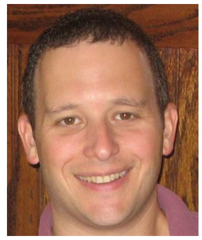
A Survey of Buffer Management Policies for Packet Switches[1]

Michael H. Goldwasser[2]



Abstract

Over the past decade, there has been great interest in the study of buffer management policies in the context of packet transmission for network switches. In a typical model, a switch receives packets on one or more input ports, with each packet having a designated output port through which it should be transmitted. An online policy must consider bandwidth limits on the rate of transmission, memory constraints impacting the buffering of packets within a switch, and variations in packet properties used to differentiate quality of service. With so many constraints, a switch may not be able to deliver all packets, in which case some will be dropped.

In the online algorithms community, researchers have used competitive analysis to evaluate the quality of an online policy in maximizing the value of those packets it is able to transmit. In this article, we provide a detailed survey of the field, describing various models of the problem that have been studied, and summarizing the known results.

1 Introduction

Packet switches play an integral role at the lower levels of network communication. For this reason, the development of effective switching policies is of great importance. To analyze these policies, the networking community has historically relied on real-world trace data, or stochastic models for network traffic. The introduction of these problems to the theory community began with three papers that applied competitive analysis to the domain: the INFOCOM 2000 paper by Aiello et al. [AMRR00], the PODC 2000 paper by Mansour, Patt-Shamir, and Lapid [MPSL00], and the STOC 2001 paper by Kesselman et al. [KLM+01]. Early algorithmic work in this area is surveyed by Azar [Aza04] and by Epstein and van Stee [EvS04], and subsequent results are mentioned in past SIGACT News columns by Jawor [Jaw05] and Chrobak [Chr08]. However, this remains an active area of research and our goal with this article is to provide an updated survey of algorithmic work related to switching policies and buffer management problems.

A typical model for a network switch contains a set of input ports and a set of output ports. For a slotted time model, one packet may arrive per input port at a given time step, with each packet designating a particular output port through which it is to be delivered. At each step, some constant number of packets can be transmitted through each output port. The switch's internal fabric allows for the transfer of packets from the input ports to the output ports, with the precise latency depending on the structure of the fabric and the switching policy. A schematic diagram of a switch is given in Figure 1.

[2]Dept. of Mathematics and Computer Science, Saint Louis University, St. Louis, MO 63103. `goldwamh@slu.edu`.

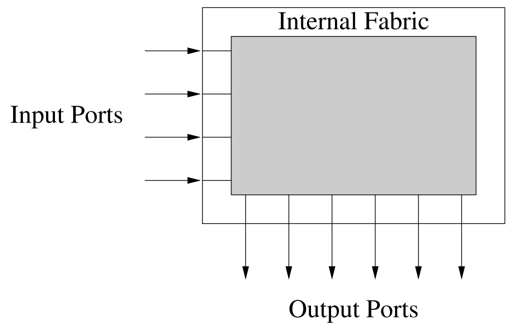


Figure 1: Schematic of a typical network switch

Because of the latency and bandwidth constraints, packets must be buffered at various places within the switch and some packets may be dropped. Networks with quality-of-service (QoS) considerations can be modeled with $w_p > 0$ designating the weighted value of packet p. For all models that we consider, the goal of a policy is to maximize the weighted throughput of the switch, that is, the sum of the values of those packets that are successfully transmitted. Competitive analysis is used to measure the effectiveness of an online switching policy relative to the optimal (offline) performance on the same instance [ST85, KMRS88]. We defer to the book by Borodin and El-Yaniv for formal definitions of deterministic and randomized competitiveness in the context of a maximization problem [BE98].

In this article, we survey research involving various models of buffer management problems. Packets might be buffered at the input ports, the output ports, within the switch fabric, or some combination of these. Models vary based on buffering constraints such as whether buffers have a maximum capacity, and if so, whether multiple buffers within the switch have independent dedicated memory or are sharing a common memory pool. Other factors include whether packets can be dropped once they have been added to a buffer, and whether they can be reordered within a buffer.

Our survey begins by examining the fundamental case of a single output port with a dedicated buffer. Results for this model apply to the more general case when there are multiple output ports with each port maintaining an *independent* buffer of size B. In such a scenario, a buffer management policy can be applied individually to each port. An arbitrary number of packets may arrive per time step, but an output port is limited to transmitting some constant number of packets per step (typically, one). In Section 2, we consider a model in which a buffer has a maximum capacity of B packets and the subset of packets that are transmitted must be sent in first-in, first out (FIFO) order. The second model, considered in Section 3, relaxes the restrictions on the buffer having an explicit capacity and FIFO semantics. Instead, each individual packet specifies a deadline by which it must be transmitted or dropped. This *bounded-delay* model is motivated by networks that guarantee a differentiated quality of service in regard to end-to-end transmission time.

In Section 4, we consider more general models for buffer management within a switch. The first case is when each output port has a dedicated FIFO buffer, but with a maximum capacity on the *combined* buffer sizes. Another model relies on packets being buffered at the *input* ports, with each input port maintaining an independent FIFO buffer of size B, and with a constraint on the internal switch fabric such that at most one packet can be transmitted from the system per time step. We also examine research involving more general *CIOQ switches*, with packets able to be buffered at both input and output ports, and *crossbar switches*, which have additional buffers available in the internal switch fabric.

In Section 5, we review work that relies on resource augmentation in the competitive analysis of buffer management problems. In this line of research, online policies are developed for switches that are assumed to have a larger buffer or faster throughput than that used in a reference solution.

In Section 6, we examine several closely related models introduced in recent years. Most notably, we discuss a SODA 2009 paper by Bienkowski et al. introducing a generalization of the bounded-delay model based on "collecting weighted items from a dynamic queue" [BCD+09], a SODA 2008 paper by Fiat, Mansour, and Nadav introducing a model in which the value of a packet degrades due to time spent in the buffer [FMN08], and an IPDPS 2009 paper by Kesselman, Patt-Shamir, and Scalosub introducing a model with inter-packet dependencies [KPSS09]. We conclude in Section 7, with a list of open questions for future research.

2 FIFO Model

In this section, we consider the following model for managing a single output port. Packets arrive, one at a time, destined for the particular output port. At regular intervals, the switch is given an opportunity to transmit one packet through the port (or more generally, up to m packets, for some constant bandwidth $m \geq 1$). Transmitted packets must be sent in FIFO order, and a dedicated buffer can hold at most B packets at any point in time. If there are B previously buffered packets and a new one arrives, a packet (possibly the arriving one) must be dropped to adhere to the capacity requirement[3]. Two submodels have been considered. In a *nonpreemptive* model, an arriving packet must be immediately dropped or else accepted into the FIFO buffer and eventually transmitted. In a *preemptive* model, packets can be inserted into the buffer yet later dropped. We note that an offline solution can always be constructed without preemptions, since it need only to buffer packets that will eventually be transmitted. Furthermore, the FIFO requirement is irrelevant in the offline setting, as any sequence of transmittals can be reordered to FIFO without changing the number of packets that are buffered at any given time.

2.1 Nonpreemptive Model

Aiello et al. introduce the nonpreemptive model by studying a restricted *two-valued* case in which all packets have value either 1 or $\alpha > 1$ [AMRR00, AMRR05]. They prove that any policy, even with randomization, can be at best $(2 - \frac{1}{\alpha})$-competitive. The proof relies on two possible constructions, both beginning with the arrival of B low-valued packets. In the first construction, nothing else arrives and so the optimal solution is to accept all packets. In the second construction, B high-valued packets are subsequently released during the same time step, in which case the optimal solution is to reject all of the low-valued packets and accept the high-valued ones. However, an online policy must accept some number of low-valued packets before knowing whether more packets are coming. Formally, let random variable x denote the number of low-valued packets accepted by an online policy. With the first instance, the randomized competitiveness is $\frac{B}{E[x]}$. On the second

[3]We note that some researchers consider a slotted-time model in which the switching policy considers the entire batch of arriving packets during a given time-slot, perhaps with flexibility in reordering those packets. Furthermore, some authors let B denote the maximum size of the buffer as it exists *after* the transmission that ends a time-slot, thereby allowing strictly more than B packets to be held for transient times when considering new arrivals.

instance, the competitive ratio is at most $\frac{B\alpha}{E[x+(B-x)\alpha]} = \frac{B\alpha}{B\alpha-(\alpha-1)E[x]}$. The lower bound of $2-\frac{1}{\alpha}$ comes by taking the worse of the two ratios, with the balance achieved when $E[x] = \frac{B\alpha}{2\alpha-1}$.

Aiello et al. analyze the competitive ratio of five online policies, considering the specific cases of $\alpha = 1$, $\alpha = 2$, and $\alpha = \infty$, but none of those policies match the lower bound. Andelman, Mansour, and Zhu provide a deterministic policy achieving optimal $(2-\frac{1}{\alpha})$-competitiveness [AMZ03, Zhu04]. Their Ratio Partition policy accepts each high-valued packet when possible, and subsequently "marks" the earliest $\frac{\alpha}{\alpha-1}$ low-valued packets in the queue (if any). A newly released low-valued packet is accepted so long as doing so would still leave a number of empty buffer slots that is at least $\frac{\alpha-1}{\alpha}$ times the number of currently unmatched low-valued packets.

For the more general case in which packets may have arbitrary values between 1 and $\alpha > 1$, Andelman, Mansour, and Zhu prove the optimal competitiveness to be $\Theta(\log\alpha)$ [AMZ03, Zhu04]. The lower bound is a natural extension of the two-valued construction. Rather than offering B packets of value 1 followed possibly by B packets of value α, an adversary proceeds by offering B packets of value 1, then B packets of value $(1+\epsilon)$, then B of value $(1+\epsilon)^2$ and so on. The optimal strategy is to wait for the last wave of B packets, but an online policy cannot predict the number of waves so it must accept enough packets from each wave to insure its competitiveness if that wave were the last. An analysis leads to the deterministic lower bound of $1+\ln(\alpha)$; the journal version of this paper contains a variant of this construction using Yao's technique to provide a *randomized* lower bound of $\frac{\log\alpha+2}{2}$. On the positive side, they provide an asymptotically optimal $e\ln(\alpha)$-competitive deterministic policy by dividing the range $[1,\alpha]$ into n classes of values, and reserving a portion of the buffer capacity for each of these classes. The upper bound is improved to $2+\ln(\alpha)+O(\ln^2(\alpha)/B)$ by Andelman and Mansour [AM03], matching the lower bound of $1+\ln(\alpha)$ up to a constant additive factor.

