Delay Analysis of Combined Input-Crosspoint Queueing Switches

Ge Nong, Ning Situ and Mounir Hamdi

***Abstract*— The switch architecture with the combined input-crosspoint queueing (CICQ) scheme has been recognized as a practical promising solution for building cost-effective high-performance switches. In an $N \times N$ CICQ switch, the switching fabric is a nonblocking buffered crossbar, a large input buffer is provided at each input and a relatively small internal buffer is provided at each crosspoint of the buffered crossbar. Each input buffer is logically organized as N virtual output queues (VOQs). In this paper, we build the queueing model for evaluating the delay performance of a CICQ switch under i.i.d uniform 2-state Markov Modulated Bernoulli Process (2-MMBP) bursty traffic. The accuracy of the queuing model is examined via computer simulation, by investigating the mean cell delay in a switch as a function of the switch size, the internal buffer size, the mean offered load and the mean burst length. The numerical results show that our queueing model can well analyze the reality.**

***Index Terms*— Combined input-crosspoint queueing switch, queueing analysis, simulation, performance modelling.**

I. Introduction

The combined input-crosspoint queueing (CICQ) switch architecture has attracted lots of research interests from the academic and industrial communities [1]–[5] and has been well recognized as a practical promising choice for building scalable high-speed switches. A CICQ switch uses VOQ at each input port and a buffered crossbar (BX) as the switching fabric. In an $N \times N$ BX, an internal buffer situates at the crosspoint of each pair of input and output, resulting in a total of N^2 internal buffers. Each internal buffer is typically very small, ranging from several to tens cells. In a CICQ switch, the major buffer space for queueing packets is provided by the input buffers at the input ports, i.e. the VOQ buffers; the internal buffers at the BX are mainly used for designing distributed scheduling algorithms that can resolve input/output ports contentions faster and better.

When the performance of a switch is evaluated, two very important measures are the throughput and the mean cell delay of the switch under a given traffic, i.e. the delay-throughput performance. A lot of simulation studies have been contributed on evaluating the delay and throughput performances of CICQ switches with various scheduling algorithms and traffic [1], [6]. In contrast, there are only a few analytical studies on the delay and throughput performances of CIOQ switches. For examples, the *throughput* performance of a CICQ switch under uniform Bernoulli or bursty traffic is analyzed in [7]–[9]; and a network calculus-based analysis for the *delay bound* of a CICQ switch is done in [10] using a service-curve based model. However, to the best of our knowledge, so far, there is no *queueing analysis model* published for evaluating the delay performance of a CICQ switch under Bernoulli or bursty traffic. While there are quite many queueing analysis models available for evaluating the delay performances of the other sorts of switches, e.g., output queueing, IQ (FIFO and VOQ) and shared-centralized queuing switches, there is a need in practice for us to develop a queueing model to theoretically analyze the delay performance of a CICQ switch. Our contribution in this paper is dedicated to this.

We build in this paper a queueing model for evaluating the delay performance of a CICQ switch under i.i.d uniform 2-state MMBP (Markov Modulated Bernoulli Process) bursty traffic, where both the input and the output scheduling employ the random selection policy. The rest of this paper is organized as the following. We first describe in Section II the system model and then build in Section III the queuing model. In Section IV, the accuracy of the queueing model is examined by computer simulation.

II. The Switch Model

A. Switch Architecture

The general CICQ switch model introduced in [1] is adopted here, which includes four main components:

- Input buffers: each input port has an input buffer organizing as N VOQs for queuing arriving cells, one queue for each output port.
- Crosspoint buffer: a crosspoint buffer (XB) is provided at each crosspoint inside the buffered crossbar.
- Flow control: it is a common practice to employ some kind of flow control mechanism to avoid overflows at each XB. Such a flow control scheme ensures that when a XB is full, the corresponding VOQ is not eligible for input arbitration.
- Input/output arbitration: each input arbiter selects according to the input arbitration policy a nonempty and eligible VOQ for sending a cell to the corresponding XB, and each output arbiter selects according to the output arbitration policy a cell from the corresponding nonempty XB to depart.

From the above description, we see that the total delay of a cell in a CICQ switch consists of two parts: the delay in the input buffer and that in the crosspoint buffer. In our queueing

Ge Nong is with the Computer Science Department, Sun Yat-Sen University, GuangZhou 510275, PRC. Email: issng@mail.sysu.edu.cn. His work is supported by the GuangDong Natural Science Foundation research grant number 06023193.

Ning Situ is with the Computer Science Department, University of Houston, TX 77004, USA. Email: stnstone@cs.uh.edu.

