RZ 3650 (# 99660) 02/20/2006
Computer Science 12 pages

Research Report

Performance of a Speculative Transmission Scheme for Arbitration Latency Reduction

Ilias Iliadis and Cyriel Minkenberg

IBM Research GmbH
Zurich Research Laboratory
8803 Rüschlikon
Switzerland
{ili, sil}@zurich.ibm.com

Performance of a Speculative Transmission Scheme for Arbitration Latency Reduction

Ilias Iliadis, *Senior Member, IEEE,* and Cyriel Minkenberg

***Abstract*— Low latency is a critical requirement in some switching applications, specifically in parallel computer interconnection networks. The minimum latency in switches with centralized arbitration comprises two components, namely, the control-path latency and the data-path latency, which in a practical high-capacity, distributed switch implementation can be far greater than the cell duration. We introduce a *speculative transmission* scheme to significantly reduce the average control-path latency by allowing cells to proceed without waiting for a grant, under certain conditions. It operates in conjunction with a traditional centralized matching algorithm to achieve a high maximum utilization and incorporates a reliable delivery mechanism to deal with failed speculations. An analytical model is presented to investigate the efficiency of the speculative transmission scheme employed in a non-blocking $N \times NR$ input-queued crossbar switch with R receivers per output. Using this model, performance measures such being the mean delay and the rate of successful speculative transmissions are derived. The results demonstrate that the control-path latency can be almost entirely eliminated for loads up to 50%. Our simulations confirm the analytical results.**

Index Terms—Electrooptic switches, packet switching, arbiters, scheduling, modeling.

I. Introduction

A key component of massively parallel computing systems is the interconnection network (ICTN). To achieve a good system balance between computation and communication, the ICTN must provide low latency, high bandwidth, low error rates, and scalability to high node counts (thousands), with low latency being the most important requirement.

Although optics hold a strong promise towards fulfilling these requirements, a number of technical and economic challenges remain. Corning Inc. and IBM are jointly developing a demonstrator system to solve the technical issues and map a path towards commercialization. For background information on this project—the Optical Shared MemOry Supercomputer Interconnect System (OSMOSIS)—and for a detailed description of the architecture we refer the reader to [1].

A. OSMOSIS architecture

The routing fabric of OSMOSIS (Fig. 1) is entirely optical and has no buffering capability. It operates in a synchronous, time-slotted fashion with fixed-size packets (cells). The switching function is implemented using fast semiconductor optical amplifiers (SOAs) in a broadcast-and-select (B&S) structure using a combination of eight-way space- and eight-way wavelength-division multiplexing, thus providing bidirectional connectivity for 64 nodes. Electronic buffers store cells at the ingress of the switch, resulting in an input-queued (IQ) architecture. To prevent head-of-line (HOL) blocking, the input queues are organized as virtual output queues (VOQs).

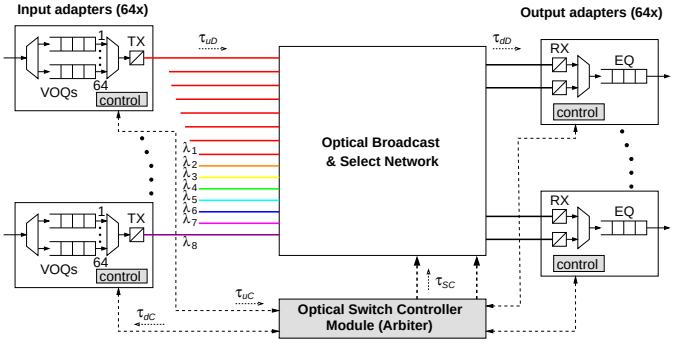


Fig. 1. High-level OSMOSIS architecture.

The B&S switch fabric structure is the optical equivalent of an electronic crossbar switch. To resolve crossbar input and output contention, central arbitration is required, which is also electronic. In addition to a low minimum latency, OSMOSIS must also be able to achieve a high maximum throughput. Therefore, the arbiter must implement an appropriate bipartite graph matching algorithm able to sustain close to 100% throughput. Using appropriate deep pipelining techniques [2], [3], it is possible to obtain maximal matchings even for switches with many ports and short cells.

The input adapters receive cells from the incoming links and store them according to their destinations in the VOQs. Upon cell arrival, a request is issued to the arbiter via the control channel (CC), which is operated in a slotted fashion with the same time slot duration as that of the data path. When the round-trip time (RTT, expressed in time slots) is greater than 1, both the data and the control path must be operated in a pipelined fashion to maintain 100% utilization without increasing the cell size. This implies that multiple cells and request/grants may be in flight on the data and control paths, respectively. To cope with a long RTT without loss of performance, we employ an incremental VOQ state update protocol that allows deep pipelining of requests and grants without a performance penalty [3]. A special OSMOSIS feature is the presence of two receivers per output, which allows up to two cells to be delivered to the same output in one time slot. This is achieved by using an asymmetric 64 × 128 B&S structure, with two receivers per output adapter.

The authors are with IBM Research GmbH, Zurich Research Laboratory, Säumerstrasse 4, CH-8803 Rüschlikon, Switzerland.

This research is supported in part by the University of California under subcontract number B527064.

B. Control-path latency

This classic centrally-arbitrated, crossbar-based IQ architecture, however, incurs a latency penalty: The *minimum* latency of a cell in the absence of contention comprises two components, namely, the control-path latency (upstream: τ_{uC}, downstream: τ_{dC}) and the data-path latency ($\tau_{\mathrm{uD}}, \tau_{\mathrm{dD}}$). The former consists of the latency from the issuance of a request to the receipt of the corresponding grant, whereas the latter consists of the transit latency from the input adapter to the output adapter. The switch-configuration-path latency (τ_{SC}) represents the latency from the issuance of a configuration command by the arbiter until the SOAs are switched accordingly. These latencies comprise serialization and deserialization (serdes) delays, propagation delays (time of flight) on the physical medium, and processing delays in the switch and the adapter. The processing delays typically include header parsing delays, routing delays, arbitration delays, pipelining delays, etc. In an output-queued (OQ) switch, on the other hand, the minimum latency comprises only the data-path latency. The difference is that in an IQ switch, a newly arriving cell must first request permission to proceed and then wait for a grant, whereas in an OQ switch, a cell can immediately proceed to its output when there is no contention.

The physical implementation and packaging aspects of OSMOSIS (and high-capacity switches in general) have important consequences [4] that imply that the above latencies are significant. In the OSMOSIS demonstrator, we estimate the involved data- and control-path RTTs to be around 600 ns, resulting in a minimum cell latency of approx. 1.2 μs [5], which is much larger than the cell duration (51.2 ns). This already exceeds our latency target of 1 μs without taking into account the latencies of the driver software stack and the host channel adapter.

Parallel ICTNs often operate at low utilization, or are subjected to highly orchestrated (by the programmer or compiler) traffic patterns. Under such conditions, the mean latency is dominated by the intrinsic control- and data-path latencies rather than by queuing delays. Hence, optimizing latency for such cases improves overall system performance.

The main contribution of this work is a hybrid crossbar arbitration scheme that combines *arbitrated* and *speculative* modes of operation, such that at low utilization most cells can proceed speculatively without waiting for a grant, thus achieving up to 50% latency reduction. Moreover, the arbitration mode ensures high utilization without excessive collisions of speculative cells in the B&S switch fabric.

First, we review related work in Sec. II. Section III specifies the operational details of speculation and how it interacts with traditional crossbar arbitration. We address all ensuing issues, such as collisions, retransmissions, as well as out-of-order and duplicate deliveries. In Sec. IV, an analytical performance model of the proposed scheme is developed, and a closed-form expression for the average delay through the switch is derived. Section V presents numerical results demonstrating the efficiency of the proposed scheme. It also presents simulation results, which confirm the validity of the analytical model developed. Finally, we conclude in Sec. VI.

II. Related Work

There are alternative ways to avoid the arbitration latency issue described above. The main options are: (1) Bring the arbiter closer to the adapters, (2) use provisioning (circuit switching), (3) use a buffered switch core, or (4) eliminate the arbiter altogether.

Although one can attempt to locate the arbiter as close to the adapters as possible, a certain distance determined by the system packaging limitations and requirements will remain [4]. Although the RTT can be minimized, the fundamental problem of non-negligible RTTs remains valid.

One can also do without cell-level allocation and rely on provisioning to resolve contention. Of course, this approach has several well-known drawbacks, such as a lack of flexibility, inefficient use of resources, and long set-up times when a new connection is needed, which make this approach unattractive for parallel computer interconnects.

An alternative approach is to provide buffers in the switch core and employ some form of link-level flow control (e.g., credits) to manage them. As long as an adapter has credits, it can send immediately without having to go through a centralized arbitration process. However, as optical buffering technology is currently neither practically nor economically feasible and the key objective of OSMOSIS is to demonstrate the use of optics, this is not an option.

The last alternative is the load-balanced Birkhoff–von–Neumann switch [6], which eliminates the arbiter entirely. It consists of a distribution and a routing stage, with a set of buffers at the inputs of the second stage. Both stages are reconfigured periodically according to a sequence of N permutation matrices. The first stage *uniformizes* the traffic regardless of destination, and the second stage performs the actual switching. Its main advantage is that, despite being crossbar-based, no centralized arbiter is required. Although this architecture has been shown to have 100% throughput under a technical condition on the traffic, it incurs a worst-case latency penalty of N time slots: if a cell arrives at an empty VOQ just after the VOQ had a service opportunity, it has to wait for exactly N time slots for the next opportunity. The mean expected latency penalty is $N/2$ time slots plus a minimum transit latency intrinsically added by the second stage. Moreover, missequencing can occur. This approach results in overall lower latency if the total architecture-induced latency penalty can be expected to be less than the control-path latency in a traditional IQ switch. In the OSMOSIS system this is not the case, hence we choose the traditional architecture.