2.2 Preemptive Model

The *preemptive* version of the FIFO model was introduced by Mansour, Patt-Shamir, and Lapid in the context of video streaming [MPSL00, MPSL04]. The lower bounds presented in Section 2.1 do not apply, because an online policy can hold low-valued packets but later preempt them in favor of newly-arrived high-valued packets. As an illustrative example, consider the following construction showing that a classic Greedy policy is not significantly better than 2-competitive [KLM+01, KLM+04]. At time 0 for a slotted model, $B-1$ packets with value 1 arrive followed by a single packet of value $\alpha > 1$. For each time slot from 1 through $B-2$, an additional packet with value α arrives. Finally, at time $B-1$, a burst of B additional α-packets arrives. Prior to the final burst, the buffer capacity is never a direct constraint; it is possible to accept all packets, always transmitting the earliest (low-valued) packet. Therefore, at the onset of time $B-1$, such a greedy policy will have transmitted the original $B-1$ low-valued packets and will currently have $B-1$ high-valued packets buffered. Unfortunately, it can effectively accept only one of the final burst of B packets, achieving a total value of $B-1+\alpha B$. In contrast, Opt will have ignored all of the low-valued packets, scheduling a high-valued packet at each time slot from 0 to $B-2$, and then accepting all B of the final burst. The competitive ratio on this instance is thus $\frac{\alpha(2B-1)}{B-1+\alpha B}$. For large α and B this approaches 2. For fixed α but large B it approaches $2-\frac{2}{\alpha+1}$, and for fixed B but large α it approaches $2-\frac{1}{B}$.

deterministic		randomized			
upper	lower	upper	lower	citation	comments
	1.282			[KLM+01, KLM+04]	
$2 - \frac{2}{\alpha+1}$				[KLM+01, KLM+04]	Greedy
1.894§				derived from [KM01]	best of Greedy and β-Preemptive
1.544§				derived from [KM03]	improved analysis of β-Preemptive
1.304				[LPS02, LPS03]	Mark&Flush
		1.25	1.197	[And05]	Randomized Mark&Flush
1.282				[EW06, EW09]	Account Strategy (ACC)

§ memoryless

Table 1: Bounds for the two-valued, preemptive FIFO model with $m = 1$ and $B \to \infty$

Two-valued model

The preceding lower bound on the greedy policy relied on jobs with value either 1 or $\alpha > 1$. This construction forms the basis for a lower bound against any deterministic policy for the two-valued model. To have better performance, a policy must preempt some of the low-valued packets prior to time $B - 1$. Yet, an adversary can immediately terminate the construction if a policy were to preempt too many of the low-valued packets. These factors are balanced as follows. For a fixed α and B, let t denote the first time at which an α-valued packet would be transmitted when following a given online deterministic policy on the base construction. We consider two possible instances. In the first case, the construction ends with a single high-valued packet arriving at time t. In this case, Opt achieves $B - 1 + \alpha(t + 1)$, while the online policy achieves $t + \alpha(t + 1)$ for a ratio of $\frac{B-1+\alpha(t+1)}{t+\alpha(t+1)}$. In the second instance, a burst of B high-valued packets arrives at time t. In this case, the online policy achieves $t + \alpha B$, while Opt achieves $\alpha(B + t)$ by sending t high-valued packets from 0 through $t - 1$ followed by the final burst of B packets. For a fixed t, an adversary can select the worse of the two instances, providing a lower bound of $\max\left(\frac{B-1+\alpha(t+1)}{t+\alpha(t+1)}, \frac{\alpha(B+t)}{t+\alpha B}\right)$. For a fixed α and B, an online policy can select t to minimize that maximum. An adversary can thus choose an α that leads to the strongest such lower bound. Kesselman et al. perform numerical analysis[4] concluding that the adversary should pick $\alpha \approx 4.01545$, in which case the competitive ratio is no better than approximately 1.282 [KLM+01, KLM+04].

Table 1 presents a summary of all two-valued results; we defer discussion of the randomized case until later in this section. The earliest policy to be analyzed is the deterministic Greedy policy, defined (for general values) as follows. The switch always accepts a new packet if there is available space in the buffer. Preemptions only occur when a new packet risks overflowing the buffer, in which case the lowest-valued packet, among the new packet and all those currently buffered, is dropped. Kesselman et al. give a tight analysis of Greedy as $(2 - \frac{2}{\alpha+1})$-competitive for the two-valued model [KLM+01, KLM+04]. Kesselman and Mansour propose another policy for a related model of *loss-bounded* analysis [KM01, KM03]; those results, if converted to the throughput-competitiveness, lead to a 1.544-competitive upper bound for any α (we discuss this conversion shortly). Lotker and Patt-Shamir provide a policy with competitive ratio of $1 + \frac{1}{\sqrt{\alpha}} + O(\frac{1}{\alpha})$ that is

[4]In the published result, Kesselman et al. present a slightly different construction and claim a lower bound of $\max\left(\frac{B+\alpha t}{t+\alpha t}, \frac{\alpha(B+t)}{t+\alpha B}\right)$, but that analysis seems to unfairly favor the adversary. The ratios that we present match those of Englert [Eng08].

never more than 1.304 for any α [LPS02, LPS03]. Englert and Westermann settle the two-valued case, providing a deterministic policy with competitiveness that tends to 1.282 for large B and never worse than 1.303 for any B [EW06, EW09]. Englert provides a slightly modified policy that achieves optimality for any combination of α and B [Eng08].

The optimal policy of Englert and Westermann is rather simple in design [EW06, EW09]. Named the ACCOUNT STRATEGY (ACC), it is defined for parameter $x \geq 1$, where x represents the desired competitiveness. It maintains an account a that amasses $(x-1)$ units of credit for each unit of value that is, or will be, transmitted by the switch. One unit of that credit can subsequently be used to preempt a low-valued packet that reaches the front of the queue. Formally, a newly released packet p is accepted if there is room in the buffer or if p is a high-valued packet that can be added at the expense of the earliest low-valued packet in the buffer. If high-valued p has been added to the buffer, account a is immediately increased by $(x-1) \cdot \alpha$. Note that a high-valued packet is never preempted, so its value is guaranteed; in contrast, a low-valued packet added to the buffer is not guaranteed to be transmitted, so it cannot be immediately credited on its arrival. After considering all new packets at a given time step, the strategy determines a packet to send. So long as $a \geq 1$ and a low-valued packet is at the front of the queue, that packet is preempted and the account is decreased by 1. If a low-valued packet is eventually sent (presumably, when $a < 1$), then the account is credited with $x-1$ units as a reward for the transmitted value. As a technical condition, the account is reset to 0 whenever the buffer becomes empty, or entirely filled with high-valued packets. The analysis proceeds in regions defined by those events upon which the account is reset. If a low-valued packet in the optimal solution is mistakenly preempted, the unit of credit that allowed its preemption can be paired with $\frac{1}{x-1}$ units of achieved value, tending toward a competitive ratio of $\left(1+\frac{1}{x-1}\right)/\frac{1}{x-1} = x$. Preempting more aggressively would put the online strategy at risk of losing its competitiveness if nothing else were to arrive. The optimal choice of $x \approx 1.282$ stems from a balancing condition in the case when the online strategy later rejects high-valued packets with its buffer full of other high-valued packets (that may have been transmitted earlier in the optimal schedule).

We note that the earlier work of Kesselman and Mansour considered the competitiveness when minimizing the value of dropped packets [KM01, KM03]. Although this measure is dual to throughput maximization, competitive ratios are not the same for the two objectives (a distinction common to most pairs of minimization and maximization problems). Kesselman and Mansour show that for the minimization problem, the optimal deterministic policy is α-competitive, even for the two-valued model. They introduce what they term *loss-bounded* analysis in which the lost value of an online policy is compared relative to the loss of the offline optimal solution together with an additive factor related to the throughput of a policy. In that context, they propose and analyze a β-PREEMPTIVE GREEDY policy that allows an accepted packet with value w_p to immediately offset the preemption of a leading subset of packets from the FIFO queue having combined value less than or equal to w_p/β. In this regard, it seems similar to the Englert and Westermann policy, but it is a *memoryless* policy as the credit for a new large job cannot be saved for later use (the β-PREEMPTIVE policy also has some similarity to the best-known policy for the general model, discussed shortly). Kesselman and Mansour perform their loss-bounded analysis of this policy and show how to convert that back to traditional throughput maximization bounds. In the conference version of their work, the throughput-competitive bound for their policy is $\sqrt{\alpha}/(\sqrt{\alpha}-2)$. Choosing the better of this policy and the standard GREEDY policy leads to an upper bound that is at most 1.894 for any α. The improved analysis in the journal version of the paper shows that β-PREEMPTIVE

deterministic		randomized			
upper	lower	upper	lower	citation	comments
4				[MPSL00, MPSL04]	Greedy
2				[KLM+01, KLM+04]	Greedy (applies to $m \geq 1$)
	1.282			[KLM+01, KLM+04]	Lower bound for two-valued system
	1.414			[AMZ03, Zhu04]	
	1.419			[KMvS03, KMvS05]	
	1.434§			[AMZ03, Zhu04]	
1.983				[KMvS03, KMvS05]	Preemptive-Greedy (PG)
1.75				[BFK+04]	Better analysis of (modified) PG
			1.25†	[BCC+04, CCF+06]	implied from s-uniform model
		1.75		[And05]	Simple "comparison based" policy
1.732				[EW06, EW09]	Better analysis of (modified) PG

§ For special case of $B = 2$, † For special case as $B \to \infty$

Table 2: Bounds for the general-valued, preemptive FIFO model
(for $m = 1$ and any B, unless noted otherwise)

is $(1 + \sqrt{\alpha})/\sqrt{\alpha}$-competitive; choosing the better of this policy and Greedy leads to an overall upper bound of 1.544 for any α. Although this does not match the optimal 1.282-competitiveness, it remains the best-known *memoryless* policy for the two-valued case.

General values

In contrast to the *nonpreemptive* model, it is possible to achieve constant competitiveness with general-valued packets in the preemptive model. However, the precise constant remains unknown. The complete progression of results for this model is summarized in Table 2. The strongest lower bound for the general case is 1.419, as shown by Kesselman, Mansour, and van Stee [KMvS03, KMvS05]. Their construction proceeds in phases, starting with B packets of value 1, and then a series of packets of value $(1 + \epsilon)$, continuing until the policy dictates that one of the higher-valued packets is to be sent. At that time, the adversary starts releasing a series of packets of value $(1+\epsilon)^2$, until one of those is to be transmitted, continuing in such phases with a geometrically increasing sequence of packet values. For any value of B, an adversary can terminate such a construction to force competitiveness of 1.419 or higher. A slightly stronger lower bound of 1.434 is known for the special case of $B = 2$ [AMZ03, Zhu04].

On the positive side, two deterministic policies have been successfully analyzed with general values. The first of these is the Greedy policy, as discussed in the preceding discussion of the two-valued case. Initial analysis by Mansour, Patt-Shamir, and Lapid shows that Greedy is at worst 4-competitive with general values [MPSL00, MPSL04]. Kesselman et al. prove a tight bound of 2-competitive, even for bandwidth $m > 1$ [KLM+01, KLM+04]. They provide a more detailed analysis parameterized by the values of m, B, and α, and show that in the case of a tie when preempting the lowest-valued packet, the best strategy is to drop the *earliest* of those packets. An alternative proof of the 2-competitiveness, due to Kimbrel [Kim04b], is presented in Epstein and van Stee's survey on buffer management [EvS04].

Kesselman, Mansour, and van Stee [KMvS03, KMvS05] introduce a policy named PREEMPTIVE-GREEDY (PG), not to be confused with the earlier β-PREEMPTIVE GREEDY policy [KM01, KM03]. PG behaves similarly to GREEDY, except that when a packet p with value w_p arrives, the switch immediately drops the earliest packet in the queue (if any) that has a value less than or equal to w_p/β (recall that the policy of [KM01, KM03] allowed for a group of packets with combined weight w_p/β to be preempted, but only from the front of the queue). Kesselman et al. show that PG is 1.983-competitive with $\beta = 15$, and that for any setting of β, the policy is at best $\phi \approx 1.618$-competitive. Bansal et al. consider a minor modification [BFK$^+$04]. When choosing a packet to preempt in the first step, they choose the first one in the queue that has a value both less than w_p/β and less than the packet immediately following it (if any). They prove that their modified policy is 1.75-competitive for $\beta \geq 4$. Englert and Westermann improve the analysis to show that the modified PG is 1.732-competitive for $\beta = 2 + \sqrt{3}$, and that the policy is at best 1.707 for any choice of β [EW06, EW09].