Mounir Hamdi is with the Computer Science Department, Hong Kong University of Science and Technology, Hong Kong. Email: hamdi@cs.ust.hk.

1-4244-1251-X/07/$25.00 2007 IEEE.

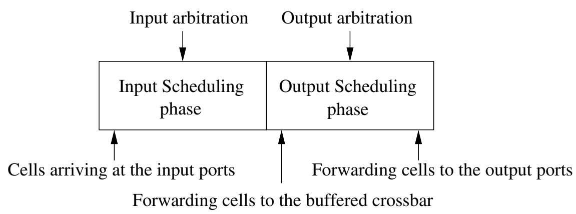

Fig. 1. The timing of events in the switch.

model developed below, these two parts are solved individually and summed up as the total delay.

B. Assumptions

We lists the below general assumptions for developing the queueing model:

1) The switch is symmetric $N \times N$ and each input/output port operates at the same speed.
2) The sizes of each VOQ and each XB are infinite and s, respectively.
3) The switch is operating synchronously over fixed size time slots, where each time slot is normalized as the time interval for transmitting a cell at the input/output port speed;
4) Each time slot comprises two phases in sequence: the input scheduling phase and the output scheduling phase;
5) Both the input and the output arbitrations uses the random selection policy, i.e. to select one randomly from all participating candidates of a contention;
6) Cell losses in the switch are rare, which implies that the mean arrival and the mean departure rates at any buffer in the switch are equal;
7) The destinations of arriving cells at the input ports are uniformly distributed, i.e., each arriving cell is destined to any output port with a probability $1/N$;
8) The traffic at each input port is i.i.d, with a mean load $N\lambda$;
9) At most one cell can arrive at each input port only at the beginning of a time slot;
10) At most one cell can depart at each output port only at the end of a time slot.

The timing of events in the switch model is summarized by Fig. 1.

III. Queueing Model Under 2-MMBP Traffic

A. General Idea

The queueing model in [11] for the VOQ switch under uniform 2-state Markov-Modulated Bernoulli Process (2-MMBP) traffic is built by solving the underlying Quasi-Birth-Death (QBD) Markov chains of the systems. Naturally, for the CICQ switch under uniform 2-MMBP traffic, a queuing model can be built as a QBD with each state encoded as a 5-tuplet (g, L_i, W_i, L_x, W_x), where g, L_i, W_i, L_x and W_x are the state of the 2-MMBP traffic at the tagged VOQ, the lengths of the tagged VOQ, the tagged input virtual HOL queue, the tagged crossbar queue (XQ) and the tagged crossbar virtual HOL queue, respectively, as shown in Fig. 2. However, such a state space will grow exponentially when the sizes of the switch and the XB increase. To avoid a computation intractable huge state space, instead of modeling the system by a single Markov chain, we will use two decoupled queueing systems to build the queuing model of the whole system, one for the tagged XQ and one for the tagged VOQ, respectively.

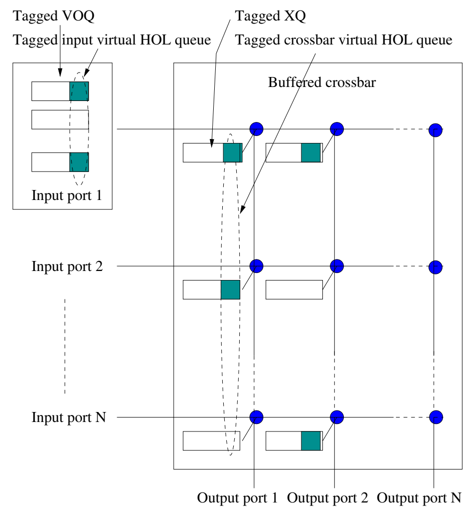

Fig. 2. The queueing models for the tagged input buffer and the tagged crossbar buffer

To decouple the tagged XQ and the tagged VOQ from each other, we build the queueing models for them separately. Specifically, the tagged VOQ is modeled as a 2-MMBP/G/1 queueing system which mean service time can be computed when the queueing model for the tagged XQ is solved. In the rest of this paper, we will further exploit this idea to build the queueing model for the switch under i.i.d uniform 2-MMBP bursty traffic. In general, we will employ an iterative computation approach to solve firstly the queueing model for the tagged XQ and then that for the tagged VOQ, by holding the fact that the throughputs of the studied systems will equal the offered loads when the systems approach steady states.