III. Speculative Transmission

Our objective is to eliminate the control-path latency in the absence of contention. To this end, we introduce a *speculative transmission* (STX) scheme. The principle behind STX is related to that of the original ALOHA and Ethernet protocols: Senders compete for a resource without prior arbitration. If there is a collision, the losing sender(s) must retry their data transmissions in a different time slot.

However, the efficiency of ALOHA-like protocols is very poor (18.4% for pure ALOHA and 36.8% for slotted ALOHA

[7, Sec. 4.2.1]) because under heavy load many collisions occur, reducing the effective throughput. Therefore, we propose a novel method to combine arbitrated and speculative (non-arbitrated) transmissions in a crossbar switch. The objective is to achieve reduced latency at low utilization owing to the speculative mode of operation and achieve high maximum throughput owing to the arbitrated mode of operation.

We consider the presence of multiple (R) receivers per output port, allowing up to R cells to arrive simultaneously. Although in OSMOSIS $R = 2$, we are interested in the general case with $1 \leq R \leq N$ here. We exploit this feature to improve the STX success rate. The first receiver is for either an arbitrated or a speculative cell. The extra $R-1$ receivers can accommodate additional speculative cells. Correspondingly, the STX arbitration can acknowledge multiple STX requests per output per time slot.

The following rules govern the design of the STX scheme:

- $\mathcal{R}1$ Upon cell arrival, a request for arbitration is issued to the central arbiter.
- $\mathcal{R}2$ An adapter is eligible to perform an STX in a given time slot if it has no grant for an arbitrated transmission in that time slot.
- $\mathcal{R}3$ When multiple cells collide, R cells proceed and the remaining cells are dropped. If the number of colliding cells is smaller than or equal to R, all cells proceed.
- $\mathcal{R}4$ When speculative cells collide with an arbitrated one, the arbitrated cell always wins, allowing up to $R-1$ speculative cells to proceed.
- $\mathcal{R}5$ Every cell may be speculatively transmitted at most once.
- $\mathcal{R}6$ Every speculative cell remains stored in its input adapter until it is either acknowledged as a successful STX or receives an arbitrated grant.
- $\mathcal{R}7$ The arbiter acknowledges every successful speculative cell to the sending input. However, when a grant arrives before the acknowledgment (ACK), a cell may be transmitted a second time. These are called *duplicate* cells as opposed to the *pure* cells, which are transmitted through grants but are not duplicate.
- $\mathcal{R}8$ Every grant is either *regular*, *spurious*, or *wasted*. It is regular if it is used by the cell that initiated it. A grant corresponding to a successfully speculatively transmitted and acknowledged cell is spurious when used by another cell residing in the same VOQ, resulting in a *spurious transmission*, or wasted if the VOQ is empty. If it is wasted, the slot can be used for a speculative transmission.

In the remainder of this section, we will explain the rationale behind these rules and elaborate on them.

A. STX policy

According to $\mathcal{R}2$, an adapter performs an STX in a given time slot t_0 if it receives no grant at t_0 and it has an eligible cell. If it receives a grant, it performs the corresponding arbitrated transmission. $\mathcal{R}2$ allows the STX scheme to operate in conjunction with regular arbitrated transmissions, with the arbitrated taking precedence over the speculative ones. Accordingly, we distinguish between arbitrated and speculative cells.

When an adapter is eligible to perform an STX, it selects a non-empty VOQ according to a specific STX policy, dequeues its HOL cell and stores it in a retransmission buffer, marks the cell as speculative, and sends it to the crossbar. On the control path, it sends a corresponding speculative request (SRQ) indicating that a cell has been sent speculatively to the selected output. Both the cell and the request comprise a unique sequence number to enable reliable, in-order delivery.

The *STX policy* defines which VOQ the adapter selects when it is eligible to perform an STX. This policy can employ, e.g., a random, oldest cell first (OCF), or youngest cell first selection. In the remainder of the paper we consider the OCF policy. It chooses the cell that has been waiting longest at the input adapter for an STX opportunity.

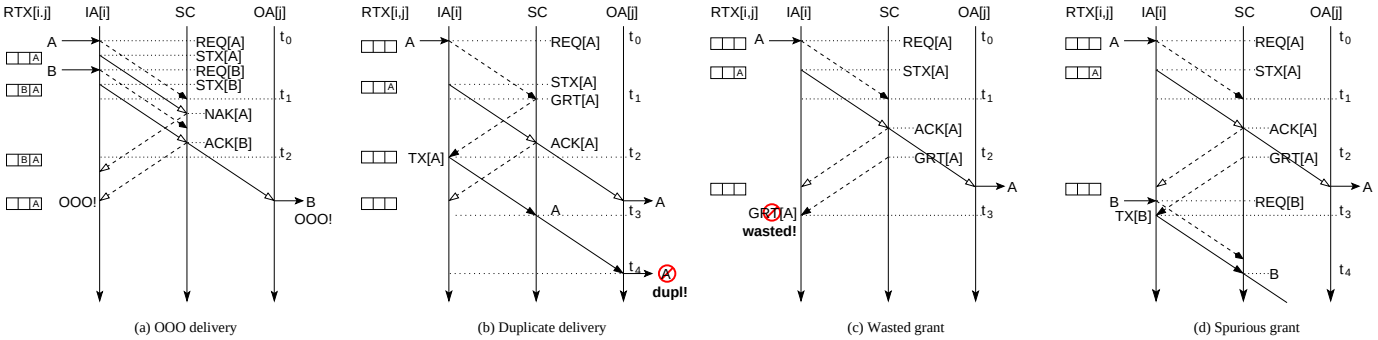
B. Collisions

An important consequence of STX is the occurrence of *collisions* in the switch fabric: As STX cells are sent without prior arbitration, they may collide with either other STX cells or arbitrated cells destined to the same output, and as a result they may be dropped.

In OSMOSIS, it is possible to always allow one cell to "survive" the collision, because the colliding cells do not share a physical medium until they arrive at the crossbar. The arbiter knows about incoming STX cells from the accompanying SRQs on the control path, and it also knows which arbitrated cells have been scheduled to arrive in the current time slot. Therefore, it can arbitrate between arriving STX cells if necessary and configure the crossbar to allow one to pass, while dropping the others. Therefore, one transmission is always successful, even in the case of a collision. This is an important difference to ALOHA or Ethernet, where *all* colliding cells are lost.

When multiple STX cells collide, we can forward up to R of them, but when an arbitrated cell collides with one or more STX cells, the arbitrated cell always takes precedence to ensure that STX does not interfere with the basic operation of the underlying matching algorithm (see $\mathcal{R}3$ and $\mathcal{R}4$). Note also that the matching algorithm ensures that collisions between arbitrated cells can never occur.

The collision arbitration operates as follows. The arbiter only grants an STX request if the corresponding output has not been matched, as indicated by the current matching $\mathcal{M}$. When there are multiple STX requests for an unmatched output, one is granted randomly. Granting an STX request does not affect the operation of the matching algorithm, e.g., in the case of i-SLIP, the round-robin pointers are not updated. The arbiter notifies the sender of a successful STX request by means of an acknowledgment (ACK). Of course, it also issues the regular grants according to matching $\mathcal{M}$. These grants may cause duplicate cell transmissions as described in $\mathcal{R}7$. The arbiter does not generate explicit negative acknowledgments (NAK) for dropped cells.



(a) OOO delivery (b) Duplicate delivery (c) Wasted grant (d) Spurious grant

Fig. 2. Scenarios leading to OOO delivery, duplications, wasted and spurious grants; RTX = retransmission queue, IA = input adapter, SC = switch core, OA = output adapter, RTT = 8, $t_{i+1} = t_i + \mathrm{RTT}/2$.

C. Retransmission

Collisions imply cell losses and out-of-order (OOO) delivery, which in turn imply a need for link-level retransmissions and ACKs, as this loss probability is orders of magnitude higher than that due to transmission errors. Reliability and ordering can be restored by means of a reliable delivery (RD) scheme. Any RD scheme requires that an STX cell remain in the input adapter buffer until successfully transmitted. The ACKs are generated by the arbiter for every successful STX cell and include the sequence number of the acknowledged cell. $\mathcal{R}6$ specifies that a speculative cell remains stored in the adapter until either of the following two events occurs:

- The cell is positively acknowledged, i.e., an ACK arrives with the corresponding sequence number. The cell is dequeued and dropped.
- A grant for this output arrives *and* the cell is the oldest unacknowledged STX cell. When a grant arrives and there are any unacknowledged STX cells for the granted output, the oldest of these is dequeued and retransmitted. Otherwise, the HOL cell of the VOQ is dequeued and transmitted, as usual. This rule implies that unacknowledged STX cells take precedence over other cells in the VOQ, to expedite their reliable, in-order delivery.

According to $\mathcal{R}5$, unacknowledged STX cells are *never* eligible for STX, because they have already been transmitted speculatively once. Allowing only one STX attempt per cell reduces the number of STXs, which increases their chance of success. Moreover, if an STX cell fails, the potential gain in latency has been lost in any case, so retrying the same cell serves no purpose. This is also the reason that we do not use explicit NAKs.

According to $\mathcal{R}7$ and $\mathcal{R}8$, a non-wasted grant can be classified in two orthogonal ways: It is either pure or duplicate, and it is either regular or spurious depending on whether it is used by the cell that initiated it.

