PERFORMANCE ANALYSIS OF SPEEDED-UP HIGH-SPEED PACKET SWITCHES

Aniruddha S. Diwan[1], Roch A. Guérin[2], and Kumar N. Sivarajan[1] *

[1] *Indian Institute of Science*
Electrical Comm. Engg. Dept.
Information & Acoustics Bldg.
Bangalore 560 012, India
{diwan,kumar}@ece.iisc.ernet.in

[2] *University of Pennsylvania*
Dept. Elec. Eng., Rm. 367 GRW
200 South 33rd Street
Philadelphia, PA 19104
guerin@ee.upenn.edu

Abstract In this paper, we study the performance of high-speed packet switches, where the switch fabric operates at a slightly higher speed than the links, i.e., a speeded-up switch. As link speeds keep increasing, the speedup of N (number of links) needed for pure output queueing (which is the best) becomes a significant technical challenge. This is one of the main reasons for the renewed interest in *moderately speeded-up switch fabrics*. The aim of this paper is to highlight the result that only a moderate speed-up factor (less than two) is sufficient to achieve full input link utilization. In particular, we emphasize that this holds, even without relying on a central switch controller making intelligent decisions on which packets to schedule through the switch. As shown in recent works, i.e., [10, 12, 8, 2, 11], there are clearly benefits to using intelligent controllers, but they do come at a cost. Instead, in this paper we focus on what can be achieved by relying *simply* on switch speedup. We do so by means of analysis and simulations for average queue length in switches with speedup. We also present simulation results on delay performance.

Keywords: Packet Switch, Speedup Factor, HOL Blocking, Matrix Analytic

1. BASIC SWITCH ARCHITECTURE

Consider the packet switch of figure 1. This switch has N input ports and N output ports and employs a combination of input and output queueing. We assume a *nonblocking* switch fabric so that all contentions inside the switch for packets destined to *different* output ports are avoided. Note that contention remains inevitable among packets addressed to the *same* output port. Fixed-length packets, or cells, arrive at the input ports of the packet switch. This cell structure is only internal to the switch, so that the links could carry variable size packets. However, in the rest of the paper, we focus on the case of fixed-size packets. Each packet

* A part of this work was done when R. A. Guérin and K. N. Sivarajan were at IBM T. J. Watson Res. Ctr., NY.

D. H. K. Tsang et al. (eds.), *Broadband Communications*

contains an identifier that indicates which output (we limit ourselves to the case of unicast flows) j, $1 \leq j \leq N$, it is destined for. In this paper, we are concerned with the performance analysis of such a switch fabric when it is *speeded-up*. That is, the switch fabric runs at a speed greater than that of the input and output links. For simplicity, we assume in our analysis that all input and output links run at the same speed.

Another important characteristic of the switch we consider is that each input makes independent decisions on which packet to send. As a result, the queueing structure at the inputs is a simple FIFO queue. This is in contrast to the switch fabrics of [10, 12, 8, 2, 11], which assume that packets are sorted according to their destination, and possibly priority, at the inputs, so that a central controller can make a selection based on the best possible combination of packets to send through the switch.

The operation of the simple switch fabric which we consider assumes, therefore, independent transmission attempts from all inputs. The basic packet transmission time through the switch fabric is called a *switch-slot*, and switch-slot boundaries at all the links are synchronized. For our analysis, we assume that packet arrivals on all N links are statistically identical, and packet destinations are independent from packet to packet and uniformly distributed across all N outputs. We denote as a *link-slot* the time it takes for a packet to arrive on the link. We also assume that the statistics of packet arrivals on all input links are identical. Because of potential contentions, it may take several switch slots to transfer a packet through the switch, as only packets with distinct destinations can be transferred within a given switch-slot. However, the impact on the inputs depends on the relation between switch-slots and link-slots, or in other words on the speedup of the switch. In switches with a speedup less than N, some queueing will occur on the inputs, and in our analysis we assume infinite buffers on inputs (and outputs), so that no packets are lost. In cases when packets from multiple inputs contend for the same destination in a switch-slot, one packet is chosen according to a *contention resolution policy*.

2. SWITCH THROUGHPUT

Let us denote the transmission time of a packet on any of the links by $1/R_l$ and the transmission time of a packet inside the switch by $1/R_s$. Thus R_s and R_l are the transmission rates (in packets per second) of the switch and input links, respectively. When the switch fabric is speeded up, $R_l < R_s$, and the ratio R_s/R_l is termed the *speed-up factor*. Let us denote the expected number of packet arrivals on each input link per link-slot by p. We term p the *input link utilization*. The expected number of packet arrivals on each input link per switch-slot is then $q = \frac{R_l}{R_s}p$. We say that a switch is *saturated* if it has a packet available for transmission through the switch fabric, in every switch-slot. We define the *saturation throughput* of a switch fabric as the expected number of packets per input link that is transmitted by the switch in each switch-slot, when all switch inputs are saturated. Clearly the saturation throughput will depend on the distribution of the packet destinations and

the speed-up factor. At one extreme, for a speed-up factor of 1, if all packets on all input links are destined to a single output port i, the saturation throughput is $1/N$. When packets on input link $i, 1 \leq i \leq N$, are always destined for output port i, the saturation throughput is 1. The saturation throughput has been widely studied in the case when, for each packet, each output port is equally likely to be the destination. As stated in section 1, this is the case we assume in this paper. In this case, the saturation throughput can also be interpreted as the *output link utilization when the input links are saturated.* (Also recall our assumption of independent and statistically identical input links.)

