

# LQD is 1.5-competitive for 3-port Shared-Memory Switches

Nicolaos Matsakis

Department of Computer Science and Centre for Discrete Mathematics  
and its Applications (DIMAP), University of Warwick  
`nmatsakis@dcs.warwick.ac.uk`

**Abstract.** We show that the Longest Queue Drop algorithm is 1.5-competitive for shared-memory switches with three output ports. This improves upon the previous best upper bound of  $(2 - \frac{1}{M} \cdot \lfloor M/3 \rfloor - \frac{3}{M})$  for the competitive ratio of the Longest Queue Drop algorithm, for 3-port shared-memory switches, where  $M$  denotes the shared-memory size.

## 1 Introduction

The area of memory management is strongly related to that of online computation, due to the unpredictability of future requests that naturally arises in related problems. Hence, the fact that a great deal of research has been dedicated to online algorithms improving the throughput of devices that incorporate buffers, comes as no surprise.

A shared-memory switch is a buffer equipped with a number of input and output ports; in our case with three output ports. Assuming that time is slotted, packets arrive at any time step to any of the input ports and must be immediately accepted by the buffer or be immediately rejected. Each arriving packet is labelled with a single output port of the switch. Any rejected packet is inevitably lost and any accepted packet is stored in the buffer. At any time step, one packet is transmitted by each output port of the switch, to which at least one packet stored in the buffer is destined.

Since there may exist at most three different types of packets stored in the buffer at any time, we may assume that the packets stored in the buffer which are destined to the same output port, are arranged in their own queue. In other words, we may have at most three queues at any time step.

An algorithm is called *preemptive* if an eviction from the buffer of an already accepted packet is possible. Otherwise, the algorithm is called *non-preemptive*.

In this paper, we discuss the Longest Queue Drop algorithm (LQD), for the case of shared-memory switches equipped with three output ports. According to LQD, which is a preemptive online algorithm, any packet is accepted by the buffer if there exists free buffer space at the time step when this packet arrives. If the buffer is full at the time step when the packet arrives, a packet is preempted from the longest queue in the buffer at this point in time, releasing buffer space for the arriving packet which is accepted. After all packet acceptances and preemptions

take place, at each time step, one packet of each queue is forwarded from the buffer to its designated output port to be transmitted.

The LQD algorithm was introduced in [7] and there exists a considerable amount of literature regarding it (for example [2, 3]).

### 1.1 Related Work

Hahne et al. [5] show a lower bound of  $4/3$  for the competitive ratio of any deterministic online algorithm, for the problem of maximizing the number of transmitted packets from a shared-memory switch equipped with any number of output ports. Aiello et al. [1] show that LQD is 2-competitive and at least  $\sqrt{2}$ -competitive, when the number of output ports  $N \geq 2$  of the switch is assumed to be arbitrary.

Kobayashi et al. [6], show a tight upper bound of  $\frac{4M-4}{3M-2} < \frac{4}{3}$  for the LQD competitive ratio for 2-port shared-memory switches, where  $M \in \mathbb{Z}^+$  denotes the buffer size. In [6], an upper bound of  $(2 - \frac{1}{M} \cdot \lfloor M/N \rfloor - \frac{N}{M})$  is, also, shown for the LQD competitive ratio. This bound tends to  $5/3$  as  $M \rightarrow \infty$  when  $N = 3$ , being larger than 1.5 for  $M \geq 19$  when  $N = 3$ .

We show that LQD is 1.5-competitive for 3-port shared-memory switches. We assume that  $M \geq N = 3$  throughout the following analysis. We note, however, that several of our lemmas and corollaries hold for switches equipped with any number of output ports  $N \leq M$ .

## 2 Preliminaries

### 2.1 An Optimal Offline Algorithm

Both LQD and OPT are assumed to use their own buffers, each of size  $M$ . Time proceeds in discrete time steps and  $T \geq 1$  will denote the latest time step when a packet exists in any buffer. Analysis will proceed by comparing the two buffer contents. As  $[n]$  we denote the set  $\{1, \dots, n\}$  for a positive integer  $n$ .

Let ALG denote any algorithm for the problem of maximizing the number of transmitted packets from a shared-memory switch equipped with any number of output ports  $N \leq M$ . A packet accepted by the ALG buffer will be called ALG packet, in short.

All definitions that follow until the end of the current paragraph, refer to the buffer content of ALG, after all packet acceptances, rejections and preemptions have taken place, at each time step: We denote as  $p_{i,\text{ALG}}^t$  the length of a queue  $i \in [N]$  in the ALG buffer at a time step  $t \in [T]$ . A queue  $i \in [N]$  is said to be *active* in the ALG buffer at a time step  $t \in [T]$  if  $p_{i,\text{ALG}}^t > 0$ , else  $i$  is *inactive* in the ALG buffer at  $t$ . As  $\ell_{\text{ALG}}^t(q) \in [M]$ , we denote the queue position of a packet  $q$  in the ALG buffer, at a time step  $t \in [T]$ .

Let  $t_1 \in [T]$  and  $t_2 \in [T]$  be two time steps such that  $t_1 \leq t_2$ . As *period* we call the set of consecutive time steps from  $t_1$ , inclusive, to  $t_2$ , inclusive and we denote the period as  $[t_1, t_2]$ .

If a packet destined to a queue  $i \in [N]$ , is rejected or preempted from the LQD buffer at a time step  $t \in [T]$ , we say that  $i$  *overflows* in the LQD buffer at  $t$ . At any time step when the LQD buffer overflows, this buffer is full.

**Observation 1** *The lengths of two queues in the LQD buffer, that overflowed in the LQD buffer at the same time step  $t \in [T]$ , may differ by at most 1 at  $t$ .*

We denote as  $h_\sigma(\text{ALG}) \geq 0$  the number of packets that ALG transmits in the period  $[1, T]$ , where  $\sigma$  denotes the incoming packet sequence.

