + Pl_{11n}(k,t)rlt_{11n}(k,t) \\ + Pl_{11lb}(k,t)rlt_{11lb}(k,t) + Pl_{11hb}(k,t)rlt_{11hb}(k,t)\big] \quad (2 \leqslant k \leqslant K). \quad (43)$$

The total offered load Gt, which is the probability that a packet arrives within a clock cycle to each input of the network, is actually the sum of the low priority offered load, Gl, and the high priority offered load, G.

$$Gt = Gl + G. \quad (44)$$

Since there is no preceding stage to the first stage, $ql(1,t)$ is specified manually as the offered load of the low priority traffic to the network inputs.

$$ql(1,t) = Gl. \quad (45)$$

References

[1] K. Liu, D.W. Petr and V.S. Frost, Design and analysis of a bandwidth management framework for ATM-based broadband ISDN, *IEEE Communications Magazine* **35**(5) (1997), 138–145.

[2] I. Cidon, R. Guerin and A. Khamisky, On protective buffer policies, *IEEE/ACM Transactions on Networking* **2**(3) (1994), 240–246.

[3] S.P. Morgan, Queueing disciplines and passive congestion control in byte-stream networks, *IEEE Trans. Commun.* **39**(7) (1991), 1097–1106.

[4] D-Link DES-3250TG 10/100Mbps managed switch, http://www.dlink.co.uk/DES-3250TG.htm.

[5] Intel® Express 460T standalone switch, http://www.intel.com/support/express/switches/460/30281.htm.

[6] J.S.C. Chen and R. Guerin, Performance study of an input queueing packet switch with two priority classes, *IEEE Trans. Commun.* **39**(1) (1991) 117–126.

[7] S.L. Ng and B. Dewar, Load sharing replicated buffered banyan networks with priority traffic (unpublished).

[8] J.H. Patel, Processor memory interconnections for multiprocessors, *IEEE Trans. Comput.* **C-30** (1981), 771–780.

[9] http://www.cisco.com/application/pdf/en/us/guest/products/ps5763/c1031/cdccont_0900aecd800f8118.pdf.

[10] http://newsroom.cisco.com/dlls/2004/next_generation_networks_and_the_cisco_carrier_routing_system_overview.pdf.

[11] Y.-C. Jenq, Performance analysis of a packet switch based on single-buffered banyan network, *IEEE Journal Selected Areas of Commun.* **SAC-1** (6) (1983), 1014–1021.

[12] T. Szymanski and S. Shaikh, Markov chain analysis of packet-switched banyans with arbitrary switch sizes, queue sizes, link multiplicities and speedups, in: *Proc. INFOCOM 89*, 1989.

[13] H. Yoon, K.Y. Lee, and M.T. Lui, Performance analysis of multibuffered packet-switching networks in multiprocessor systems, *IEEE Trans. Comput.* **39** (1990), 319–327.

[14] J.S. Turner, Queueing analysis of buffered switching networks, *IEEE Trans. Commun.* **41**(2) (1993), 412–420.

[15] S.H. Hsiao and C.Y.R. Chen, Performance analysis of single-buffered multistage interconnection networks, *IEEE Trans. Commun.* **42**(9) (1994), 2722–2729.

[16] T.H. Theimer, E.P. Rathgeb and M.N. Huber, Performance analysis of buffered banyan networks, *IEEE Trans. Commun.* **39**(2) (1991), 269–277.

[17] H. Mun and H.Y. Youn, Performance analysis of finite buffered multistage interconnection networks, *IEEE Trans. Comput.* **43**(2) (1994), 153–161.

[18] K.S. Chan, K.L. Yeung and S.C.H. Chan, A refined model for performance analysis of buffered banyan networks with and without priority control, *IEICE Trans. Commun.* **E82-B**(1) (1999), 48–59.

[19] D.M. Dias and J.R. Jump, Analysis and simulation of buffered delta networks, *IEEE Trans. Comput.* **C-30**(4) (1981), 273–282.

[20] N. Chrysos and M. Katevenis, Multiple priorities in a two-lane buffered crossbar, in: *Proc. IEEE Globecom 2004 Conference*, Dallas, TX, USA, 2004.