RZ 3499 (# 99397) 06/23/03
Computer Science 19 pages

# Research Report

## Resequencing Worst-Case Analysis for Parallel Buffered Packet Switches

Ilias Iliadis and Wolfgang E. Denzel

IBM Research
Zurich Research Laboratory
8803 Rüschlikon
Switzerland

# Resequencing Worst-Case Analysis for Parallel Buffered Packet Switches

Ilias Iliadis and Wolfgang E. Denzel
IBM Research, Zurich Research Laboratory, 8803 Rüschlikon, Switzerland

**Abstract**

This paper considers a general parallel buffered packet switch (PBPS) architecture which is based on multiple packet switches operating independently and in parallel. A load-balancing mechanism is used at each input to distribute the traffic to the parallel switches. The buffer structure of each of the parallel packet switches is based on either a dedicated, a shared, or a buffered-crosspoint output-queued architecture. As in such PBPS multipath switches packets may get out of order when they travel independently in parallel through these switches, a resequencing mechanism is necessary at the output side. This paper addresses the issue of evaluating the minimum resequence-queue size required for a deadlock-free lossless operation. An analytical method is presented for the exact evaluation of the worst-case resequencing delay and the worst-case resequence-queue size. The results obtained reveal their relation, and demonstrate the impact of the various system parameters on resequencing.

# I. INTRODUCTION

The capacity demand of the early generations of high-performance packet switches or routers could sufficiently be covered with single-stage switch-fabric architectures. Even the majority of the current switch generations is still based on single-stage architectures because the continued advances in VLSI technology made it possible, for the most part, to keep pace with the increasing capacity requirements. However, a further proliferation in capacity, in particular an increase of the number of ports as opposed to an increase of the port bandwidth, will eventually require the transition to multistage packet switch architectures. Owing to the complications associated with this transition, an intermediate solution that allows an easier migration is of great interest.

This paper considers such a solution that is based on multiple single-stage switches which provide multiplied capacity by independently operating in parallel. This concept has been referred to as parallel switch architecture in [1], parallel packet switch (PPS) in [2, 3], and distributed packet switch in [4]. A motivation was to cope with the problem arising when line rates run faster than the switch fabric does. At the ingress side, incoming packets (or cells) are evenly distributed across the lower-capacity parallel switches (planes). We refer to this function as *plane load-balancing*. At the egress side, the output traffic of all parallel switches is merged back, by the reverse, the *plane server* function, as illustrated in Figure 1.

Further advantages of the parallel packet switch concept include the following: the possibility for a smooth migration from the existing single-stage switches and the flexibility regarding the number of parallel switches. The migration is facilitated because the individual switches can work fully independently and need not be synchronized among each other. Therefore, the concept can be applied to any switch by developing only a single new component, namely a higher-speed switch adaptor or line card that incorporates the plane load-balancing and plane server schemes. Secondly, using more switches than the minimum number required for accommodating the total bandwidth results in an effective speedup that improves the overall performance. In addition, cost savings are achieved by providing fault tolerance based on an $N + 1$-type redundancy rather than a full duplication typically needed in single-stage switches.

As in multistage-multipath switches, a resequencing problem arises also in this context as packets of a given flow may get out of sequence when they travel independently through various buffered switch planes. Consequently, a resequencing mechanism is needed at the egress side. Note that the resequencing problem can be avoided by using a static scheme in which all packets of a given flow or virtual connection are forced to follow the same path [5]. This scheme, however, has the drawback that it limits the bandwidth per flow/connection to that of the port bandwidth of a single switch, and therefore is not considered here. Another approach for avoiding resequencing is by using unbuffered switch modules [4, 6]. This relies on buffering exclusively at the ingress side in the form of virtual output queuing (VOQ), which is also used in buffered switches to enhance performance. This approach, however, cannot be applied for the migration of existing buffered switches. The preservation of the packet sequence integrity also holds in the original work on PPS in [2], which is driven by the desire to mimic an ideal output queueing switch with parallel output-queued switches. This turns out to require not only a centralized control but also a speedup factor of two, which apparently is not very practical as the authors admit in a subsequent paper [3]. In this work, they propose a distributed control which cannot guarantee the preservation of packet sequence and hence requires resequencing. The cost for resequencing is now tolerated because it provides the possibility to obtain higher capacity with a minimum of lower-speed parallel hardware and without centralized control. It is intuitively clear that the disturbance of the packet sequence, as well as the throughput and delay performance, depend on how evenly the traffic load is balanced across the parallel switches. A good balance and a high efficiency are best achieved by dynamic load-balancing mechanisms that dispatch each packet independently [5].

The dynamic load-balancing mechanisms can be *state-independent* or *state-dependent*. In the former case, packets are dispatched to the planes without considering any information related to the switches. Typical examples are the round-robin [6, 7], random [8], and cyclic [1] mechanisms. In the state-dependent case, each packet is dispatched to an eligible plane that is determined by making use of switch-related information, such as buffer (queue) occupancies [4], or the knowledge of the traffic already arrived [2, 3]. Clearly, the state-

dependent mechanisms are more complex than the state-independent ones; however, improved resequencing behavior is to be expected. It turns out that their analysis is quite involved and therefore requires consideration in a separate paper. Accordingly, we focus here on state-independent load-balancing schemes.

In this paper we consider the general parallel buffered packet switch (PBPS) architecture, which is based on multiple buffered packet switches operating independently and in parallel. The buffer structure of each of the parallel packet switches is based on either a dedicated, a shared, or a crosspoint output-queued architecture. Our particular interest is on the buffered-crosspoint architecture, such as the single-stage distributed packet-routing switch described in [9], which for practical reasons uses a round-robin column-scheduling mechanism. More specifically, we are interested in exploring the possibility of migrating this switch according to the PBPS concept and in assessing the resequencing aspect. We address the issue of evaluating the worst-case resequence-queue size as this determines the minimum output buffer size required for a deadlock-free lossless[1] operation.

We note that a significant body of work has been done in resequencing analysis in various other contexts; examples include [10-13] and the references therein. These resequencing results however are limited to the resequencing delay and the resequence-queue occupancy. To the best of our knowledge, none of the earlier studies has considered the issue of the maximum resequence-queue size because it was assumed that if the resequence buffer is full, subsequently arriving out-of-sequence packets are dropped. In contrast, we do not allow packets to be dropped in the architectures we consider. This in turn may result in a deadlock situation because when an egress output buffer is full with resequence packets, none of them can depart from the buffer, and no packets can enter the buffer because there is no free space left. Consequently, the knowledge of the worst-case resequence-queue size is a prerequisite for an appropriate buffer dimensioning and safe operation of the switching system. This issue is further elaborated in Section II-A. Moreover, early works dealt with the resequencing incurred by a *single flow* either in isolation [11, 12] or in the presence of multiple flows [10]. In contrast to previous works, here we consider the combined *multiple flow* effect arising in the PBPS context, as at an output adaptor there can be many independent flows simultaneously under resequencing, all of which should be jointly taken into account.

More recent works have considered the worst-case resequence-queue size issue but only in terms of upper bounds because, as this paper demonstrates, the issue of deriving the worst-case resequence-queue size is far from straightforward. In [14, 15] the resequence buffer is bounded by first deriving the worst-case resequencing delay. Despite the fact that the worst-case resequencing delay can be used to derive an upper bound of the worst-case resequence-queue size, it is not readily apparent how the former relates to the latter. Does the worst-case resequence-queue size imply the worst-case resequencing delay? How tight are the upper bounds derived on the basis of the worst-case resequencing delay? How does the worst-case resequence-queue size depend on the nature of the load-balancing mechanism, the output-queued structure, the number of priorities, the switch speedup, and the other parameters? To address these issues, we present a methodology for the exact evaluation of the absolute worst-case resequence-queue size in the PBPS context over the spectrum of all plane load-balancing mechanisms. As it turns out, there are very simple and very general answers to these questions, as summarized by the following results.

- For a small number of planes, the upper bound established on the basis of the worst-case resequencing delay is substantially different from the exact worst-case resequence-queue size.
- In the case of dedicated and shared output-queued architectures, the maximum resequence-queue size is proportional to the resequencing delay by a constant factor and, therefore, the resequencing delay corresponding to the worst-case resequence-queue size is the maximum possible one.
- In the case of the crosspoint output-queued architecture with round-robin column scheduling, the resequencing delay corresponding to the worst-case resequence-queue size is not necessarily the maximum possible one.
- The worst-case resequence-queue sizes for the dynamic state-independent load-balancing mechanisms con-

[1] Lossless operation can be achieved by employing either a credit or a backpressure type of a feedback flow control mechanism from the switch planes to the input adaptors, and from the output adaptors to the switching planes.

sidered (random, cyclic, round-robin) are all equal to the absolute worst-case resequence-queue size.
- The worst-case resequence-queue sizes for the dedicated, shared, and crosspoint output-queued architectures are the same, despite some strikingly different characteristics.
- In the presence of priority classes, the worst-case resequence-queue size is bounded for the highest-priority traffic only. Consequently, additional measures are needed to ensure a safe operation for the low-priority traffic classes.

This paper is organized as follows. In Section II, after presenting our model of a PBPS based on buffered-crosspoint switches, it is demonstrated that the knowledge of the worst-case resequence queue is a prerequisite for the safe operation of this type of switches that do not allow packet losses. A detailed description of the state-independent load-balancing mechanisms considered, as well as of the operation of the resequence queue is provided. Section III presents a general analytical method for the derivation of the absolute worst-case resequencing delay and resequence-queue occupancy under the joint effect of multiple resequence flows in the absence of priority traffic classes for the dedicated, shared, and crosspoint output-queued architectures. It is then demonstrated that the absolute results obtained also apply in the case of the state-independent plane load-balancing schemes considered. The circumstances under which the worst-case resequence-queue size arises are identified. The issue of multiple priorities is also addressed. The concluding remarks follow in Section IV.