B. The Tagged XQ

The 2-MMBP traffic governed by two parameters α and β is depicted in Fig. 3, which is widely used as a model for bursty traffic with arrivals correlated in a long term. A state transition of the offered 2-MMBP traffic at each input port can occur only at the beginning of each time slot. A 2-MMBP can generate a cell in two cases: (1) from the ON state back to the ON state with a probability α, and (2) from the OFF state to the ON state with a probability $1-\beta$. Each cell is uniformly destined for any output port with a probability of $1/N$. Given the mean burst length τ and the mean arrival rate $N\lambda$ of the traffic at each input port, the parameters α and β of a 2-MMBP can be calculated as $\alpha = 1 - 1/\tau$ and $\beta = (1 - N\lambda(2-\alpha))/(1 - N\lambda)$, respectively [11].

A 2-MMBP actually consists of two interleaving Bernoulli processes for the states ON and OFF, each with its own cell generation probability: α for the state ON and $1-\beta$ for the

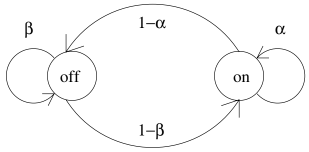

Fig. 3. The 2-MMBP bursty traffic model.

state OFF, respectively. In Fig. 2, noticed that when the tagged XQ is not full, a HOL cell of the tagged VOQ can be blocked only by cells at input port 1, which is analogous to increase the probability for a cell to be destined to output port 1 when the cell is generated by the 2-MMBP at the input 1. To take this characteristics into account in the traffic model for the tagged XQ, by biasing the probability for a cell arriving at input port 1 to be destined for output port 1 from $1/N$ to $\eta_2/(N\lambda)$, we have the below traffic model assumed for cells arriving at the tagged XQ in a CICQ switch under a 2-MMBP traffic with a mean load $N\lambda$ and a mean burst length τ.

Traffic Model 1: The cells arriving process at the input 1 of the buffered crossbar is a 2-MMBP with $\alpha = (1 - 1/\tau)$, $\beta = (1 - N\lambda(2-\alpha))/(1 - N\lambda)$, each arriving cell is destined for the output 1 with a probability $\gamma = \eta_2/(N\lambda)$ if the tagged XQ is not full, or else with a probability 0.

With the above traffic model, the tagged XQ is offered with a mean load of η_2 when it is not full. As examined by the simulations presented later, this assumed traffic model for the tagged XQ can render us a very accurate queueing model for the buffered crossbar under uniform 2-MMBP traffic. To develop the queueing model, η_2 is supposed to be a known parameter, which can be computed using an iterative computation method once the queueing model is built. In the rest of this subsection, two Markov chains are built and solved for the tagged XQ in respect to whether the size of crossbar buffer is larger than 1 or not, i.e., for $s > 1$ and $s = 1$, respectively.

1) Markov Chain for $s > 1$: Each state of the Markov chain Z is expressed as a triplet (L, G, W) sampled at the end of each time slot, where L, G and W refer to the length of the *tagged XQ*, the state of the *2-MMBP traffic at the tagged XQ* (modeled by Traffic Model 1) and the length of the *tagged crossbar virtual HOL queue (XVHQ)*, respectively. The state-space of this three-dimension Markov chain is

$$\{(0, g, 0), (l, g, w) | l \in [1, s], g \in [0, 1], w \in [1, N]\}$$

and all states are ordered in the lexicographic order, i.e.

$$((0, g, 0), (1, 0, 1), \ldots, (s, 0, N), (s, 1, 1), \ldots, (1, 1, N)$$

where $g = 0$ and $g = 1$ represent that the 2-MMBP at the tagged XQ are in the states OFF and ON, respectively. The transition probability matrix T of the Markov chain Z is defined below:

$$T = \begin{pmatrix} C_1 & C_2 & 0 & & & & & & & \\ C_0 & A_1 & A_2 & 0 & & & & & & \\ 0 & A_0 & A_1 & A_2 & 0 & & & & & \\ 0 & 0 & A_0 & A_1 & A_2 & 0 & & & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & & & \\ 0 & 0 & 0 & \cdots & \cdots & \cdots & 0 & A_0 & A_1 & A_2 \\ 0 & 0 & 0 & \cdots & \cdots & \cdots & 0 & 0 & D_0 & D_1 \end{pmatrix}$$