There are several methods of achieving reliable, in-order delivery in the presence of STX, e.g., Stop & Wait, Go-Back-N (GBN), and Selective Retry (SR). Here, we consider SR.

SR allows a predetermined maximum number of cells per output to be unacknowledged at each input at any given time. STX cells are stored in retransmission (RTX) queues (one RTX queue per VOQ). The output adapter accepts cells in any order and performs resequencing to restore the correct cell order. To this end, it has a resequencing queue (RSQ) per input to store OOO cells until the missing ones arrive. The input adapter accepts ACKs in any order. This implies that only the failed STX cells need to be retransmitted, hence the name Selective Retry, as opposed to retransmitting the entire RTX queue as is done with GBN. SR requires resequencing logic and buffers at every output adapter. In addition, the RTX queues require a random-out organization, because cells can be dequeued from any point in the queue. However, SR minimizes the number of retransmissions, thus improving performance.

D. STX scenarios

In the following sections, we will explain the STX operations in more detail and describe some special scenarios with timeline diagrams, see Fig. 2.

1) Out-of-order delivery: Allowing multiple STXs from the same VOQ implies that cells may be delivered out of order. Figure 2(a) illustrates how this can happen, with $\mathrm{RTT} = 8$. Cell A arrives at t_0 and submits a regular request. At $t_0 + 1$, cell A is sent speculatively. Cell B arrives at $t_0 + 2$, submits a request, and is sent speculatively at $t_0 + 3$. Cell A is not successful, but cell B is. Because cell B has been sent speculatively before cell A has received a grant, cell B arrives at the output adapter before cell A, i.e., out of order. Owing to the RSQ at the output adapter, B is not discarded at $t_2 + 3$, but stored in the corresponding RSQ. The required size of the resequencing buffer is discussed in more detail in Sec. III-D.4.

2) Duplicate delivery: Cells may be delivered in duplicate. This happens when a successful STX cell is retransmitted because a grant for the corresponding VOQ arrives before the ACK. The output adapter simply drops all duplicate deliveries. Any cell with a sequence number smaller than or equal to that of the last cell successfully delivered in-order to the output is a duplicate. Figure 2(b) depicts this scenario. Cell A, which arrived at t_0, is sent speculatively at $t_0 + 3$. The speculation is successful, so A arrives at the output at $t_2 + 3$. However, a grant arrives before the ACK, causing A to be sent again. The duplicate arrives at the output at t_4 and is discarded.

3) Wasted and spurious grants: Another issue is that grants may be *wasted*. This happens when a grant arrives for a VOQ that is currently empty because the last cell made a successful speculation. Figure 2(c) illustrates this scenario. The speculation of A at $t_0 + 2$ is successful, and A is removed

from the VOQ when the ACK arrives at $t_2 + 3$. The grant issued by the arbiter at t_2 finds an empty VOQ and therefore goes to waste. A related scenario, shown in Fig. 2(d), leads to *spurious* grants, which can reduce the latency of an arbitrated transmission. Here, the newly arrived cell B is transmitted in response to the grant for the preceding cell A. In effect, B did not have to wait for a full RTT to obtain a grant.

4) Retransmission and resequencing window: We now address the dimensioning of the RTX and RSQ buffers. We must allow for up to RTT back-to-back STX transmissions to achieve immediate full link utilization in the absence of contention, thus requiring an RTX buffer of size $B_{\mathrm{RTX}} = \mathrm{RTT}$ cells. In addition to the selection policy described in Sec. III-A, the decision to attempt an STX for a given VOQ also depends on the state of the RTX buffer. With SR, to ensure that the resequencing buffer does not overflow, cells may be transmitted speculatively as long as the difference between the sequence numbers of the cells at the HOL of the VOQ and the HOL of the RTX buffer is less than or equal to the resequencing buffer size B_{RSQ}. To ensure that the additional RSQ condition does not constrain the link utilization, B_{RSQ} should be chosen equal to or greater than B_{RTX}; hence, $B_{\mathrm{RSQ}} = B_{\mathrm{RTX}} = \mathrm{RTT}$ is the optimal choice.

IV. System Analysis

Without loss of generality, we assume for the purpose of our analysis and simulations that $\tau_{\mathrm{uD}} = \tau_{\mathrm{dD}} = \tau_{\mathrm{uC}} = \tau_{\mathrm{dC}} = \mathrm{RTT}/2$ (although in practice these delays may differ), that $\tau_{\mathrm{SC}} = 0$, and that RTT is equal for all adapters. We proceed using the following nomenclature:

N	switch size (number of ports),
R	number of receivers,
RTT	round-trip delay (in number of time slots),
λ	input load,
X_g	delay from the time a grant is requested until it returns to the input adapter,
σ	rate of cell departures from input adapter due to grants,
σ_{d}	rate of duplicate cell departures from input adapter due to grants,
σ_{p}	rate of pure cell departures from input adapter due to grants,
P_{S}	probability that a cell is speculatively transmitted,
$P_{\mathrm{s\|S}}$	probability that a speculative cell is also successfully transmitted through the switch fabric,
P_{Ss}	probability that a cell is successfully speculatively transmitted through the switch fabric,
μ	probability that at any given slot a cell can be served,
θ	impatience time, relative deadline, waiting time of a cell until it receives a grant,
$f_\theta(\tau)$	pdf of the impatience time,
U	offered waiting time, i.e. waiting time of a cell for speculative transmission if no grant ever arrives,
$f_U(\tau)$	pdf of the offered waiting time,
α_{d}	probability of missing deadline, i.e., transmission of a cell due to a grant,
P_{w}	probability that a grant is wasted,
Q	probability that a grant is spurious, or, equivalently, that a cell is spuriously transmitted,
P_{sw}	probability that a grant is either spurious or wasted,
D	switch delay.

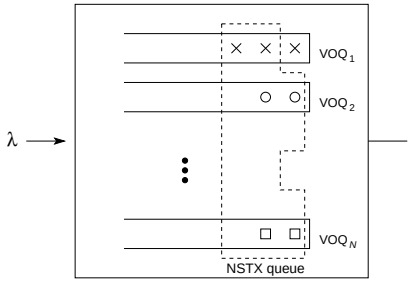


Fig. 3. Input adapter.

The objective of this study is to develop an analytic model for the derivation of the average switch delay, i.e., the delay from the arrival instant of a cell at an input adapter to its departure from the output buffer to its destination port. This model takes into account all the delay components except for the resequencing delay.

We consider an $N \times NR$ nonblocking input-queued crossbar switch. We assume a synchronous slotted operation, with the slot being the time required to transmit a cell. We also assume uniform Bernoulli traffic, with λ denoting the probability of a cell arrival at a given input queue in an arbitrary slot, or equivalently, the arrival rate or throughput, as shown in Fig. 3. This, in turn, implies that the distribution of the total number A over all the inputs of cell arrivals in a slot that are destined to a particular output is binomial, i.e.,

$$P(A = n) = \binom{N}{n} \left(\frac{\lambda}{N}\right)^n \left(1 - \frac{\lambda}{N}\right)^{N-n} , \qquad (1)$$

with $n \in [0, N]$. The first two moments are then given by

$$E(A) = \overline{A} = \lambda \ , \text{ and } E(A^2) = \overline{A^2} = \lambda^2 + \lambda(1 - \frac{\lambda}{N}) \ . \quad (2)$$

Cell arrivals generate requests which are sent to the arbiter, where they arrive after a delay of $\tau = \mathrm{RTT}/2$. The processing of these requests at the arbiter is modeled by a discrete Geo^X/D/1 queue, as depicted in Fig. 4. We assume that the arbiter is ideal in that at each slot it always matches one input to one output (provided there is an input request for that output). The mean of the sojourn time T_A in this queue is given by [8]

$$E(T_A) = \overline{T_A} = 1 + \frac{\overline{A^2} - \overline{A}}{2\overline{A}(1 - \overline{A})} \ . \qquad (3)$$

Consequently, the time X_g required from the instant a cell arrives at an input adapter until the corresponding issued grant returns to the input adapter is equal to $\mathrm{RTT} + T_A$ with mean

$$E(X_g) = \overline{X_g} = \mathrm{RTT} + \overline{T_A} \ . \qquad (4)$$

Owing to the speculative transmission scheme, the cell may have been speculatively transmitted, in the mean time. Cells waiting at the input adapter to be speculatively transmitted constitute an equivalent queue referred to as *NSTX queue*.

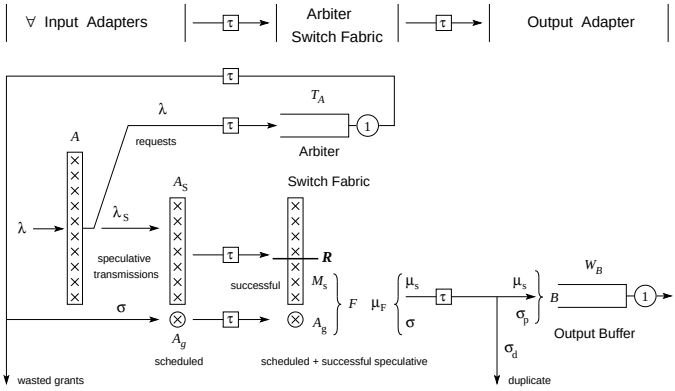


Fig. 4. System model for a given output ($\tau = \mathrm{RTT}/2$).