## II. Preliminaries

### A. Switch Model

The main subject of this paper is the evaluation of the resequence-queue size and the corresponding egress output buffer size required in the context of a PBPS architecture. The system model is illustrated in Figure 1. There are $K$ ($K \geq 2$) parallel switch planes, each of which consisting of an output-queued switch of size $N \times N$ running at a nominal speed of one. Figure 1 specifically illustrates a buffered-crosspoint switch. The ingress side comprises $N$ input adaptors, each performing the *plane load-balancing* function. The egress side comprises $N$ output adaptors, each performing the *plane server* function. Both the input and the output adaptors have a buffering capability. As shown in Figure 1, the input and output adaptors consist of $m$ individual ingress-input and egress-output ports, respectively, running at a relative speed of $s$. Note that the architecture proposed allows us to realize both a fast switch (by choosing a high value of $s$ with an appropriate number of planes $K$) as well as a larger switch size ($mN \times mN$ vs. $N \times N$). Owing to the deployment of multiple planes, the internal switch speed need not be higher than the ingress/egress port speed. The *speedup* or *expansion factor* of the configuration considered is then given by

$$S = \frac{K}{ms}\,. \tag{1}$$

For stability reasons, the expansion factor should be greater than or equal to one.

The architecture considered provides the capability of switching variable-length frames by segmenting the frames at the ingress into fixed-length segments and subsequently transmitting them through the switch via fixed-size switch packets (or cells). In the following a buffer space unit is taken to be equal to the fixed size of a switch packet. We also refer to the transmission time of a switch packet at the nominal switch port speed as a *packet cycle*. It is moreover assumed that the round-trip times between input adaptors and planes, and between planes and output adaptors are negligible. The system supports a number of $P$ priority classes. Priorities are obeyed at all points of queueing, i.e. at the VOQs of the input adaptors, in the buffers of the switch planes, and at the output adaptors. A *flow* is the set of fixed-length segments traveling between a pair of ingress-input and egress-output ports through the system. A given flow may contain multiple streams of individual IP sessions (e.g. TCP or UDP). As mentioned above, these segments are transported by switch packets; consequently, a stream of switch packets between a pair of input/output switch ports can transport up to $m^2$ flows, corresponding to the combinations of the $m$ ingress-inputs and $m$ egress-outputs of the corresponding input and output adaptors, respectively. Also, up to $mN$ flows may arrive at a given egress-output port, and

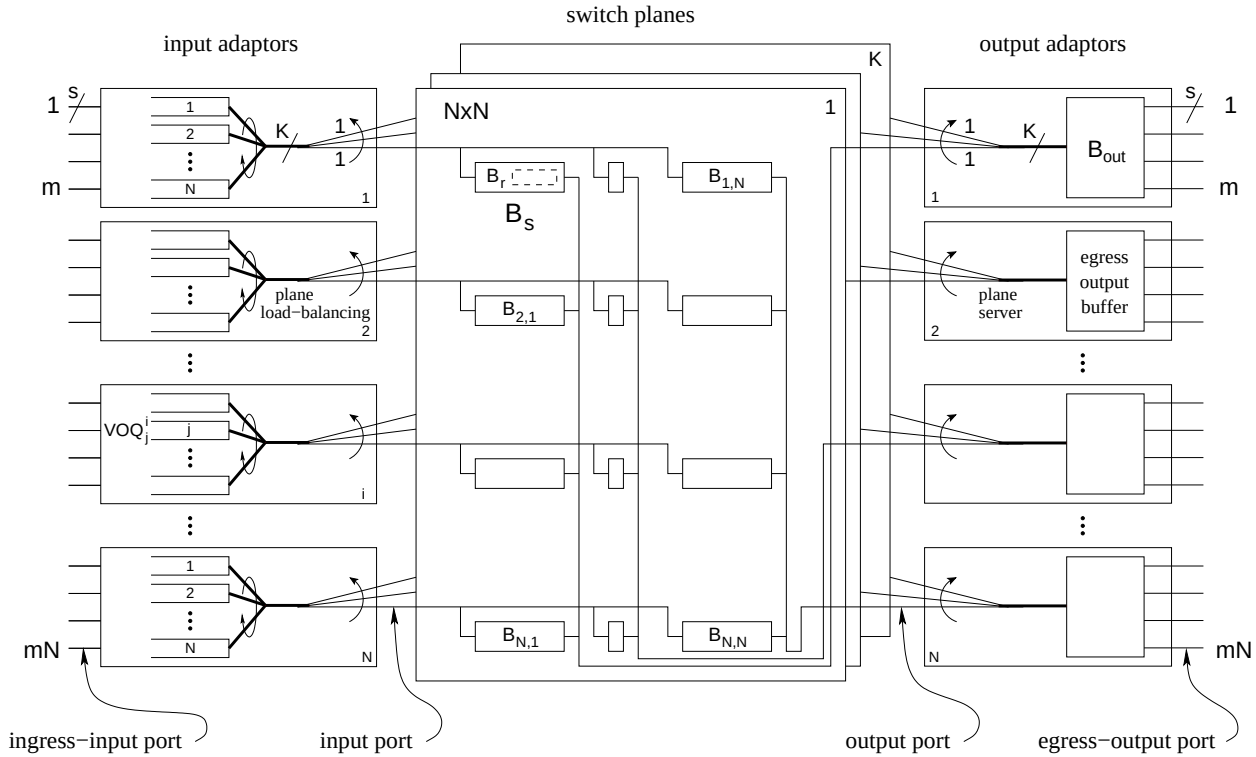


Fig. 1. The PBPS/buffered-crosspoint switch.

up to $m^2N$ flows may arrive at a given output adaptor. Note that a stream of switch packets at a switch input port may contain packets belonging to different flows interleaved, either because of the simultaneous arrival of two frames at two different ingress-input ports (in case of $m > 1$) or because of a series of short acknowledgments destined for different outputs (in case of $m = 1$), or because of a combination thereof. The switch-related information of packets is their incoming and outgoing ports, their sequence number as well as their priority. At the output adaptors, however, the packets (and the corresponding segments) are associated with the ingress-input/egress-output flows so that resequencing can be performed at the flow level.

Within the switch planes, there is a buffer for every input-output pair, also referred to as crosspoint buffer. Crosspoint buffers of a given column correspond to the same output port and, for ease of implementation, are assumed to be served in a round-robin manner. Each crosspoint buffer distinguishes priorities and has a size of $B_s$ units, which is shared among the $P$ priority classes. Let $B_r$ be the maximum buffer occupancy with packets of the highest priority.

Each input adaptor contains $N$ Virtual Output Queues (VOQs) corresponding to the $N$ switch output ports. Each VOQ can be fed at a maximum rate of $ms$ packets per packet cycle. VOQs are served according to priorities, and according to a round-robin scheme within a priority. In one packet cycle, the *plane load-balancing scheme* is assumed to have the capability to dispatch one packet to each plane, i.e., up to $K$ packets to the corresponding planes, as shown in Figure 1. Furthermore, the $K$ packets could be taken from the same VOQ if the remaining VOQs are empty (this is reflected by the thick lines in the figure). Each output adaptor contains an egress output buffer of size $B_{\text{out}}$ units shared among the $m$ egress-output ports and fed from the $K$ planes. In one packet cycle, the *plane server scheme* can transfer one packet from each plane to an output adaptor, implying a maximum rate of $K$ packets per packet cycle.

In addition to the buffered-crosspoint queued architecture depicted in Figure 1, this paper also considers the dedicated and shared output-queued architectures. The dedicated output-queued architecture assumes a single buffer per output port that corresponds to all crosspoint buffers of the column associated with this port in the buffered-crosspoint architecture. The shared output-queued architecture assumes a single buffer for all output ports that corresponds to all crosspoint buffers. Note that a buffered-crosspoint queued architecture with a

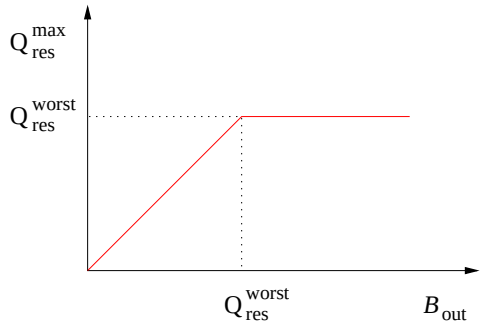


Fig. 2. Maximum resequence-queue size vs. output buffer size.

FIFO column scheduling is logically equivalent to a dedicated output-queued architecture. Consequently, the results reported for the dedicated output-queued architecture apply also to this case. Thus, in the remainder of this paper, when referring to a buffered-crosspoint queued architecture, we assume a round-robin column scheduling. Let $B$ denote the maximum buffer occupancy in a plane with packets of the highest priority that are destined to an output port associated with an output adaptor. In the case of the buffered-crosspoint architecture, it holds that $B = NB_r$. Note that packets can join this buffer (of a given plane) at a maximum rate of $N$ packets per cycle.

According to the scheme described above, packets are transferred from the input to the output adaptors over multiple switch planes. As a result, packets (and segments) corresponding to a given flow may arrive at an output adaptor in an order different from the one in which they originally departed from the input adaptor. As FIFO delivery is required, out-of-sequence packets must wait at the output adaptor for the *missing packets* to arrive so they can be put back into proper sequence. Therefore, in addition to queueing and transmission delay, out-of-sequence packets incur resequencing delay due to the reordering of the output stream at the receiving output adaptor. Out-of-sequence packets stored in an output buffer constitute a virtual *resequence queue*. When a missing packet arrives, it *releases* the corresponding resequence packets from the resequence queue. This release is not an actual packet movement but merely a pointer manipulation. Consider an arbitrary output adaptor and let $Q_{\text{res}}^{\max}$ denote the maximum resequence-queue size in the output buffer of size $B_{\text{out}}$. Clearly, it holds that $Q_{\text{res}}^{\max} \leq B_{\text{out}}$. For small values of $B_{\text{out}}$, we expect $Q_{\text{res}}^{\max}$ to be equal to $B_{\text{out}}$ as the first few out-of-sequence packets can easily fill the output buffer. The fact that no further departures are possible implies that no subsequent arrivals are possible, and therefore the system is deadlocked. However, for sufficiently large values of $B_{\text{out}}$, we expect $Q_{\text{res}}^{\max}$ to be smaller than $B_{\text{out}}$ and, in fact, to converge to the worst-case resequence-queue size, denoted by $Q_{\text{res}}^{\text{worst}}$ as shown in Figure 2. Clearly, the minimum output buffer size required for safe operation is equal to the worst-case resequence-queue size incremented by one packet, i.e. $B_{\text{out}} \cong Q_{\text{res}}^{\text{worst}}$. The objective of this work is the evaluation of this value.