Randomized Policies

We conclude this section by discussing the use of randomization in the preemptive FIFO model. For the two-valued version of the problem, Andelman demonstrates that randomization can be used to improve the competitiveness [And05]. While the optimal deterministic competitiveness is known to be 1.282, Andelman presents a policy that is $(1 + \frac{1}{\sqrt{\alpha}} - \frac{1}{\alpha}) \leq 1.25$-competitive against an oblivious adversary. He also presents a randomized lower bound of 1.197 against an oblivious adversary. His RANDOMIZED MARK&FLUSH policy makes an a priori selection between two deterministic policies: MARK&FLUSH or GREEDYHIGH. The MARK&FLUSH policy, due to Lotker and Patt-Shamir [LPS02, LPS03], is 1.304-competitive and was the best-known policy at the time RANDOMIZED MARK&FLUSH was developed. The GREEDYHIGH policy is one that rejects all low-valued packets and greedily executes high-valued packets.

For the general-valued case, Andelman presents another barely random policy that achieves a competitive ratio of 1.75 against an oblivious adversary. While this bound is no better than what was known at the time for the (deterministic) PG policy, the randomized policy is significantly simpler. In particular, it is *comparison based*, as its behavior depends on the relative order of the packets' values but not on the actual values. The policy makes an a priori selection between one of two deterministic policies: GREEDY or HALFGREEDY. The standard GREEDY policy is already known to be 2-competitive in the worst case. The HALFGREEDY policy accepts new packets using the greedy rule for dropping the least valuable in the case of overflow. In deciding what packet to transmit, it relies on an online computation of the optimal solution. In the opening of Section 2, we note that an optimal solution can be effectively computed while ignoring the FIFO requirement. This computation can be done online by always buffering the B most valuable pending packets and transmitting the most valuable among those. The HALFGREEDY policy marks packets in its current buffer that have already been sent by the non-FIFO optimal schedule. If the number of marked packets is less than or equal to $\frac{B}{2}$, it continues by sending the earliest packet in the buffer. Otherwise, it sends the earliest of those marked packets (dropping all unmarked packets ahead of it). While the 1.75-competitiveness of this randomized policy does not surpass the best-known deterministic bound, there is much room for improvement; the strongest lower bound for this setting is 1.25 (due to a similar bound on the s-uniform model introduced in the following section).

3 Bounded-Delay Model

In this section, we consider a model in which each packet has an additional parameter specifying the maximum delay that it can withstand in the switch. In particular a packet with *span* s must be transmitted in one of the first s time slots after its arrival. As originally modeled by Kesselman et al. [KLM$^+$01, KLM$^+$04], the switch is not constrained by a FIFO requirement and there is no explicit constraint on the buffer capacity (subsequent papers have reintroduced a capacity restriction; see Section 6.4). This bounded-delay model for a switch can be viewed as a classic scheduling problem, namely that of maximizing the weighted throughput of unit-length jobs with integral release times and deadlines. Using standard scheduling nomenclature, the bounded-delay switch model with bandwidth m is equivalent to $Pm \mid r_j, p_j = 1 \mid \sum w_j(1 - U_j)$. We note that the offline version of this problem can be solved in polynomial time by reducing it to an instance of the assignment problem, with each time-slot for each machine matched to a job that can be feasibly scheduled at that time. To better understand the bounded-delay model, several restricted variants have been considered.

- **s-bounded**

  In this model, the switch has a priori knowledge that each packet's span will be at most s. This model is essentially the same as the general case, with the added possibility that a policy, or its analysis, can be tuned based on knowledge of s.

- **agreeable deadlines (a.k.a. similarly ordered)**

  In this model, packets' inherent deadlines must be ordered consistently with their arrivals. That is, if packet p_j arrives after packet p_i, then packet p_j must have a deadline that is later than or equal to that of p_i.

  It is worth noting that all 2-bounded instances have agreeable deadlines. Any previously buffered packet in a 2-bounded setting must have imminent deadline, and therefore no later than the deadline of a newly arriving packet.

- **s-uniform**

  In this model, all packets must have a common span of s. As such, this is itself a special case of agreeable deadlines. Kesselman et al. note that the optimal throughput for an s-uniform instance is identical to that for the same arrival sequence in the FIFO model with buffer size s [KLM$^+$01, KLM$^+$04]. As a consequence, any online policy that is c-competitive for the FIFO model is trivially c-competitive for the s-uniform model, as no packets remain in the buffer for more than s time-slots. The converse is not necessarily so, as an online policy for the s-uniform model might send a high-value packet out of order, deferring the decision of whether to transmit an earlier, lower-valued packet.

In addition to these restrictions on packet parameters, the achievable competitiveness may depend on further restrictions placed upon an online policy. In particular, an online policy is *memoryless* if each decision that it makes is based solely on the set of pending packets at that time, and *scale-invariant* if its behavior on instance $\mathcal{I}$ is the same as on an instance $\mathcal{I}'$ with identical structure but packet values that are scaled by a positive constant. Table 3 summarizes the history of research for the various combinations of conditions. We discuss the most significant of those results in the remainder of this section.

restriction	deterministic upper	deterministic lower	randomized upper	randomized lower	citation
General	$2^{\S\dagger}$				GREEDY [Haj01, KLM+01, KLM+04]
			$1.582^{\S\P}$		RMIX [BCC+04, CCF+06]
	1.939				GENFLAG [CJST04, CJST07]
	1.854				DP_{ϕ} [LSS07]
	$1.893^{\S}$				[EW07]
	1.828				[EW07]
		1.618		1.25	implied from 2-bounded model
				$1.33^{\P}$	implied from 2-bounded model
s-bounded	$2-\frac{2}{s}+o(\frac{1}{s})^{\S}$				EDF_{1/λ_s} [BCC+04, CCF+06]
4-bounded	$1.732^{\S}$				$EDF_{\sqrt{3}}$ [BCC+04, CCF+06]
3-bounded	$1.618^{\S}$				$EDF_{\phi-1}$ [BCC+04, CCF+06]
2-bounded	$1.618^{\S\dagger}$				β-EDF [KLM+01, KLM+04]
		$1.17^{\dagger}$			[KLM+01, KLM+04]
		1.414			[KLM+01, KLM+04]
		1.618			[Haj01, AMZ03, CF03]
				1.25	[CF03]
			$1.25^{\S}$		R2B [BCC+04, CCF+06]
				$1.33^{\P}$	[BCJ08]
			$1.33^{\S\P}$		RAND [BCJ08]
agreeable deadlines		1.618		1.25	implied from 2-bounded model
				$1.33^{\P}$	implied from 2-bounded model
	1.838				SIMFLAG [CJST04, CJST07]
	$1.618^{\S *}$				MG [LSS05]
			$1.33^{\S}$		RG [Jeż10]
s-uniform	$1.618^{\S}$		$1.33^{\S}$		implied from agreeable deadlines
		1.377			implied from 2-uniform case
				1.25	limit as $s \to \infty$ [CCF+06]
2-uniform	$1.434^{\S}$				β-EDF [KLM+01, KLM+04]
		$1.11^{\dagger}$			[KLM+01, KLM+04]
		1.25			[KLM+01, KLM+04]
	$1.414^{\S}$				[AMZ03, Zhu04]
		1.366			[AMZ03, Zhu04]
		$1.414^{\S}$		1.17	[BCC+04, CCF+06]
	1.377				SWITCH [CJST04, CJST07]
		1.377			[CJST04, CJST07]
				$1.2^{\P}$	[BCJ08]
			$1.25^{\S}$		implied from 2-bounded
			$1.33^{\S\P}$		implied from 2-bounded

§ restricted to memoryless, scale-invariant policies

† applies for general m

* applies for any buffer size B

¶ versus adaptive adversary

Table 3: Progress for the bounded-delay model with $m = 1$

3.1 Lower Bounds

The strongest known lower bounds for the general model are all proven within the far more restricted domain of 2-bounded instances. That model suffices to present the quintessential dilemma for an online policy, the choice between a lower-valued packet with imminent deadline and a higher-valued packet with later deadline. No online policy can be strictly better than $\phi \approx 1.618$-competitive for the 2-bounded model [Haj01, AMZ03, CF03]; we discuss the lower bound construction in more detail in the following paragraph. The lower bound is known to be tight for the 2-bounded model [KLM+01, KLM+04] and the more general agreeable deadline model [LSS05]. It remains the strongest known lower bound for the general model. Similar techniques have been used to prove randomized lower bounds for the 2-bounded model of 1.25 versus an oblivious adversary [CF03], and 1.33 versus an adaptive adversary [BCJ08]. These bounds are also known to be tight for the 2-bounded case [BCC+04, CCF+06, BCJ08], and they remain the strongest known lower bounds for the most general case.

To present the deterministic lower bound of ϕ, we adopt the notation of Chin and Fung [CF03]. For arbitrarily small ϵ and arbitrarily large n, they define a sequence of increasing values for $0 \leq i \leq n$ with $v_i = (1-\epsilon)\phi^i + \epsilon(\phi+1)^i > \phi^i$. Starting with time $t = 0$, an adversary releases two packets at time t, one with value v_t and span 1, and the other with value v_{t+1} and span 2. This continues so long as the online policy chooses to transmit the lower-valued of the available packets (i.e., that with value v_t at time t). If the policy ever dictates that the higher-valued packet would be sent, the construction is terminated. To ensure a finite instance, the construction terminates at time n with the release of a single packet with value v_n and span 1. The analysis considers three cases. If an online policy dictates that the larger-valued packet be sent at time 0, then the competitive ratio is at least $\frac{v_0+v_1}{v_1} = \frac{1+\phi+\epsilon}{\phi+\epsilon} > \phi - \epsilon$. If the policy were to transmit the larger available packet for the first time at $t \geq 1$, the switch will have transmitted values $v_0 + \cdots + v_{t-1}$ prior to time t, and v_{t+1} at time t. In contrast, the optimal solution for this instance will be to send values $v_1 + \cdots + v_t$ prior to time t, followed by both v_t and v_{t+1}. This leads to a competitive ratio of at least $\frac{v_1+\cdots+v_t+v_t+v_{t+1}}{v_0+\cdots+v_{t-1}+v_{t+1}}$. The chosen values telescope in a way that ensures that this ratio is at least $\phi - \epsilon$. The final case to consider is when the policy never chooses the larger-valued packet and the construction ends at time n, in which case the ratio is $\frac{v_1+\cdots+v_n+v_n}{v_0+\cdots+v_{n-1}+v_n} \to \phi$ as $n \to \infty$.

Chin and Fung use a similar approach to produce the randomized lower bound of 1.25 versus an oblivious adversary [CF03]. Rather than releasing packets with values near ϕ^t and ϕ^{t+1} at time t, they use values 2^t and 2^{t+1} respectively. They apply Yao's principle to prove a randomized lower bound against an oblivious adversary, by proving a similar deterministic lower bound against a random distribution of instances. Specifically, they build a random distribution including the $n+1$ instances that result when terminating the arrival sequence at times $0, \ldots, n$. The instance ending at time t is chosen with probability $2^{-(t+1)}$ for $t < n$ and 2^{-n} for $t = n$. For this distribution, any deterministic policy will have expected value at most $2n + 1$, while the expected value of Opt is $\frac{5n}{2} + 1$. For arbitrarily large n, this provides the lower bound of $\frac{5}{4}$. The lower bound of 1.33 against an adaptive adversary uses a similar style of construction, but with the adaptive adversary intentionally making decisions that are more likely to differentiate its behavior from that of the randomized online policy.