When the switch is not saturated we have to specify the process by which packets arrive on the links, in addition to the distribution of their destinations. The packets that cannot be transmitted through the switch immediately upon arrival, due to HOL blocking, are queued in an infinite buffer at each input link. We define the *stability throughput* as the maximum link utilization for which these input queues are stable. It has been implicitly assumed in much of the literature on input queueing that, as long as the arrival rate of packets on each input link is less than the saturation throughput of the switch, these input queues are stable [5]. However, that this is an assumption requiring proof has been recognized by Jacob and Kumar and their proof for the case of Bernoulli arrivals (successive packets have independent destinations) with identical rates on all input links may be found in [6].

Saturation-Stability Property: For a specified distribution of the packet destinations, we will say that an arrival process satisfies the *Saturation-Stability Property* if the input link queues are stable whenever the expected number of arrivals on each input link per switch-slot is less than the saturation throughput of the switch. We will assume that the packet arrival processes arising in our discussion satisfy the Saturation-Stability property.

We now ask: *For a given input link speed R_l what is the switch speed R_s required in order to achieve an input link utilization of unity, i.e., 100% (stability) throughput?* Note that, due to our assumptions, specifically the symmetry among the output links and the equality of the transmission speeds on the input and output links, the stability of the output queues is assured. Our answer is embodied in the following proposition.

Proposition 1 *The input link utilization of an input queueing switch fabric running at a rate $R_s > R_l/\gamma$, where R_l is the rate on the input links and γ is the saturation throughput of the switch with the same distribution for the packet destinations as that of the packets arriving on any input link, can be made arbitrarily close to one, provided the arrival process satisfies the Saturation-Stability Property.*

Proof: The expected number of arrivals on each input link *per switch-slot* is $q = \frac{R_l}{R_s}p$. If $R_s > R_l/\gamma, q < \gamma p < \gamma$ for all p, and the input queues are stable since the arrival process satisfies the Saturation-Stability Property by assumption. Thus the input link utilization p can be made arbitrarily close to 1 and the switch achieves a (stability) throughput of 100%.

3. QUEUE LENGTH ANALYSIS

In this section we analyze the queue lengths at various queueing points in a speeded-up switch fabric. The time it takes to transmit a packet fully on any link is called a *link-slot*. A link-slot consists of r switch-slots. The time it takes to transfer a packet from an input port to an output port, i.e., inside the switch fabric, is s switch-slots. In other words, the link speed is less by a factor of r/s than the switch fabric speed and we say that the speed-up factor of the switch fabric is r/s. Note that this type of model is applicable to any speed-up factor of the form r/s and is thus a general model. It can be seen that pure input-queued switches ($r = s = 1$) and pure output-queued switches ($r = N, s = 1$) are also special cases of the above model.

3.1 PACKET ARRIVAL PROCESS

We now describe the packet arrival process which is considered for analyzing the speeded-up switch fabric. This arrival process is a suitable adaptation of the uniform i.i.d. Bernoulli arrival process. Further, for analytical simplicity, the arrival process considered here is over synchronized switch-slots. Such a switch-slot arrival process leads to the Markov chain of figure 2. An ON switch-slot corresponds to an arrival. An OFF switch-slot corresponds to no arrival. The state transition probabilities are given in figure 2. The destination output port of a packet is chosen uniformly randomly out of N output ports. The packet arrival rate, λ, in packets per switch-slot can be obtained by solving for the steady-state vector of this Markov chain and we get,

$$\lambda = \frac{p}{1 + (r-1)p} \tag{1}$$

and the rate of packet arrivals in packets per link-slot is $r\lambda$.

3.2 ASYMPTOTIC ANALYSIS OF SWITCH PERFORMANCE

The packet switch with the input and output queues forms a complex queueing network. It seems that the exact queueing analysis of this complex queueing network is intractable for finite N. In order to get a handle on the problem, we follow the approach outlined in [7], and try to analyze the different parts of the network separately. For this, consider the travelogue of a tagged packet from the moment it arrived at an input queue to the moment it is transmitted fully on its destination output link. The tagged packet encounters queueing at three points in the network. It is first queued in the input queue where it has arrived. The packets in each input queue are transfered on a FIFO (first in first out) basis. When all the packets in front of this tagged packet in its input queue are transmitted across the switch, the tagged packet enters the HOL position of this input queue. At this point, there may be packets in the HOL position of other input queues whose destination is the same as that of the tagged packet. There may also be a packet which is already

in the transfer process to the destination of the tagged packet, in which case the destination is busy. The switch can start transferring only one of the former type of packets in a switch slot to the destination and then *only if* the destination is free. This is called *HOL contention.* This is the second queueing point in the network. Note that the HOL packets destined for a given output form a contention (or virtual) queue corresponding to that output. Therefore the tagged packet is queued in the HOL position, i.e., in its contention queue. Which packet from a given contention queue is to be transferred to the corresponding output (if it is free) in the next s switch-slots is decided by a *contention resolution policy.* The third queueing occurs in the output queue because the tagged packet is transmitted on the output link only when all the packets that arrived before it are transmitted fully on the output link.