### *B. Load-Balancing Mechanisms*

Each input adaptor contains a plane load-balancing function that dynamically dispatches the outgoing packets to the $K$ available paths (planes). A packet can be dispatched to a given switch plane if the corresponding flow control permits it. A packet is said to be *eligible* if there is a plane that can accept it. We also refer to the planes to which an eligible packet can be dispatched as *eligible planes*. Within a packet cycle, there can be at most one packet dispatched to any given plane. Therefore, a high efficiency is achieved when packets are being dispatched to all $K$ planes within each packet cycle.

As mentioned in the Introduction, various dynamic path-selection algorithms with varying degree of complexity are conceivable. In this paper we focus on the less complex state-independent load-balancing mechanisms and in particular on the following schemes mentioned in the Introduction:

*Round-Robin*: a round-robin mechanism is applied to the aggregate stream of eligible packets; each packet is dispatched to the plane pointed by the counter that is incremented until the first eligible plane is found.

*Random*: packets are randomly dispatched to the eligible planes.

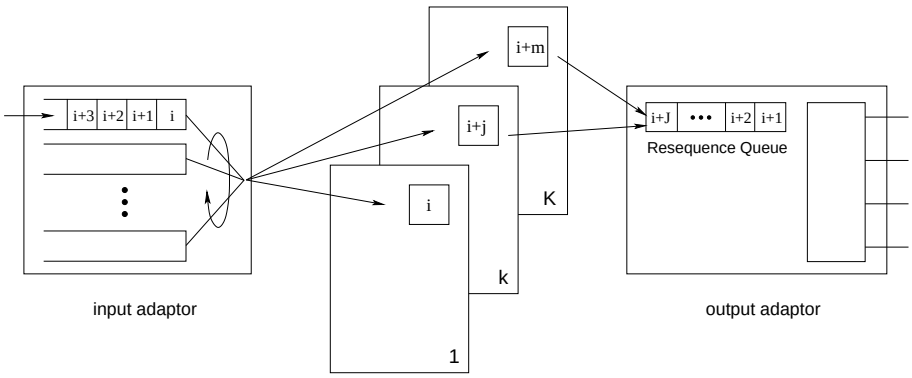


Fig. 3. Resequence queue.

*Cyclic*: It is applied in switch input adaptors that operate in slotted fashion. According to this scheme, successive slots are cyclically preassigned to planes, regardless of the arrival pattern of the packets.

Note that the above schemes cannot exclude the possibility that an arbitrary number of successive packets of a given flow are dispatched to the same plane. Owing to this property, severe load asymmetries between planes can occur, as discussed in detail in Appendix A. Reducing these asymmetries requires more complex schemes that do not possess this property. These include per-flow round-robin and state-dependent schemes, which result in improved resequence behavior (e.g. reduced resequence buffer sizes) as simulations we conducted have shown. Because of space limitations, these schemes are evaluated in a subsequent work.

At the egress side of the switch, we have considered various mechanisms for the *plane server scheme*. In contrast to the load-balancing schemes at the ingress side, simulations we conducted on similar schemes at the egress side revealed that more complex state-dependent plane sever schemes did not result in any advantage over the round-robin scheme. Consequently, in the remainder we assume that the *plane server scheme* operates in a round-robin fashion at the egress side of the switch.

### *C. The Resequence Queue*

In the PBPS context many flows can be simultaneously under resequencing at an arbitrary output adaptor. Let us consider a specific flow, and in particular a typical sequence of packets $i, i+1, i+2, \ldots$, belonging to this flow. In Figure 3 it is shown that packets $i+1, \ldots, i+J$ have already arrived and have been stored at the output buffer of the corresponding destination adaptor, forming the resequence queue while waiting for packet $i$ to arrive. Note that the sequence $i+1, \ldots, i+J$ may not necessarily contain all consecutive packets, as some packets $i+j$, with $1 < j < J$ may have been dispatched to plane 1 and therefore reside behind packet $i$.

Let us now consider the case where the output buffer is not full, and suppose that packets $i+1, \ldots, i+K-1$ were transferred within one packet cycle from planes 2, ..., $K$, respectively, to the output buffer along with packet $p$ transferred from plane 1. As packet $i$ remains in plane 1, packets $i+1, \ldots, i+K-1$ join the resequence queue. The same effect occurs when the output buffer is full, albeit on a different time scale. The longest period arises when all packets in the output buffer are destined to a particular egress-output port such that the output buffer is emptied at a rate of $s$ packets per packet cycle. In this case, packets $p$, $i+1, \ldots, i+K-1$, will be transferred to the output buffer in $K/s$ successive packet cycles. This period constitutes an *arbitration cycle* during which the plane server scheme transfers one packet from each of the $K$ planes to the output buffer. Clearly, the resequence-queue size increases by the same amount in both cases, although the arbitration cycle lasts one packet cycle in the former and $K/s$ packet cycles in the latter case. This indicates that as far as the resequence-queue size is concerned, time should be measured in relative (arbitration cycles) rather than absolute terms (packet cycles). More formally,

**Arbitration cycle:** The period it takes for the plane server scheme to check and serve (if possible) all $K$ planes provided they are not all empty.

**Remark 1.** From the above it follows that a delay of one arbitration cycles corresponds to a resequencing delay of at most $K/s$ packet cycles occurring when the egress output buffer is constantly filled with packets that are all destined to the same egress-output port.

In the remainder, unless otherwise indicated, the term cycle will refer to an arbitration cycle.

## III. Analysis of Multiple-Flows

In the PBPS context, many flows can be simultaneously under resequencing at an arbitrary (tagged) output adaptor all of which should be jointly taken into account. First, a general analytical method is presented for the derivation of the absolute worst-case resequencing delay $C_{\text{worst}}$ and resequence-queue occupancy $Q_{\text{res}}^{\text{worst}}$ under the joint effect of *multiple resequence flows* in the absence of priority traffic classes (i.e. $P = 1$) for the dedicated, shared, and crosspoint output-queued architectures. We then demonstrate that the absolute results obtained apply in the case of the three state-independent plane load-balancing schemes considered. Finally the results are extended to the case of multiple priorities.

### A. Worst-Case Resequencing Delay

We consider the crosspoint output-queued architecture, and describe here a scenario resulting in the maximization of the resequencing delay. At the instant when packet $i$ is dispatched to plane 1, buffers $B_{j,1}$ ($j = 1, \ldots, N$) of the first plane corresponding to the first output adaptor, as well as $B_{\text{out}}$ are full. This is indicated in Figure 4 by the gray packets, all destined to the first egress output port of the first output adaptor.

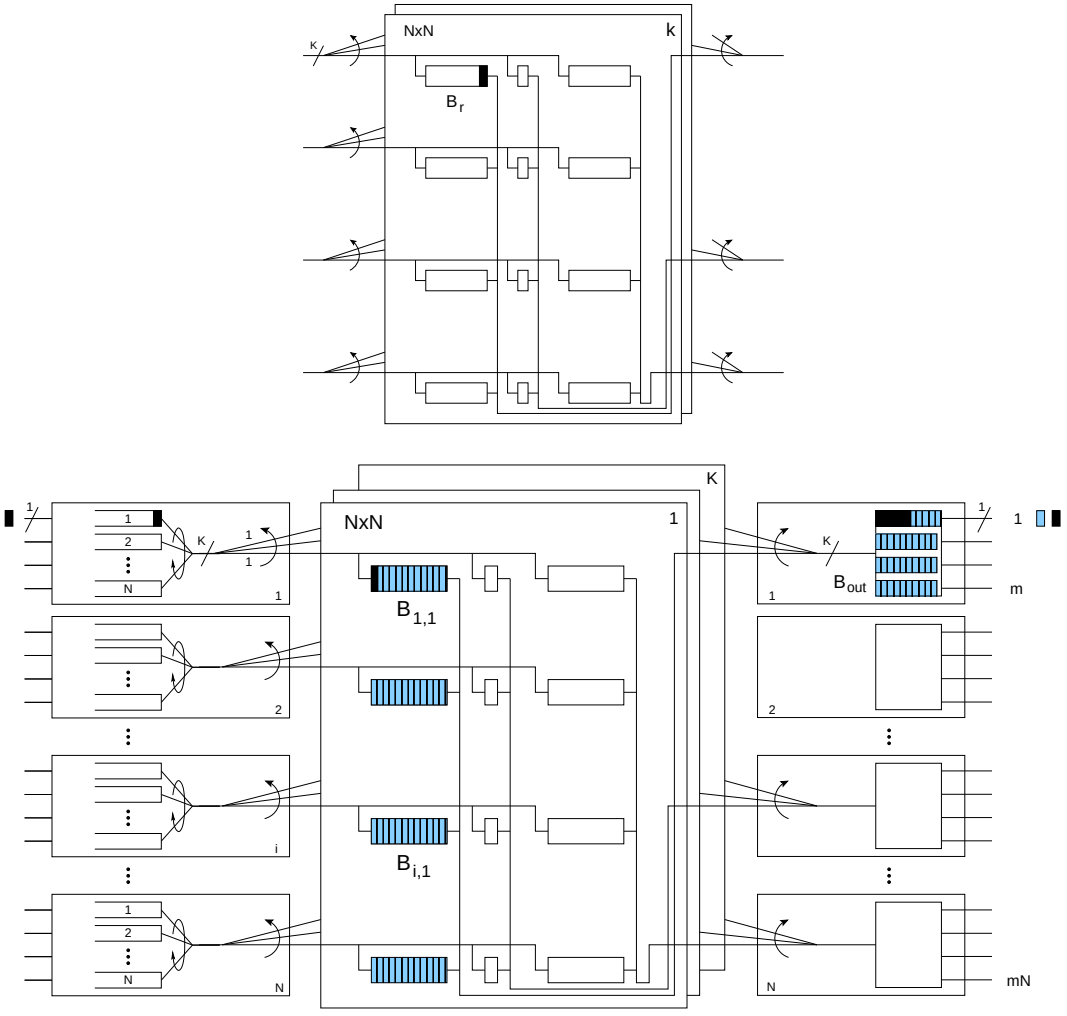


Fig. 4. Worst-case resequencing delay for a PBPS/crosspoint-queued switch configuration.

At the same time, a subsequent packet, say packet $i+1$, is dispatched to an empty plane, say plane 2. In the next cycle, packet $i+1$ and a gray packet will be dispatched to the output buffer. Packet $i$ will arrive at the output buffer after all gray packets have already arrived, i.e. after $NB_r$ cycles. Thus, the worst-case resequencing delay (expressed in arbitration cycles) is given by

$$C_{\text{worst}} = NB_r\,, \tag{2}$$

or, by virtue of Remark 1, $KNB_r/s$ packet cycles.