We define as  $T'$ -balanced any algorithm ALG' such that if  $p_{i,\text{LQD}}^t \geq 1$  then  $p_{i,\text{ALG}'}^t \geq 1$ , for each queue  $i \in [N]$  and at each time step  $t \in [T']$ , where  $T' \leq T$ .

Since  $M \geq N$ , no (offline or online) algorithm can keep active at time step  $t = 1$  strictly more queues than the number of active queues of LQD at  $t = 1$ , for any incoming packet sequence  $\sigma$ . Hence, no algorithm can transmit strictly more packets at  $t = 1$  than the number of packets that LQD transmits at  $t = 1$ , for the same  $\sigma$ . Taking into consideration the fact that LQD is, trivially, an 1-balanced algorithm, we have the next observation:

**Observation 2** *Let  $\text{ALG}_{0b}$  denote any algorithm that keeps inactive at  $t = 1$  an active queue in the LQD buffer at  $t = 1$ , for an incoming sequence  $\sigma$ . Then, there exists an 1-balanced algorithm  $\text{ALG}_{1b}$  such that  $h_\sigma(\text{ALG}_{1b}) \geq h_\sigma(\text{ALG}_{0b})$ .*

**Lemma 1.** *There exists an optimal offline algorithm which is  $T$ -balanced and which never preempts any packet.*

*Proof.* Let  $\text{ALG}_1$  be an offline algorithm which is  $t'$ -balanced but not  $(t' + 1)$ -balanced, where  $t' \in [T - 1]$ . We assume that  $\text{ALG}_1$  never preempts any packet, since otherwise we can modify  $\text{ALG}_1$  so that it never preempts any packet without decreasing  $h_\sigma(\text{ALG}_1)$ , where  $\sigma$  is the incoming packet sequence.

It is  $p_{j,\text{ALG}_1}^{t'+1} = 0$  and  $p_{j,\text{LQD}}^{t'+1} > 0$  for a queue  $j \in [N]$ , because  $\text{ALG}_1$  is  $t'$ -balanced but not  $(t' + 1)$ -balanced. But then, there has to exist a queue  $k \neq j$  and a time step  $t'' \leq t' + 1$  so that:

- It holds  $p_{k,\text{LQD}}^{t''} < p_{k,\text{ALG}_1}^{t''}$  or
- The sum of the empty  $\text{ALG}_1$  buffer space at  $t''$  and  $p_{k,\text{ALG}_1}^{t''}$  is strictly greater than  $p_{k,\text{LQD}}^{t''}$ .

In case two or more different pairs of queues and time steps  $\{k, t''\}$  exist, as defined above, then we choose the latest time step  $t''$  in the period  $[1, t' + 1]$  and we break ties arbitrarily for choosing the queue  $k$ , at this chosen  $t''$ .

We, now, modify  $\text{ALG}_1$  so that one more packet for  $j$  is accepted in the period  $[1, t' + 1]$ . For this, one packet less for  $k$  (if  $p_{k,\text{LQD}}^{t''} < p_{k,\text{ALG}_1}^{t''}$ ) may need to be accepted by the modified algorithm, at the latest overflow time step in the LQD buffer of  $k$  in the period  $[1, t'']$ . This gives us a new  $\hat{t}$ -balanced algorithm  $\text{ALG}_2$  (for some  $\hat{t} \geq t'$ ) for which it holds  $h_\sigma(\text{ALG}_2) \geq h_\sigma(\text{ALG}_1)$  because  $j$  becomes active in the  $\text{ALG}_2$  buffer at  $t' + 1$ .

Working as above, we obtain a sequence of algorithms  $\text{ALG}_1, \dots, \text{ALG}_v$  (for a  $v \in \mathbb{Z}^+$ ), where  $\text{ALG}_1, \dots, \text{ALG}_{v-1}$  are not  $T$ -balanced,  $\text{ALG}_v$  is  $T$ -balanced and it holds  $h_\sigma(\text{ALG}_l) \geq h_\sigma(\text{ALG}_{l-1})$  for each  $l \in [v]$ . By their design, none of the algorithms  $\text{ALG}_1, \dots, \text{ALG}_v$  preempts any packet from any queue. It follows that  $h_\sigma(\text{ALG}_v) \geq h_\sigma(\text{ALG}_1)$  which by Observation 2, completes the proof.  $\square$

## 2.2 Classifying the Queues and Packets

A queue  $i \in [N]$  is called as *established* at  $t \in [T]$ , if  $p_{i,\text{LQD}}^t = 0$  and  $p_{i,\text{OPT}}^t > 0$ . If  $p_{i,\text{LQD}}^t > p_{i,\text{OPT}}^t$  then  $i$  is a *free* queue at  $t$ , else if  $1 \leq p_{i,\text{LQD}}^t \leq p_{i,\text{OPT}}^t$  then  $i$  is a *dominating* queue at  $t$ . We denote as  $D^t, E^t, F^t$ , the sets of dominating, established and free queues, respectively, at  $t \in [T]$ .

An OPT packet  $q$  of a queue  $i \in D^t$  at a time step  $t \in [T]$ , for which it holds  $\ell_{\text{OPT}}^t(q) > p_{i,\text{LQD}}^t$ , is called *extra* packet at  $t$ . An OPT packet of an established queue at a  $t \in [T]$  is, also, called *extra* packet at  $t$ . An LQD packet of a queue  $i \in F^t$  at a time step  $t \in [T]$ , is called *free* packet at  $t$ . An LQD packet of a dominating queue at a  $t \in [T]$  is called *common* packet, at  $t$ .

We denote as  $e(t) \geq 0$  the number of extra packets in the OPT buffer at  $t \in [T]$  and as  $f(t) \geq 0$  the number of free packets in the LQD buffer, at  $t$ .