According to $\mathcal{R}5$, a cell is speculatively transmitted once. Therefore, when a cell is speculatively transmitted, it is removed from the NSTX queue regardless of whether the speculative transmission is successful. Clearly, cells are also removed from this queue, and also from the input adapter, when they are transmitted owing to corresponding grant arrivals. In the snapshot depicted in Fig. 3, the cell 'x' located at the head of the first VOQ has been speculatively transmitted (but no positive acknowledgment has arrived yet), whereas the remaining cells have not.

The rate at which grants arrive at an input adapter is equal to λ. Let us now denote by σ the (yet unknown) rate at which cells depart from the input adapter owing to grants. Therefore, the probability P_{w} that a grant is wasted is given by

$$P_{\mathrm{w}} = 1 - \frac{\sigma}{\lambda} \, . \tag{5}$$

Furthermore, σ represents the probability that a slot cannot be used for a speculative transmission, because a cell is transmitted as a result of a grant arrival. In general, σ can be written as the sum

$$\sigma = \sigma_{\mathrm{d}} + \sigma_{\mathrm{p}} \, , \tag{6}$$

where the first term represents the rate of duplicate cells and the second term the rate of pure cells, i.e., those that were transmitted through grants and were not duplicate.

Let λ_{S} denote the average arrival rate of speculatively transmitted cells at the switch fabric that are destined to a given output. Owing to the uniform destination assumption, this rate is the same for all output ports. The probability P_{S} that a cell is speculatively transmitted is given by

$$P_{\mathrm{S}} = \frac{\lambda_{\mathrm{S}}}{\lambda} \, , \tag{7}$$

the ratio of the departure rate λ_{S} of speculatively transmitted cells from an input adapter to the arrival rate λ.

Note also that, owing to the independence of the input adapters, the distribution of the number A_{S} of speculative cell arrivals in a slot at the switch fabric that are destined to a particular output is binomial with mean $E(A_{\mathrm{S}}) = \lambda_{\mathrm{S}}$, i.e.,

$$P(A_{\mathrm{S}} = n) = \binom{N}{n} \left(\frac{\lambda_{\mathrm{S}}}{N}\right)^{n} \left(1 - \frac{\lambda_{\mathrm{S}}}{N}\right)^{N-n} \, , \tag{8}$$

with $n \in [0, N]$. Owing to the uniform destination assumption, the mean is the same for all output ports. The first two moments are then given by

$$\begin{aligned} E(A_{\mathrm{S}}) &= \overline{A_{\mathrm{S}}} = \lambda_{\mathrm{S}} \, , \\ E(A_{\mathrm{S}}^2) &= \overline{A_{\mathrm{S}}^2} = \lambda_{\mathrm{S}}^2 + \lambda_{\mathrm{S}}(1 - \frac{\lambda_{\mathrm{S}}}{N}). \end{aligned} \tag{9}$$

Let A_{g} denote the number of granted cell arrivals in a slot at the switch fabric that are destined to a particular output. According to the matching algorithm, A_{g} can be either zero or one. Assuming that the stochastic process $\{A_{\mathrm{g}}\}$ over successive slots is Bernoulli, and given that $E(A_{\mathrm{g}}) = \sigma$, it holds that

$$P(A_{\mathrm{g}} = n) \; = \; \begin{cases} 1 - \sigma \, , & \text{for} \;\; n = 0 \, , \\ \sigma \, , & \text{for} \;\; n = 1 \, . \end{cases} \tag{10}$$

Let A_{gp} and A_{gd} denote the number of pure and duplicate cell arrivals in a slot at the switch fabric that are destined to a particular output, respectively. Therefore it holds that

$$A_{\mathrm{gd}} + A_{\mathrm{gp}} = A_{\mathrm{g}}. \tag{11}$$

Owing to (6) it holds that

$$P(A_{\mathrm{gp}} = n) \; = \; \begin{cases} 1 - \sigma_{\mathrm{p}} \, , & \text{for} \;\; n = 0 \, , \\ \sigma_{\mathrm{p}} \, , & \text{for} \;\; n = 1 \, . \end{cases} \tag{12}$$

and

$$P(A_{\mathrm{gd}} = n) \; = \; \begin{cases} 1 - \sigma_{\mathrm{d}} \, , & \text{for} \;\; n = 0 \, , \\ \sigma_{\mathrm{d}} \, , & \text{for} \;\; n = 1 \, , \end{cases} \tag{13}$$

Note, however, that A_{gd} and A_{gp} are not independent Bernoulli processes given that they have to satisfy (11). We proceed by assuming that

$$A_{\mathrm{gp}} = Y A_{\mathrm{g}} \quad \text{and} \quad A_{\mathrm{gd}} = (1 - Y) A_{\mathrm{g}} \, , \tag{14}$$

where $\{Y\}$ is a Bernoulli stochastic process over successive slots with

$$P(Y = n) \; = \; \begin{cases} 1 - \frac{\sigma_{\mathrm{p}}}{\sigma} \, , & \text{for} \;\; n = 0 \, , \\ \frac{\sigma_{\mathrm{p}}}{\sigma} \, , & \text{for} \;\; n = 1 \, . \end{cases} \tag{15}$$

From the above definitions, it follows that the number F of successfully transmitted cells (both speculative and granted) in a slot through the switch fabric that are destined to a particular output is given by

$$F = \min(A_{\mathrm{S}} + A_{\mathrm{g}} \, , R) \, . \tag{16}$$

The distribution of F is obtained as follows

$$P(F = n) = \tag{17}$$
$$\begin{cases} P(A_{\mathrm{S}} = 0 \, \cap \, A_{\mathrm{g}} = 0) \, , & \text{for } n = 0, \\ P((A_{\mathrm{S}} = n \, \cap \, A_{\mathrm{g}} = 0) & \\ \quad \cup \, (A_{\mathrm{S}} = n - 1 \, \cap \, A_{\mathrm{g}} = 1)) \, , & \text{for } 1 \leq n < R, \\ P((A_{\mathrm{S}} \geq R \, \cap \, A_{\mathrm{g}} = 0) & \\ \quad \cup \, (A_{\mathrm{S}} \geq R - 1 \, \cap \, A_{\mathrm{g}} = 1)) \, , & \text{for } n = R. \end{cases}$$

As A_{S} and A_{g} are independent, using (10) and (17) yields

$$P(F=n) = \begin{cases} (1-\sigma)P(A_{\mathrm{S}}=0)\,, & \text{for } n=0, \\ (1-\sigma)P(A_{\mathrm{S}}=n) & \\ \quad +\sigma P(A_{\mathrm{S}}=n-1)\,, & \text{for } 1\le n<R, \\ (1-\sigma)P(A_{\mathrm{S}}\ge R) & \\ \quad +\sigma P(A_{\mathrm{S}}\ge R-1)\,, & \text{for } n=R. \end{cases} \tag{18}$$

The rate μ_{F} of transmitted cells through the switch fabric is equal to the average number $E(F)$ of transmitted cells per slot through the switch fabric given by

$$\mu_{\mathrm{F}} = E(F) = \sum_{n=1}^{R} n\,P(F=n)\,. \tag{19}$$

Similarly, the number of successful speculative cells M_{s} in a slot through the switch fabric that are destined to a particular output is given by

$$M_{\mathrm{s}} = \min(A_{\mathrm{S}}+A_{\mathrm{g}}\,,R) - A_{\mathrm{g}}\,. \tag{20}$$

The distribution of M_{s} is obtained as follows

$$P(M_{\mathrm{s}}=n) = \begin{cases} P((A_{\mathrm{S}}=0\,\cap\,A_{\mathrm{g}}=0) & \\ \quad \cup\,A_{\mathrm{g}}=1)\mathbf{1}_{\{R=1\}} & \\ \quad +P(A_{\mathrm{S}}=0)\mathbf{1}_{\{R>1\}}\,, & \text{for } n=0\,, \\ P(A_{\mathrm{S}}=n)\,, & \text{for } 1\le n<R-1\,, \\ P((A_{\mathrm{S}}=R-1\,\cap\,A_{\mathrm{g}}=0) & \\ \quad \cup\,(A_{\mathrm{S}}\ge R-1\,\cap\,A_{\mathrm{g}}=1))\,, & \text{for } n=R-1\,, \\ P(A_{\mathrm{S}}\ge R\,\cap\,A_{\mathrm{g}}=0)\,, & \text{for } n=R\,. \end{cases} \tag{21}$$

As A_{S} and A_{g} are independent, using (10) and (21) yields

$$P(M_{\mathrm{s}}=n) = \begin{cases} [(1-\sigma)P(A_{\mathrm{S}}=0)+\sigma]\,\mathbf{1}_{\{R=1\}} & \\ \quad +P(A_{\mathrm{S}}=0)\mathbf{1}_{\{R>1\}}\,, & \text{for } n=0\,, \\ P(A_{\mathrm{S}}=n)\,, & \text{for } 1\le n<R-1\,, \\ (1-\sigma)P(A_{\mathrm{S}}=R-1) & \\ \quad +\sigma P(A_{\mathrm{S}}\ge R-1)\,, & \text{for } n=R-1\,, \\ (1-\sigma)P(A_{\mathrm{S}}\ge R)\,, & \text{for } n=R\,. \end{cases} \tag{22}$$

The rate μ_{s} of successful speculative cells transmitted through the switch fabric is equal to the average number $E(M_{\mathrm{s}})$ of successfully transmitted cells per slot through the switch fabric given by

$$\mu_{\mathrm{s}} = E(M_{\mathrm{s}}) = \sum_{n=1}^{R} n\,P(M_{\mathrm{s}}=n)\,. \tag{23}$$