Similarly, in the case of a dedicated/shared output-queued architecture, the worst-case resequencing delay (expressed in arbitration cycles) is given by

$$C_{\text{worst}} = B\,, \tag{3}$$

or, by virtue of Remark 1, $KB/s$ packet cycles.

**Remark 2.** The worst-case resequencing delay does not depend on $m$, the number of individual ports per adaptor.

Note that in obtaining these results no assumptions were made regarding the load-balancing scheme. Let us now address the following issue. Is it possible, under the deployment of a plane load-balancing scheme, to arrive at the scenario described above, whereby the loading asymmetry between plane 1 and any of the remaining planes is so extreme? Note that the plane load-balancing scheme ensures that the load is equally distributed among the planes. Consequently, at first glance, this scenario, in which all the gray packets reside in one of the planes, seems rather unlikely. Nevertheless, in Appendix A it is shown that this scenario is feasible under any of the state-independent load-balancing schemes considered.

### *B. Worst-Case Resequence-Queue Size*

Let us now consider a snapshot of the output resequence queue at a random arbitration cycle $\tau$, and let us assume that there are $F$ flows under resequencing indexed by $f = 1, \ldots, F$. Let us denote by $P^{(f,i)}_{k,t,\tau}$ a packet that entered the $k$-th plane at time $t$ and was subsequently dispatched to the egress output buffer in cycle $\tau$, that corresponds to the $f$-th flow, and whose sequence number is equal to $i$. To distinguish resequence packets $P^{(\cdot,\cdot)}_{\cdot,\cdot,\cdot}$ residing in the resequence queue from the rest of the packets, the notation $R^{(\cdot,\cdot)}_{\cdot,\cdot,\cdot}$ is used ($R$ instead of $P$).

The resequence packets are placed into the rows and columns of the table depicted in Figure 5 according to the cycle in which and the plane from which they arrive at the output adaptor. In cycle $\tau$, a resequence packet $R^{(i,4)}_{3,\theta,\tau}$ is dispatched from the third plane to the output buffer. This packet belongs to the $i$-th flow, and its sequence number is equal to four. According to Figure 5, this packet waits for the missing packet $P^{(i,3)}_{K,\cdot,\tau+1}$, which is located in the $K$-th plane and will be dispatched to the output buffer in the next cycle $\tau+1$. Furthermore, $R^{(i,4)}_{3,\cdot,\cdot}$ as well as $R^{(i,2)}_{1,\cdot,\cdot}$ wait for packet $P^{(i,1)}_{2,\epsilon,\upsilon}$ which is located in the second plane and will be dispatched to the output buffer in cycle $\upsilon$. Finally, $P^{(i,3)}_{K,\cdot,\tau+1}$ will also join the resequence queue in cycle $\tau+1$ as it will have to wait for $P^{(i,1)}_{2,\epsilon,\upsilon}$, which will arrive later in cycle $\upsilon$.

The missing packets will arrive after cycle $\tau$ and are therefore indicated in subsequent columns. From the set of missing packets corresponding to a given flow, the one with the smallest sequence number is called the *principal missing packet*. For example, in Figure 5, $P^{(i,3)}_{K,\eta,\tau+1}$ and $P^{(i,1)}_{2,\epsilon,\upsilon}$ are missing packets corresponding to flow $i$, with the latter being the principal one. As there are $F$ flows under resequencing, there are $F$ principal missing packets. These are indicated in Figure 5 with a sequence number equal to one.

A relation between the parameters associated with resequence and non-resequence packets is given by the following remark.

**Remark 3.** Let $R^{(f,i+j)}_{k_1,t_1,\tau_1}$ be a resequence packet that waits for packet $P^{(f,i)}_{k_2,t_2,\tau_2}$ of the same flow $f$. This implies that $t_2 \leq t_1$ and $\tau_1 < \tau_2$. Furthermore, the two packets cannot stem from the same plane, i.e. $k_1 \neq k_2$,

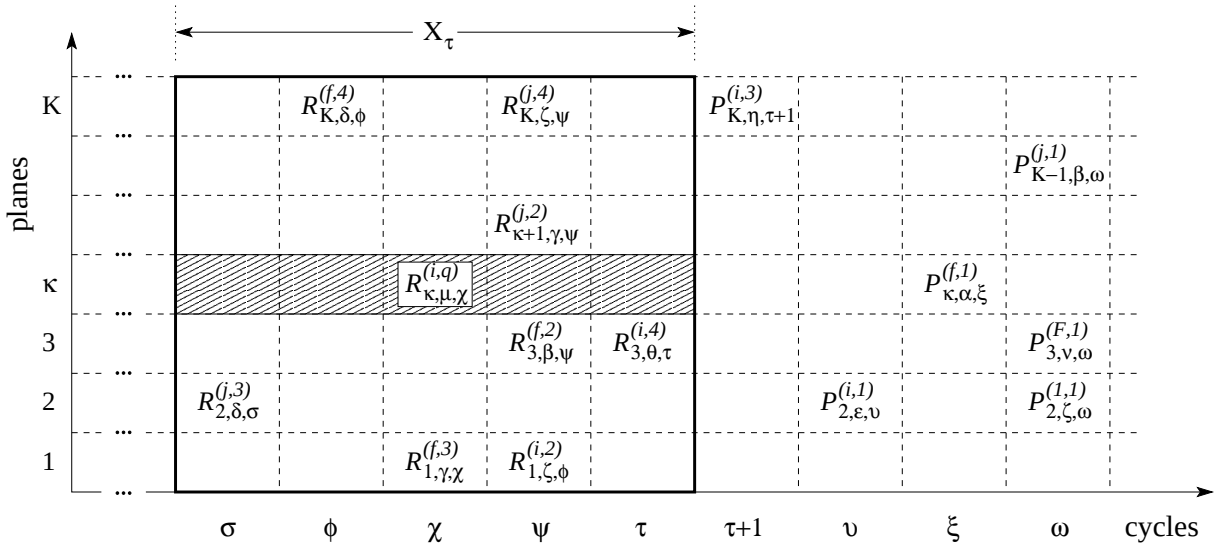


Fig. 5. Snapshot of the output resequence queue.

as in all output-queued architectures considered, a packet of a flow cannot bypass, within a plane, another packet of the same flow.

Let us now focus on packets arriving at the output adaptor from a given plane, say the $k$-th plane. Let $P^{(\cdot,\cdot)}_{k,t_1,\tau_1}$ and $P^{(\cdot,\cdot)}_{k,t_2,\tau_2}$ be two such packets, with $t_1 \leq t_2$. In the following remark, we establish a key differentiating property between the dedicated/shared and the crosspoint output-queued architectures.

**Remark 4.** The dedicated/shared output-queued architecture ensures a FIFO delivery because packets maintain their positions relative to one another. Thus, packets depart from the plane and enter the output adaptor in the same order in which they entered the plane. The same applies to the case of a crosspoint-queued architecture with FIFO column scheduling. In contrast, a crosspoint-queued architecture with round-robin column scheduling does not necessarily guarantee a FIFO delivery. The FIFO property is guaranteed only among packets stemming from the same crosspoint buffer (or input adaptor). Consider that the two packets enter two different crosspoint buffers. Owing to the round-robin service mechanism of the crosspoints buffers of the column corresponding to the output adaptor, the departure sequence of the two packets depends on the difference of the number of packets located in front of them in the corresponding crosspoint buffers. More specifically, $P^{(\cdot,\cdot)}_{k,t_2,\tau_2}$ will depart prior to $P^{(\cdot,\cdot)}_{k,t_1,\tau_1}$ if at the instant it enters its crosspoint buffer the number of packets in the buffer is smaller than the corresponding number of packets in front of $P^{(\cdot,\cdot)}_{k,t_1,\tau_1}$ in its crosspoint buffer. Thus,

$$\begin{aligned} &t_1 \leq t_2 \;\Leftrightarrow\; \tau_1 < \tau_2 \;, && \text{for a dedicated/shared output-queued architecture and} \\ &&& \text{for a crosspoint output-queued architecture for packets} \\ &&& \text{stemming from the same crosspoint buffer, and} \\ &t_1 \leq t_2 \;\not\Leftrightarrow\; \tau_1 < \tau_2 \;, && \text{for the crosspoint output-queued architecture considered.} \end{aligned} \tag{4}$$

We now proceed by defining the **resequencing degree** $X_\tau$ as the maximum of the resequencing delay cycles corresponding to the resequence packets at cycle $\tau$, i.e.

$$X_\tau \;\triangleq\; \tau - \sigma_\tau + 1 \;,$$

where $\sigma_\tau$ denotes the earliest dispatch cycle for the resequence packets located in the resequence queue at cycle $\tau$, formally

$$\sigma_\tau \;\triangleq\; \min_{\theta} \left\{ R^{(\cdot,\cdot)}_{\cdot,\cdot,\theta}(\tau) \right\} \;.$$

Let $R^{(f,j)}_{k_j,t_j,\sigma_\tau}$ be such a resequence packet and let $P^{(f,i)}_{k_i,t_i,x}$ be its corresponding principal missing packet. Clearly, at cycle $\sigma_\tau$ the principal missing packet is stored at the buffer of plane $k_i$ and remains there throughout the $X_\tau$ interval. Note that the *resequencing degree* is bounded above by the worst-case resequencing delay $C_{\text{worst}}$ (expressed in arbitration cycles),

$$X_\tau \;\leq\; C_{\text{worst}}\;, \qquad \forall\, \tau \geq 0\;, \tag{5}$$

with the equality holding iff two successive packets of a flow are simultaneously transferred to two planes, with the buffer of the first of them being full with packets destined to the output adaptor and the buffer of the other being empty.

From the above it follows that the resequence-queue size is equal to the number of resequence packets located inside the *resequence rectangle* depicted in Figure 5. The remaining empty slots do not contribute to the resequence-queue size because they correspond either to no packet arrivals or to arrivals of packets that are not out-of-sequence and, therefore, not shown. Note that the resequence-queue size $Q_{\text{res}}(\tau)$ at cycle $\tau$ is trivially bounded above by $K\times X_\tau$, the maximum possible number of packets inside the resequence rectangle. Similarly, the worst-case resequence-queue size $Q^{\text{worst}}_{\text{res}}$ is trivially bounded above by $K\times B$, the maximum possible number of packets inside the resequence rectangle when the length of the rows is the maximum possible, which, by virtue of (3) and (5), is equal to $B$. This type of bounds has been applied in previous works, such as [1, 14, 15]. We now proceed to derive tight upper bounds determining the worst-case resequence-queue size.