By Lemma 1, a queue cannot be inactive in the OPT buffer and active in the LQD buffer at the same time step. This gives us the next corollary:

**Corollary 1.** *Any queue at any time step  $t \in [T]$  can either belong to set  $D^t$  or to set  $F^t$  or to set  $E^t$  or be inactive in both buffers at  $t$ .*

By Corollary 1, the ratio of transmitted packets between OPT and LQD, for an incoming packet sequence  $\sigma$ , is defined as:

$$r_\sigma = \frac{h_\sigma(\text{OPT})}{h_\sigma(\text{LQD})} = 1 + \frac{\sum_{t=1}^{t=T} |E^t|}{h_\sigma(\text{LQD})} \geq 1 \quad (1)$$

The numerator of the fraction in the right hand side of (1) equals the number of transmitted extra packets in  $[1, T]$ , i.e., the number of OPT packets which are extra packets when transmitted by the OPT buffer. The denominator equals the number of LQD packets transmitted in  $[1, T]$ .

**Observation 3** *Since  $M \geq N$ , if a queue  $i \in [N]$  is inactive in the LQD buffer at a time step  $t \in [T - 1]$  and at least one packet arrives destined to  $i$  at  $t + 1$ , then  $i$  will be active in the LQD buffer at  $t + 1$ .*

**Lemma 2.** *If  $i \in E^{t+1}$  for any queue  $i \in [N]$  at any time step  $t \in [T - 1]$ , then it is  $i \in D^t \cup E^t$ .*

*Proof.* First, we show that if  $i \in E^{t+1}$  then  $p_{i,\text{OPT}}^t > 0$ : Assume that  $p_{i,\text{OPT}}^t = 0$  and, therefore, that it holds  $p_{i,\text{LQD}}^t = 0$  due to Lemma 1. If no packet arrives to  $i$  at  $t + 1$ , then  $i \notin E^{t+1}$  by the definition of an established queue. Otherwise, if

at least one packet arrives destined to  $i$  at  $t + 1$  then  $i$  has to be active in the LQD buffer at  $t + 1$ , by Observation 3; therefore it is  $i \notin E^{t+1}$ .

We, now, show that if  $i \in E^{t+1}$  then  $i \notin F^t$ : Assume that  $i \in F^t$ . The length of a free queue in the LQD buffer at any time step is at least 2, by the definition of a free queue and Lemma 1; hence since  $i \in F^t$  then  $i$  has to be active in the LQD buffer at  $t + 1$  and, therefore, it has to be  $i \notin E^{t+1}$ .

By Corollary 1 and the arguments of the first two paragraphs of this proof, the lemma follows.  $\square$

**Lemma 3.** *If  $i \in F^t$  for any queue  $i \in [N]$  at any time step  $t \in [T - 1]$ , then it holds  $i \in F^{t+1} \cup D^{t+1}$ .*

*Proof.* The length of a free queue in the LQD buffer is at least 2, by the definition of a free queue and Lemma 1. It follows that  $p_{i,\text{LQD}}^t \geq 2$ . But then,  $i$  has to be active in the LQD buffer at  $t + 1$ , which by Corollary 1, completes the proof.  $\square$

**Lemma 4.** *It holds  $f(t) \geq e(t) + \sum_{i \in F^t} p_{i,\text{OPT}}^t$  at any time step  $t \in [T]$ .*

*Proof.* (see Appendix).  $\square$

Assume, now, that  $i \in E^t$  at a time step  $t \in [T]$  for a queue  $i \in [N]$ . We denote as  $t_{1,i}^{\lambda_i} \in [T]$  the latest time step in the period  $[1, t]$  when  $i$  overflowed in the LQD buffer as a dominating queue and we know that such a time step exists by Lemma 2 and Observation 3. Also, we denote as  $t_{2,i}^{\lambda_i} \leq t$  the first time step after  $t_{1,i}^{\lambda_i}$  when  $i$  becomes an established queue, where the superscript  $\lambda_i \in \mathbb{Z}^+$  will be defined shortly. This gives us a strict total order:  $t_{1,i}^1 < t_{2,i}^1 < t_{1,i}^2 < t_{2,i}^2 < \dots$ , where the superscript denotes each pair of two consecutive time steps in the total order, starting from pair  $\{t_{1,i}^1, t_{2,i}^1\}$ , then pair  $\{t_{1,i}^2, t_{2,i}^2\}$  etc. Finally, let us say that we have  $A_i \in \mathbb{Z}^+$  pairs of time steps in this total order, i.e., it holds  $t_{1,i}^1 < t_{2,i}^1 < \dots < t_{1,i}^{A_i} < t_{2,i}^{A_i}$  and, hence,  $\lambda_i \in [A_i]$ .

For example, assume that a queue  $i \in [N]$  becomes established for the first time in  $[1, T]$  at time step 10. Then, it is  $t_{2,i}^1 = 10$  (note that time step 10 may be equal to  $t_{1,j}^{\lambda_j}$  or  $t_{2,j}^{\lambda_j}$  for another queue  $j \neq i$  and for a  $\lambda_j \in [A_j]$ ). By Lemma 2 and Observation 3, it follows that  $i$  has overflowed in the LQD buffer as a dominating queue, for at least one time step before time step 10 and assume that the latest time step in the period  $[1, 10]$  of such an overflow is time step 7, i.e.,  $t_{1,i}^1 = 7$ . Assume, also, that the first time step after  $t_{2,i}^1 = 10$  when  $i$  overflows as a dominating queue in the LQD buffer is time step 30 and the first time step after time step 30 when  $i$  becomes an established queue is time step 40. It follows that  $t_{2,i}^2 = 40$  and that  $t_{1,i}^2$  is the latest time step in  $[30, 39]$  when  $i$  overflows as a dominating queue in the LQD buffer. This concludes our example.