Flow conservation implies that

$$\mu_{\mathrm{s}} = \mu_{\mathrm{F}} - \sigma\,. \tag{24}$$

The probability $P_{\mathrm{s|S}}$ that a speculative cell is also successfully transmitted through the switch fabric is given by

$$P_{\mathrm{s|S}} = \frac{\mu_{\mathrm{s}}}{\lambda_{\mathrm{S}}}\,, \tag{25}$$

the ratio of the average number of successful speculative cells per slot through the switch fabric to the the average number of speculative cells per slot. From (7) and (25), it follows that the probability P_{Ss} that a cell is successfully speculatively transmitted through the switch fabric is given by

$$P_{\mathrm{Ss}} = P_{\mathrm{S}}\,P_{\mathrm{s|S}} = \frac{\mu_{\mathrm{s}}}{\lambda}\,. \tag{26}$$

At the output queue, the arriving duplicate cells are dropped such that the net arrival rate is equal to $\mu_{\mathrm{s}}+\sigma_{\mathrm{p}}$. Flow conservation implies that the net arrival rate is equal to the departure rate λ, which yields

$$\sigma_{\mathrm{p}} = \lambda - \mu_{\mathrm{s}}\,. \tag{27}$$

Combining (6), (24) and (27) yields

$$\sigma_{\mathrm{d}} = \mu_{\mathrm{F}} - \lambda\,. \tag{28}$$

We proceed to calculate the distribution of the net number B of cells entering the output queue in a slot excluding the duplicate cells which are dropped. It holds that $B = M_{\mathrm{s}} + A_{\mathrm{gp}}$, with M_{s} and A_{gp} not being independent. From (20) and (11), it now follows that

$$B = \min(A_{\mathrm{S}}+A_{\mathrm{gp}}\,,R-A_{\mathrm{gd}})\,. \tag{29}$$

Consequently,

$$P(B=n) = \begin{cases} P((A_{\mathrm{S}}=0\,\cap\,A_{\mathrm{g}}=0) & \\ \quad \cup\,A_{\mathrm{gd}}=1)\,\mathbf{1}_{\{R=1\}} & \\ \quad +P(A_{\mathrm{S}}=0\,\cap\,A_{\mathrm{gp}}=0)\mathbf{1}_{\{R>1\}}\,, & \text{for } n=0\,, \\ P((A_{\mathrm{S}}=n\,\cap\,A_{\mathrm{gp}}=0) & \\ \quad \cup\,(A_{\mathrm{S}}=n-1\,\cap\,A_{\mathrm{gp}}=1))\,, & \text{for } 1\le n<R-1\,, \\ P((A_{\mathrm{S}}=R-1\,\cap\,A_{\mathrm{g}}=0) & \\ \quad \cup\,(A_{\mathrm{S}}=R-2\,\cap\,A_{\mathrm{gp}}=1)\,\cup & \\ \quad \cup\,(A_{\mathrm{S}}\ge R-1\,\cap\,A_{\mathrm{gd}}=1))\,, & \text{for } n=R-1\,, \\ P((A_{\mathrm{S}}\ge R\,\cap\,A_{\mathrm{g}}=0) & \\ \quad \cup\,(A_{\mathrm{S}}\ge R-1\,\cap\,A_{\mathrm{gp}}=1))\,, & \text{for } n=R\,. \end{cases} \tag{30}$$

Given that A_{S} is independent of A_{g}, A_{gd} and A_{gp}, and then making use of (10), (12) and (13), Eq. (30) yields

$$P(B=n) = \begin{cases} [(1-\sigma)P(A_{\mathrm{S}}=0)+\sigma_{\mathrm{d}}]\,\mathbf{1}_{\{R=1\}} & \\ \quad +(1-\sigma_{\mathrm{p}})P(A_{\mathrm{S}}=0)\,\mathbf{1}_{\{R>1\}}\,, & \text{for } n=0\,, \\ (1-\sigma_{\mathrm{p}})P(A_{\mathrm{S}}=n) & \\ \quad +\sigma_{\mathrm{p}}P(A_{\mathrm{S}}=n-1)\,, & \text{for } 1\le n<R-1\,, \\ (1-\sigma)P(A_{\mathrm{S}}=R-1) & \\ \quad +\sigma_{\mathrm{p}}P(A_{\mathrm{S}}=R-2) & \\ \quad +\sigma_{\mathrm{d}}P(A_{\mathrm{S}}\ge R-1)\,, & \text{for } n=R-1\,, \\ (1-\sigma)P(A_{\mathrm{S}}\ge R) & \\ \quad +\sigma_{\mathrm{p}}P(A_{\mathrm{S}}\ge R-1)\,, & \text{for } n=R\,. \end{cases} \tag{31}$$

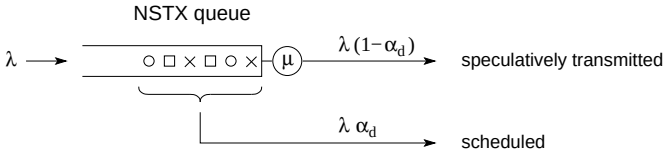


Fig. 5. Impatience queueing model for the NSTX queue.

The first two moments are then given by

$$E(B) = \overline{B} = \sum_{n=1}^{R} n\, P(B=n)\ , \qquad E(B^2) = \overline{B^2} = \sum_{n=1}^{R} n^2\, P(B=n)\ . \tag{32}$$

The mean of the waiting time W_B in the output queue is given by [8]

$$E(W_B) = \overline{W_B} = \frac{\overline{B^2} - \overline{B}}{2\overline{B}(1-\overline{B})}\ . \tag{33}$$

A. Input Adapter

We now present the model for deriving the probability that a cell is speculatively transmitted, as well as the remaining measures of interest. Cells arriving at an input adapter join the equivalent non-speculatively-transmitted (NSTX) queue and issue a request for arbitration. This queue contains the cells that are contending for speculative transmission, as illustrated in Fig. 5. In a given slot one of these cells can be served, i.e., speculatively transmitted, if there is no non-wasted grant present. Thus, the probability μ that in any given slot a cell can be served is given by

$$\mu = 1 - \sigma\ , \tag{34}$$

where both μ and σ are (yet to be determined) functions of λ. It is assumed that the cell to be served is the one that has been waiting the longest time, i.e., we assume a FCFS serving discipline. Let w represent the time that a cell has been waiting in the queue until it is served. Note that a cell is not guaranteed to be served. It may be removed from the queue owing to the arrival of a corresponding grant at the input adapter while it is waiting in the NSTX queue. This situation can be modeled by a queueing model where each customer has a strict deadline before which it is available for service and after which it must leave the system. In the context of queueing theory, customers with limited waiting time are usually referred to as "*impatient customers*". The switch model presented corresponds to a general discrete-time customer impatience model with a FCFS service discipline. The corresponding service times are geometrically distributed with parameter μ. Also, the deadlines of customers are effective only until the beginning of their service. As such a discrete-time model has not yet been analyzed in the literature, we consider instead the continuous counterpart model that assumes Poisson arrivals and exponentially distributed service times [9]. Note that, for small values of the load λ and the service rate μ, the model is accurate because the geometric interarrival and service time distributions approach the exponential ones. Let θ represent the *relative deadline*, i.e., the time a cell has been waiting in the queue until it is removed, and let $f_\theta(\tau)$ be the corresponding probability density function (pdf). If the removal of a cell is due to a grant that corresponds to its original request, then the time elapsed is roughly equal[1] to $\overline{X_g}$ given by (4). However, there is also a possibility that the cell is spuriously transmitted at an earlier time. We assume that the probability of this event is equal to Q and that the time is uniformly distributed in the interval $(0, \overline{X_g})$. The pdf of the customer impatience is therefore given by

$$f_\theta(\tau) = \frac{Q}{\overline{X_g}} + (1-Q)\delta(\tau - \overline{X_g})\ , \quad \text{for } 0 \le \tau \le \overline{X_g}\ . \tag{35}$$

Consequently, the mean impatience is obtained by

$$\overline{\theta} = E(\theta) = \int_0^\infty \tau f_\theta(\tau) d\tau = (1 - \frac{Q}{2})\,\overline{X_g}\ . \tag{36}$$

The performance measures of such a system are obtained by the following theorems.

Theorem 1: The pdf of the distribution of the *offered waiting time* U, which is the time an arriving customer with infinite (no) deadline must wait before its service commences, is given by

$$\begin{aligned} f_U(\tau) &= \lambda p_0 e^{\lambda \int_0^\tau \overline{F_\theta}(x)dx - \mu\tau} + p_0\,\delta(\tau) \\ &= \begin{cases} p_0\delta(\tau) + \lambda p_0 e^{-a\tau - b\tau^2}\ , & 0 \le \tau \le \overline{X_g}\ , \\ \lambda p_0 e^{\lambda\overline{\theta} - \mu\tau}\ , & \tau \ge \overline{X_g}\ , \end{cases} \end{aligned} \tag{37}$$

where

$$a \triangleq \mu - \lambda\ , \qquad b \triangleq \frac{\lambda Q}{2\,\overline{X_g}}\ , \qquad c \triangleq \frac{a}{2b}\ , \tag{38}$$

and

$$p_0 = \left(1 + \lambda\left\{\sqrt{\frac{\pi}{4b}}\, e^{\frac{a^2}{4b}}\left[\mathrm{erf}(\sqrt{b}\,(\overline{X_g} + c)) - \mathrm{erf}(\sqrt{b}\,c)\right] + \frac{1}{\mu}\, e^{\lambda\overline{\theta} - \mu\overline{X_g}}\right\}\right)^{-1}\ . \tag{39}$$

Proof: See Appendix A. ∎

Theorem 2: The probability of *missing the deadline*, α_{d}, which corresponds to the transmission of a cell due to a grant, is given by

$$\alpha_{\mathrm{d}} = 1 - \frac{\mu}{\lambda}\,(1 - p_0)\ , \tag{40}$$

where p_0 is given by (39).