### B.1 Dedicated/Shared Output-Queued Architecture

LEMMA 1: For the dedicated and shared output-queued architectures, the resequence-queue size $Q_{\text{res}}(\tau)$ at cycle $\tau$ is bounded above by

$$Q_{\text{res}}(\tau) \leq (K-1)X_\tau\;, \tag{6}$$

with the equality holding iff throughout the $X_\tau$ interval, at each cycle there are $K-1$ resequence packets stemming from the same $K-1$ planes.

*Proof:* Let us consider the missing packets and in particular a missing packet that has entered its plane at the earliest time compared with the others. Formally,

$$P^{(f^*,i^*)}_{k^*,t^*,s^*} \;\triangleq\; \min_t \left\{ P^{(f,i)}_{k,t,s} \quad \text{s.t.} \quad 1\leq f\leq F\;, \quad 1\leq k\leq K\;, \quad s>\tau \right\}\;. \tag{7}$$

It follows that $P^{(f^*,i^*)}_{k^*,t^*,s^*}$ is a principal missing packet. Suppose that from the missing packets depicted in the right-hand side of the resequence rectangle in Figure 5, $P^{(f,1)}_{k,\alpha,\xi}$ is the one that arrived the earliest, i.e. for all these $P^{(\cdot,\cdot)}_{\cdot,x,\cdot}$ packets it holds that $\alpha\leq x$. Then, $f^*=f$, $i^*=1$, $k^*=k$, $t^*=\alpha$, and $s^*=\xi$. We now consider the row of the resequence rectangle corresponding to the plane $k^*=k$, and establish the following lemma.

*Proposition 1:* The $X_\tau$ slots corresponding to the $k^*$-th row (plane) of the resequence rectangle do not contain any resequence packet.

*Proof:* For the purpose of contradiction, suppose that there exists a slot that contains a resequence packet $R^{(i,q)}_{k^*,\mu,\xi}$ which waits for $P^{(i,1)}_{k',\epsilon,\upsilon}$. These correspond to packets $R^{(i,q)}_{k,\mu,\xi}$ and $P^{(i,1)}_{2,\epsilon,\upsilon}$ shown in Figure 5. According to Remark 3, it holds that $k'\neq k^*$ and

$$\epsilon \leq \mu\;. \tag{8}$$

Packet $R^{(i,q)}_{k^*,\mu,\xi}$ arrives at the output adaptor prior to $P^{(f^*,i^*)}_{k^*,t^*,s^*}$, which by virtue of (4) implies that

$$\mu \leq t^*\;. \tag{9}$$

From definition (7) it follows that the entry time $\epsilon$ of packet $P^{(i,1)}_{k',\epsilon,\upsilon}$ satisfies the following inequality,

$$t^* \leq \epsilon\;. \tag{10}$$

Combining (8)-(10) yields $\epsilon = \mu = t^*$, which implies that during the same cycle, first $P^{(i,1)}_{k',\epsilon,\upsilon}$ and $R^{(i,q)}_{k^*,\mu,\xi}$ were dispatched to planes $k'$ and $k^*$, respectively, and then $P^{(f^*,i^*)}_{k^*,t^*,s^*}$ was dispatched to plane $k^*$. This, however, leads to contradiction because it implies that $P^{(i,1)}_{k',\epsilon,\upsilon}$ has entered its plane earlier than $P^{(f^*,i^*)}_{k^*,t^*,s^*}$. Consequently, the shaded row of Figure 5 does not contain resequence packets. ■

Proposition 1 implies that the number of resequence packets in the resequence rectangle can be at most $KX_\tau - X_\tau$, with the equality holding iff throughout the $X_\tau$ interval, there are $K-1$ resequence packets at each cycle stemming from the remaining $K-1$ planes. This completes the proof of the lemma. ■

The worst-case resequence-queue size is now expressed by the following theorem.

**Theorem 1:** For the dedicated and shared output-queued architectures, the worst-case resequence-queue size $Q^{\text{worst}}_{\text{res}}$ is given by

$$Q^{\text{worst}}_{\text{res}} = (K-1)B \ , \tag{11}$$

achieved by a set of flows corresponding to a single input/output pair of adaptors, and with a corresponding resequencing degree $X_\tau$ equal to the worst-case resequencing delay $B$.

*Proof:* From (5) and (6) it follows that $Q_{\text{res}}(\tau) \leq (K-1)B$, with the equality holding iff equalities hold in both (5) and (6). The plane with the full buffer described in the condition of (5) corresponds to the $k^*$-th row of Proposition 1 of Lemma 1. On the other hand, the equality in (6) implies that throughout the $X_\tau$ interval, at each cycle there are $K-1$ resequence packets stemming from the remaining planes, which in turn excludes having a second plane with full buffer. This implies that all the $K-1$ resequence packets of the first cycle, as well as subsequent cycles, correspond to the same flow (in terms of input/output ports), which potentially corresponds to several flows from the various ingress-input ports of an input adaptor to the various egress-output ports of the output adaptor. Consequently, this leads us to the unique worst-case resequence-queue size scenario in which packets of such flows are dispatched to a plane having a full buffer, and at the same time subsequent packets are dispatched to the remaining planes where the buffers are empty. Note that in this scenario it holds that the resequencing degree is equal to the worst-case resequencing delay. ■

**Remark 5.** Note that the worst-case resequence-queue size implies a maximum resequencing degree.

### B.2 Buffered-Crosspoint Output-Queued Architecture

We begin by noting that Lemma 1 established in the case of a dedicated/shared output-queued architecture does not necessarily hold in the case of the crosspoint output-queued architecture considered, owing to the failure of the FIFO property as (4) no longer implies (9). We therefore proceed by following a different methodology which in the end reveals that indeed Lemma 1 does not hold.

Let us consider a snapshot of the worst-case output resequence-queue size at cycle $\tau$ as shown in Figure 6. Consider the principal missing packets and let us group them based on the input adaptor from which they were dispatched to the planes. Suppose that there are $G$ ($G \leq N$) such groups formed, and without loss of generality assume that they correspond to the first $G$ input adaptors.

Let us also consider the principal missing packets of the $j$-th group and in particular packet $P_j$, which was the first of these packets to enter its plane. This allows us to assign all resequence packets of the flows corresponding to this group to a single flow corresponding to $P_j$, without affecting the number of resequence packets. Consequently, applying this assignment to each of the $G$ groups results in a worst-case scenario in which there is a single principal missing packet for each such group. Let $P$ be the set $\{P_1, \ldots, P_G\}$ of the $G$ principal missing packets. We now denote by $q_k$ the number of the packets $\{P_{k_1}, \ldots, P_{k_j}, \ldots, P_{k_{q_k}}\}$ in $P$ that are dispatched to the $k$-th plane, such that $\sum_{k=1}^{K} q_k = G$. Let $\mathbf{q}$ be the vector $(q_1, \ldots, q_K)$ and $|\mathbf{q}|$ be the number of planes that contain such packets, i.e. $|\mathbf{q}| = \sum_{k=1}^{K} \mathbf{1}_{q_k>0}$, with $1 \leq |\mathbf{q}| \leq \min(K, G)$.

Let us now consider a plane, say the $k$-th plane, for which $q_k > 0$. From the above we deduce that the principal missing packets $\{P_{k_1}, \ldots, P_{k_j}, \ldots, P_{k_{q_k}}\}$ reside in the $k_1$-th, …, $k_{q_k}$-th crosspoint buffer of the column of the $k$-th plane. Furthermore, the worst-case resequence-queue size assumption implies that they are residing in the $k$-th plane throughout the $X_\tau$ interval. As a consequence, these crosspoints contribute packets

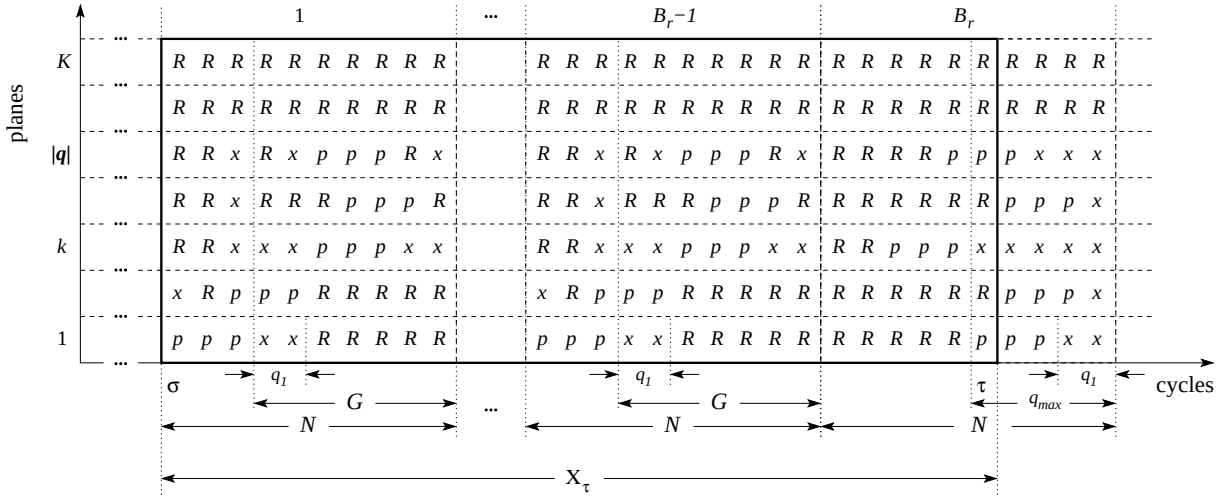


Fig. 6. Snapshot of the output resequence queue (crosspoint-queued architecture).

to the $k$-th row of the resequence rectangle, none of which is a resequence packet. The proof is similar to the proof given for the dedicated/shared case, with (4) now being repeatedly applied to each of the $q_k$ crosspoint buffers, and is therefore omitted. These packets are indicated by the symbol $x$ in Figure 6, where it is assumed that $q_k = 5$ and $q_1 = 2$. The number of resequence packets in the $k$-th row of the $X_\tau$ region is maximized when the remaining $G - q_k$ crosspoint buffers dispatch resequence packets (indicated by the symbol $R$ in Figure 6). Moreover, the worst-case resequence-queue size assumption implies a high resequencing degree which translates into a maximum length of the rows of the rectangle. This is achieved when the remaining ($G + 1$-th, ..., $N$-th) crosspoint buffers also constantly dispatch (nonresequence) packets (indicated by the symbol $p$ in Figure 6), and when at the arrival epochs of packets $P_{k_1}, \ldots, P_{k_{q_k}}$ the corresponding $q_k$ crosspoints are full, i.e. contain $B_r - 1$ packets. Owing to the round-robin service mechanism of the crosspoints buffers, the $k$-th row (starting at cycle $\sigma$) contains $B_r - 1$ identical $N$-cycle-long segments followed by the last segment, which contains the principal missing packet(s), as depicted in Figure 6. Clearly, for a given vector $\mathbf{q}$, both the resequencing degree and the number of resequence packets are maximized when for instance the resequence packet(s) are at the beginning and the missing packet(s) are at the end of the last segment. It will later be shown that a vector $\mathbf{q}$ that maximizes the resequence-queue size does not necessarily maximize the resequencing delay.