We analyze the queue-length of these three types of queues for infinite input and output queues, and for the asymptotic case ($N \to \infty$) in the following subsections. Contention at the HOL positions is resolved thus: If k HOL packets contend for a particular output in a switch-slot, one of the k packets is chosen uniformly randomly from those k packets. The other packets have to wait until that switch-slot in which the output becomes free and a new selection is made among the packets that are then waiting. The arrival process of section 3.1 is assumed to satisfy the saturation-stability property of section 2. For this, the arrival rate, λ, should be less than the saturation throughput of the switch. Observe that a switch with only input queues is exactly the same as a pure input-queued switch on a switch-slot basis. The saturation throughput is 0.586 for the uniform i.i.d. Bernoulli arrival model [7] and tends to 0.5 for bursty arrivals [9] over the switch-slots. Therefore as long as $\lambda < 0.5$, the switch input queues are stable and, due to the symmetry among the output links and the equality of the transmission speeds on the input and output links, the switch output queues are also stable.

3.2.1 Contention Queue Analysis. The destination output of a packet is selected uniformly randomly among the N outputs as mentioned before. The situation at the HOL positions of the input queues is similar to that at the HOL positions of a pure input-queued switch. That is, in both cases packets encounter the HOL blocking phenomenon. We assume that each output contention process tends to be independent asymptotically (as $N \to \infty$). This is indeed the case for a pure input-queued switch with uniform i.i.d. Bernoulli traffic [7]. With this assumption the contention queues can be analyzed separately.

We say that an input queue is unbacklogged in a given switch-slot, say mth, if and only if, either a packet transfer ended in the $(m-1)$th switch-slot or it was empty during the $(m-1)$th switch-slot. A packet occupying HOL position in mth switch-slot is said to be unbacklogged if the corresponding input queue is unbacklogged. Otherwise the packet is said to be backlogged. Now consider any one contention queue, say corresponding to output i. In a given switch-slot, say mth, the contention queue contains backlogged as well as unbacklogged packets. A packet whose transfer is in progress in the mth switch-slot is also considered as

a backlogged packet. We denote the number of backlogged packets by B_m^i and the number of unbacklogged packets by A_m^i. We assume that, as $N \to \infty$, the steady-state number of packets moving to the head of unbacklogged input queues in each switch-slot, and destined for output i, (A^i), becomes Poisson at rate, $\overline{F}/N = \rho_0$, where $\overline{F}$ is the mean steady-state number of new packets at the HOL positions. In other words, $\lim_{N\to\infty} \Pr\{A^i = k\} = e^{\rho_0}{\rho_0}^k/k!$, and $\rho_0 = \lambda$ below saturation. That this "Poisson process" assumption is correct, is substantiated by the simulated performance [4].

(B_m^i) can be thought of as the system queue length in mth switch-slot. The above Poisson process assumption and the form of B_m^i suggest that the contention queue can be modeled as a discrete-time $BP/D_s/1$ queue with random order of service. BP represents discrete-time batch-Poisson process. D_s represents deterministic service time which, in this case, is s switch-slots (the transmission time of a packet inside the switch). This queueing model can be analyzed to get the steady-state queue-length distribution in a random switch-slot and the delay distribution of the packets in the queue. Each output contention process is assumed to be independent as $N \to \infty$ and all the output contention processes are identical. Hence it is sufficient to analyze any one of them. Note that when the speed-up factor is of the form $r/1$, i.e., integer speed-up factor r, the contention queue is modeled as $BP/D_1/1$. This is the same queueing model as that of the contention queue in [7]. There this model is referred to as discrete $M/D/1$ queue. Thus the analysis of this discrete $M/D/1$ developed in [7] is directly applicable to the case of *integer* speed-up factors. The analysis of the contention queue in [7] is a special case of the analysis developed below.

Distribution of Queue-length in a Random Switch-Slot. We cannot model the queue-length of the contention queue *alone* by a DTMC. If we know how many switch-slots of service time have been completed for the current packet in transfer out of the deterministic s switch-slots of service time, we can model the queue-length by a DTMC. The switch status is T_0 at the beginning of a switch-slot if the switch is not busy at a switch-slot boundary. In this status the switch is ready to serve any packet in this switch-slot. If a packet arrives when the status is T_0 and the contention queue is empty, the packet starts service immediately from this switch-slot. If say, n packets are in the contention queue waiting for service, one of them starts service from this switch-slot according to the contention resolution policy if no packet arrives in this switch-slot. If a packet arrives in this switch-slot, one of the total $(n+1)$ packets starts service from this switch-slot. The status T_i, where $1 < i \leq s$, at the beginning of a switch-slot indicates that there is a packet transfer in progress and that first i switch-slots of service have been completed. A packet is eligible to get service in the same switch-slot in which it arrives.