Proof: Immediate from (3.32) of [9]. ∎

Corollary 1: The probability that a cell is speculatively transmitted is given by

$$P_{\mathrm{S}} = 1 - \alpha_{\mathrm{d}}\ . \tag{41}$$

Theorem 3: The probabilities that a grant is spurious or wasted are given by

$$Q = \frac{P_{\mathrm{SA}}\,(1 - P_{\mathrm{na}})}{1 - (1 - P_{\mathrm{SA}})(1 - P_{\mathrm{na}})}\ . \tag{42}$$

and

$$P_{\mathrm{w}} = \frac{P_{\mathrm{SA}}\,P_{\mathrm{na}}}{1 - (1 - P_{\mathrm{SA}})(1 - P_{\mathrm{na}})}\ . \tag{43}$$

[1]This holds when RTT $\gg \overline{T_A}$, with the variance of the queueing delay around the value $\overline{T_A}$ being relatively small.

respectively, where $P_{\rm SA}$ and $P_{\rm na}$ are given by

$$P_{\rm SA} = p_0 \left\{ 1 + \frac{\lambda}{2} \sqrt{\frac{\pi}{b}}\, e^{\frac{a^2}{4b}} \left[\mathrm{erf}(\sqrt{b}\,(\overline{X_g} - \mathrm{RTT} + c)) - \mathrm{erf}(\sqrt{b}\,c) \right] \right\} P_{\rm s|S} \ , \tag{44}$$

and

$$P_{\rm na} = \left(1 - \frac{\lambda}{N}\right)^{\overline{X_g}} \ . \tag{45}$$

Proof: See Appendix B. ∎

B. Delay Evaluation

We now proceed with the evaluation of the various measures as well as of the mean switch delay. As depicted in Fig. 4, there is a loop in the flow in that the requests from the input adapters are sent to the arbiter, the output of which is fed back to the input adapters. This suggests that the measures of interests cannot be directly obtained, but they will have to be derived using an iterative procedure.

Indeed, let us examine the expression for Q given by (42) and (44). From (36), (38) and (39), we note that this expression is also a function of Q given that b, c, $\overline{\theta}$, and therefore p_0 are functions of Q. Consequently, (42) leads to a fixed-point iteration for the evaluation of Q. It turns out that σ needs to be specified beforehand, along with μ and a through (34) and (38), respectively. The procedure assumes an initial value for Q, say $Q_{\rm old}$, and its new value, $Q_{\rm new}$, is derived according to (42) based on the following sequence of derivations:

$$Q_{\rm old} \left\{ \begin{array}{l} \stackrel{(36)}{\longrightarrow} \overline{\theta} \\ \stackrel{(38)}{\longrightarrow} b, c \end{array} \right\} \stackrel{(39)}{\longrightarrow} p_0 \stackrel{(40)}{\longrightarrow} \alpha_{\rm d} \stackrel{(41)}{\longrightarrow} P_{\rm S} \stackrel{(7)}{\longrightarrow} \lambda_{\rm S} \stackrel{(8)}{\longrightarrow} A_{\rm S} \stackrel{(22)}{\longrightarrow} M_{\rm s} \stackrel{(23)}{\longrightarrow} \mu_{\rm s} \stackrel{(25)}{\longrightarrow} P_{\rm s|S} \stackrel{(44)}{\longrightarrow} P_{\rm SA} \stackrel{(42)}{\longrightarrow} Q_{\rm new} \tag{46}$$

Iterating these steps using repeated substitution leads to the derivation of $Q(\sigma)$, which denotes the equilibrium fixed point for Q for a given fixed σ. Next we proceed to evaluating the yet unknown value of σ. We apply a similar iterative procedure by starting with an initial value $\sigma_{\rm old}$ and deriving the new value $\sigma_{\rm new}$ according to the following sequence of derivations:

$$\sigma_{\rm old} \stackrel{(34)}{\longrightarrow} \mu \stackrel{(46)}{\longrightarrow} Q(\sigma_{\rm old}) \stackrel{(39)(40)}{\longrightarrow} p_0, \alpha_{\rm d} \stackrel{(41)(7)}{\longrightarrow} P_{\rm S}, \lambda_{\rm S} \stackrel{(8)(22)}{\longrightarrow} A_{\rm S}, M_{\rm s} \stackrel{(23)(25)}{\longrightarrow} \mu_{\rm s}, P_{\rm s|S} \stackrel{(44)}{\longrightarrow} P_{\rm w} \stackrel{(5)}{\longrightarrow} \sigma_{\rm new} \tag{47}$$

By iterating these steps using repeated substitution, the equilibrium fixed point for σ can be obtained along with all other performance measures of interest. The mean switch delay can now be derived as follows.

Theorem 4: The mean delay $\overline{D}$ is given by

$$\overline{D} \ = \ \mathrm{RTT} + \overline{W_B} + \overline{\theta} - P_{\rm s|S} \left(\overline{\theta}\, I_0 - I_1 + \frac{Q}{2\overline{X_g}}\, I_2 \right) \ , \tag{48}$$

where

$$I_0 = p_0 \left\{ 1 + \lambda\, e^{\frac{a^2}{4b}} \sqrt{\frac{\pi}{4b}} \left[\mathrm{erf}(\sqrt{b}\,(\overline{X_g} + c)) - \mathrm{erf}(\sqrt{b}\,c) \right] \right\} \ , \tag{49}$$

$$I_1 = -\, p_0\, \lambda\, e^{\frac{a^2}{4b}} \left\{ c \sqrt{\frac{\pi}{4b}} \left[\mathrm{erf}(\sqrt{b}\,(\overline{X_g} + c)) - \mathrm{erf}(\sqrt{b}\,c) \right] + \frac{1}{2b} \left[e^{-b(\overline{X_g}+c)^2} - e^{-bc^2} \right] \right\} \ , \tag{50}$$

$$I_2 = p_0\, \lambda\, e^{\frac{a^2}{4b}} \left\{ \frac{(1+2c^2b)\sqrt{\pi}}{4b\sqrt{b}} \left[\mathrm{erf}(\sqrt{b}\,(\overline{X_g} + c)) - \mathrm{erf}(\sqrt{b}\,c) \right] + \frac{(c - \overline{X_g})\, e^{-b(\overline{X_g}+c)^2} - c\, e^{-bc^2}}{2b} \right\} \tag{51}$$

and $\overline{W_B}$, $\overline{\theta}$, $P_{\rm s|S}$, $\overline{X_g}$, and $\{a,\, b,\, c\}$, are given by (33), (36), (25), (4), and (38), respectively.

Proof: See Appendix C. ∎

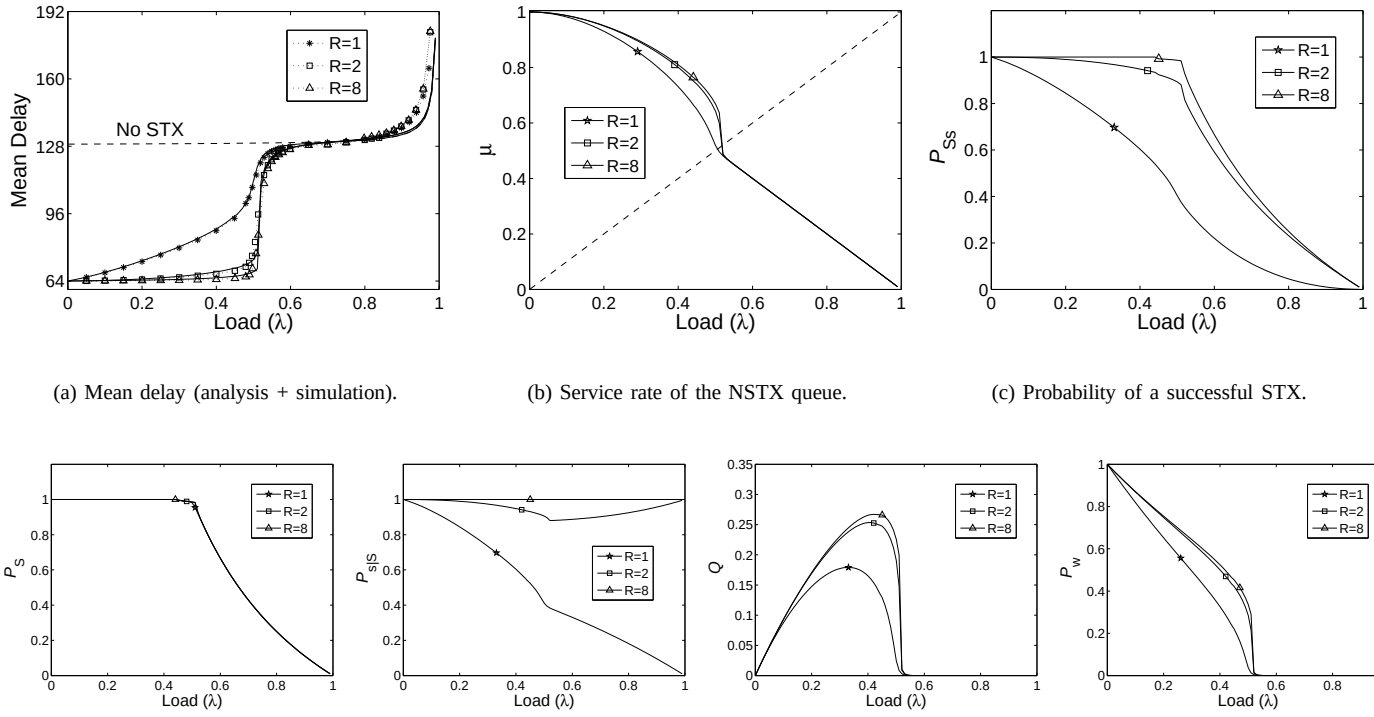
Also, the mean switch delay when the speculative transmission scheme is not used is given by $\overline{X_g} + \mathrm{RTT}$ or $\overline{T_A} + 2\,\mathrm{RTT}$.