The other interesting set of lower bounds are those for the 2-uniform case. Note that the 2-bounded constructions do not apply to the uniform setting, as they require some packets with span 1 and some with span 2. A similar dilemma can be forced by releasing two packets and allowing the switch to transmit one. At the next time step, one additional packet can be released, at which point the old packet has remaining span 1 and the new one span 2. But the corresponding lower bounds are weaker due to the additional packet sent in the first step. The first polices considered for the 2-uniform problem were all memoryless and scale-invariant, with the best being a $\sqrt{2} \approx 1.414$-competitive policy of Andelman, Mansour, and Zhu [AMZ03, Zhu04]. At that time, there remained a gap between that result and the strongest lower bound. Chin et al. provide a construction that shows that $\sqrt{2}$ is indeed the best possible deterministic competitiveness for policies that are memoryless and scale-invariant [BCC$^+$04, CCF$^+$06]. Their lower bound construction relies on forcing some settings that are scaled versions of a previous choice made by the policy, and therefore with predictable outcomes. Without the extra requirements, Chrobak et al. prove that the deterministic competitiveness is 1.377, demonstrating a lower bound and matching (memory-based) policy [CJST04, CJST07].

3.2 Upper Bounds

In this section, we describe several policies for the bounded-delay model, highlighting a few common themes. Most policies are defined in a way that ensures that each transmitted packet p is the earliest-deadline pending packet having value greater than or equal to w_p; an exchange argument shows that if there were another pending packet q with earlier deadline and at least as great a value, then q could be transmitted in lieu of p without loss of generality. Commonly, policies compare the merits of transmitting earliest-deadline packet e (in case of tie, the heaviest of the early packets) relative to the heaviest-valued packet h (in case of tie, the earliest of the heavy packets). Some policies base their immediate decision at a given time on the maximum-weight feasible subset of pending packets (the so called, *optimal provisional schedule*). For these policies, e is typically defined to be the earliest-deadline packet from the provisional subset (as opposed to the full set); heaviest h always belongs to the optimal provisional schedule.

Agreeable Deadlines

We begin by examining a deterministic policy of Li, Sethuraman, and Stein that achieves optimal competitiveness of $\phi \approx 1.618$ for the case of agreeable deadlines [LSS05]. Intuitively, the policy strikes a balance between scheduling the most valuable packet (regardless of deadline) and an early-deadline packet (so long as its value is reasonable). Formally, their Modified Greedy (MG) policy is defined for a model in which the buffer is bounded by size B (although B can be arbitrarily large in general). At each time step, it computes the optimal provisional schedule for the given buffer size, and then considers the earliest-deadline e and the heaviest-valued h from that subset. If $w_e \geq w_h/\phi$, e is sent; otherwise, the earliest-deadline packet f satisfying $w_f \geq \max(\phi w_e, w_h/\phi)$ is sent. The analysis technique is novel. It considers each action of MG relative to an Opt-like adversary, but after each time-step it artificially modifies the adversary's buffer to match that of MG. This is done while guaranteeing that the modifications to the buffer do not disadvantage the adversary, or else by awarding artificial value to the adversary as compensation.

Jeż [Jeż10] provides a simplification to MG that sends heaviest h (rather than defined f) in the case when $w_e < w_h/\phi$. Similar analysis shows that it remains ϕ-competitive. Jeż also provides a new randomized policy that is 1.33-competitive against an oblivious adversary. This Random Greedy (RG) policy is similar to MG, but rather than choose deterministically between e and h, it transmits e with probability $\frac{w_e}{w_h}$, and h otherwise. The $\frac{4}{3}$-competitive analysis uses the same technique, manipulating an adversary's buffer after each step to force it to a similar state as the random RG. This analysis requires an assumption at times that the adversary schedules its pending packets in EDF order. For this reason, the analysis only applies to an oblivious adversary, as an adaptive adversary cannot be restricted in such fashion.

General Model

For the general model, the Greedy policy that always transmits the most valuable pending packet is 2-competitive [Haj01, KLM+01, KLM+04]. The proof can be shown with a standard charging scheme and the analysis is tight. There has been a progression of results improving on this upper bound, but there remains a gap between the 1.618 lower bound.

The first deterministic policy to break the 2-competitive barrier for the general case is GenFlag, by Chrobak et al. [CJST04, CJST07]. This always sends either the earliest-deadline e or heaviest-valued h, without restricting consideration to the optimal provisional schedule. The policy relies on one bit of memory about the previous time step to determine the policy for the current step. They prove that a tuned version of the policy is $\frac{64}{33} \approx 1.939$-competitive.

Li, Sethuraman, and Stein provide a 1.854-competitive policy named DP [LSS07]. Its origin stems from the authors' MG policy for agreeable deadlines [LSS05]. Both policies compute the optimal provisional schedule and consider sending the earliest deadline packet e from that set, or else some packet f with weight meeting a threshold condition relative to the heaviest packet h. The hallmark feature of DP is the introduction of *dummy packets* into the buffer to encode relevant information about the past. Those dummy packets are never sent, but are used to influence the decision making process. In particular, whenever it sends a packet f other than e or h, a dummy packet h' is generated and paired with the real h. This dummy packet is given weight w_h/α for parameter α, and deadline d_f. Furthermore, a status bit is set for the real h that has the effect of artificially reducing its value to w_h/α. This makes h less likely to be considered the "heaviest" packet in future steps, yet if it is chosen as h again, it is automatically sent. If the policy ever chooses an f that is a dummy packet, its corresponding real packet will be sent. The authors prove that DP has competitive ratio upper bounded by $\max\{\alpha, \frac{\alpha+1}{\alpha}, \frac{3}{\alpha}, \frac{3\alpha}{\alpha+1}\}$. This expression is minimized at $3/\phi \approx 1.854$ by setting $\alpha = \phi$. This analysis is not believed to be tight; the worst known example for DP forces a competitive ratio of 1.764.

Englert and Westermann's 1.828-competitive deterministic policy is the best to date [EW07]. They begin by providing a 1.893-competitive *memoryless* policy, the only such policy known to be better than 2-competitive. As an initial step, they show that in the restricted two-valued case, a memoryless policy can achieve optimal competitiveness of $\sqrt{2}$. For the general-valued case, the authors introduce the concept of *suppressed packets* as follows. Consider an optimal provisional schedule. For each packet p of that schedule, consider what would happen if p were transmitted next and the remaining provisional schedule recomputed. If there exists a packet p' that is not in the original provisional schedule, but is contained in the recomputed scheduled, that p' is being *suppressed* by p. If p were to be considered as the "heaviest" packet and transmitted in lieu of some earlier-deadline packet, not only would the switch receive the benefit of w_p but perhaps a future benefit due to the increased value of the remaining provisional schedule containing suppressed p'. With this in mind, when choosing the role of the "heaviest" packet, the policy considers for each p the sum $w_p + \frac{1}{2}w_{p'}$. The packet that maximizes this expression is denoted as h. The earliest deadline packet e from the provisional schedule is sent when $w_e \geq \frac{1}{\beta}(w_h + \frac{w_{h'}}{2})$ where h' is the packet suppressed by h, and β is a parameter of the policy; otherwise h is sent. The proof of 1.893-competitiveness relies on a detailed case analysis.

The better variant of Englert and Westermann's policy considers *levels* of the provisional optimal schedule that are delimited by tight packets in the schedule (those sent precisely at their deadline). For each level, the policy computes the amortized value δ that would be sent per time-step if the provisional schedule were to be followed until that tight deadline. The largest of these amortized constants is maintained from the preceding time-step, hence the use of memory. To achieve 1.828-competitiveness, they increase the contribution of suppressed packets when choosing h, and use the level-based amortizations when deciding whether to transmit e or h.

The best known *randomized* policy is RMix by Chin et al. [BCC+04, CCF+06]. This simple policy picks a real $x \in [-1, 0]$ uniformly at random at each step, and transmits the earliest-deadline packet p with $w_p \geq e^x \cdot w_h$ where h is the heaviest currently pending packet. The original analysis uses a potential function to show that RMix is $\frac{e}{e-1} \approx 1.582$-competitive against an oblivious adversary. Jeż [Jeż09a] proves the same competitiveness against an *adaptive* adversary, using an analysis in which the adversary's buffer is manipulated to match that of the online policy (akin to Li, Sethuraman, and Stein's analysis of MG [LSS05]).

4 Multiple Input/Output Models

The basic FIFO model from Section 2 applies to a switch with a single output port, or a multi-port switch in which each output port has its own dedicated buffer with capacity B. In this section, we look at several other models for multi-port switches.

4.1 Multiple Output Queues with Shared Memory

In this section, we consider an $N \times N$ switch in which there is a FIFO queue associated with each output port and a constraint that the *combined* size of those buffers is at most B. It is presumed that the internal switch fabric has speedup N, so that all packets received in the input ports can immediately be buffered in the output queues. At the end of each time-step, one packet is sent from the front of each nonempty output queue. In a nonpreemptive variant of the problem, once a packet is added to an output queue, it must be sent; in a preemptive variant of the problem, existing packets can be dropped from an output queue as desired. Both versions of the problem have been studied, but only for the restricted case in which all packets have unit value.

For the *preemptive* version, a common queuing policy is Longest Queue Drop (LQD). If the combined buffer space is not yet full, an incoming packet is accepted. If the combined buffer is full, the packet is tentatively accepted and then a packet (possibly the newest) is dropped from the back of whichever queue is the longest. Hahne, Kesselman, and Mansour prove that LQD has competitiveness somewhere between 2 and $\sqrt{2}$ [HKM01, AKM08]. They also provide a general lower bound of $\frac{4}{3}$ for the deterministic competitiveness with $N = 2$. Kobayashi, Miyazaki, and Okabe match this lower bound for $N = 2$ and large B, showing that LQD is $\frac{4B-4}{3B-2}$-competitive [KMO07]. They also provide an improved analysis of $2 - o(1)$ for the performance of LQD for large B.

Kesselman and Mansour study the *nonpreemptive* version with unit values [KM02, KM04]. They show that it is impossible to achieve constant competitiveness in this model, but achieve positive results with resource augmentation, comparing the performance of their policy with buffer capacity $B' > B$ to the optimal performance with buffer capacity B. We discuss those results in Section 5.

There does not appear to be a study of this model using general-valued packets. However, the standard classify-and-randomly-select technique [ABFR94] can transform any c-competitive policy for the unit-valued case to a randomized $2c$-competitive policy for the two-valued case or to a randomized $O(c \log \alpha)$-competitive policy for general values in the range $[1, \alpha]$.

4.2 Multiple Input Queues

Azar and Richter consider a model for an $N \times 1$ switch with each input queue having an independent FIFO buffer of size B [AR03, AR05]. At each time-step, the internal switch fabric can transfer one packet from a chosen input queue to the output port (thus the internal switch fabric has speedup $S = 1$). Such a model had earlier been studied without an explicit bound on the buffer capacity, but with a goal of minimizing the maximum buffer size while delivering all packets [BFLN02, BFLN03]. Returning to the goal of maximizing throughput with fixed buffers, there are two algorithmic considerations: admission control into the queues and a scheduling policy determining from which queue the next output packet is drawn. As a result, this problem is interesting even when all packets have unit value (a case that is trivially solvable in the single-queue model).