We club the queue-length in a switch-slot and the switch-status of that switch-slot to form a state. This state can be modeled by a 2-D DTMC over the state space $\{(i,j) : i \geq 0, j \in \{T_0, T_1, \cdots, T_{s-1}\}\}$. Then queue-lengths in successive

switch-slots follow this 2-D DTMC. It is assumed that the number of arrivals in the contention queue during each switch-slot has Poisson probabilities. Therefore the state transition probability matrix takes the form,

$$\mathbf{P} = \begin{bmatrix} B_0 & B_1 & B_2 & B_3 & \cdots \\ C & A_1 & A_2 & A_3 & \cdots \\ \tilde{0} & A_0 & A_1 & A_2 & \cdots \\ \tilde{0} & \tilde{0} & A_0 & A_1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{2}$$

where B_0 is an 1×1 matrix, $B_i,\ i > 0$ are $1 \times s$ matrices, C is an $s \times 1$ matrix and $A_i,\ i \geq 0$ are $s \times s$ matrices. Note that the $\mathbf{P}$ matrix is a structured $M/G/1$-type of matrix. Please refer to [4] for calculation of the entries of the matrices. Algorithms to solve for the steady-state probabilities vector of such matrices by matrix analytic methods have been implemented in a software package called TELPACK [1]. We have used TELPACK to compute the steady-state probability vector of the $\mathbf{P}$ matrix. The steady-state probability vector of the $\mathbf{P}$ matrix is of the form, $\pi = [\pi^0 \pi^1 \pi^2 \cdots \pi^i \cdots]$, where $\pi^0 = [q_{0,T_1}]$ and, $\pi^i = [q_{i,T_0} q_{i,T_2} \cdots q_{i,T_s}],: i > 0$. q_{i,T_j} is the steady-state probability of the queue-length being i and the switch status being T_j in a random switch-slot. We now can compute the steady state probability vector of queue-length in a random switch-slot which is, $\tilde{l} = [l_0 l_1 l_2 \cdots l_i \cdots]$, where l_i is the probability that the queue-length is i in a random switch-slot. The l_is are given by, $l_0 = q_{0,T_0}$ and $l_i = \sum_{j=0}^{s} q_{i,T_j}$.

Distribution of Delay. Karol *et al* [7] have developed a simple numerical method for computing the delay distribution of a discrete-time $BP/D_1/1$ queue, with packets served in random order (Appendix III of [7]). In this section we extend this scheme to the case of a discrete-time $BP/D_s/1$ queue, with packets served in random order. The number of packet arrivals at the beginning of each switch-slot is Poisson distributed with rate λ and each packet requires s switch-slots of service time. We focus our attention on a particular "tagged" packet in the system, during a given switch-slot. Let p^m_{k,T_j} denote the probability, conditioned on there being a total of k packets in the system and the switch status being T_j during the given switch-slot, that the remaining delay is m switch-slots until the tagged packet completes service.
The p^m_{k,T_j} can be obtained by recursion on m as follows,

$$p^m_{1,T_0} = \begin{cases} 1: & m = s \\ 0: & m \neq s \end{cases} \tag{3}$$

$$p^s_{k,T_0} = \frac{1}{k}: \quad k \geq 1 \tag{4}$$

$$p^m_{k,T_0} = (k-1)p^s_{k,T_0} \cdot \sum_{j=0}^{\infty} p^{m-s}_{k-1+j,T_0} \cdot \frac{e^{-s\lambda}(s\lambda)^j}{j!} \quad : \begin{cases} m = x \cdot s \\ x > 1, \\ k > 1 \end{cases} \tag{5}$$

Averaging over k, the delay D has probabilities as follows:
For $m = s \cdot i$ such that $i \geq 1$ and i integer,

$$\Pr\{D = m\} = \sum_{k=1}^{\infty} p^m_{k,T_0} \cdot \sum_{n=0}^{\infty} q_{n,T_0} \cdot \frac{e^{\lambda}\lambda^{k-n-1}}{(k-n-1)!} \tag{6}$$

For $m = s \cdot i + j$ such that $i \geq 1$ and $1 \leq j \leq (s-1)$ and i, j integers,

$$\Pr\{D = m\} = \sum_{k=1}^{\infty} p^{m-j}_{k,T_j} \cdot \sum_{n=1}^{\infty} q_{n,T_j} \cdot \frac{e^{(s-j+1)\lambda}((s-j+1)\lambda)^{k-n}}{(k-n)!} \tag{7}$$

For all the remaining m, the delay probability is zero. The q_{i,T_j} are obtained in section 3.2.1. With (6)–(7) we get the delay distribution of a packet in the system. The moments of the delay distribution are determined numerically from the delay probabilities in (6)–(7).