V. Numerical Results

We consider a single-stage 64×64 switch with $\mathrm{RTT} = 64$. The STX policy is OCF with SR reliable delivery. The input and output buffers are assumed to be infinite. The mean delay curves (excluding the resequencing delay) corresponding to R = 1, 2, and 8 are analytically evaluated using (48) and are depicted in Fig. 6a. The dashed line indicates the switch delay when the speculative transmission scheme is not used.

We also developed a simulation model to verify the analytical results. We use a steady-state simulation method to determine the mean throughput and delay with uniform Bernoulli traffic. The arbitration algorithm used in the simulations is i-SLIP with six iterations. Simulations were conducted using the Akaroa2 parallel simulation management tool to run 12 independent replications of the model to obtain confidence intervals on the sampled data. The confidence intervals achieved are better than 0.3%, with 99% confidence, on the throughput and better than 5%, with 95% confidence, on the mean delay. The mean resequencing delay was evaluated by means of simulation and turned out to be relatively small. Therefore, the conclusions drawn below based on the analytic model also apply when resequencing is taken into account.

First, we note that for loads of less than 80% the analytical results are in excellent agreement with the simulated ones depicted by dotted lines and symbols. For higher loads, there is a divergence because the exact behavior of the i-SLIP matching algorithm is not captured by the simple Geo^X/D/1 queue that models the arbiter. Consequently, at high loads, the delay derived based on this queue underestimates the delay of the i-SLIP matching algorithm. The results demonstrate that for light loads there is a significant delay reduction from 128 (2*RTT) to 64 (RTT) time slots. This is due to the fact that all cells are speculatively transmitted and are successful because of the absence of contention at low loads, as shown in Fig. 6c. Furthermore, returning grants are wasted because cells have already been successfully speculatively transmitted



(a) Mean delay (analysis + simulation).

(b) Service rate of the NSTX queue.

(c) Probability of a successful STX.

(d) Probability of a cell being speculatively transmitted.

(e) Probability of an STX cell being successfully transmitted.

(f) Probability of a spurious transmission.

(g) Probability of a grant being wasted.

Fig. 6. Analytic performance characteristics for $R = 1, 2, 8$ and RTT = 64 as a function of the arrival rate.

and there are no subsequent arrivals to make use of them. The delay reduction diminishes as the load increases, although it remains significant for loads of less than 50%. The results also demonstrate that for higher loads the delay increases sharply. The key to explaining this behavior is the NSTX queue. For loads exceeding 50%, the service rate μ of this queue, as derived in (34) and depicted in Fig. 6b, is smaller than the arrival rate λ, which in turn implies a high increase of the queue occupancy (which is not infinite because cells are removed owing to expiration of their deadline). This also translates into a drastically reduced possibility of speculative transmissions and therefore to a sharply increased delay. The probability of a cell being successfully speculatively transmitted, as derived from (26), is shown in Fig. 6c. For $R \geq 2$ and loads of less than 50%, this probability is high, but drops sharply for loads exceeding 50%. The introduction of two receivers results in a significant performance improvement compared with a single receiver. However, the performance gain achieved from the introduction of additional receivers is minimal.

Figures 6(d,e) show the probabilities of speculation and of the success of a speculation derived from (7) and (25), respectively. The former does not depend on R, as the number of speculation opportunities depends only on the overall utilization, whereas the probability of success increases drastically by increasing R to 2. With R = 8, basically every speculation is successful. The product of these two curves yields Fig. 6c. Fig. 6f shows that the effect of spurious grants is non-negligible, implying that speculation also reduces latency on up to 25% (for $R > 1$) of the arbitrated transmissions. Figure 6g, finally, shows that below 50% load most grants are wasted, whereas beyond 50% load almost none are.

VI. Conclusions and Future Work

This work was motivated by the need to achieve low latency in an input-queued centrally-arbitrated cell switch for high-performance computing applications; specifically, the aim is to reduce the control-path latency incurred between issuance of a request and arrival of the corresponding grant.

The proposed solution features a combination of speculative and arbitrated transmission modes, coupling the advantages of uncoordinated transmission, i.e., not having to wait for a grant, hence low latency, with those of coordinated transmission, i.e., high maximum utilization.

An analytical model has been developed to evaluate the efficiency of this scheme with an oldest-cell-first speculation policy and selective retries, in the context of an $N \times NR$ crossbar switch with R receivers per output, assuming uniform i.i.d. arrivals. This model provides analytical results based on a fixed-point iterative method, which yields various performance measures of interest, such as the mean delay, the rates of speculative, pure, and duplicate transmissions, as well as the rate of successful speculative transmissions. In particular, the model captures the effect of non-negligible RTTs.

Our analysis and simulation results both confirm that this scheme achieves a significant latency reduction of up to 50% at traffic loads up to 50%. Employing two receivers per output instead of one drastically increases the speculative efficiency, but the additional gain of more than two receivers is minimal.

Here, we have only considered the oldest-cell-first policy. Simulation experiments indicate that the performance benefit of STX significantly improves when a different policy, e.g. random or youngest-cell-first, is employed. We are currently extending the analytical model to cover these policies. Although the results presented here form a solid baseline, the performance under bursty and non-uniform traffic patterns also remains to be studied. We are also studying the performance impact of using other, less costly, reliable delivery schemes such as stop-and-wait or go-back-N.

Finally, the proposed STX scheme entails cost in terms of bandwidth overhead and hardware complexity. The additional bandwidth required to implement STX consists of speculative requests and acknowledgments on the control channel and cell sequence numbers on the data channel. In terms of hardware, STX requires one RTX queue per VOQ and logic to implement the STX policy in the input adapters, resequencing buffers and logic in the output adapters, and speculative arbitration logic in the central arbiter. As this cost is not negligible, implementing STX is only worthwhile in applications where latency is crucial. The STX scheme is currently being implemented in FPGAs for the OSMOSIS system. The results of this effort will be reported in a subsequent publication.

Acknowledgments

The authors thank the sponsors and acknowledge the technical contributions of everybody involved at IBM, Corning Inc., Photonic Controls LLC, and G&O.

Appendix A
Impatience Model

Proof of Theorem 1.

From (35), it follows that the complementary cumulative distribution function $\overline{F_\theta}(\tau)$ of the impatience time θ equals

$$\overline{F_\theta}(\tau) = P(\theta > \tau) = \begin{cases} 1 - Q\,\frac{\tau}{\overline{X_g}}\,, & 0 \le \tau < \overline{X_g}\,, \\ 0\,, & \tau \ge \overline{X_g}\,. \end{cases} \tag{52}$$

From (52) it follows that

$$\int_0^\tau \overline{F_\theta}(x)dx = \begin{cases} \tau - \frac{Q}{2\overline{X_g}}\,\tau^2\,, & 0 \le \tau \le \overline{X_g}\,, \\ (1-\frac{Q}{2})\,\overline{X_g}\,(\overset{(36)}{=}\overline{\theta}\,)\,, & \tau \ge \overline{X_g}\,. \end{cases} \tag{53}$$

The probability p_0 that the queue is empty is derived from (3.27) of [9] together with (53) as follows

$$\begin{aligned} p_0 &= \left(1 + \lambda \int_0^\infty e^{\lambda \int_0^\tau \overline{F_\theta}(x)\,dx - \mu\tau} d\tau\right)^{-1} \\ &= \left(1 + \lambda \left[\int_0^\infty e^{\lambda\left(\tau - \frac{Q}{2\overline{X_g}}\,\tau^2\right) - \mu\tau} d\tau + \int_{\overline{X_g}}^\infty e^{\lambda\overline{\theta} - \mu\tau} d\tau\right]\right)^{-1} \\ &= \left(1 + \lambda \left\{\frac{1}{2}\sqrt{\frac{\pi}{b}}\,e^{\frac{a^2}{4b}}\left[\mathrm{erf}(\sqrt{b}\,(\overline{X_g}+c)) - \mathrm{erf}(\sqrt{b}\,c)\right]\right.\right. \\ &\qquad \left.\left. + \frac{1}{\mu}\,e^{\lambda\overline{\theta} - \mu\overline{X_g}}\right\}\right)^{-1}, \end{aligned} \tag{54}$$

where a, b and c are defined in (38). The pdf $f_U(\tau)$ is derived from (3.30) of [9] together with (53) and is given by (37). ■

Appendix B
Spurious/Wasted Grants

Proof of Theorem 3.