The overwhelming majority of research on this model has focused on the unit-valued case, with competitive bounds depending on the choice of B and N. Albers and Jacobs undertake an experimental evaluation of many of the following policies [AJ07, AJ10]. We note that with unit values, there is never a reason to preempt one packet in favor of another. The primary issue becomes the choice from which queue to transmit. For the deterministic case, Azar and Richter show that any work-conserving policy is 2-competitive, and that no deterministic policy can be better than $\left(2 - \frac{1}{N}\right)$-competitive for $B = 1$ or $(1.366 - \Theta(1/N))$-competitive for arbitrary B [AR03, AR05]. Albers and Schmidt show that a natural greedy policy in which a packet is always sent from the longest queue is no better than $2 - \frac{1}{B}$-competitive for $N \gg B$, regardless of how ties are broken [AS04, AS05]. They provide a different policy named SemiGreedy that is 1.889-competitive for any B with $N \gg B$, and optimally 1.857-competitive for $B = 2$. They also strengthen the deterministic lower bound to $\frac{e}{e-1} \approx 1.582$ for any B if $N \gg B$. Azar and Litichevskey achieves this bound for large B, providing a deterministic policy with a ratio that tends to $\frac{e}{e-1}$ [AL04, AL06b].

If randomization is allowed, Azar and Richter provide a $\frac{e}{e-1} \approx 1.582$-competitive policy for $B > \log N$, and a lower bound of $(1.46 - \Theta(1/N))$ for $B = 1$ [AR03, AR05]. Albers and Schmidt strengthen the lower bound to 1.466 for any B and large N, and 1.231 for $N = 2$ [AS04, AS05]. Schmidt provides a randomized policy that is 1.5-competitive for any B and N, and he improves upon some bounds for the case of $N = 2$ [Sch05]. The optimal competitiveness of 1.231 for $N = 2$ is achieved for any B by Bienkowski and Mądry [BM08].

With general packet values in $[1, \alpha]$, the seminal work of Azar and Richter includes a meta-policy that can be used to turn any c-competitive policy for a standard *single-queue* FIFO model into a $2c$-competitive policy for the respective multi-queue model [AR03, AR05]. For example, in the general-valued, nonpreemptive case the single-queue $(2 + \ln(\alpha))$-competitive policy for large B [AM03] can be converted to a $(4 + 2\ln(\alpha))$-competitive policy for the corresponding multi-queue model. Kobayashi, Miyazaki, and Okabe provide a different meta-policy that converts a c-competitive result for the unit-valued, multi-queue model to a policy for the two-valued, multi-queue model with competitiveness at most $\min(\alpha c, \frac{\alpha c(2-c)+c^2-2c+2}{\alpha(2-c)+c-1})$ [KMO09]. Applying this framework with known results for unit-valued models allows them to improve many bounds for the two-valued system; see Table 1 of that paper for a summary of current results. Finally, a few policies have been developed directly for the general-valued, preemptive multi-queue model. Azar and Richter provide a deterministic 3-competitive policy named `TransmitLargestHead (TLH)` which greedily buffers the B most valuable packets at each input port and transmits the packet that is the highest-value of those currently at the head of a buffer [AR04b]. Itoh and Takahashi improve the analysis for TLH to show $(3 - \frac{1}{\alpha})$-competitiveness [IT06].

4.3 Combined Input and Output Queued (CIOQ) Switches

An $N \times N$ CIOQ switch has buffer space available at each input port and each output port. In a typical model, each buffer has a fixed capacity, but not all buffers have the same capacity. The internal fabric of a switch is responsible for transferring packets from the input ports to the output ports. A switch with speedup S proceeds with S internal transfer cycles per one external time-step, with a limitation that each input queue releases at most one packet and each output queue accepts at most one packet during a single transfer cycle. Figure 2 contains a diagram of a CIOQ switch.

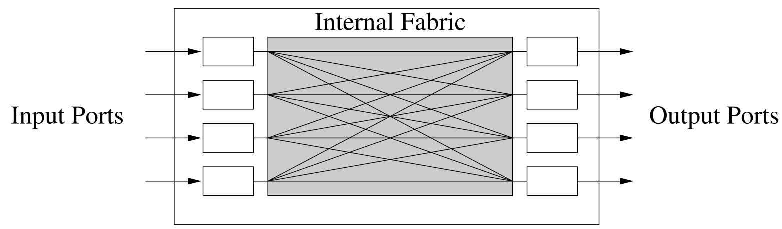


Figure 2: Schematic of a CIOQ switch

Commonly, *virtual output queuing (VOQ)* is used whereby each input port maintains N independent buffers, one dedicated to each output port. For example queue $VOQ_{i,j}$ holds packets that arrive at input port i, waiting to be transferred to output port j. Each virtual queue must obey FIFO semantics, but packets arriving at an input port that are destined for distinct output ports may proceed through the switch in any order.

Kesselman and Rosén were the first to consider competitiveness of switching policies for the CIOQ models [KR03, KR06]. For the case of unit-valued packets, they provide a nonpreemptive switch policy with VOQ that is 3-competitive for any speedup, and 2-competitive for $S = 1$. For the case where there are up to k distinct packet values in the range $[1, \alpha]$, they provide two *preemptive* policies with VOQ that are respectively $4S$-competitive and $8 \min(k, 2 \log \alpha)$-competitive. Azar and Richter improve on these results, proving constant competitive bounds for arbitrary speedup and packet values using VOQ. Specifically, they propose a deterministic policy β-PG (Preemptive Greedy) and show that it is 8-competitive for parameter $\beta = 3$ [AR04a, AR06]. Kesselman, Kogan, and Segal improve that analysis, showing that β-PG is at most 7.5-competitive for $\beta = 3$ and at most 7.47-competitive for $\beta = 2.8$ [KKS08b]. Kesselman and Rosén also consider CIOQ switches with *Priority Queuing (PQ)* buffers that transmit the packet of highest value at a given step, providing a 6-competitive policy for any speedup [KR08].

4.4 Crossbar Switches

The internal fabric of a switch can be implemented using a crossbar architecture, as shown in Figure 3. This supports non-blocking communication during an internal transfer cycle for any valid matching between input and output ports. Furthermore, a buffered crossbar switch contains additional dedicated buffers at every junction between input port i and output port j.

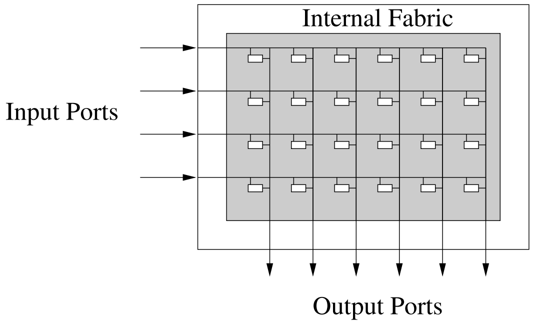


Figure 3: Schematic of a buffered crossbar switch

A competitive analysis in this model has been undertaken by Kesselman, Kogan, and Segal. They propose a 4-competitive policy for the case of unit-sized, unit-valued packets with FIFO buffers, and an 18-competitive policy for general-valued packets but with Priority Queuing [KKS08a]. For the FIFO case, they also present a nonpreemptive 7-competitive switch policy for the case of *variable-length* packets that have uniform value density (that is, a packet's value is proportional to its size), and a preemptive 21-competitive policy for unit-sized packets with general values [KKS08c]. These bounds hold for arbitrary speedup.

4.5 Multiple Output Ports with Reconfiguration Penalty

Azar et al. [AFG+09] introduce a model for a switch with N output ports, such that the switch is only able to transmit a single packet per time-step (rather than one packet per port). They assume that there is a currently "active" output port that can transmit packets, and that there is a certain amount of reconfiguration time required to change to a different port (one unit of idle time, in their model). There is no explicit buffer constraint on the switch, but packets have deadlines by which they must be transmitted or dropped (i.e., the bounded-delay model).

Azar et al. consider unit-valued packets. They show that the offline case is NP-hard, even for the s-uniform model, and they provide a deterministic, online policy that is $\left(\frac{1}{1-4\sqrt{N/s}}\right)$-competitive where s is the *minimum* span of all packets (i.e., $d_p - r_p \geq s$ for all p). They note that for $s \gg N$ this ratio approaches 1. In contrast, they provide a lower bound on the competitiveness of any policy (even with randomization) of $\frac{16}{15}$ for $s \leq 2N$, and more generally $1/(1-\frac{N}{8s})$.

5 Resource Augmentation

In this section, we review the use of resource augmentation in the models surveyed in Sections 2–4. For the FIFO model of Section 2, Kim considers augmenting an online policy with increased transmission rate as well as a larger buffer [Kim04a]. He examines the necessary and sufficient augmentations to allow an online policy to be optimal (i.e., 1-competitive). For the nonpreemptive *two-valued model*, a buffer of size $2B$ can be used to guarantee optimality (by reserving half of the capacity for high-valued packets, and half for low-valued packets). Even with an arbitrary increase in bandwidth, any policy requires a buffer of size at least $(2-\frac{1}{\alpha})B$ to guarantee optimality, and that the bandwidth must be at least 2-fold more than the offline when using a buffer of precisely that size. He provides a policy matching this buffer size with speed 2. For the nonpreemptive *general-valued* case, he provides an optimal policy using a speedup of $2(\lfloor \log \alpha \rfloor + 1)$ and a buffer of size $2(\lfloor \log \alpha \rfloor + 1)B$. Finally, for the *preemptive* general-valued case, he proves that the standard GREEDY policy is optimal if augmented with any speedup $s \geq 2$ and a buffer of size $\frac{s}{s-1}B$.

Jeżabek considers resource augmentation for the bounded-delay model of Section 3, examining the effect of an online policy with bandwidth m versus the offline with bandwidth 1 [Jeż09b]. He shows that for any constant m, it is impossible to be 1-competitive with bandwidth m, and he provides an online policy that is $(1 + \frac{1}{2^m-1})$-competitive for any m. In recent (unpublished) work, he shows that with agreeable deadlines, there exists a 1-competitive policy for $m = 2$ [Jeż09c].

For the multi-output switch with shared memory, as described in Section 4.1, Kesselman and Mansour provide the following results [KM02, KM04]. They allow their policy a buffer capacity of $B(\ln(N)+1)$ and describe a HARMONIC policy in which the largest k current queues are restricted to using at most $\frac{H_k}{\ln(N)+1} \cdot B$ cumulative buffer space where H_k is the k^{th} harmonic number. An arriving packet is accepted so long as these constraints remain valid for all k. They prove the HARMONIC policy to be 2-competitive versus an offline policy with buffer capacity B. They generalize this to a PARAMETRIC HARMONIC policy that achieves c-competitiveness using a buffer of size $B \log_c N$. Finally, they show that it is impossible to be c-competitive with a buffer size less than $B\frac{\log_{2c} N}{2c}$.

For the multi-input FIFO model as described in Section 4.2, Albers and Schmidt [AS04, AS05] consider resource augmentation. If each input port is allowed a FIFO buffer of size $(1 + c)B$, any greedy policy becomes $\frac{c+2}{c+1}$-competitive (versus the optimal solution with a single buffer B). If an online policy were allowed to transmit m packets per time-step, they show it is possible to be $(1 + \frac{1}{m})$-competitive versus the offline optimal that transmits one packet per time-step.