3.2.2 Input Queue Analysis. As $N \to \infty$, successive packets in an input queue i experience the same service time distribution because their destination addresses are independent and are equiprobable. The number of switch-slots elapsed between the entry of a tagged packet in the HOL position of its input queue and the exit of that packet from the input queue, is equal to the delay of the packet in the contention queue. It is as if the tagged packet is served for that many switch-slots in its input queue. In other words, the service time distribution of a packet in an input queue is the delay distribution of a packet in the contention queue. With the arrival process of section 3.1 and the service time distribution of section 3.2.1, we can model the input queue as a $G/G/1$ queue. The transition matrix $\mathcal{P}$ of the 2-D DTMC has the form,

$$\mathcal{P} = \begin{bmatrix} B_0 & B_1 & B_2 & B_3 & \cdots \\ A_0 & A_1 & A_2 & A_3 & \cdots \\ \tilde{0} & A_0 & A_1 & A_2 & \cdots \\ \tilde{0} & \tilde{0} & A_0 & A_1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{8}$$

Please refer to [4] for calculation of the entries of the matrices. To get the steady-state probability vector of the $\mathcal{P}$ matrix, we need to solve the set of equations, $\tilde{\pi} = \tilde{\pi}\mathcal{P}$ and $\|\tilde{\pi}\| = 1$. where $\tilde{\pi}$ is a vector of the form, $\tilde{\pi} = [\pi^0\pi^1\pi^2 \cdots \pi^i \cdots]$ and π^i is a vector of the form, $\pi^i = [\pi_{i,0}\pi_{i,1}\pi_{i,3} \cdots \pi_{i,r-1}]$. $\pi_{i,j}$ is the steady-state probability of the state (i, j) of the 2-D DTMC. Note that the $\mathcal{P}$ matrix is a structured $M/G/1$-type of matrix. We use TELPACK [1] to compute the steady-state probability vector $\tilde{\pi}$ of our $\mathcal{P}$ matrix. The steady-state probability vector $\tilde{\pi}$ enables us to compute the

steady-state distribution of queue-length as seen by a departing packet. The steady-state queue-length vector is of the form, $\tilde{q} = [q_0 q_1 q_2 \cdots q_i \cdots]$ where, $q_i = \sum_{j=0}^{r-1} \pi_{i,j}$. The moments of the distribution of the queue-length as seen by a departing packet are then numerically computed using the above equations. The transition rates into and out of each state in a discrete-state stochastic process must be identical. By using the time-average interpretations for the steady-state queue-length probabilities, we see that the distribution of queue-length as seen by a departing packet is the same as the distribution of queue-length as seen by an arriving packet [13, pp. 387–388]. Note that in [13, pp. 387–388], the Markov chain is one-dimensional. In our case it is two-dimensional. But as we are interested in the steady-state probability vector of levels (i.e., queue-lengths), the arguments are applicable in this case also. We compare the results of the above analysis with the results of simulations in section 5. As it is not possible to compute the distribution of queue-length in a random slot, we cannot analyze the delay of a packet in the input queue. In section 5, we show simulation results for the delay of a packet in the input queue.

3.2.3 Output Queue Analysis. We have assumed that, as $N \to \infty$, all contention queue processes are identical and independent. It then follows that all output queue systems are also identical and independent. It is sufficient to analyze only one output queue. It is difficult to characterize the departure process of a contention queue due to its general queueing model structure. The exact arrival process at the output queue is then unknown. We use an approximate ON-OFF type of arrival model to analyze the output queue. An exact analysis like that of input queues may be difficult in this case. A packet arrives in the output queue after s switch-slots in a busy period. This is because the contention queue service time is s switch-slots. Therefore we model the arrival process as follows: A packet arrives at output queue i in a switch-slot only if the switch status is T_{s-1} at the beginning of the switch-slot (section 3.2.1). There are always $(s-1)$ idle switch-slots (corresponding to the switch status T_0 to T_{s-2}) before an arrival in the output queue. The group of all these switch-slots is called a *busy cluster*. In the busy period of the contention queue, packets depart regularly at the end of the busy clusters. When the contention queue is empty in a switch-slot, no packet arrives in output queue i in that switch-slot. We construct the ON-OFF arrival model such that, $E[B]$, the mean busy period of the contention queue is equal to s times the mean number of busy clusters. The mean OFF period is equal to, $E[I]$, the mean idle period of contention queue i. The ON-OFF Markov process is depicted in figure 3. The ON state and the series of associated $OFF_j, 1 \leq j \leq s-1$ form a busy cluster. It is easy to see that $p = 1 - s/E[B]$ and $p = 1 - 1/E[I]$ since the mean busy period of the contention queue is equal to s times the mean number of busy clusters. The mean OFF period is equal to the mean idle period of contention queue i. Note that for *integer* speed-up factors $s = 1$. There are no idle slots in a busy cluster and a busy

cluster consists of only ON slot. The ON-OFF Markov chain of figure 3 reduces to the simple ON-OFF Markov chain of figure 4.