where $C_1 + C_2 e_1 = 1$, $C_0 + (A_1 + A_2)e_1 = (A_0 + A_1 + A_2)e = (D_0 + D_1)e_1 = e_1$ and e_1 is a column vector of ones of size $2N$. The steady-state probability vector of Z is given by $\Pi = (\pi_{(0,g)}, \pi_{(1,g)}, ..., \pi_{(s,g)})$ where each element $\pi_{(l,g)} = (\pi_{(l,g,1)}, \pi_{(l,g,2)}, ..., \pi_{(l,g,N)})$, $l \in [1, s]$, is a row vector of size N, and $\pi_{(0,g)}$ is a scalar. Let's denote $\pi_0 = \pi_{(0,0)} + \pi_{(0,1)}$. The steady-state probabilities for the states in level l are denoted by $\pi_l = (\pi_{(l,0)}, \pi_{(l,1)})$, where $\pi_{(l,0)}$ and $\pi_{(l,1)}$ are the probability vectors with the traffic at the tagged XQ being in the states 0 and 1, respectively.

let $P_{blo,W_t(w')|W_{t-1}(g,w)}$ ($P'_{blo,W_t(w')|W_{t-1}(g,w)}$, respectively) denote the joint probability that the HOL cell of the tagged XQ is *blocked* and there is (is not, respectively) a new cell arrival at the tagged XQ at the beginning of the current output scheduling phase, and similarly $P_{suc,W_t(w')|W_{t-1}(g,w)}$ ($P'_{suc,W_t(w')|W_{t-1}(g,w)}$, respectively) denote the joint probability that the HOL cell of the tagged XQ is *transmitted* and there is (is not, respectively) a new cell arrival at the tagged XQ at the beginning of the current output scheduling phase, given the following: 1) at the end of the last time slot, the traffic source at the tagged XQ is in state g and the length of the tagged XVHQ is $w \in [1, N]$, and 2) at the end of the current time slot, the lengths of the tagged XVHQ is $w' \in [1, N]$.

In case that there is a new cell arriving at the tagged XQ at the beginning of the current output scheduling phase, we define six matrices $B^{(g)}$, $B_0^{(g)}$ and $S^{(g)}$ as Eq.(1)(2)(3), respectively, where $g \in [0, 1]$. For the case that there is no cell arriving at the tagged XQ, we define another six matrices $B'^{(g)}$, $B_0'^{(g)}$ and $S'^{(g)}$ similar to $B^{(g)}$, $B_0^{(g)}$ and $S^{(g)}$, by replacing $P_{blo,W_t(w')|W_{t-1}(g,w)}$ in $B^{(g)}$ with $P'_{blo,W_t(w')|W_{t-1}(g,w)}$, $P_{blo,W_t(w')|W_{t-1}(g,0)}$ in $B_0^{(g)}$ with $P'_{blo,W_t(w')|W_{t-1}(g,0)}$ and $P_{suc,W_t(w')|W_{t-1}(g,w)}$ in $S^{(g)}$ with $P'_{suc,W_t(w')|W_{t-1}(g,w)}$, respectively.

Using these definitions, the sub-matrices in the transition matrix T are computed as

$$C_0 = \begin{pmatrix} H_0 e & (S'^{(0)} - H_0)e \\ H_1 e & (S'^{(1)} - H_1)e \end{pmatrix}, \quad C_1 = \begin{pmatrix} \beta & 1 - \beta - B_0^{(0)} e \\ 1 - \alpha & \alpha - B_0^{(1)} e \end{pmatrix},$$

$$C_2 = \begin{pmatrix} z_r & B_0^{(0)} \\ z_r & B_0^{(1)} \end{pmatrix}, \quad A_0 = \begin{pmatrix} H_0 & S'^{(0)} - H_0 \\ H_1 & S'^{(1)} - H_1 \end{pmatrix},$$

$$A_1 = \begin{pmatrix} G_0 & S^{(0)} + B'^{(0)} - G_0 \\ G_1 & S^{(1)} + B'^{(1)} - G_1 \end{pmatrix}, \quad A_2 = \begin{pmatrix} z_m & B^{(0)} \\ z_m & B^{(1)} \end{pmatrix},$$

$$D_0 = \begin{pmatrix} H_0 & S^{(0)} + S'^{(0)} - H_0 \\ H_1 & S^{(1)} + S'^{(1)} - H_1 \end{pmatrix}, \quad D_1 = \begin{pmatrix} G_0 & B^{(0)} + B'^{(0)} - G_0 \\ G_1 & B^{(1)} + B'^{(1)} - G_1 \end{pmatrix},$$