**Lemma 5.** *The queue  $i \in [N]$  is a dominating queue at each time step of the period  $[t_{1,i}^{\lambda_i}, t_{2,i}^{\lambda_i} - 1]$ , for any  $\lambda_i \in [A_i]$ .*

*Proof.* (see Appendix).  $\square$

**Lemma 6.** *The number of extra packets of any queue  $i \in [N]$  cannot strictly increase between any two consecutive time steps of the period  $[t_{1,i}^{\lambda_i}, t_{2,i}^{\lambda_i}]$ , for any  $\lambda_i \in [A_i]$ , i.e., it holds  $p_{i,\text{OPT}}^t - p_{i,\text{LQD}}^t \geq p_{i,\text{OPT}}^{t+1} - p_{i,\text{LQD}}^{t+1}$ , where  $t \in [t_{1,i}^{\lambda_i}, t_{2,i}^{\lambda_i} - 1]$ .*

*Proof.* (see Appendix).  $\square$

**Lemma 7.** *The number of extra packets of any queue  $i \in [N]$  at any time step  $t_{2,i}^{\lambda_i}$  (for any  $\lambda_i \in [A_i - 1]$ ) upper-bounds the number of transmitted extra packets of  $i$  in the period  $[t_{2,i}^{\lambda_i}, t_{1,i}^{\lambda_i+1} - 1]$ . Also, the number of extra packets of  $i$  at  $t_{2,i}^{\lambda_i}$  upper-bounds the number of transmitted extra packets of  $i$  in the period  $[t_{2,i}^{\lambda_i}, T]$ .*

*Proof.* (see Appendix).  $\square$

### 2.3 Introducing the Packet Connections

We shall work similarly to Aiello et al. [1], assigning connections between extra packets and LQD packets.

**Definition 1.** *A connection between an extra packet  $e$  and an LQD packet  $e'$  which is assigned at a time step  $t \in [T]$  when both packets are in their respective buffers, will be called valid if  $\ell_{\text{LQD}}^t(e') \leq \ell_{\text{OPT}}^t(e)$ .*

All connections will be valid, hence a valid connection will be usually called from now as, simply, connection. We say that an extra packet  $u$  is *connected* with an LQD packet  $u'$  at a time step  $t \in [T]$ , if:

- These packets were assigned the connection between them at a  $t' \in [1, t]$  and
- These two packets remain connected together at each time step of  $[t', t]$  and
- These two packets are in their respective buffers at  $t$ .

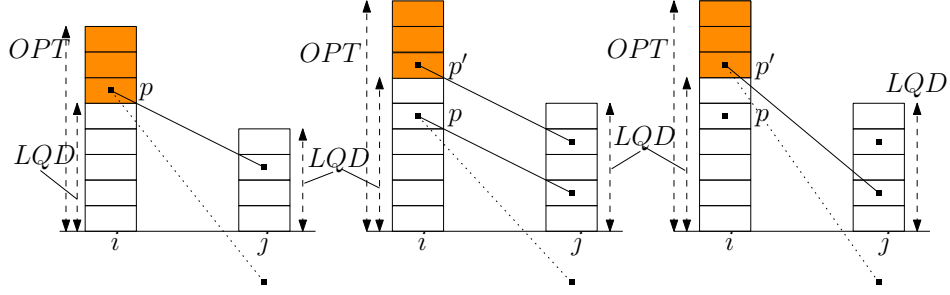
**Fact 1** *From the time step  $t_c \in [T]$  when an extra packet  $x$  is assigned a connection with an LQD packet  $y$ , these two packets are assumed to stay connected at each later time step, until any of the following takes place, in any order:*

1. *The LQD packet  $y$  is transmitted, or*
2. *The LQD packet  $y$  is preempted, or*
3. *The extra packet  $x$  becomes a non-extra packet at a time step  $t'_c > t_c$ , i.e.,  $x$  is still in the OPT buffer at  $t'_c$  but it is not an extra packet at  $t'_c$ .*

In the first case of Fact 1, that is if the LQD packet  $y$  is transmitted at a time step  $t_s > t_c$ , we delete the connection between  $x$  and  $y$  at  $t_s$ , i.e.,  $x$  and  $y$  are no more connected in the period  $[t_s, T]$  (note that by Definition 1, the LQD packet  $y$  will be transmitted not later than the extra packet  $x$  is transmitted). Finally, we shall say that  $x$  is *associated* with the transmitted LQD packet  $y$ , in  $[t_s, T]$ .

In the second case of Fact 1, that is in the case the LQD packet  $y$  is preempted at a time step  $t_p > t_c$ , we delete the connection between  $x$  and  $y$  at  $t_p$  and we assign at  $t_p$ , a new connection between  $x$  and the newly accepted LQD packet. Since the newly accepted LQD packet cannot be located at a greater queue position than that of the preempted LQD packet  $y$  at the preemption time step  $t_p$ , this new connection is valid.

For the third case of Fact 1, we first need Definitions 2 and 3:



**Fig. 1.** In the figure we have two queues ( $i$  and  $j \neq i$ ) and in orange color we identify the extra packets of  $i$ . *Left:* At time step  $t'_c - 1$  the extra packet  $p$  is connected with an LQD packet (bold line segment) and associated with a transmitted LQD packet (dotted line segment). *Center:* At  $t'_c$  the OPT packet  $p$  becomes a non-extra packet because  $i$  accepts packets in the LQD buffer; therefore we assign (on the *right*, again at time step  $t'_c$ ) to an inferior extra packet  $p'$  of the same queue  $i$  the connection and association that  $p$  had, after deleting the single connection of  $p'$

**Definition 2.** As  $\text{con}(e, t) \geq 0$ , we define the number of LQD packets that an extra packet  $e$  (which is in the OPT buffer at a time step  $t \in [T]$ ) is connected with at  $t$ , plus the number of transmitted LQD packets associated with  $e$  at  $t$ .

**Definition 3.** Let  $p$  and  $p' \neq p$  be two extra packets belonging to the same queue  $i \in [N]$  in the OPT buffer, at a time step  $t \in [T]$ . We shall say that  $p'$  is inferior to  $p$  at  $t$ , if it is  $\ell_{OPT}^t(p') > \ell_{OPT}^t(p)$  and it is  $\text{con}(p', t) < \text{con}(p, t)$ .