Let P_{sw} denote the probability that a grant is either spurious or wasted, i.e.,

$$P_{\text{sw}} \;\triangleq\; Q + P_{\text{w}}\,. \tag{55}$$

This occurs when the cell that has initiated the grant does not make use of the grant because it is either transmitted owing to a spurious grant, or it is successfully speculatively transmitted and acknowledged prior to a grant arrival. Let P_{SA} denote the probability that a cell is successfully speculatively transmitted and acknowledged prior to its grant arrival. Then, the probability $1 - P_{\text{sw}}$ that a grant is regular is equal to the product of $1-Q$, the probability that a cell is not transmitted due to a spurious grant, times $1-P_{\text{SA}}$, the probability that a cell is not successfully speculatively transmitted and acknowledged prior to its grant arrival, i.e. $1 - P_{\text{sw}} = (1-Q)(1-P_{\text{SA}})$. Consequently,

$$P_{\text{sw}} = Q + (1-Q)P_{\text{SA}}\,. \tag{56}$$

Let τ denote the period the cell has waited at the input adapter before it gets transmitted. The conditional pdf $P_{\text{SA}}(\tau)$ of a cell being speculatively transmitted after a waiting time of τ and acknowledged (after time $\tau + \text{RTT}$) prior to its grant arrival (after time $\overline{X_g}$) is given by

$$P_{\text{SA}}(\tau) = \begin{cases} f_U(\tau)P_{\text{s|S}}\,, & \text{for } \; 0 \le \tau < \overline{X_g} - \text{RTT}\,, \\ 0\,, & \text{otherwise}\,. \end{cases} \tag{57}$$

Unconditioning on τ, we obtain P_{SA} from (57)

$$P_{\text{SA}} = \left[\int_0^{\overline{X_g}-\text{RTT}} f_U(\tau)d\tau\right] P_{\text{s|S}}\,. \tag{58}$$

Substituting (37) into (58), after some manipulations, yields (44). A grant is wasted when upon its arrival, say at instant t, the cell that initiated it (which arrived at time $t - \overline{X_g}$) does not make use of it and there are no subsequent cell arrivals to the corresponding VOQ during the interval $(t-\overline{X_g}, t)$. As the latter event is independent of the former, it holds that

$$P_{\text{w}} = P_{\text{sw}}\,P_{\text{na}}\,, \tag{59}$$

where P_{na} denotes the probability that there are no cell arrivals to a VOQ during the interval $(t-\overline{X_g}, t)$. Owing to the uniform destination assumption, the process according to which cells arrive at a particular VOQ is Bernoulli with parameter λ/N. Consequently, the probability of no cell arrival during an interval of $\overline{X_g}$ successive slots is given by (45). Combining (55) and (59) yields

$$Q = P_{\text{sw}}\,(1 - P_{\text{na}})\,, \tag{60}$$

Plugging (56) into (60) and solving for Q yields (42). Combining (59), (60) and (42) yields P_{w} as given by (43). ■

APPENDIX C
MEAN DELAY

Proof of Theorem 4.

Let us first consider the delay D_i from the instant a cell arrives at the input adapter until the instant it is transmitted through the switch fabric. We consider the following cases:

Case 1) The cell is speculatively transmitted from the adapter, after a waiting period of ω slots, and it is successfully transmitted through the switch fabric. This implies that the offered waiting time is equal to ω, the impatience of the cell exceeds ω, and therefore the pdf of this event is given by

$$P_1(\omega) = \overline{F_\theta}(\omega) f_U(\omega) P_{\text{s|S}} \ . \tag{61}$$

The corresponding delay is equal to $\omega + \text{RTT}/2$.

Case 2) The cell is speculatively transmitted from the adapter, after a waiting period of ω slots, but it is not successfully transmitted through the switch fabric. It is subsequently transmitted through a grant after having waited y slots. This implies that the offered waiting time is equal to ω, the impatience of the cell is equal to y exceeding ω, and therefore the pdf of this event is given by

$$P_2(y) = \left(\int_0^y f_U(\omega) d\omega \right) f_\theta(y) (1 - P_{\text{s|S}}) \ . \tag{62}$$

The corresponding delay is equal to $y + \text{RTT}/2$.

Case 3) The cell is not speculatively transmitted from the adapter but instead transmitted through a grant after having waited y slots. This implies that the impatience of the cell is equal to y, the offered waiting time exceeds y, and therefore the pdf of this event is given by

$$P_3(y) = \left(\int_y^\infty f_U(\omega) d\omega \right) f_\theta(y) \ . \tag{63}$$

The corresponding delay is equal to $y + \text{RTT}/2$. Note that $\int_0^\infty P_1(\omega) d\omega + \int_0^\infty [P_2(y) + P_3(y)] dy = 1$.

Combining the three cases by unconditioning on ω and y, (61), (62) and (63), after some manipulations and using (36), yield the mean delay as follows:

$$\begin{aligned} \overline{D_i} \ &= \ \frac{\text{RTT}}{2} + \overline{\theta} \\ &- P_{\text{s|S}} \int_0^\infty \left[\int_\omega^\infty y f_\theta(y) dy - \omega \overline{F_\theta}(\omega) \right] f_U(\omega) d\omega \ . \end{aligned} \tag{64}$$

Let us now consider the delay D_o from the instant a cell is transmitted through the switch fabric until the instant it starts its transmission at the corresponding output port. The mean D_o is given by

$$\overline{D_o} \ = \ \frac{\text{RTT}}{2} + \overline{W_B} \ , \tag{65}$$

with $\overline{W_B}$ given by (33). Thus, by virtue of (64) and (65), the mean switch delay is given by

$$\begin{aligned} \overline{D} \ &= \ \overline{D_i} + \overline{D_o} = \text{RTT} + \overline{W_B} + \overline{\theta} \\ &- P_{\text{s|S}} \int_0^\infty \left[\int_\omega^\infty y f_\theta(y) dy - \omega \overline{F_\theta}(\omega) \right] f_U(\omega) d\omega \ . \end{aligned} \tag{66}$$

Substituting (35) into (66) and using (36), after some manipulations, yields

$$\begin{aligned} \overline{D} \ &= \ \overline{D_i} + \overline{D_o} = \text{RTT} + \overline{W_B} + \overline{\theta} \\ &- P_{\text{s|S}} \int_0^{\overline{X_g}} \left(\overline{\theta} - \omega + \frac{Q}{2\overline{X_g}} \omega^2 \right) f_U(\omega) d\omega \ . \end{aligned} \tag{67}$$

which in turn yields (48) by denoting

$$I_j \ \triangleq \ \int_0^{\overline{X_g}} \omega^j f_U(\omega) d\omega \ , \quad \text{for} \ \ j = 0, 1, 2 \ . \tag{68}$$

By making use of (37), the quantities I_0, I_1, and I_2 defined above are derived by (49), (50), and (51), respectively. ■

REFERENCES

[1] R. Hemenway, R. Grzybowski, C. Minkenberg, and R. Luijten, "Optical-packet-switched interconnect for supercomputer applications," *OSA J. Opt. Netw.*, vol. 3, no. 12, pp. 900–913, Dec. 2004.

[2] E. Oki, R. Rojas-Cessa, and H. Chao, "A pipeline-based approach for maximal-sized matching scheduling in input-buffered switches," *IEEE Commun. Lett.*, vol. 5, no. 6, pp. 263–265, June 2001.

[3] C. Minkenberg, I. Iliadis, and F. Abel, "Low-latency pipelined crossbar arbitration," in *Proc. IEEE GLOBECOM 2004*, Dallas, TX, Dec. 2004, paper no. GE15-2.

[4] C. Minkenberg, R. Luijten, F. Abel, W. Denzel, and M. Gusat, "Current issues in packet switch design," *ACM Computer Commun. Rev.*, vol. 33, no. 1, pp. 119–124, Jan. 2003.

[5] C. Minkenberg, F. Abel, P. Müller, R. Krishnamurthy, and M. Gusat, "Control path implementation of a low-latency optical HPC switch," in *Proc. Hot Interconnects 13*, Stanford, CA, Aug. 17–19 2005, pp. 29–35.

[6] C.-S. Chang, D.-S. Lee, and Y.-S. Jou, "Load-balanced Birkhoff-von Neumann switches, part I: one-stage buffering," *Elsevier Computer Communications*, vol. 25, pp. 611–622, 2002.

[7] A. Tanenbaum, *Computer Networks*, 3rd ed. Prentice Hall, 1996.

[8] H. Takagi, *Queueing Analysis, Volume 3: Discrete-Time Systems*. North Holland, 1993.

[9] A. Movaghar, "On queueing with customers impatience until the beginning of service," *Queueing Systems*, vol. 29, pp. 337–350, 1998.

Ilias Iliadis received a B.S. degree in Electrical Engineering in 1983 from the National Technical University of Athens, Greece, an M.S. degree in 1984 from Columbia University, New York, as a Fulbright Scholar, and a Ph.D. degree in Electrical Engineering in 1988, also from Columbia University. He has been at the IBM Zurich Research Laboratory since 1988. He was responsible for the performance evaluation of the IBM's PRIZMA switch chip. His research interests include performance evaluation of computer communication networks, traffi c control and engineering for IP and ATM networks, switch architectures, and stochastic systems. He holds several patents.

Dr. Iliadis is a member of IFIP Working Group 6.3, Sigma Xi, and the Technical Chamber of Greece. He has served as a Technical Program Co-Chair for the IFIP Networking 2004 Conference.

Cyriel Minkenberg received MS and PhD degrees in electrical engineering from the Eindhoven University of Technology, The Netherlands, in 1996 and 2001, respectively. Since 2001, he has been a research staff member at the IBM Zurich Research Laboratory, where he has contributed to the design and evaluation of the IBM PowerPRS switch family. Currently, he is responsible for the architecture and performance evaluation of the crossbar scheduler for the OSMOSIS optical supercomputer interconnect.