In the above matrices, e is a column vector of ones of size N, z_r is a row vector of zeros of size N, z_m is an $N \times N$ matrix

$$B_0^{(g)} = \begin{pmatrix} P_{blo,W_t(1)|W_{t-1}(g,0)} & P_{blo,W_t(2)|W_{t-1}(g,0)} & \cdots & P_{blo,W_t(n)|W_{t-1}(g,0)} \end{pmatrix} \tag{1}$$

$$B^{(g)} = \begin{pmatrix} P_{blo,W_t(1)|W_{t-1}(g,1)} & P_{blo,W_t(2)|W_{t-1}(g,1)} & \cdots & P_{blo,W_t(n)|W_{t-1}(g,1)} \\ P_{blo,W_t(1)|W_{t-1}(g,2)} & P_{blo,W_t(2)|W_{t-1}(g,2)} & \cdots & P_{blo,W_t(n)|W_{t-1}(g,2)} \\ P_{blo,W_t(1)|W_{t-1}(g,3)} & P_{blo,W_t(2)|W_{t-1}(g,3)} & \cdots & P_{blo,W_t(n)|W_{t-1}(g,3)} \\ \vdots & \vdots & \cdots & \vdots \\ P_{blo,W_t(1)|W_{t-1}(g,n)} & P_{blo,W_t(2)|W_{t-1}(g,n)} & \cdots & P_{blo,W_t(n)|W_{t-1}(g,n)} \end{pmatrix} \tag{2}$$

$$S^{(g)} = \begin{pmatrix} P_{suc,W_t(1)|W_{t-1}(g,1)} & P_{suc,W_t(2)|W_{t-1}(g,1)} & \cdots & P_{suc,W_t(n)|W_{t-1}(g,1)} \\ P_{suc,W_t(1)|W_{t-1}(g,2)} & P_{suc,W_t(2)|W_{t-1}(g,2)} & \cdots & P_{suc,W_t(n)|W_{t-1}(g,2)} \\ P_{suc,W_t(1)|W_{t-1}(g,3)} & P_{suc,W_t(2)|W_{t-1}(g,3)} & \cdots & P_{suc,W_t(n)|W_{t-1}(g,3)} \\ \vdots & \vdots & \cdots & \vdots \\ P_{suc,W_t(1)|W_{t-1}(g,n)} & P_{suc,W_t(2)|W_{t-1}(g,n)} & \cdots & P_{suc,W_t(n)|W_{t-1}(g,n)} \end{pmatrix} \tag{3}$$

of zeros, $H_0 = (S^{(0)}\beta)/((1-\beta)\gamma)$, $H_1 = (S^{(1)}(1-\alpha))/(\alpha\gamma)$, $G_0 = (B^{(0)}\beta)/((1-\beta)\gamma)$ and $G_1 = (B^{(1)}(1-\alpha))/(\alpha\gamma)$. The success and blocking probability are computed by Eq.(6)-(17), where the the parameters in the formulas are defined as below:

- η_3: the probability that there is a new cell arriving at the tagged XQ at the beginning of an output scheduling phase given that the tagged XQ is empty at the end of the last time slot,

$$\eta_3 = \frac{(\pi_{(0,0)}(1-\beta) + \pi_{(0,1)}\alpha)\gamma}{\pi_{(0,0)} + \pi_{(0,1)}}.$$

- p_0: the probability that the tagged XQ is empty at the beginning of an output scheduling phase,

$$p_0 \quad = \quad (1-(1-\beta)\gamma)\pi_{(0,0)} + (1-\alpha\gamma)\pi_{(0,1)}.$$

  For denotation convenience, let $p_1 = 1 - p_0$.

- l_1: the probability that given the tagged XQ is nonempty at the end of the last time slot, the length of the tagged XQ is 1 at the beginning of the current output scheduling phase,

$$l_1 \quad = \quad \frac{\pi_{(1,0)}e(1-(1-\beta)\gamma) + \pi_{(1,1)}e(1-\alpha\gamma)}{1-\pi_0} \tag{4}$$

- l_2: the probability that given the tagged XQ is nonempty at the beginning of an output scheduling phase, the length of the tagged XQ is 1,

$$l_2 \quad = \quad \frac{\pi_{(0,0)}(1-\beta)\gamma + \pi_{(0,1)}\alpha\gamma}{p_1} + \frac{\pi_{(1,0)}(1-(1-\beta)\gamma) + \pi_{(1,1)}(1-\alpha\gamma)}{p_1} \tag{5}$$