Therefore, in the third case of Fact 1, that is if  $x$  is an extra packet in  $[t_c, t'_c - 1]$  but not an extra packet at  $t'_c$ , then:

- If there exists an inferior extra packet of  $x$  at  $t'_c$ , then we choose the inferior extra packet  $x'$  of  $x$  which is located at a smaller queue position than any inferior extra packet of  $x$  at  $t'_c$  and:
  - We delete all connections and associations of both  $x$  and  $x'$  at  $t'_c$  and
  - We assign to  $x'$  at  $t'_c$  all connections and associations that we deleted from  $x$  at  $t'_c$  and we set  $\text{con}(x', t'_c) = \text{con}(x, t'_c)$ .
- Else we delete all connections and associations of  $x$  at  $t'_c$ .

### 3 Analysis

**Definition 4.** A valid connection between an extra packet  $e$  of a queue  $i \in [N]$  and an LQD packet  $c$ , which is assigned at a time step  $t_{1,i}^{\lambda_i}$  (for any  $\lambda_i \in [A_i]$ ), is called strong connection, if  $c$  is a common packet of  $i$  at this  $t_{1,i}^{\lambda_i}$  time step.

By Observation 1 and Definition 1 we have Observation 4.

**Observation 4** *A connection that is assigned to an extra packet of a queue  $i \in [N]$  at any of its  $t_{1,i}^{\lambda_i}$  time steps ( $\lambda_i \in [A_i]$ ) is valid.*

**Lemma 8.** *Assume that an extra packet of a queue  $i \in [N]$  is assigned a strong connection with an LQD packet  $c$  at a  $t_{1,i}^{\lambda_i}$  time step (for a  $\lambda_i \in [A_i]$ ). Then,  $c$  will never be preempted and will never become a free packet until its transmission.*

*Proof.* (see Appendix).  $\square$

**Definition 5.** *Assume that  $i \in D^t$  at a time step  $t \in [T]$ . Then, the upmost  $h$  extra packets of  $i$  at  $t$  (where  $h \in \mathbb{Z}^+$ ) are the  $h$  extra packets of  $i$  that occupy the  $h$  greatest queue positions of  $i$  at  $t$ .*

**Definition 6.** *An extra packet  $s$  of a queue  $i \in [N]$  for which it holds  $\ell_{OPT}^t(s) > 2 \cdot p_{i,LQD}^t$  at a time step  $t \in [T]$ , is called upper extra packet of  $i$  at  $t$ . Any other extra packet of  $i$  at  $t$  is called lower extra packet of  $i$  at  $t$ .*

Let  $l_i^t \geq 0$  and  $u_i^t \geq 0$  denote the numbers of lower and upper extra packets, respectively, that a queue  $i \in [N]$  has at a time step  $t \in [T]$ .

**Definition 7.** *If  $2 \cdot p_{i,LQD}^t < p_{i,OPT}^t$  at a time step  $t \in [T]$  for a queue  $i \in [N]$ , then  $i$  is called primary dominating queue at  $t$  else if  $2 \cdot p_{i,LQD}^t \geq p_{i,OPT}^t$  then  $i$  is called secondary dominating queue at  $t$ .*

By Observation 1 and since  $M \geq N$ , we have Observation 5:

**Observation 5** *At any  $t \in [T]$  when a packet of a queue  $i \in [N]$  is preempted or rejected from the LQD buffer, it holds  $p_{i,LQD}^t \geq \lfloor M/N \rfloor$ .*

**Lemma 9.** *If a queue  $i \in [N]$  overflows in the LQD buffer at a time step  $t \in [T]$  and  $i$  is a primary dominating queue at  $t$ , then it holds  $|F^t| = 2$ .*

*Proof.* (see Appendix).  $\square$

**Lemma 10.** *If  $i \in F^t$  and a dominating queue overflows in the LQD buffer at  $t$ , then  $p_{i,LQD}^t \leq \lceil M/2 \rceil$ .*

*Proof.* Assume for contradiction that  $p_{i,LQD}^t \geq M/2 + 1$ . By Observation 1, the length of the dominating queue that overflows in the LQD buffer at  $t$ , is at least equal to the length of  $i$  in the LQD buffer at  $t$ , minus 1. Taking the sum of the packet numbers for these two queues at  $t$  in the LQD buffer, it follows that at least  $M + 1$  packets exist in the LQD buffer at  $t$ , which cannot happen.  $\square$

**Lemma 11.** *If  $|F^t| = 1$  then it is  $e(t) \leq \lceil M/2 \rceil$ , at any time step  $t \in [T]$ .*

*Proof.* (see Appendix).  $\square$

According to the connection assignment process that we describe in Section 3.1, if an extra packet  $r$  is connected with at least one LQD packet at a time step  $t \in [T]$ , then we may assign a new connection to  $r$  at  $t$ , only if each connection that  $r$  currently has, is strong. This gives us the next lemma.



**Lemma 12.** *Each extra packet of any queue  $i \in [N]$  may be connected with at most one LQD packet of a queue other than  $i$ , at any time step of  $[1, T]$ .*

*Proof.* The lemma follows by Lemma 8 and Fact 1.  $\square$

**Lemma 13.** *Any extra packet may be connected with at most one free packet, at any time step of  $[1, T]$ .*

*Proof.* The lemma follows by Lemma 12 and the definition of a free packet.  $\square$

### 3.1 Assigning the Packet Connections

We shall, now, assign connections between extra packets and LQD packets, proving that the next invariant holds at every time step  $t \in [1, T]$ :

**Invariant 1** *No LQD packet is connected with two or more extra packets at  $t$ .*

Hence, assume that a queue  $i \in [N]$  overflows in the LQD buffer at a  $t_{1,i}^{\lambda_i}$  time step (for a  $\lambda_i \in [A_i]$ ). For simplicity, let  $t_{1,i} = t_{1,i}^{\lambda_i}$  and  $t_{2,i} = t_{2,i}^{\lambda_i}$ . We distinguish between the cases that  $i$  is primary dominating or secondary dominating, at  $t_{1,i}$ :

#### Case 1: The queue $i$ is primary dominating at $t_{1,i}$

By Lemma 9, it is  $|F^{t_{1,i}}| = 2$  and, hence, the only dominating or established queue at  $t_{1,i}$  is  $i$ . It follows that all LQD packets are currently not connected at  $t_{1,i}$ , since it is only  $i$  that has extra packets in the OPT buffer at  $t_{1,i}$ .