The service time of the output queue is r switch-slots and is deterministic (the transmission time of a packet on the output link). We analyze this output queue model. The queue-length in a random switch-slot cannot be modeled by a DTMC. We look at the departure-switch-slots. The output queue-length as seen by a departing packet in a switch-slot together with the ON-OFF chain status in that switch-slot form a 2-D DTMC over the state space $\{(i,j) : i \geq 0, j \in \{ON, OFF_1, OFF_2, \cdots, OFF_{s-1}, OFF\}\}$ where i is the queue-length in a departure-switch-slot and j is the state of the ON-OFF process. Let $ON \equiv s$ and $OFF \equiv 0$ and $OFF_i \equiv i$. We say that i is the level of the 2-D DTMC and j is the phase of the 2-D DTMC. Let $m = \lceil r/s \rceil$. The state transition matrix is of the form,

$$\mathcal{P} = \begin{bmatrix} B_0 & B_1 & B_2 & \cdots & B_m & \tilde{0} & \tilde{0} & \tilde{0} & \cdots \\ C & A_1 & A_2 & \cdots & A_m & \tilde{0} & \tilde{0} & \tilde{0} & \cdots \\ \tilde{0} & A_0 & A_1 & \cdots & A_{m-1} & A_m & \tilde{0} & \tilde{0} & \cdots \\ \tilde{0} & \tilde{0} & A_0 & \cdots & A_{m-2} & A_{m-1} & A_m & \tilde{0} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{9}$$

Note that the $\mathcal{P}$ matrix is a structured $M/G/1$-type matrix. Please refer to [4] for calculation of the entries of the matrices. The steady-state probability vector of the $\mathcal{P}$ matrix is computed by using TELPACK. The steady-state probability vector is of the form, $\pi = [\pi^0 \pi^1 \pi^2 \cdots \pi^i \cdots]$ where, $\pi^i = [\pi_{i,0} \pi_{i,1} \cdots \pi_{i,s}]$ The steady-state probability vector of queue-lengths as seen by a departing packet is of the form, $\tilde{q} = [q_0 q_1 q_2 \cdots q_i \cdots]$ where $q_i = \pi_{i,0} + \pi_{i,1} + \cdots + \pi_{i,s}$. The results of the above analysis are compared with the results of simulations in the next section of this chapter. We are unable to analyze the delay performance of this output queue. Nevertheless, in section 5 of this chapter, we show the results of simulations for the delay of a packet in the output queue.

4. MEAN DELAY SIMULATIONS

In this section we study the simulation results of the mean delay of the speeded-up packet switches for the arrival process of section 3.1. We have considered a 64×64 nonblocking speeded-up switch fabric. We plot the total average delay of a packet in the switch obtained through simulations in figure 5. The average delays for the cases of pure input-queued and pure-output queued switches are also plotted in figure 5. It can be seen from the plot that the delay curves for speed-up factors 2 and 3 are close to the delay curve of the pure output-queued switch. So even a moderate speed-up of 2 quickly overcomes the HOL blocking phenomenon of pure input-queueing. To illustarte this, we have plotted the input and output queue delays separately for a speed-up factor of 2 in figure 6. We also consider bursty traffic for delay simulations with the following model. A burst of successive packets

is destined to the same output port but the destinations of each burst are chosen uniformly from among the N output ports. The burst sizes have the following *modified geometric distribution*: If the mean burst size is b, the size of a burst is $1 + B$ where B is a geometric r. v. with mean $b - 1$. The mean queueing delays achieved by output queueing and input queueing with a speedup factor of 2 are shown in Fig 7. Fig 8 shows the mean queueing delays for an average burst size $b = 50$. This emphasizes the fact that speed-up of 2 is sufficient even for bursty traffic.

5. QUEUE LENGTH ANALYSIS VS SIMULATION

Results for three speed-up factors are studied: $1.5(= 3/2)$, $2(= 2/1)$ and $3(= 3/1)$. The analysis of the input and output queues gives the mean queue-length as seen by a departing packet. We obtain the same data from simuation for comparison. We plot the mean input queue-length obtained by analysis and simulation in figure 9. It can be seen that the results are in good agreement. Note that even though the input queue analysis is exact, it is asymptotic, and thus we compare it with simulations to judge the effect of finite switch size (64×64). The mean output queue-length as seen by a departure is plotted in figure 10. The reason for a slight discrepancy here is the approximation of the arrival process to the output queue by the ON-OFF process. Observe that the discrepancy is more pronounced for the speed-up factor 1.5. For speed-ups of 2 and 3 the results of both cases are in good agreement. Thus the approximation works better for higher speed-up factors. Overall, the asymptotic analysis results conform well with the simulation results.

6. CONCLUSIONS

In this paper, we have used matrix-geometric techniques to investigate the impact of speedup on the performance of combined input and output queueing packet switches. The important observation is that only a moderate speedup factor (about two) is necessary to approach not only the throughput but also the delay performance of pure output-queued switches. Moreover, these results hold for random scheduling of the head-of line packets; no complex scheduling and/or matching algorithms are required within the switch. This is practically significant for the design and operation of high speed switch fabrics, especially with the continued increase in link speeds and switch sizes.

References

[1] A. Agrawal and N. Oguz.
Telpack Software, http://www.cstp.umkc.edu/org/tn/telpack/home.html, ftp://ftp.cstp.umkc.edu/telpack/software/.

[2] A. Charny, P. Krishna, N. Patel and R. Simcoe. Algorithms for providing bandwidth and delay guarantees in input-buffered crossbars with speedup. Proceed-

ings of the Sixth International Workshop on Quality of Service (IWQoS'98), 1998.