2) Computing the Mean Cell Delay in the Tagged XQ: Let $\overline{Q}$ be the mean queue length of the tagged XQ, we have

$$\overline{Q} = \sum_{l=1}^{s} l\pi_l e_1 \tag{18}$$

Given the tagged XQ has a throughput of λ, from $\overline{Q}$, the mean cell delay $\overline{D}$ in the tagged XQ can be computed by Little's Law as

$$\overline{D} = \frac{\overline{Q}}{\lambda} \tag{19}$$

3) The Iterative Computation Algorithm: An iterative computation method is employed to calculate the parameter η_2 and solve the queueing models in both cases for $s > 1$ and $s = 1$. Notice that when a queueing model in this context is steady under an offered load, its throughput must equal to the load, which implies

$$\begin{aligned} \lambda \quad = \quad & \gamma((1-\beta)(\textstyle\sum_{l=1}^{s-1} l\pi_{(l,0)}e + \pi_{(0,0)}) + \\ & \alpha(\textstyle\sum_{l=1}^{s-1} l\pi_{(l,1)}e + \pi_{(0,1)})) \end{aligned} \tag{20}$$

The queueing model can be solved by an iterative computation method. First, we initiate η_2, $\pi_{(0,0)}$ and $\pi_{(0,1)}$ by some values, e.g., $\eta_2 = \lambda$ and $\pi_{0,0} = \pi(0,1) = 1/(2sN)$, and we get the Markov chain Z with its transition matrix T being computed using the initiated parameters η_2, $\pi_{(0,0)}$ and $\pi_{(0,1)}$. Then, we solve Z for new $\pi_{(0,0)}$ and $\pi_{(0,1)}$ and use Eq.(20) to update η_2. This process is repeated until the three parameters are converged to their steady values. At last, from the solved Z, we can computed the mean cell delay in the tagged XQ using Eq.(18)(19).

4) Markov Chain for $s = 1$: In general, the Markov Chain Z in this case can be constructed similar to that of the previous case for $s > 1$, with some slight differences: (1) the state $(1, g, N)$ will not appear when $s = 1$, for each XQ winning an output arbitration in a time slot will be empty at the end of the time slot; and (2) we don't need π_1 in computing the transition probabilities, for both the probabilities l_1 and l_2 equal to 1 in this case. Therefore, the Markov Chain Z in this case is three-dimensional given by

$$\{(0, g, 0), (1, g, w) | g \in [0, 1], w \in [1, N-1]\},$$

and all states are ordered in the lexicographic order, i.e.

$$(0, 0, 0), (0, 1, 0), (1, 0, 1), (1, 0, 2), \ldots, (1, 1, N-1).$$

Similarly, the transition matrix T can be constructed like that in the immediately previous case for $s > 1$.

Fig. 4 shows the numerical results from the queueing models and the simulations for the mean cell delays in the buffered crossbar as the functions of the XB sizes and the mean offered loads, for two switch sizes of 16 and 32, and two mean burst lengths of 16 and 32. Because that for a given mean burst length τ the maximum mean offered load is $\tau/(\tau+1)$, the

$$P_{suc,W_t(w')|W_{t-1}(0,w)} = (1-\beta)\gamma \frac{1}{w'} \binom{N-w}{w'-w} \eta_3^{(w'-w)} (1-\eta_3)^{(N-w')}, \quad 0 < w \le w' \le N \tag{6}$$

$$P_{suc,W_t(w')|W_{t-1}(1,w)} = \alpha\gamma \frac{1}{w'} \binom{N-w}{w'-w} \eta_3^{(w'-w)} (1-\eta_3)^{(N-w')}, \quad 0 < w \le w' \le N \tag{7}$$

$$\begin{aligned} P_{blo,W_t(w')|W_{t-1}(0,w)} \quad = \quad & (1-\beta)\gamma(\frac{w-1}{w'} \binom{N-w}{w'-w} \eta_3^{(w'-w)} (1-\eta_3)^{(N-w')} (1-l_1) \\ & + \frac{w-1}{w'+1} \binom{N-w}{w'+1-w} \eta_3^{(w'+1-w)} (1-\eta_3)^{(N-w'-1)} l_1 \\ & + \frac{w'-w+1}{w'+1} \binom{N-w}{w'+1-w} \eta_3^{(w'+1-w)} (1-\eta_3)^{(N-w'-1)}), \quad 1 \le w-1 < w' < N \end{aligned} \tag{8}$$

$$\begin{aligned} P_{blo,W_t(w')|W_{t-1}(1,w)} \quad = \quad & \alpha\gamma(\frac{w-1}{w'} \binom{N-w}{w'-w} \eta_3^{(w'-w)} (1-\eta_3)^{(N-w')} (1-l_1) \\ & + \frac{w-1}{w'+1} \binom{N-w}{w'+1-w} \eta_3^{(w'+1-w)} (1-\eta_3)^{(N-w'-1)} l_1 \\ & + \frac{w'-w+1}{w'+1} \binom{N-w}{w'+1-w} \eta_3^{(w'+1-w)} (1-\eta_3)^{(N-w'-1)}), \quad 1 \le w-1 < w' < N \end{aligned} \tag{9}$$