We assign one connection to each lower extra packet of  $i$  at  $t_{1,i}$  with a different common packet of  $i$ . The number of common packets of  $i$  at  $t_{1,i}$  suffices so that each lower extra packet of  $i$  is assigned one connection, by Definition 6. We, also, assign one connection to every extra packet of  $i$  with a different free packet at  $t_{1,i}$ . The number of free packets at  $t_{1,i}$  suffices to assign one connection to each extra packet of  $i$  with a different free packet, due to Lemma 4.

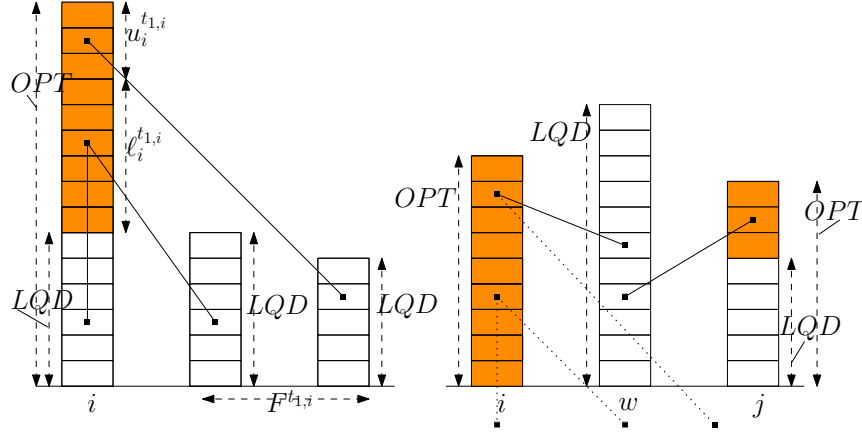
Wrapping up, we assigned two connections to each lower extra packet of  $i$  at  $t_{1,i}$  and one connection to each upper extra packet of  $i$  at  $t_{1,i}$ , so that Invariant 1 holds at  $t_{1,i}$ . Any connection assigned is valid, by Observation 4.

**Lemma 14.** *No packet in the LQD buffer at time step  $t_{2,i} > t_{1,i}$  may be connected with a connection which was assigned at time step  $t_{1,i}$ .*

*Proof.* (see Appendix).  $\square$

If  $l_i^{t_{1,i}} \geq p_{i,\text{OPT}}^{t_{2,i}}$  then the connection assignment for Case 1 is complete. Otherwise, if  $l_i^{t_{1,i}} < p_{i,\text{OPT}}^{t_{2,i}}$ , we distinguish between sub-cases (A1), (A2) and (A3):

**(A1)** If  $|F^{t_{2,i}}| = 2$ , then it is only queue  $i$  that has extra packet(s) at  $t_{2,i}$ . By this and Lemma 14, no LQD packet is currently connected at  $t_{2,i}$ . Hence, we



**Fig. 2.** An example of the connection assignment of Case 1. Left: The queue  $i$  overflows at  $t_{1,i}$ . Each lower extra packet of  $i$  is assigned two connections and each upper extra packet of  $i$ , one connection. On the right we have sub-case (A2): At  $t_{2,i}$  all connected LQD packets from  $t_{1,i}$  have left the buffer (dotted line segments) and we assign one connection to each of the  $(p_{i,\text{OPT}}^{t_{2,i}} - l_i^{t_{1,i}})$  upmost extra packets of  $i$  at  $t_{2,i}$ . Even if another dominating queue  $j$  exists at  $t_{2,i}$  and each of  $j$ 's extra packets is connected with one free packet, Invariant 1 is not violated at  $t_{2,i}$ .

assign a connection at  $t_{2,i}$  to each of the upmost  $p_{i,\text{OPT}}^{t_{2,i}} - l_i^{t_{1,i}}$  extra packets of  $i$  at  $t_{2,i}$ , with a different free packet. By Lemma 4, the number of free packets suffices for these connections to be assigned at  $t_{2,i}$  and Invariant 1 holds at  $t_{2,i}$ .

(A2) If  $|F^{t_{2,i}}| = 1$ , then it is  $e(t_{2,i}) \leq \lceil M/2 \rceil$ , by Lemma 11. We denote the single free queue we have at  $t_{2,i}$  as  $w \neq i$  and the third queue that we may have at  $t_{2,i}$  (which can be dominating or established) as  $j \neq \{i, w\}$  (see Figure 2). By Observation 5 it is  $p_{i,\text{LQD}}^{t_{1,i}} \geq \lfloor M/3 \rfloor$ , which by Definition 6 gives us  $l_i^{t_{1,i}} \geq \lfloor M/3 \rfloor$ . By the last inequality and since  $e(t_{2,i}) \leq \lceil M/2 \rceil$ , it is:

$$2 \cdot l_i^{t_{1,i}} \geq e(t_{2,i}) - 1 \quad (2)$$

But the number of extra packets in the OPT buffer at  $t_{2,i}$  is equal to the number of extra packets of  $i$  at  $t_{2,i}$  plus the number of extra packets of  $j$  at  $t_{2,i}$ :

$$e(t_{2,i}) = (p_{j,\text{OPT}}^{t_{2,i}} - p_{j,\text{LQD}}^{t_{2,i}}) + p_{i,\text{OPT}}^{t_{2,i}} \quad (3)$$

By (2), (3) and since we have assumed that  $l_i^{t_{1,i}} < p_{i,\text{OPT}}^{t_{2,i}}$ , it holds:

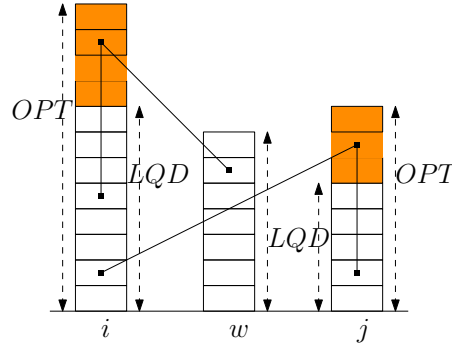
$$l_i^{t_{1,i}} \geq p_{j,\text{OPT}}^{t_{2,i}} - p_{j,\text{LQD}}^{t_{2,i}} \quad (4)$$

Hence, we assign a valid connection to each of the upmost  $(p_{i,\text{OPT}}^{t_{2,i}} - l_i^{t_{1,i}})$  extra packets of  $i$ , without violating Invariant 1 at  $t_{2,i}$ . This is because each of

the upmost  $(p_{i,\text{OPT}}^{t_{2,i}} - l_i^{t_{1,i}})$  extra packets of  $i$  at  $t_{2,i}$ , is located at a queue position at  $t_{2,i}$  which is at least equal to  $l_i^{t_{1,i}}$ . But, by (4) and Lemma 13, the number of free packets already connected with extra packets of  $j$  at  $t_{2,i}$  is at most  $l_i^{t_{1,i}}$ .

(A3) It cannot be  $|F^{t_{2,i}}| = 0$ , since at least one extra packet of  $i$  exists in the OPT buffer, at  $t_{2,i}$ . Therefore, by Lemma 4, it has to be  $|F^{t_{2,i}}| \geq 1$ .

**Case 2: The queue  $i$  is secondary dominating at  $t_{1,i}$**



**Fig. 3.** An example of the connection assignment of Case 2: One connection to each extra packet of  $i$  is assigned with a common packet of  $i$  at  $t_{1,i}$ . A second connection is assigned with a free packet, at the same time step

We distinguish between sub-cases (B1), (B2) and (B3), for the first connection to each extra packet of  $i$  at  $t_{1,i}$ :

(B1) If  $|F^{t_{1,i}}| = 2$  then all LQD packets are currently not connected at  $t_{1,i}$ . Hence, we assign at  $t_{1,i}$  one connection to each extra packet of  $i$  with a different common packet of  $i$ . The number of extra packets of  $i$  at  $t_{1,i}$  is at most equal to the number of common packets of  $i$  at  $t_{1,i}$ , by Definition 7. It follows that the number of common packets of  $i$  suffices for these connections at  $t_{1,i}$  and Invariant 1 holds at  $t_{1,i}$ . Each connection is valid, by Observation 4.

(B2) If  $|F^{t_{1,i}}| = 1$ , then one other dominating (or established) queue  $j \neq i$  may exist at  $t_{1,i}$ . Let us denote as  $w \neq \{i, j\}$  the single free queue we have at  $t_{1,i}$ . By Lemma 4 it is  $e(t_{1,i}) \leq f(t_{1,i}) - 1$ . Also, it is  $f(t_{1,i}) = p_{w,\text{LQD}}^{t_{1,i}} \leq p_{i,\text{LQD}}^{t_{1,i}} + 1$  due to Observation 1. By the last two inequalities, we get:

$$e(t_{1,i}) \leq p_{i,\text{LQD}}^{t_{1,i}} \quad (5)$$

By (5) and Lemma 12, the number of common packets of  $i$  that are not currently connected at  $t_{1,i}$  with extra packets of  $j$ , is at least equal to the number of extra packets of  $i$  at  $t_{1,i}$ . Hence, we assign a connection to each extra packet of  $i$  with a different common packet of  $i$  at  $t_{1,i}$ , without violating Invariant 1 at  $t_{1,i}$ .

**(B3)** It cannot be  $|F^{t_{1,i}}| = 0$ , for the same reason as in sub-case (A3).

Finally, by Lemmas 4 and 13, we assign a second connection to each extra packet of  $i$  with a different and currently not connected free packet at  $t_{1,i}$ . This second connection is valid by Observation 4.

### 3.2 Upper-bounding the LQD Competitive Ratio, when $N = 3$

The connection assignment process that we described in Cases 1 and 2, is applied for every queue  $k \in [N]$  and at each  $t_{1,k}^{\lambda_k}$  time step ( $\lambda_k \in [A_k]$ ).

**Lemma 15.** *Invariant 1 holds at every time step of  $[1, T]$ .*

*Proof.* Invariant 1 may be violated only at a time step when a connection between an extra packet and an already connected LQD packet is assigned. But we showed in both Cases 1 and 2 that at any time step when a connection is assigned to an extra packet of  $i \in [N]$ , Invariant 1 is not violated.  $\square$

**Lemma 16.** *For any transmitted extra packet  $e$ , it holds  $con(e, t) = 2$  where  $t \in [T]$  denotes the transmission time step of  $e$ .*

*Proof.* (see Appendix).  $\square$

By Lemmas 15 and Lemma 16, it follows that  $2 \cdot \sum_{t=1}^T |E^t| \leq h_\sigma(\text{LQD})$ , for any incoming packet sequence  $\sigma$ . Therefore, by (1), we have that LQD is 1.5-competitive for 3-port shared-memory switches.