**Remark 6.** Depending on the values of $N, G$ and $\mathbf{q}$, it may not always be possible that in **all** of the first $|\mathbf{q}|$ rows of the last segment the packets are arranged in the form $R, \ldots, R, p, \ldots, p, x, \ldots, x$, as shown in Figure 6. Consequently, the expressions subsequently derived constitute upper bounds, which are tight for some special combinations of $N, G$ and $\mathbf{q}$, as will be discussed below.

The number of resequence packets in the $k$-th row of the last segment cannot exceed the quantity $\min(N + 1 - q_{\max}, G - q_k)$, where $q_{\max} = \max_{1 \le k \le K}\{q_k\}$, as depicted in Figure 6. Furthermore, by considering all the planes, it follows that

$$X_\tau \;=\; B_r N + 1 - \max_{1 \le k \le K}\{q_k\}\;. \tag{12}$$

Let us now consider a plane, say the $i$-th plane, for which $q_i = 0$. The worst-case resequence-queue size implies that the first $G$ crosspoints constantly provide resequence packets corresponding to the $G$ resequence flows and that the remaining $N - G$ crosspoints are empty so that the $i$-th row is filled with resequence packets. This is illustrated in the top two rows in Figure 6.

From the above it follows that the number of resequence packets $Q_{\text{res}}$ is bounded above by

$$Q_{\rm res} \;\leq\; (B_r-1)\;\left\{\sum_{k=1}^{K}[\mathbf{1}_{q_k>0}(G-q_k)+\mathbf{1}_{q_k=0}N]\right\}+ \\ +\sum_{k=1}^{|\mathbf{q}|}\min(N+1-\max_{1\leq k\leq K}\{q_k\},G-q_k)+(K-|\mathbf{q}|)(N+1-\max_{1\leq k\leq K}\{q_k\})\;,$$

and by using the fact that $\sum_{k=1}^{K} q_k = \sum_{k=1}^{K} \mathbf{1}_{q_k>0}\; q_k = G$ and $\sum_{k=1}^{K} \mathbf{1}_{q_k>0} = |\mathbf{q}|$,

$$Q_{\rm res} \;\leq\; f(G,\mathbf{q}) \;\triangleq\; (B_r-1)f_B(G,\mathbf{q})+f_L(G,\mathbf{q})\;, \tag{13}$$

with

$$f_B(G,\mathbf{q}) \;\triangleq\; (K-|\mathbf{q}|)N+(|\mathbf{q}|-1)G\;, \tag{14}$$

and

$$f_L(G,\mathbf{q}) \;\triangleq\; \sum_{k=1}^{|\mathbf{q}|}\min(N+1-\max_{1\leq k\leq K}\{q_k\},G-q_k)+(K-|\mathbf{q}|)(N+1-\max_{1\leq k\leq K}\{q_k\})\;. \tag{15}$$

From (13) it follows that the worst-case resequence-queue size $Q_{\rm res}^{\rm worst}$ is bounded above by

$$Q_{\rm res}^{\rm worst} \;\leq\; M(N,K) \;\triangleq\; \max_{G,\mathbf{q}} f(G,\mathbf{q})\;. \tag{16}$$

In order to derive the function $M(N,K)$, we first evaluate the following two tight upper bounds,

$$M_L(N,K) \;\triangleq\; \max_{G,\mathbf{q}} f_L(G,\mathbf{q})\;, \tag{17}$$

$$M_B(N,K) \;\triangleq\; \max_{G,\mathbf{q}} f_B(G,\mathbf{q})\;. \tag{18}$$

We begin with the evaluation of $M_L(N,K)$. Conditioning first on $G$ and then on $|\mathbf{q}|$, (17) yields

$$M_L(N,K) \;=\; \max_{G,\mathbf{q}} f_L(G,\mathbf{q}) = \max_{1\leq G\leq N}\left\{\max_{1\leq q\leq \min(K,G)}\left\{\max_{\mathbf{q}\text{ s.t }|\mathbf{q}|=q} f_L(G,\mathbf{q})\right\}\right\}\;,$$

or

$$M_L(N,K) \;=\; \max_{1\leq G\leq N}\left\{\max_{1\leq q\leq \min(K,G)}\{f_L(G,q)\}\right\}\;, \tag{19}$$

with

$$f_L(G,q) \;\triangleq\; \max_{\mathbf{q}\text{ s.t }|\mathbf{q}|=q}\{f_L(G,\mathbf{q})\}\;. \tag{20}$$

We now establish the following Lemmas.

LEMMA 2: Consider the vector $\mathbf{q}^*(G,q)$ with $G+q-q\lceil\frac{G}{q}\rceil$ elements having the value $\lceil\frac{G}{q}\rceil$, $q\lceil\frac{G}{q}\rceil-G$ elements having the value $\lceil\frac{G}{q}\rceil-1$, and the remaining $K-q$ elements being equal to zero. The vector $\mathbf{q}^*(G,q)$ is maximal, i.e. $f_L(G,\mathbf{q}^*(G,q))=f_L(G,q)$, and also results in the largest resequencing degree.

*Proof:* See Appendix C. ■

*Corollary 1:* It holds that

$$f_L(G,q) \;=\; (K-1)N-(q-1)(N-G)-(K-q)\left(\left\lceil\frac{G}{q}\right\rceil-1\right)\;. \tag{21}$$

*Proof:* Immediate from Lemma 2 and (15). ■

LEMMA 3: It holds that

$$M_L(N,K) = \max_{G,\mathbf{q}} f_L(G,\mathbf{q}) = (K-1)N \ ,$$

for either $G = N$ and $q = \min(K,N)$, or $G = q = 1$.

*Proof:* See Appendix C. ■

LEMMA 4: It holds that

$$M_B(N,K) = \max_{G,\mathbf{q}} f_L(G,\mathbf{q}) = (K-1)N \ ,$$

for either $G = N$ or $q = 1$.

*Proof:* See Appendix C. ■

We now proceed by establishing the following theorem based upon which the result sought is obtained.

**Theorem 2:** The worst-case resequence-queue size is given by

$$Q_{\text{res}}^{\text{worst}} = (K-1)NB_r \ , \tag{22}$$

for either $G = N$ and $|\mathbf{q}| = \min(K,N)$, with a corresponding resequencing degree $X_\tau = NB_r + 1 - \left\lceil \frac{N}{K} \right\rceil$, or $G = |\mathbf{q}| = 1$, with a corresponding resequencing degree $X_\tau = NB_r - 1$.

*Proof:* From (13), (17), (18), and by combining the results and conditions given by Lemma 3 and Lemma 4, (16) yields $Q_{\text{res}}^{\text{worst}} \leq M(N,K) = (B_r - 1)M_B(N,K) + M_L(N,K) = B_r(K-1)N$. We now demonstrate that for each of these conditions the equality holds. According to Remark 6, a sufficient condition for equality is when the packets in the first $|q|$ rows of the last segment are arranged in the form $R,\ldots,R,p,\ldots,p,x,\ldots,x$. For condition $G = N$ and $|\mathbf{q}| = \min(K,N)$, this is possible when, for each $k$ ($1 \leq k \leq |\mathbf{q}|$), packets $P_{k_1},\ldots,P_{k_{q_k}}$ reside in $q_k$ consecutive crosspoints in the $k$-th plane, and at the beginning of a segment the round-robin mechanism serves the $(q_k+1)$-th (modulo $N$) crosspoint. For condition $G = |\mathbf{q}| = 1$, this is possible when, at the beginning of a segment, the round-robin mechanism of the plane in which the only missing packet is located serves the crosspoint following the one in which the missing packet is located. A detailed description of the corresponding worst-case scenario is provided in Appendix B. In conclusion, the function $M(N,K)$ does indeed represent the worst-case resequence-queue size. The resequencing degree corresponding to each of the conditions is obtained from (12) by considering the maximal vector $\mathbf{q}^*(G,|\mathbf{q}|)$ as defined by Lemma 2. ■

**Remark 7.** In contrast to the dedicated/shared output-queued architecture case, here the worst-case resequence-queue size does not necessarily imply a maximum resequencing degree. This is because, in the case where $K < N$ and the worst-case resequence-queue size is obtained by the vector $\mathbf{q}^*(N,K)$, the corresponding resequencing degree is $X_\tau = NB_r + 1 - \left\lceil \frac{N}{K} \right\rceil$ which is less than the worst-case resequencing delay $NB_r$. However, the resequencing degree corresponding to the worst-case resequence-queue size obtained by the vector $\mathbf{q}^*(1,1)$ is equal to the worst-case resequencing delay $NB_r$. Note also that in the former case Lemma 1, which was established for a dedicated/shared output-queued architecture, does not hold.

### B.3 Comparison of the Output-Queued Architectures

In this section we comment on the results given by Theorems 1 and 2. Note that by making use of the relation $B = NB_r$, (22) of Theorem 2 can be rewritten as

$$Q_{\text{res}}^{\text{worst}} = (K-1)B \ . \tag{23}$$

Consequently, the above expression for the worst-case resequence-queue size applies to both a crosspoint architecture and a dedicated/shared architecture. There are, however, two differences between the dedicated/shared and the crosspoint architecture considered. First, the worst-case resequence-queue size in the former case is always achieved by a single flow, whereas in the latter case it can also be achieved by $N$ flows. Second, according to Remark 7, the worst-case resequence-queue size implies the worst-case resequencing delay only in the former case. In the latter case, the worst-case resequence-queue size does not necessarily imply a maximum resequencing delay.