6 Closely Related Models

In this section, we describe several models related to buffer management that have been introduced in recent years. Perhaps the most significant of these, described in Section 6.1, is a SODA 2009 paper of Bienkowski et al. that generalizes the bounded-delay model. In Section 6.2, we discuss a SODA 2008 paper of Fiat, Mansour, and Nadav which introduces a model in which the value of packets degrades over time.

6.1 Collecting Weighted Items from a Queue

Bienkowski et al. introduce the following generalization of the bounded-delay buffer management problem [BCD+09]. There exists an ordered queue of weighted items. At each time step, some number of packets may expire from the front of the queue while other items may arrive at arbitrary locations. After that, an online agent is given a chance to collect one item taken anywhere from the queue. The agent has no a priori knowledge of when events will occur and the goal is to maximize the total weight of the collected items. They also consider a special case of a FIFO queue in which all new arrivals enter at the back of the queue.

The proposed model generalizes the bounded-delay problem in the following way. The queue represents the currently pending packets, ordered according to deadline. Therefore, the items at the front are the first to expire, while new arrivals may generally be placed in arbitrary locations (depending upon the relative deadlines). The special case of the FIFO queue corresponds to the agreeable deadlines version of the bounded-delay model, as a newly arriving packet must have the greatest deadline. Because of the correspondence, this paper provides great insight on the status of the bounded-delay model. In the original bounded-delay model, policies often consider pending packets' deadlines when making a decision (e.g., computing an optimal provisional schedule). In the new model, the queue provides implicit knowledge of the relative order of the deadlines for pending items, but knowledge of an item's actual deadline becomes apparent only after it expires.

It can trivially be shown that a Greedy policy is 2-competitive for this new model, and the 1.618-competitive lower bound for the buffer management problem applies. The authors give a stronger lower bound of 1.633 for the new problem. This new lower bound is shown for a very restricted *decremental* case in which all items arrive at the onset of the game and the only unknowns are the implicit deadlines of the items.

The authors provide a 1.89-competitive policy PrudentMark and show that their analysis is essentially tight. While this policy is not quite as good as the best-known 1.828-competitive policy for the bounded-delay model [EW07], it must operate with the more limited knowledge afforded by the new model. For the FIFO queue model, they provide a 1.737-competitive policy (in place of the 1.618-competitive policy for the bounded-delay model with agreeable deadlines).

Another very interesting result concerns a special case of *nondecreasing weights*, in which the weights of arriving items are nondecreasing. For this case, they provide a ϕ-competitive MarkAnd-Pick policy. This is an optimal result, as the existing buffer management lower bounds satisfy this nondecreasing weight property. This implies that if a stronger lower bound exists for the buffer management problem, it must rely on a construction that is not nondecreasing.

Finally, they consider *memoryless* policies which make decisions based only on the configuration of the current queue (as opposed to past expirations). For this restriction, they show a deterministic lower bound of 2, matching the upper bound of Greedy. They also claim an *adaptive-adversary* randomized lower bound of 1.582 for memoryless policies, which is met by the RMix policy [BCC+04, CCF+06, Jeż09a], adapted for the weighted queue collection problem.

6.2 Latency-Sensitive FIFO Model

Fiat, Mansour, and Nadav introduce a nonpreemptive FIFO model in which there is no explicit limit on the buffer capacity, yet for which each buffered packet loses a unit of value for each time-step that it spends in the buffer [FMN08]. If all packets have the same initial value R, they provide a deterministic 1.618-competitive policy which is matched by a *randomized* lower bound of 1.618. Their policy uses a simple threshold, stating that a newly arriving packet is accepted into the buffer so long as the current size of the buffer is at most R/ϕ^2.

When packets have general values, they provide a memoryless deterministic policy DT based on the use of a *doubling threshold*. A new packet is accepted if its value is at least twice the current buffer size. They show that this policy is 5.25-competitive and that this analysis is tight. They pair this result with a deterministic lower bound of 3, and a deterministic *memoryless* lower bound of 4.1. They provide a more complex policy IT based on what they denote as *incremental thresholds*. An IT policy is defined by a sequence of values $a_0, a_1, a_2, \ldots$ such that a packet is accepted into the queue if its value is at least $a_B \cdot B$ for current buffer size of B (in contrast to the doubling threshold, for which $a_B = 2$ for all B). They show that a sequence of a_i values can be constructed leading to a competitiveness that can be made arbitrarily close to 4.9. They conjecture that the optimal competitiveness for a deterministic, memoryless policy is $\phi^3 \approx 4.23$, and that there exists some sequence of a_i values such that IT achieves this ratio.

6.3 Packet Dependencies

Kesselman, Patt-Shamir, and Scalosub introduce the following variant of the single-buffer FIFO model, motivated by the fact that individual network packets often belong to a larger data context [KPSS09]. They assume that each packet identifies a *data frame* to which it belongs, and that packets of a frame are only useful if sufficiently many of them are delivered. Formally, they define the k-of-n frame throughput maximization problem (denoted as k-of-n FTM), in which the switching policy is only rewarded for frames that have at least k of their n packets transmitted. The goal is to maximize the number of successfully transmitted *frames* (as opposed to packets). They focus primarily on the special case where $k = n$ and denote this as k-FTM.

For the offline version of k-FTM, they show that a simple algorithm produces a $(k + 1)$-approximation for any buffer size, but that producing an $o(k/\ln k)$-approximation is NP-hard for $k \geq 3$ and $B = 1$ due to a reduction to k-dimensional matching.

For the online version of k-FTM, they show that it is impossible to have bounded competitiveness, even for $k = 2$. For this reason, they consider a restriction that instances be what they term *order-respecting*. This means that the frames can be linearly ordered in such a way that if frame i is before frame i' in this order, than the j^{th} packet of frame i must arrive before the j^{th} packet of frame i' for all $1 \leq j \leq n$. For this restricted version, they provide a nonpreemptive, deterministic policy that is $O(k^2)$-competitive, and a lower bound of $\Omega(k)$ against the competitiveness of any deterministic policy (preemptive or nonpreemptive). They have limited results for the more general k-of-n FTM problem, but are able to relate the offline approximability of this problem with $B = 1$ and k near 1 or n, to the $(k+1)$-FTM problem. They conclude with a discussion of open questions.

6.4 Bounded-Delay Model with Maximum Buffer Requirement

The original bounded-delay model of Section 3, as defined by Kesselman et al. [KLM$^+$01, KLM$^+$04], does not have an explicit bound on the capacity of the buffer. However, it is natural to consider a model with individual packet deadlines and capacity constraints for a buffer. For example, the 1.618-competitive policy for agreeable deadlines by Li, Sethuraman, and Stein [LSS05] is described in a model with a fixed sized buffer. Li considers the more general bounded-delay model including a constraint that at most B packets can be buffered at any time [Li09a]. He provides a deterministic, memoryless 3-competitive online policy and a randomized, memoryless $\phi^2 \approx 2.618$-competitive policy. Fung improves on this by providing a 2-competitive deterministic policy named GreedyQueue (GRQ) [Fun09]. Subsequently, Li tightens the analysis of his deterministic policy to be 2-competitive as well, and provides a lower bound of $2 - \frac{1}{B}$ for a class of policies that are based on always sending a packet that belongs to the optimal provisional schedule [Li09c].

Azar and Levy consider a variant of the multi-input switch described in Section 4.2, in which there are individual packet deadlines and buffer capacities [AL06a]. The greedy policy is not constant competitive for this model. For the nonpreemptive model, they provide a 2-competitive deterministic policy for *unit* values, and a randomized policy which is $O(\log \alpha)$-competitive for general values. In the preemptive case, Azar and Levy provide a 9.82-competitive deterministic policy for general values. Li improves on this bound, providing a 4.723-competitive deterministic policy, and a lower bound of 2 for the problem [Li09b].

7 Open Questions

Buffer management problems remain a rich domain for future research. We conclude by highlighting some of the most significant open problems.

- **FIFO Model**
  - For the general-valued preemptive model, the best-known deterministic upper bound is 1.732, achieved by PG [EW06, EW09]. The best-known lower bound is 1.419 [KMvS03, KMvS05].
  - For the general-valued preemptive model, does randomization help? At one point in time, the 1.75-competitive random policy of [And05] equaled the best known deterministic bound for the problem, but that has since been surpassed. The strongest lower bound, due to a reduction from the s-uniform model, is 1.25 for an oblivious adversary.
  - What is the impact of memory in the FIFO model? For the two-valued case, the optimal 1.282-competitive Account Strategy of [EW06, EW09] requires memory in the form of its account, while the best known memoryless policy is 1.544-competitive [KM03], as described in Section 2.2.

    For the general-valued case, both the Greedy and PG policies are memoryless. Can better policies be developed that use memory? Can stronger lower bounds be proven for memoryless policies?

- **Bounded Delay Model**
  - The most significant issue is to narrow the gap for the general version of this model. There is currently a deterministic upper bound of 1.828 [EW07] versus the lower bound of 1.618 [Haj01, AMZ03, CF03].
  - One approach to narrowing the general gap is to focus on the s-bounded case. It is known that 1.618 is the optimal deterministic competitiveness for both the 2-bounded and 3-bounded cases. For the 4-bounded case, there remains a gap between the upper bound of 1.732 from [BCC$^+$04, CCF$^+$06] and the 1.618 lower bound.
  - There remains a large gap for the deterministic competitiveness of the s-uniform model The best upper bound is the 1.618-competitive MG policy for the more general agreeable deadline case [LSS05]. The strongest lower bound of 1.377 comes from the special case of the 2-uniform model.
  - As a warm-up to settling the gap for the general s-uniform case, what can be said about the *two-valued* version of that problem? The 1.282-competitive policy for the FIFO model applies to this case, but the matching FIFO lower bound does not apply (high-value packets can be scheduled without necessarily dropping earlier low-value packets).
  - The 1.582-competitive randomized RMix [BCC$^+$04, CCF$^+$06] is currently the best known policy for the general case. This is opposite a lower bound of 1.33 against a randomized adaptive adversary.

- **Multi-Queue Models**
  - Section 4.1 describes the model for multiple output queues with shared memory. For the preemptive version of that problem, Hahne, Kesselman, and Mansour prove that LQD has competitiveness somewhere between 2 and $\sqrt{2}$ for general B [HKM01, AKM08]. Kobayashi, Miyazaki, and Okabe provide a refined upper bound of $2 - o(1)$ where the $o(1)$ term depends depends on B but tends to zero for large B [KMO07]. What is the true competitiveness for LQD?
  - What happens in the shared-memory, multiple output queue model with *general-valued* packets? Is constant competitiveness achievable?
  - In Section 4.5, we describe a model for a multi-output switch with a single active output port and a reconfiguration penalty for changing that port. Azar et al. consider the case of maximizing throughput for unit-valued packets [AFG$^+$09]. What can be said about the *weighted* case?

- **Additional Models**
  - The paper of Bienkowski et al. [BCD$^+$09], described in Section 6.1, brings up many interesting questions about the new model and its implications on the bounded-delay model. The most general version of their model has an upper bound of 1.89 and a lower bound of 1.633, and their *FIFO queue* variant has an upper bound of 1.737 versus the same lower bound of 1.633.
  - In Section 6.2, we discuss a latency-sensitive model introduced by Fiat, Mansour, and Nadav [FMN08]. For the nonpreemptive case with general values, they provide a memoryless upper bound of 5.25 against a general lower bound of 3 (and a memoryless lower bound of 4.1). They also conjecture that their *incremental threshold* framework can be better tuned to a $\phi^3 \approx 4.23$-competitive algorithm.
  - What can be said about the *preemptive* version of the latency-sensitive model of Fiat, Mansour, and Nadav [FMN08]?