$$P_{blo,W_t(w-1)|W_{t-1}(0,w)} = (1-\beta)\gamma l_1, \quad 1 < w \le N \tag{10}$$

$$P_{blo,W_t(w-1)|W_{t-1}(1,w)} = \alpha\gamma l_1, \quad 1 < w \le N \tag{11}$$

$$P_{blo,W_t(N)|W_{t-1}(0,w)} = (1-\beta)\gamma \frac{w-1}{N} \eta_3^{(N-w)} (1-l_1), \quad 1 < w \le N \tag{12}$$

$$P_{blo,W_t(N)|W_{t-1}(1,w)} = \alpha\gamma \frac{w-1}{N} \eta_3^{(N-w)} (1-l_1), \quad 1 < w \le N \tag{13}$$

$$\begin{aligned} P_{blo,W_t(w')|W_{t-1}(0,0)} \quad = \quad & (1-\beta)\gamma(\frac{w'-1}{w'} \binom{N-1}{w'-1} p_1^{(w'-1)} p_0^{(N-w')} (1-l_2) \\ & + \frac{w'}{w'+1} \binom{N-1}{w'} p_1^{w'} p_0^{(N-1-w')} l_2), \quad 0 < w' \le N \end{aligned} \tag{14}$$

$$\begin{aligned} P_{blo,W_t(w')|W_{t-1}(1,0)} \quad = \quad & \alpha\gamma(\frac{w'-1}{w'} \binom{N-1}{w'-1} p_1^{(w'-1)} p_0^{(N-w')} (1-l_2) \\ & + \frac{w'}{w'+1} \binom{N-1}{w'} p_1^{w'} p_0^{(N-1-w')} l_2), \quad 0 < w' \le N \end{aligned} \tag{15}$$

$$P_{blo,W_t(N)|W_{t-1}(0,0)} = (1-\beta)\gamma \frac{N-1}{N} p_1^{(N-1)} (1-l_2) \tag{16}$$

$$P_{blo,W_t(N)|W_{t-1}(1,0)} = \alpha\gamma \frac{N-1}{N} p_1^{(N-1)} (1-l_2) \tag{17}$$

maximum mean offered load in the figures is chosen to be 0.9. For the switches under investigation, the results from the queueing models and the simulations are very close and nearly indistinguishable, which confirms that the assumptions for the queueing model are reasonable.

C. The Tagged VOQ

Under uniform 2-MMBP traffic, we model the tagged VOQ as a 2-MMBP/G/1 system. Given the mean inter-arrival time is known, this queuing system can be solved if its mean service time is known. We consider the mean service time is determined by two parameters: (1) the probability of the tagged XQ is full, and (2) the probability that the VOQ wins in the input scheduling phase. Let $\overline{t_s} = \overline{t_f} + \overline{t_{nf}}$ denote the mean service time for an HOL cell at the tagged VOQ, where $\overline{t_f}$ and $\overline{t_{nf}}$ given in Eq.(21)(22) are the geometry distributed mean times for the HOL cell sees the tagged XQ being full and non-full before it is forwarded to the tagged XQ, respectively,

$$\overline{t_f} = (1 - \pi_s e)^{-1} - 1 \tag{21}$$

$$\overline{t_{nf}} = \left(\sum_{i=0}^{N-1} \frac{1}{i+1} \xi^i (1-\xi)^{(N-1-i)} \right)^{-1} \tag{22}$$

where ξ is the probability that the tagged VOQ is valid for input arbitration in a time slot, which is given by

$$\begin{aligned} \xi = \quad & (1 - v_{00}(1-(1-\beta)\gamma) - v_{01}(1-\alpha\gamma)) \\ & (1 - (\pi_{(s,0)} + \pi_{(s,1)})e) \end{aligned} \tag{23}$$

where v_{00} and v_{01} denote the probabilities that at the end of the last time slot, the tagged VOQ is empty and the traffic is in the states OFF and ON, respectively.

This 2-MMBP/G/1 queueing system can be solved using an iterative computation method. First, we initiate $\overline{t_s}$ by some value, saying $1/\lambda$ for example. Next, we compute v_0 by solving this queueing system with the given $\overline{t_s}$. Then, we can renew the value of $\overline{t_s}$ using Eq.(21)(22). This process is repeated until t_s is converged. Given t_s, we can computed the mean cell delay in the tagged VOQ. Combining the mean cell delays in the tagged VOQ and the tagged XQ, the mean total cell delay in the CICQ switch is obtained.