[3] W. E. Denzel, A. P. J. Engbersen, and I. Illiadis. A flexible shared buffer switch for ATM at Gb/s rates. *Computer Networks and ISDN Systems*, 27(4):611-624, Jan 1995.

[4] A. S. Diwan. *Performance analysis of speeded-up high-speed packet switches.* MS thesis, Indian Institute of Science, Bangalore (INDIA), Feb 1998.

[5] I. Illiadis, W. E. Denzel. Analysis of packet switches with input and output queueing. *IEEE Transactions on Communications*, 41(5):731-740, May 1993.

[6] L. Jacob. *Performance analysis of scheduling strategies for switching and multiplexing of multiclass variable bit rate traffic in an ATM network.* PhD thesis, Indian Institute of Science, Bangalore (INDIA), Dec 1992.

[7] M. Karol, M. Hluchyj, and S. Morgan. Input versus output queueing on a space division switch. *IEEE Transactions on Communications*, 35(12):1347–1356, Dec 1987.

[8] P. Krishna, N. S. Patel, A. Charny and R. Simcoe. On the speedup required for work-conserving crossbar switches. Proceedings of the Sixth International Workshop on Quality of Service (IWQoS'98), 1998.

[9] S. Q. Li. Performance of a non-blocking space-division packet switch with correlated input traffic. In *IEEE GLOBECOM*, pages 1754–1763, 1989.

[10] N. McKeown, V. Anantharam, and J. Walrand. Achieving 100% throughput in an input-queued switch. In *IEEE INFOCOM, San Francisco, CA*, pages 296–302, March 1996.

[11] B. Prabhakar and N. McKeown. On the speedup required for combined input and output queued switching. Computer Systems Lab, Technical Report CSL-TR-97-738, Stanford University.

[12] I. Stoica and H. Zhang. Exact emulation of an output queueing switch by a combined input output queueing switch. Proceedings of the Sixth International Workshop on Quality of Service (IWQoS'98), 1998.

[13] R. W. Wolff. *Stochastic modeling and the theory of queues.* Prentice Hall, Englewood Cliffs, New Jersey, 1989.

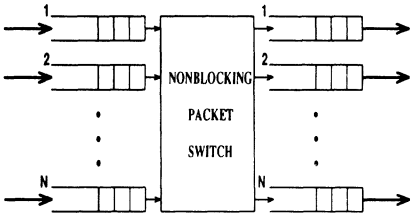

Figure 1 An $N \times N$ speeded-up packet switch with input and output queues.

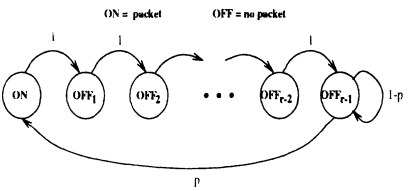

Figure 2 Markov chain modeling the arrival process in the input queues of the packet switch.

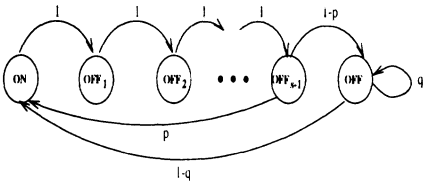

Figure 3 Approximate model of the arrival process in the output queues for a rational speed-up factor r/s.

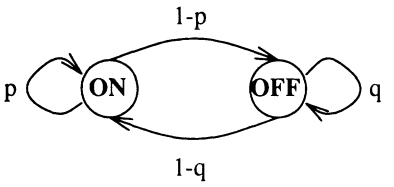

Figure 4 Approximate model of the arrival process in the output queues for an integer speed-up factor.

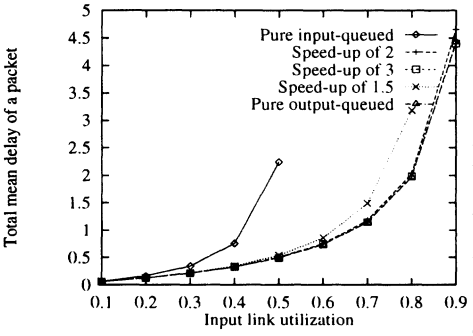

Figure 5 Mean total packet delay results for switches with speed-up factors of 1.5, 2 and 3 and delay results of pure input-queued and pure output-queued switches.

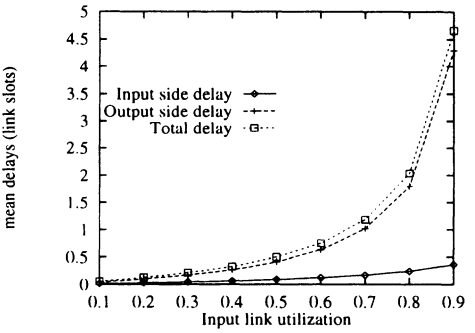

Figure 6 Mean delay in input queue and the mean delay in output queue and total mean delay of a packet for a speed-up factor of 2.