*Corollary 2:* For small values of $K$, the upper bounds established by the theorems are substantially different from the trivial upper bound $KB$ (e.g. for $K = 4$, they are 25% smaller than the trivial one).

**Remark 8.** The worst-case resequence-queue size does not depend on $m$ or $s$.

Note again that in obtaining these results no assumptions were made regarding the load-balancing scheme. As in the evaluation of the worst-case resequencing delay, it turns out that these results also apply in the case of the state-independent plane load-balancing schemes considered. The scenarios are constructed using the technique presented in Appendices A and B. This technique involves two steps. First, the creation of backpressure and backlog in the output adaptor, the switch buffers, and the input adaptors, by considering a hot-spot scenario. Second, the consideration of appropriately synchronized arrival patterns.

### C. Priorities

We consider here a system operating under the presence of multiple priorities. In this case, the results obtained in the preceding sections III-A and III-B apply to the highest-priority class. Let us now consider a lower-priority flow of packets $i, i + 1, \ldots$. The worst-case resequence-queue size and resequencing delay arise when packet $i$ is dispatched to a plane and stays there indefinitely because it is always preempted by traffic of higher priority, and at the same time packets subsequent to $i$ are transferred to the remaining planes and from there to the output adaptor. In this case, both the worst-case resequencing delay and the worst-case resequence-queue size are unbounded. Note also that this holds independently of $P$, the number of priorities. Consequently, the model considered so far is prone to deadlock situations under priority traffic and regardless of the size of the output buffers.

From the above it follows that additional measures are needed to ensure safe operation. One solution, for instance, could be to devise a mechanism to identify and release lower-priority missing packets that have been blocked for a long period of time. This amounts to overriding priorities at these release instants. As a first attempt, we have simulated numerous such schemes and found that a reduction of the maximum measured resequence-queue sizes could only be achieved by schemes of high complexity. The worst-case resequence-queue sizes associated with these schemes are subject of further study. An alternative solution, which is in contrast to the original assumption, could be based on totally ignoring priorities within the switch planes, while they are still obeyed at the input and output adaptors. This approach is quite attractive from the cost point of view, and the resequence buffer can be dimensioned according to the results presented in Section III. Furthermore, the overall impact on the QoS guarantees is minimal under normal circumstances as most of the queueing takes place at the output adaptors and very little in the switch planes. In the case of extreme circumstances, however, and given that there is no priority handling inside the switch planes, additional measures may need to be introduced at the input adaptors in order to preserve the QoS guarantees. This is a subject of further investigation.

## IV. Conclusions

The general class of parallel buffered packet switches (PBPS) based on a dedicated, shared, or buffered-crosspoint output-queued architecture has been considered. The issue of evaluating the minimum resequence-queue size required for a deadlock-free lossless operation in the presence of multiple flows was addressed. An analytical method was developed for evaluating the absolute worst-case resequence-queue size, independently of the load-balancing mechanism used, and it was proven that this is also the worst case for some typical load-balancing mechanisms, such as random, cyclic, and round-robin. It was demonstrated that for these mechanisms, the worst-case resequence-queue size depends only on the characteristics of the parallel switches, and in particular on the number of their ports, the size of the internal switch buffers, and the number of parallel planes. The circumstances under which the worst-case resequence-queue size arises were identified. In the case of a dedicated/shared architecture, the exact worst-case resequence-queue size is obtained based on the worst-case resequencing delay. Whereas earlier works derived only loose upper bounds of the worst-case resequence-queue size based on the maximum possible resequencing delay, we derived the exact values, which for a small number of planes are significantly lower than these loose upper bounds. Interest-

ingly, in the case of the crosspoint-queued architecture with round-robin column scheduling, it is found that the resequencing delay corresponding to the exact worst-case resequence-queue size is not necessarily the maximum possible one. Furthermore, despite some strikingly different characteristics of the output-queued architectures considered, such as the lack of the FIFO property of the crosspoint-queued architecture, the worst-case resequence-queue size is the same for all three architectures.

For example, for a PBPS architecture based on six 64×64 switch planes, with each having a maximum buffering capability of 1024 packets per output port, the exact worst-case resequence-queue size is equal to 5120 packets for the dedicated, shared, and crosspoint output-queueing. This value is about 17% lower than the loose upper bound of 6144 packets obtained as the product of the number of planes (6) and the worst-case resequencing delay (1024). This gap becomes even more pronounced for a smaller number of switch planes. Reducing the number of switch planes to four, results in a worst-case resequence-queue size of 3072 packets, which is 25% less than the loose upper bound of 4096 packets.

The results obtained apply to the highest-priority class only. In systems having multiple priority classes, however, additional measures are needed to ensure a safe operation. Their nature has been outlined, and the particular choice of the preferred one is a subject of future work. A further reduction of the resequence buffer size required is possible by adopting more sophisticated (state-dependent) load-balancing mechanisms. The analysis of these schemes builds upon the analytical methods developed and the results obtained in this paper, and is the subject of a future publication.

## APPENDIX A
## SCENARIO FOR EXTREME ASYMMETRY

We show here that the scenario with the extreme plane asymmetries considered in Section III-A is feasible under any of the state-independent load-balancing schemes considered.[2] It is based on the property that any of these schemes cannot exclude the possibility that successive packets of a given flow are dispatched to the same plane. While for the random scheme this is apparent, for the other two schemes this is less so. Let us for instance assume a round-robin mechanism. We begin the construction of the scenario by initially considering an empty system and flows arriving at all ingress-inputs at full rate and destined to the first egress-output of the first output adaptor. This hot-spot scenario eventually causes the egress output buffer as well as the crosspoint buffers of the first column of each plane to fill up.[3] Owing to the assumption of an expansion factor greater than or equal to one and by virtue of (1), the potential input rate of $K$ is larger than the rate of $s$ packets per packet cycle at which the egress output buffer is emptied. Therefore, the egress output buffer remains filled and exercises constant backpressure to the planes. Consequently, a period of one arbitration cycle lasts $K/s$ packet cycles. This implies that the rate at an ingress-input can be at most $K$ packets per arbitration cycle.

We now show how one arrives at the scenario described by considering appropriately synchronized arrival patterns. Suppose, without loss of generality, that each of the $N$ round-robin input load-balancers points to the first plane. The plane asymmetry is now created by changing the arrival pattern as follows. Let us consider the arbitration cycle that begins at the instant when one packet place becomes available in $B_{1,1}$ of the first plane owing to a packet transfer to the output adaptor.[4] All flows stemming from the first input adaptor are stopped, except the one between the first ingress-input port and the first egress-output port. The rate of this flow is changed to one (gray) packet every $N$ (arbitration) cycles. The first packet of this flow is transferred to buffer $B_{1,1}$ of the first plane. Consider now the instant (within the same arbitration cycle) when one packet

[2] Although this scenario refers to a crosspoint-queued architecture, a similar scenario can be constructed in the case of a dedicated/shared output-queued architecture

[3] Note that when the crosspoint buffers of the switches are filled before the egress output buffer, the input VOQs also start filling up. One then arrives at this state by allowing the egress output buffer to fill up as well and subsequently stopping the incoming traffic until the input VOQs become empty.

[4] In the case of a cyclic load-balancing mechanism, we assume that the flow out of the output adaptor is momentarily interrupted until the packet transfer occurs at a slot preassigned to the first plane.

place becomes available in $B_{1,1}$ of the second plane owing to a packet transfer to the output adaptor. A new flow between the first ingress-input port and the last egress-output port is initiated at a rate of $K-1$ packets every $N$ (arbitration) cycles. According to the dynamic state-independent schemes considered, the first packet of this flow is transferred to buffer $B_{1,N}$ of the second plane, and the $(K-1)$-th packet to the $K$-th plane. It is now evident that at the end of the arbitration cycle buffer $B_{1,1}$ of the first plane is full, whereas buffers $B_{1,1}$ of the remaining planes are no longer full (one packet place is empty). This is the first instance of asymmetry between the planes. The above sequence of arrival patterns is now successively repeated in the next $N-1$ arbitration cycles at input adaptors 2 through $N$, such that buffers $B_{i,1}$ $(i=1,\ldots,N)$ buffers of the first plane are full, in contrast to buffers $B_{i,1}$ $(i=1,\ldots,N)$ of the remaining planes, which are no longer full (one packet place is empty). Note also that at the end of this $N$-cycle interval, each of the round-robin input load-balancer has dispatched packets to all $K$ planes, and therefore points to the first plane, just as it did at the beginning of this interval. The same holds in the case of the cyclic load-balancing mechanism. Repeating the scenario described above in the subsequent $N$-cycle intervals will therefore result in the buffers $B_{i,1}$ $(i=1,\ldots,N)$ of the first plane being always full, in contrast to buffers $B_{i,1}$ $(i=1,\ldots,N)$ of the remaining planes, which are continuously depleted until they become empty.

## APPENDIX B
## WORST-CASE RESEQUENCE-QUEUE SIZE

We present here a scenario that builds upon the one presented in Appendix A and results in a worst-case resequence-queue size in the case of $G=|\mathbf{q}|=1$ of Theorem 2. This condition implies that all resequence packets belong to a single flow. This flow is assumed to correspond to the pair between the first ingress-input port and the first egress-output port, with its rate being changed again to the maximum of $s$ (black) packets per packet cycle. The first, second, third, ..., $K$-th packet of this flow are assumed to join buffers $B_{1,1}$ of the first, second, third, ..., $K$-th plane, respectively. In the following arbitration cycle, the second, third, ..., $K$-th packet of the flow (along with a packet from buffer $B_{2,1}$ of the first plane) will be transferred to the egress output buffer and join the resequence queue. Note also that the ($K$+1)-th packet of the flow will be dispatched to the second plane because buffer $B_{1,1}$ of the first plane is full. In this way, at each arbitration cycle, $K-1$ packets of the flow will be transferred from planes 2 through $K$ to the egress output buffer. All these packets join the resequence queue as they have to wait for the first packet of the flow stored in buffer $B_{1,1}$ of the first plane. This packet will be dispatched to the egress output buffer after $NB_r$ arbitration cycles, corresponding to the number of packets in the first plane upon its arrival. Consequently, the resequence-queue size will grow to $NB_r(K-1)$, which, according to (22), is the worst case.