- **General Networks**
  - All of the research discussed in this survey focuses on the throughput of a single switch. A general open problem is to extend these analyses to a *network* of switches; for example, see [AOKR03, GR05, RS07, KR08, RR09].

Acknowledgments

I wish to thank Marek Chrobak, Matthias Englert, and Alex Kesselman for the valuable feedback they provided on a preliminary draft of this article, and for the open lines of communication that ensued. Additional thanks are due to Marek for his support in publishing the survey as part of SIGACT News, and to Łukasz Jeż for finding a mistake in the final draft.

References

[ABFR94] Baruch Awerbuch, Yair Bartal, Amos Fiat, and Adi Rosén. Competitive non-preemptive call control. In *Proc. Fifth Annual ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 312–320, Arlington, Virginia, January 1994.

[AFG+09] Yossi Azar, Uriel Feige, Iftah Gamzu, Thomas Moscibroda, and Prasad Raghavendra. Buffer management for colored packets with deadliens. In Friedhelm Meyer auf der Heide and Michael A. Bender, editors, *Proc. 21st ACM Symp. on Parallelism in Algorithms and Architectures (SPAA)*, pages 319–327, Calgary, Alberta, Cananda, August 2009.

[AJ07] Susanne Albers and Tobias Jacobs. An experimental study of new and known online packet buffering algorithms. In Lars Arge and Michael Hoffmann, editors, *Proc. 15th European Symp. on Algorithms (ESA)*, volume 4698 of *Lecture Notes in Computer Science*, pages 754–765, Eilat, Israel, October 2007. Springer-Verlag.

[AJ10] Susanne Albers and Tobias Jacobs. An experimental study of new and known online packet buffering algorithms. *Algorithmica*, 2010.

[AKM08] William Aiello, Alex Kesselman, and Yishay Mansour. Competitive buffer management for shared-memory switches. *ACM Trans. on Algorithms*, 5(1), 2008.

[AL04] Yossi Azar and Arik Litichevskey. Maximizing throughput in multi-queue switches. In Susanne Albers and Tomasz Radzik, editors, *Proc. 12th European Symp. on Algorithms (ESA)*, volume 3221 of *Lecture Notes in Computer Science*, pages 53–64, Bergen, Norway, September 2004. Springer-Verlag.

[AL06a] Yossi Azar and Nir Levy. Multiplexing packets with arbitrary deadlines in bounded buffers. In Lars Arge and Rusins Freivalds, editors, *Proc. 10th Scandinavian Workshop on Algorithm Theory (SWAT)*, volume 4059 of *Lecture Notes in Computer Science*, pages 5–16, Riga, Latvia, July 2006. Springer-Verlag.

[AL06b] Yossi Azar and Arik Litichevskey. Maximizing throughput in multi-queue switches. *Algorithmica*, 45(1):69–90, 2006.

[AM03] Nir Andelman and Yishay Mansour. Competitive management of non-preemptive queues with multiple values. In Faith Ellen Fich, editor, *Proc. 17th Int. Conf. on Distributed Computing (DISC)*, volume 2848 of *Lecture Notes in Computer Science*, pages 166–180, Sorento, Italy, October 2003. Springer-Verlag.

[AMRR00] William A. Aiello, Yishay Mansour, S. Rajagopolan, and Adi Rosén. Competitive queue policies for differentiated services. In *Proc. Nineteenth IEEE Conf. on Computer Communications (INFOCOM)*, volume 2, pages 431–440, Tel-Aviv, Israel, March 2000.

[AMRR05] William A. Aiello, Yishay Mansour, S. Rajagopolan, and Adi Rosén. Competitive queue policies for differentiated services. *J. Algorithms*, 55(2):113–141, May 2005.

[AMZ03] Nir Andelman, Yishay Mansour, and An Zhu. Competitive queueing policies for QoS switches. In *Proc. 14th Annual ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 761–770, Baltimore, Maryland, January 2003.

[And05] Nir Andelman. Randomized queue management for DiffServ. In Phillip B. Gibbons and Paul G. Spirakis, editors, *Proc. 17th ACM Symp. on Parallelism in Algorithms and Architectures (SPAA)*, pages 1–10, Las Vegas, Nevada, July 2005.

[AOKR03] William Aiello, Rafail Ostrovsky, Eyal Kushilevitz, and Adi Rosén. Dynamic routing on networks with fixed-sized buffers. In *Proc. 14th Annual ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 431–440, Baltimore, Maryland, January 2003.

[AR03] Yossi Azar and Yossi Richter. Management of multi-queue switches in QoS networks. In *Proc. 35th ACM Symp. on Theory of Computing (STOC)*, pages 82–89, San Diego, California, June 2003.

[AR04a] Yossi Azar and Yossi Richter. An improved algorithm for CIOQ switches. In Susanne Albers and Tomasz Radzik, editors, *Proc. 12th European Symp. on Algorithms (ESA)*, volume 3221 of *Lecture Notes in Computer Science*, pages 65–76, Bergen, Norway, September 2004. Springer-Verlag.

[AR04b] Yossi Azar and Yossi Richter. The zero-one principle for switching networks. In László Babai, editor, *Proc. 36th ACM Symp. on Theory of Computing (STOC)*, pages 64–71, Chicago, Illinois, June 2004.

[AR05] Yossi Azar and Yossi Richter. Management of multi-queue switches in QoS networks. *Algorithmica*, 43(1–2):81–96, 2005.

[AR06] Yossi Azar and Yossi Richter. An improved algorithm for CIOQ switches. *ACM Trans. on Algorithms*, 2(4):640–660, 2006.

[AS04] Susanne Albers and Markus Schmidt. On the performance of greedy algorithms in packet buffering. In László Babai, editor, *Proc. 36th ACM Symp. on Theory of Computing (STOC)*, pages 35–44, Chicago, Illinois, June 2004.

[AS05] Susanne Albers and Markus Schmidt. On the performance of greedy algorithms in packet buffering. *SIAM Journal on Computing*, 35(2):278–304, 2005.

[Aza04] Yossi Azar. Online packet switching. In Giuseppe Persiano and Roberto Solis-Oba, editors, *Proc. Second Workshop on Approximation and Online Algorithms (WAOA)*, volume 3351 of *Lecture Notes in Computer Science*, pages 1–5, Bergen, Norway, September 2004. Springer-Verlag.

[BCC+04] Yair Bartal, Francis Y. L. Chin, Marek Chrobak, Stanley P. Y. Fung, Wojciech Jawor, Ron Lavi, Jiří Sgall, and Tomáš Tichý. Online competitive algorithms for maximizing weighted throughput of unit jobs. In Volker Diekert and Michel Habib, editors, *Proc. 21st ACM Symp. on Theoretical Aspects of Computer Science (STACS)*, volume 2996 of *Lecture Notes in Computer Science*, pages 187–198, Montpellier, France, March 2004. Springer-Verlag.

[BCD+09] Marcin Bienkowski, Marek Chrobak, Christoph Dürr, Mathilde Hurand, Artur Jeż, Łukasz Jeż, and Grzegorz Stachowiak. Collecting weighted items from a dynamic queue. In *Proc. 20th Annual ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 1126–1135, New York, New York, January 2009.

[BCJ08] Marcin Bienkowski, Marek Chrobak, and Łukasz Jeż. Randomized algorithms for buffer management with 2-bounded delay. In Evripidis Bampis and Martin Skutella, editors, *Proc. Sixth Workshop on Approximation and Online Algorithms (WAOA)*, Lecture Notes in Computer Science, pages 92–104, Germany, September 2008. Springer-Verlag.

[BE98] Allan Borodin and Ran El-Yaniv. *Online Computation and Competitive Analysis.* Cambridge University Press, New York, 1998.

[BFK+04] Nikhil Bansal, Lisa K. Fleischer, Tracy Kimbrel, Mohammad Mahdian, Baruch Schieber, and Maxim Sviridenko. Further improvements in competitive guarantees for QoS buffering. In Josep Díaz, Juhani Karhumäki, Arto Lepistö, and Donald Sannella, editors, *Proc. 31st Int. Colloquium on Automata, Languages and Programming (ICALP)*, volume 3142 of *Lecture Notes in Computer Science*, pages 196–207, Turku, Finland, July 2004. Springer-Verlag.

[BFLN02] Amotz Bar-Noy, Ari Freund, Shimon Landa, and Joseph (Seffi) Naor. Competitive online switching policies. In *Proc. 13th Annual ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 525–534, San Francisco, California, January 2002.

[BFLN03] Amotz Bar-Noy, Ari Freund, Shimon Landa, and Joseph (Seffi) Naor. Competitive on-line switching policies. *Algorithmica*, 36(3):225–247, 2003.

[BM08] Marcin Bienkowski and Aleksander Mądry. Geometric aspects of online packet buffering: An optimal randomized algorithm for two buffers. In Eduardo Sany Laber, Claudson F. Bornstein, Loana Tito Nogueira, and Luebio Fario, editors, *Proc. 8th Latin American Symp. on Theoretical Informatics (LATIN)*, volume 4957 of *Lecture Notes in Computer Science*, pages 252–263, Búzios, Brazil, April 2008. Springer-Verlag.

[CCF+06] Francis Y. L. Chin, Marek Chrobak, Stanley P. Y. Fung, Wojciech Jawor, Jiří Sgall, and Tomáš Tichý. Online competitive algorithms for maximizing weighted throughput of unit jobs. *J. Discrete Algorithms*, 4(2):255–276, 2006.

[CF03] Francis Y. L. Chin and Stanley P. Y. Fung. Online scheduling for partial job values: Does timesharing or randomization help? *Algorithmica*, 37:149–164, 2003.

[Chr08] Marek Chrobak. Online algorithms column 13. *SIGACT News*, 39(3):96–121, September 2008.

[CJST04] Marek Chrobak, Wojciech Jawor, Jiří Sgall, and Tomáš Tichý. Improved online algorithms for buffer management in QoS switches. In Susanne Albers and Tomasz Radzik, editors, *Proc. 12th European Symp. on Algorithms (ESA)*, volume 3221 of *Lecture Notes in Computer Science*, pages 204–215, Bergen, Norway, September 2004. Springer-Verlag.

[CJST07] Marek Chrobak, Wojciech Jawor, Jiří Sgall, and Tomáš Tichý. Improved online algorithms for buffer management in QoS switches. *ACM Trans. on Algorithms*, 3(4), 2007.

[Eng08] Matthias Englert. *Online Scheduling for Buffering Problems*. PhD thesis, RWTH Aachen University, May 2008.

[EvS04] Leah Epstein and Rob van Stee. Buffer management problems. *SIGACT News*, 35(3):58–66, 2004.

[EW06] Matthias Englert and Matthias Westermann. Lower and upper bounds on FIFO buffer management in QoS switches. In Yossi Azar and Thomas Erlebach, editors, *Proc. 14th European Symp. on Algorithms (ESA)*, volume 4168 of *Lecture Notes in Computer Science*, pages 352–363, Zurich, Switzerland, September 2006. Springer-Verlag.

[EW07] Matthias Englert and Matthias Westermann. Considering suppressed packets improves buffer management in QoS switches. In *Proc. 18th Annual ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 209–218, New Orleans, Louisiana, January 2007.