IV. Numerical Results

A set of simulation experiments were conducted to verify the accuracy of the built queuing model. Both the simulators and the computing programs for the queuing model are implemented by Matlab and the simulations were run long enough

(a) 16-by-16 switches (b) 32-by-32 switches

Fig. 4. Mean cell delays in the buffered crossbar vs. mean loads, 2-MMBP bursty traffic.

(a) 16-by-16 switches (b) 32-by-32 switches

Fig. 5. Mean cell delays in the switches vs. mean loads, 2-MMBP bursty traffic with a mean burst length of 16 cells.

to collect sufficient samples for the statistic results with an 90% confidence interval. In Fig. 5, we present the results for the delay performances of the switches under i.i.d. uniform 2-MMBP bursty traffic with the mean burst length of 16, with the internal buffer sizes varying from 1 to 16 and two switch sizes of 16 and 32. From these figures, in general, we can see that the results from our queueing model are tightly close to the simulation results. The significant discrepancies can be observed only when the switches are under extremely heavy loads approaching to be saturated.

Specifically, we can see from the curves in these figures, increasing the internal buffer size will bring only slight improvements to the delay-throughput performance of a CICQ switch under uniform 2-MMBP traffic. Such an observation agree with that from the simulation studies in [1], [12]. This suggests that in general, an internal buffer size of one cell is enough for a CICQ switch under uniform traffic to perform comparatively well against another with a bigger internal buffer up to tens of cells.

REFERENCES

[1] R. Rojas-Cessa, E. Oki, and H. Chao, "On the combined input-crosspoint buffered switch with round-robin arbitration," *IEEE Transactions on Communications*, vol. 53, no. 11, pp. 1945–51, Nov. 2005.

[2] F. Abel, C. Minkenberg, R. Luijten, and et al., "A four-terabit packet switch supporting long round-trip times," *IEEE Micro*, vol. 23, no. 1, pp. 10–24, Jan.-Feb. 2003.

[3] K. Yoshigoe, K. Christensen, and A. Jacob, "The RR/RR CICQ switch: hardware design for 10-Gbps link speed," in *Proceedings of the IEEE International Performance, Computing, and Communications Conference*, Phoenix, AZ, USA, Apr. 2003, pp. 481–5.

[4] L. Mhamdi and M. Hamdi, "MCBF: a high-performance scheduling algorithm for buffered crossbar switches," *IEEE Communications Letters*, vol. 7, no. 9, pp. 451–3, 2003.

[5] R. B. Magill, C. C.E. Rohrs, and R. Stevenson, "Output-queued switch emulation by fabrics with limited memory," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 4, pp. 606–15, May 2003.

[6] I. Radusincivic, M. Pejanovic, and Z. Petrovic, "Impact of scheduling algorithms on performances of buffered crossbar switch fabrics," in *Proceedings of IEEE International Conference on Communications*, vol. 4, New York, NY, USA, Apr. 2002, pp. 2416–20.

[7] M. Lin and N. McKeown, "The throughput of a buffered crossbar switch," *IEEE Communications Letters*, vol. 9, no. 5, pp. 465–7, May 2005.

[8] S. Sun, S. He, and W. Gao, "Throughput analysis of a buffered crossbar switch with multiple input queues under burst traffic," *IEEE Communications Letters*, vol. 10, no. 4, pp. 305–7, Apr. 2006.

[9] B. Zhou and M. Atiquzzaman, "Performance of ATM switch fabric using cross-point buffers," *Computer Communications*, vol. 20, no. 13, pp. 1146–59, Dec. 1997.

[10] Q. Duan, X. Li, and L. Zhang, "Network calculus-based delay analysis for qos provision in buffered crossbar switch," *Journal of Interconnection Networks*, vol. 4, no. 2, pp. 131–46, June 2003.

[11] G. Nong, M. Hamdi, and J. Muppala, "Performance evaluation of multiple input-queued ATM switches with PIM scheduling under bursty traffic," *IEEE Transactions on Communications*, vol. 49, no. 8, pp. 1329–33, Aug. 2001.

[12] J. Luo, Y. Lee, and J. Wu;, "DRR: A fast high-throughput scheduling algorithm for combined input crosspoint-queued cicq switches," in *Proc. of 13th IEEE Int. Symp. on Modeling, Analysis, and Simulation of Computer and Tele. Systems*, Sept. 2005, pp. 329–332.