## APPENDIX C
## PROPERTIES RELATED TO THE BUFFERED-CROSSPOINT ARCHITECTURE

PROOF OF LEMMA 2.

First, we show that $\mathbf{q}^*(G,q)$ results in the largest resequencing degree. This follows immediately from (12) and the fact that

$$\min_{\mathbf{q}\ \text{s.t}\ |\mathbf{q}|=q}\left\{\max_{1\le k\le K}\{q_k\}\right\}=\left\lceil\frac{G}{q}\right\rceil\ .$$

To show that $\mathbf{q}^*(G,q)$ is also maximal, it suffices to show that there is a maximal vector with $|q_i-q_j|\le 1,\ \forall\ i,j$ such that $q_i,q_j>0$. Suppose there is a maximal vector $\mathbf{q}$ with $q_i\ge q_j+2\ge 3$ for some pair $(i,j)$. We will show that the vector $\mathbf{q}'=(q'_1,\ldots,q'_K)=(q_1,\ldots,q_i-1,\ldots,q_j+1,\ldots,q_K)$ is also maximal, with $|\mathbf{q}'|=|\mathbf{q}|=q$ and $q'_i-q'_j=(q_i-q_j)-2$. Clearly, repeatedly applying this procedure results in a maximal vector with $q_i\le q_j+1$, or $|q_i-q_j|\le 1$. Let $q_{\max}=\max_{1\le k\le K}\{q_k\}$ and $q'_{\max}=\max_{1\le k\le K}\{q'_k\}$. The following cases are considered:

Case 1) $q'_{\max}=q_{\max}$. Then, $\min(N+1-q'_{\max},G-q'_i)=\min(N+1-q_{\max},G-q_i+1)=\min(N+$

$1-q_{\max}, G-q_i) + \mathbf{1}_{N+1-q_{\max}>G-q_i}$ and $\min(N+1-q'_{\max}, G-q'_j) = \min(N+1-q_{\max}, G-q_j-1) = \min(N+1-q_{\max}, G-q_j) - \mathbf{1}_{N+1-q_{\max}\geq G-q_j}$. From the above and (15) it follows that $f_L(G,\mathbf{q}') = f_L(G,\mathbf{q}) + \mathbf{1}_{N+1-q_{\max}>G-q_i} - \mathbf{1}_{N+1-q_{\max}\geq G-q_j}$. Owing to the assumption $q_i > q_j$, the event $N+1-q_{\max} \geq G-q_j$ implies $N+1-q_{\max} > G-q_i$, and consequently $\mathbf{1}_{N+1-q_{\max}>G-q_i} - \mathbf{1}_{N+1-q_{\max}\geq G-q_j} \geq 0$. From the above it follows that $f_L(G,\mathbf{q}') \geq f_L(G,\mathbf{q}) = f_L(G,q)$ and, by virtue of definition (20), $f_L(G,\mathbf{q}') = f_L(G,q)$, which implies that the vector $\mathbf{q}'$ is also maximal.

Case 2) $q'_{\max} = q_{\max} - 1$. Then, $\min(N+1-q'_{\max}, G-q'_i) = \min(N+2-q_{\max}, G-q_i+1) = \min(N+1-q_{\max}, G-q_i) + 1$, and $\min(N+1-q'_{\max}, G-q'_j) = \min(N+2-q_{\max}, G-q_j-1) = \min(N+1-q_{\max}, G-q_j) + \mathbf{1}_{N+2-q_{\max}<G-q_j} - \mathbf{1}_{N+2-q_{\max}>G-q_j}$, and $\min(N+1-q'_{\max}, G-q'_m) = \min(N+2-q_{\max}, G-q_m) \geq \min(N+1-q_{\max}, G-q_m)$ for $m \neq i, j$. From the above and (15) it follows that $f_L(G,\mathbf{q}') \geq f_L(G,\mathbf{q}) + 1 + \mathbf{1}_{N+2-q_{\max}<G-q_j} - \mathbf{1}_{N+2-q_{\max}>G-q_j} + (K-q)$. From this inequality and given that $q \leq K$, it follows that $f_L(G,\mathbf{q}') \geq f_L(G,\mathbf{q}) = f_L(G,q)$ and, by virtue of definition (20), $f_L(G,\mathbf{q}') = f_L(G,q)$, which implies that the vector $\mathbf{q}'$ is also maximal. ■

PROOF OF LEMMA 3.

Substituting (21) into (19) yields

$$M_L(N,K) = \max_{G,\mathbf{q}} f_L(G,\mathbf{q}) = \max_{\substack{1\leq G<N \\ 1\leq q\leq \min(K,G)}} \left\{ (K-1)N - (q-1)(N-G) - (K-q)\left(\left\lceil \frac{G}{q} \right\rceil - 1\right)\right\} .$$

We proceed by observing that $(q-1)(N-G) \geq 0$, with the equality holding iff $q=1$ or $G=N$. Furthermore, it holds that $(K-q)\left(\left\lceil \frac{G}{q} \right\rceil - 1\right) \geq 0$, with the equality holding iff $q=K$ or $q=G$, which is equivalent to $q = \min(K,G)$, considering that $q \leq \min(K,G)$. Consequently, $(q-1)(N-G) + (K-q)\left(\left\lceil \frac{G}{q} \right\rceil - 1\right) \geq 0$, with the equality holding iff $q=G=1$, or $G=N$ and $q=\min(K,N)$. From the above it now follows that $\max_{G,\mathbf{q}} f_L(G,\mathbf{q}) = (K-1)N$. ■

PROOF OF LEMMA 4.

Conditioning first on $G$ and then on $|\mathbf{q}|$, (18) yields

$$M_B(N,K) \;=\; \max_{G,\mathbf{q}} f_B(G,\mathbf{q}) = \max_{1\leq G\leq N} \left\{ \max_{1\leq q\leq \min(K,G)} \left\{ \max_{\mathbf{q}\text{ s.t }|\mathbf{q}|=q} f_B(G,\mathbf{q}) \right\} \right\} ,$$

and by virtue of (14),

$$\begin{aligned} M_B(N,K) = \max_{\substack{1\leq G\leq N \\ 1\leq q\leq \min(K,G)}} \{(K-q)N + (q-1)G\} \;= \\ \max\left( \max_{\substack{1\leq G<N \\ 1\leq q\leq \min(K,G)}} \{KN - G - (N-G)q\} \;, \max_{1\leq q\leq \min(K,N)} \{KN-N\} \right) \;= \\ \max\left( \max_{1\leq G<N} \{KN-N\} \,, KN-N \right) , \end{aligned}$$

where the two terms of the maximum are both equal to $(K-1)N$ and are obtained when $q=1$ and $G=N$, respectively. ■

# REFERENCES


[1] T. Aramaki, H. Suzuki, S.-i. Hayano, and T. Takeuchi, “Parallel ‘ATOM’ switch architecture for high-speed ATM networks,” *Proc. IEEE ICC ’92*, Chicago, IL, vol. 1, pp. 250-254, June 1992.

[2] S. Iyer, A. Awadallah, and N. McKeown, “Analysis of a packet switch with memories running slower than the line rate”, in *Proc. IEEE INFOCOM ’00*, Tel Aviv, Israel, vol. 2, pp. 529-537, March 2000.

[3] S. Iyer and N. McKeown, “Making parallel packet switches practical,” in *Proc. IEEE INFOCOM ’01*, Anchorage, AK, vol.3, pp. 1680-1687, April 2001.

[4] W. Wang, L. Dong, and W. Wolf, “Distributed switch architecture with dynamic load-balancing and parallel input-queued crossbars for terabit switch fabrics,” in *Proc. IEEE INFOCOM ’02*, New York, NY, vol. 1, pp. 352-361, June 2002.

[5] J.S. Turner and N. Yamanaka, “Architectural Choices in Large Scale ATM Switches,” Technical Report WUCS-97-21, Washington University, Computer Science Department, 5/1/97, or *IEICE Trans. on Commun.*, vol. E81-B, no. 2, pp. 138-151, 1998.

[6] F. M. Chiussi, J. G. Kneuer, and V. P. Kumar, “Low-cost scalable switching solutions for broadband networking: the ATLANTA architecture and chipset,” *IEEE Commun. Mag.*, vol.35, no. 3, pp. 44-53, March 1997.

[7] M. A. Henrion, G. J. Eilenberger, G. H. Petit, and P. H. Parmentier, “Multipath self-routing switch,” *IEEE Commun. Mag.*, vol. 31, no. 4, pp. 46-52, April 1993.

[8] T. Chaney, J. A. Fingerhut, M. Flucke, and J. S. Turner, “Design of a gigabit ATM switch,” in *Proc. IEEE INFOCOM ’97*, Kobe, Japan, vol. 1, pp. 2-11, April 1997.

[9] F. Abel, C. Minkenberg, R. P. Luijten, M. Gusat, and I. Iliadis, “A four-terabit packet switch supporting long round-trip times,” *IEEE Micro*, vol. 23, no. 1, pp. 10-24, Jan./Feb. 2003.

[10] I. Iliadis and Y.-C. Lien, “Resequencing in distributed systems with multiple classes,” *Proc. IEEE INFOCOM ’88*, New Orleans, LA, pp. 881-888, March 1988.

[11] I. Iliadis and L. Y.-C. Lien, “Resequencing delay for a queueing system with two heterogeneous servers under a threshold-type scheduling,” *IEEE Trans. Commun.*, vol. 36, no. 6, pp. 692-702, June 1988.

[12] I. Iliadis and L. Y.-C. Lien, “Resequencing control for a queueing system with two heterogeneous servers,” *IEEE Trans. Commun.*, vol. 41, no. 6, pp. 951-961, June 1993.

[13] N. Gogate and S. S. Panwar, “On a resequencing model for high speed networks,” *Proc. IEEE INFOCOM ’94*, Toronto, Canada, vol. 1, pp. 40-47, June 1994.

[14] J. S. Turner, “An optimal nonblocking multicast virtual circuit switch,” *Proc. IEEE INFOCOM ’94*, Toronto, Canada, vol. 1, pp. 298-305, June 1994.

[15] H. J. Chao and J. S. Park, “Architecture designs of a large-capacity Abacus ATM switch,” *Proc. IEEE CLOBECOM ’98*, Sydney, Australia, vol. 1, pp. 369-374, Nov. 1998.