[EW09] Matthias Englert and Matthias Westermann. Lower and upper bounds on FIFO buffer management in QoS switches. *Algorithmica*, 53(4):523–548, April 2009.

[FMN08] Amos Fiat, Yishay Mansour, and Uri Nadav. Competitive queue management for latency sensitive packets. In *Proc. 19th Annual ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 228–237, San Francisco, California, January 2008.

[Fun09] Stanley P. Y. Fung. Bounded delay packet scheduling in a bounded buffer. Technical Report cs.DS/0907.2741, arXiv.org, July 2009.

[GR05] Eyal Gordon and Adi Rosén. Competitive weighted throughput analysis of greedy protocols on DAGs. In *Proc. 24th ACM Symp. on Principles of Distributed Computing (PODC)*, pages 227–236, Las Vegas, Nevada, July 2005.

[Haj01] B. Hajek. On the competitiveness of online scheduling of unit-length packets with hard deadlines in slotted time. In *Proc. 35th Conf. in Information Sciences and Systems (CISS)*, pages 434–438, Baltimore, Maryland, March 2001.

[HKM01] Ellen L. Hahne, Alexander Kesselman, and Yishay Mansour. Competitive buffer management for shared-memory switches. In *Proc. Thirteenth ACM Symp. on Parallel Algorithms and Architectures (SPAA)*, pages 53–58, Heraklion, Crete, Greece, July 2001.

[IT06] Toshiya Itoh and Noriyuki Takahashi. Competitive analysis of multi-queue preemptive QoS algorithms for general priorities. *IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences*, E89-A(5):1186–1197, 2006.

[Jaw05] Wojciech Jawor. Three dozen papers on online algorithms. *SIGACT News*, 36(1):71–85, 2005.

[Jeż09a] Łukasz Jeż. Randomised buffer management with bounded delay against adaptive adversary. *CoRR*, abs/0907.2050, 2009.

[Jeż09b] Jan Jeżabek. Increasing machine speed in on-line scheduling of weighted unit-length jobs in slotted time. In *Proc. 35th Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM)*, volume 5404 of *Lecture Notes in Computer Science*, pages 329–340, Špindleruv Mlýn, January 2009. Springer-Verlag.

[Jeż09c] Jan Jeżabek. Resource augmentation for QoS buffer management with agreeable deadlines. (unpublished), 2009.

[Jeż10] Łukasz Jeż. A 4/3-competitive randomised algorithm for online scheduling of packets with agreeable deadlines. In Jean-Yves Marion and Thomas Schwentick, editors, *Proc. 27th ACM Symp. on Theoretical Aspects of Computer Science (STACS)*, Dagstuhl Seminar Proceedings, Nancy, France, March 2010. to appear.

[Kim04a] Jae-Hoon Kim. Optimal buffer management via resource augmentation. In *Proc. 15th Annual Int. Symp. on Algorithms and Computation (ISAAC)*, volume 3341 of *Lecture Notes in Computer Science*, pages 618–628, Hong Kong, China, December 2004. Springer-Verlag.

[Kim04b] Tracy Kimbrel. A simple proof of the 2-competitiveness of the greedy FIFO buffering algorithm. Technical Report RC23272, IBM Research, 2004.

[KKS08a] Alex Kesselman, Kirill Kogan, and Michael Segal. Best effort and priority queuing policies for buffered crossbar switches. In *Proc. 15th Int. Colloqium on Structural Information and Communication Complexity (SIROCCO)*, volume 5058 of *Lecture Notes in Computer Science*, pages 170–184, Villars-sur-Ollon, Switzerland, June 2008. Springer-Verlag.

[KKS08b] Alex Kesselman, Kirill Kogan, and Michael Segal. Improved competitive performance bounds for CIOQ switches. In Dan Halperin and Kurt Mehlhorn, editors, *Proc. 16th European Symp. on Algorithms (ESA)*, volume 5193 of *Lecture Notes in Computer Science*, pages 577–588, Karlsruhe, Germany, September 2008. Springer-Verlag.

[KKS08c] Alex Kesselman, Kirill Kogan, and Michael Segal. Packet mode and QoS algorithms for buffered crossbar switches with FIFO queueing. In *Proc. 27th ACM Symp. on Principles of Distributed Computing (PODC)*, pages 335–344, Toronto, Canada, August 2008.

[KLM+01] Alexander Kesselman, Zvi Lotker, Yishay Mansour, Boal Patt-Shamir, Baruch Schieber, and Maxim Sviridenko. Buffer overflow management in QoS switches. In *Proc. 33rd ACM Symp. on Theory of Computing (STOC)*, pages 520–529, Crete, Greece, July 2001.

[KLM+04] Alexander Kesselman, Zvi Lotker, Yishay Mansour, Boal Patt-Shamir, Baruch Schieber, and Maxim Sviridenko. Buffer overflow management in QoS switches. *SIAM Journal on Computing*, 33(3):563–583, 2004.

[KM01] Alexander Kesselman and Yishay Mansour. Loss-bounded analysis for differentiated services. In *Proc. 12th Annual ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 591–600, Washington, DC, January 2001.

[KM02] Alex Kesselman and Yishay Mansour. Harmonic buffer management policy for shared memory switches. In *Proc. 21st IEEE Conf. on Computer Communications (INFOCOM)*, New York City, New York, June 2002.

[KM03] Alexander Kesselman and Yishay Mansour. Loss-bounded analysis for differentiated services. *J. Algorithms*, 46(1):79–95, 2003.

[KM04] Alexander Kesselman and Yishay Mansour. Harmonic buffer management policy for shared memory switches. *Theoretical Computer Science*, 324(2–3):161–182, 2004.

[KMO07] Koji Kobayashi, Shuichi Miyazaki, and Yasuo Okabe. A tight bound on online buffer managment for two-port shared-memory switches. In Phillip B. Gibbons and Christian Scheideler, editors, *Proc. 19th ACM Symp. on Parallelism in Algorithms and Architectures (SPAA)*, pages 358–364, San Diego, California, June 2007.

[KMO09] Koji Kobayashi, Shuichi Miyazaki, and Yasuo Okabe. Competitive buffer management for multi-queue switches in QoS networks using packet buffering algorithms. In Friedhelm Meyer auf der Heide and Michael A. Bender, editors, *Proc. 21st ACM Symp. on Parallelism in Algorithms and Architectures (SPAA)*, pages 328–336, Calgary, Alberta, Cananda, August 2009.

[KMRS88] Anna Karlin, Mark Manasse, Larry Rudolph, and Daniel Sleator. Competitive snoopy paging. *Algorithmica*, 3(1):70–119, 1988.

[KMvS03] Alex Kesselman, Yishay Mansour, and Rob van Stee. Improved competitive guarantees for QoS buffering. In Giuseppe Di Battista and Uri Zwick, editors, *Proc. 11th European Symp. on Algorithms (ESA)*, volume 2832 of *Lecture Notes in Computer Science*, pages 361–372, Budapest, Hungary, September 2003. Springer-Verlag.

[KMvS05] Alex Kesselman, Yishay Mansour, and Rob van Stee. Improved competitive guarantees for QoS buffering. *Algorithmica*, 43(1–2):97–111, 2005.

[KPSS09] Alex Kesselman, Boal Patt-Shamir, and Gabriel Scalosub. Competitive buffer management with packet dependencies. In *Proc. 23rd Int. Parallel and Distributed Processing Symp.*, pages 1–12, Rome, Italy, May 2009.

[KR03] Alex Kesselman and Adi Rosén. Scheduling policies for CIOQ switches. In *Proc. Fifteenth ACM Symp. on Parallelism in Algorithms and Architectures (SPAA)*, pages 353–362, San Diego, California, June 2003.

[KR06] Alex Kesselman and Adi Rosén. Scheduling policies for CIOQ switches. *J. Algorithms*, 60(1):60–83, July 2006.

[KR08] Alex Kesselman and Adi Rosén. Controlling CIOQ switches with priority queuing and in multistage interconnection networks. *J. Interconnection Networks*, 9(1–2):53–72, 2008.

[Li09a] Fei Li. Competitive scheduling of packets with hard deadlines in a finite capacity queue. In *Proc. 28th IEEE Conf. on Computer Communications (INFOCOM)*, volume 2, pages 1062–1070, Rio de Janeiro, Brazil, April 2009.

[Li09b] Fei Li. Improved online algorithms for multiplexing weighted packets in bounded buffers. In *Proc. 5th Int. Conference on Algorithmic Aspects in Information and Management*, volume 5564 of *Lecture Notes in Computer Science*, pages 265–278, San Francisco, California, June 2009. Springer-Verlag.

[Li09c] Fei Li. Packet scheduling in a size-bounded buffer. *CoRR*, abs/0909.3637, 2009.

[LPS02] Zvi Lotker and Boaz Patt-Shamir. Nearly optimal FIFO buffer management for DiffServ. In *Proc. 21st ACM Symp. on Principles of Distributed Computing (PODC)*, pages 134–142, Monterey, California, July 2002.

[LPS03] Zvi Lotker and Boaz Patt-Shamir. Nearly optimal FIFO buffer management for two packet classes. *Computer Networks*, 42(4):481–492, 2003.

[LSS05] Fei Li, Jay Sethuraman, and Clifford Stein. An optimal online algorithm for packet scheduling with agreeable deadlines. In *Proc. 16th Annual ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 801–802, Vancouver, British Columbia, Canada, January 2005.

[LSS07] Fei Li, Jay Sethuraman, and Clifford Stein. Better online buffer management. In *Proc. 18th Annual ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 199–208, New Orleans, Louisiana, January 2007.

[MPSL00] Yishay Mansour, Boaz Patt-Shamir, and Ofer Lapid. Optimal smoothing schedules for real-time streams. In *Proc. 19th ACM Symp. on Principles of Distributed Computing (PODC)*, pages 21–29, Portland, Oregon, July 2000.

[MPSL04] Yishay Mansour, Boaz Patt-Shamir, and Ofer Lapid. Optimal smoothing schedules for real-time streams. *Distributed Computing*, 17(1):77–89, 2004.

[RR09] Harald Räcke and Adi Rosén. Approximation algorithms for time-constrained scheduling on line networks. In Friedhelm Meyer auf der Heide and Michael A. Bender, editors, *Proc. 21st ACM Symp. on Parallelism in Algorithms and Architectures (SPAA)*, pages 337–346, Calgary, Alberta, Cananda, August 2009.

[RS07] Adi Rosén and Gabriel Scalosub. Rate vs. buffer size: Greedy information gathering on the line. In Phillip B. Gibbons and Christian Scheideler, editors, *Proc. 19th ACM Symp. on Parallelism in Algorithms and Architectures (SPAA)*, pages 305–314, San Diego, California, June 2007.

[Sch05] Markus Schmidt. Packet buffering: Randomization beats deterministic algorithms. In Volker Diekert and Bruno Durand, editors, *Proc. 22nd ACM Symp. on Theoretical Aspects of Computer Science (STACS)*, volume 3404 of *Lecture Notes in Computer Science*, pages 293–304, Stuttgart, Germany, February 2005. Springer-Verlag.

[ST85] Daniel Sleator and Robert Tarjan. Amortized efficiency of list update and paging rules. *Communications of the ACM*, 28:202–208, 1985.

[Zhu04] An Zhu. Analysis of queueing policies in QoS switches. *J. Algorithms*, 53(2):137–168, 2004.