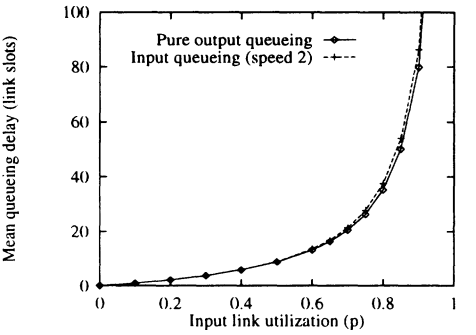

Figure 7 Delay performance of a switch with speed-up of 2 and delay performance of pure output-queued switch. mean burst size $b = 10$.

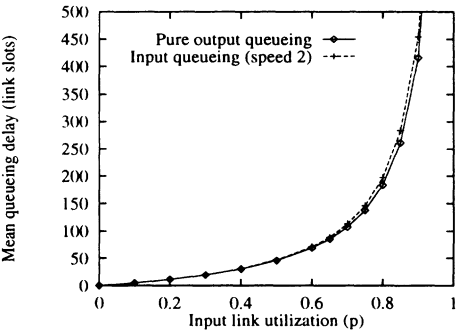

Figure 8 Delay performance of a switch with speed-up of 2 and delay performance of pure output-queued switch. mean burst size $b = 50$.

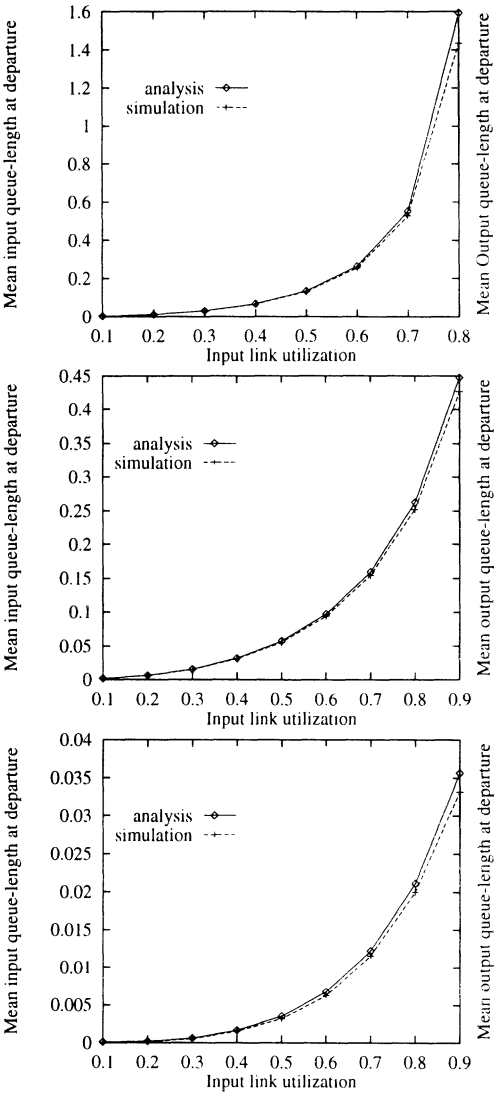

Figure 9 The mean input queue-length as seen by a departure obtained by analysis is compared to that obtained by simulation, (top) speed-up of 1.5, (middle) speed-up of 2, (bottom) speed-up of 3. The input link utilization is $r\lambda$. see (1). The arrival process is of section 3.1.

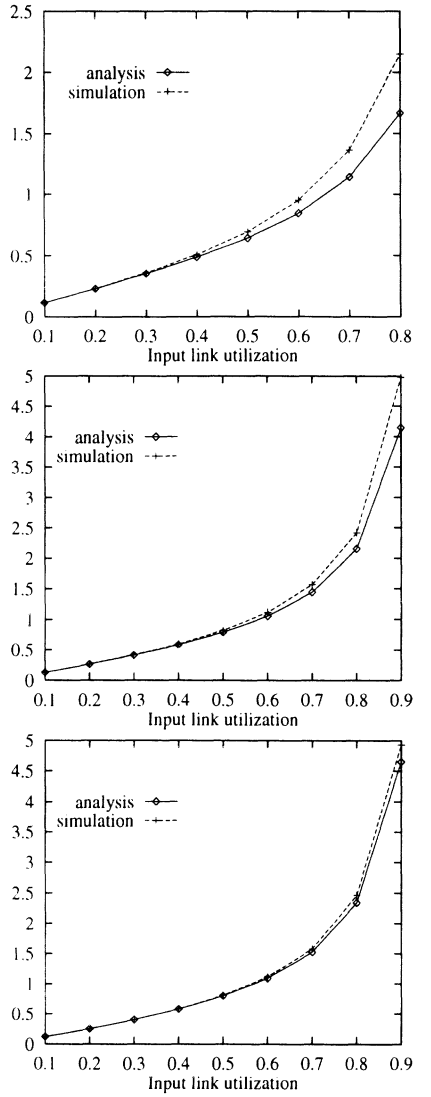

Figure 10 The mean output queue-length as seen by a departure obtained by analysis is compared to that obtained by simulation, (top) speed-up of 1.5, (middle) speed-up of 2, (bottom) speed-up of 3. The input link utilization is $r\lambda$. see (1). The arrival process is of section 3.1