## References

1. Aiello, W., Kesselman, A., Mansour, Y.: Competitive buffer management for shared-memory switches. *ACM Transactions on Algorithms*, vol. 5(1) (2008)
2. Chao, H.J., Guo, X.: *Quality of Service Control in High-Speed Networks*. Wiley-IEEE Press (2001)
3. Chao, H.J., Liu, B.: *High Performance Switches and Routers*. Wiley-IEEE Press (2007)
4. Goldwasser, M.H.: A survey of buffer management policies for packet switches. *SIGACT News*, vol. 41(1), pp. 100–128 (2010)
5. Hahne, E.L., Kesselman, A., Mansour, Y.: Competitive buffer management for shared-memory switches. In: *SPAA*, pp. 53–58 (2001)
6. Kobayashi, K.M., Miyazaki, S., Okabe, Y.: A tight bound on online buffer management for two-port shared-memory switches. In: *SPAA*, pp. 358–364 (2007)
7. Wei, S.X., Coyle, E.J., Hsiao, M.T.: An optimal buffer management policy for high-performance packet switching. In: *Proc. IEEE GLOBECOM '91*, vol. 2 (1991)

## A Appendix

### A.1 Proof of Lemma 4

First, let us say that at least one queue overflows in the LQD buffer at  $t$ . Assume for contradiction that  $f(t) < e(t) + \sum_{i \in F^t} p_{i,\text{OPT}}^t$ , that is  $\sum_{i \in F^t} p_{i,\text{LQD}}^t < \sum_{i \in D^t \cup E^t} (p_{i,\text{OPT}}^t - p_{i,\text{LQD}}^t) + \sum_{i \in F^t} p_{i,\text{OPT}}^t$ , which gives us the next inequality:

$$\sum_{i \in F^t} p_{i,\text{LQD}}^t + \sum_{i \in D^t} p_{i,\text{LQD}}^t < \sum_{i \in D^t} p_{i,\text{OPT}}^t + \sum_{i \in E^t} p_{i,\text{OPT}}^t + \sum_{i \in F^t} p_{i,\text{OPT}}^t \quad (6)$$

The left hand side of (6) equals  $M$ , since the LQD buffer becomes full at  $t$ . But the right hand side of (6) lower-bounds the OPT buffer size which has to be equal to  $M$ . Therefore, we obtain a contradiction and it holds  $f(t) \geq e(t) + \sum_{i \in F^t} p_{i,\text{OPT}}^t$ .

Now, assume that no queue overflows at  $t$  in the LQD buffer and let  $t' < t$  be the latest time step before  $t$  when a queue overflowed in the LQD buffer (if time step  $t' < t$  does not exist, then the lemma holds trivially at  $t$ , since no extra packets may exist in the OPT buffer at  $t$ ). By the argument of the first two paragraphs of this proof, it holds  $f(t') - \sum_{i \in F^{t'}} p_{i,\text{OPT}}^{t'} \geq e(t')$ . Also, it holds  $e(t') \geq e(t)$ , since no new extra packets are obtained in the period  $[t' + 1, t]$ , according to our assumption that no queue overflows in the LQD buffer, in this period. By the last two inequalities, we have:

$$f(t') - \sum_{i \in F^{t'}} p_{i,\text{OPT}}^{t'} \geq e(t) \quad (7)$$

Since no queue overflows in the LQD buffer, in the period  $[t' + 1, t]$ , it follows by Lemma 3 that any queue that is a free queue at  $t'$  will stay a free queue at every time step of the period  $[t' + 1, t]$  and that any arriving packet destined to a free queue in  $[t' + 1, t]$  will be accepted by the LQD buffer. This gives us the next inequality:

$$\sum_{i \in F^t} p_{i,\text{LQD}}^t - \sum_{i \in F^t} p_{i,\text{OPT}}^t \geq \sum_{i \in F^{t'}} p_{i,\text{LQD}}^{t'} - \sum_{i \in F^{t'}} p_{i,\text{OPT}}^{t'} \quad (8)$$

By (8) and since  $f(t) = \sum_{i \in F^t} p_{i,\text{LQD}}^t$  and  $f(t') = \sum_{i \in F^{t'}} p_{i,\text{LQD}}^{t'}$ , we have:

$$f(t) - \sum_{i \in F^t} p_{i,\text{OPT}}^t \geq f(t') - \sum_{i \in F^{t'}} p_{i,\text{OPT}}^{t'} \quad (9)$$

By (7) and (9), we have  $f(t) - \sum_{i \in F^t} p_{i,\text{OPT}}^t \geq e(t)$ , completing the proof.  $\square$

### A.2 Proof of Lemma 5

Assume that there exists a time step  $t \in [t_{1,i}^{\lambda_i}, t_{2,i}^{\lambda_i} - 1]$ , when  $i$  is an established queue. But this contradicts the definition of time step  $t_{2,i}^{\lambda_i}$ .

Assume that there exists a time step  $t \in [t_{1,i}^{\lambda_i}, t_{2,i}^{\lambda_i} - 1]$  when  $i$  is a free queue. But then,  $i$  has to overflow in the LQD buffer after  $t$  so that it becomes a dominating queue before  $t_{2,i}^{\lambda_i}$  (due to Lemma 2). But this contradicts the definition of time step  $t_{1,i}^{\lambda_i}$ .

Finally,  $i$  cannot become an inactive queue in both buffers at a time step  $t \in [t_{1,i}^{\lambda_i}, t_{2,i}^{\lambda_i} - 1]$ , since  $i$  has to overflow in the LQD buffer as a dominating queue after  $t$  and before  $t_{2,i}^{\lambda_i}$ , due to Lemma 2. But this contradicts the definition of time step  $t_{1,i}^{\lambda_i}$ .

By the arguments of the first three paragraphs of this proof and Corollary 1, the lemma follows.  $\square$

### A.3 Proof of Lemma 6

For the number of extra packets of  $i$  to strictly increase between two consecutive time steps  $t$  and  $t + 1$  both of which belonging to a period  $[t_{1,i}^{\lambda_i}, t_{2,i}^{\lambda_i}]$ , the queue  $i$  has to overflow in the LQD buffer at time step  $t + 1$  as a dominating queue. But this contradicts the definition of time step  $t_{1,i}^{\lambda_i}$  (if  $(t + 1) \in [t_{1,i}^{\lambda_i} + 1, t_{2,i}^{\lambda_i} - 1]$ ), or the definition of time step  $t_{2,i}^{\lambda_i}$  (if  $(t + 1) = t_{2,i}^{\lambda_i}$ ).  $\square$

### A.4 Proof of Lemma 7

Assume that the number of transmitted extra packets of  $i$  in the period  $[t_{2,i}^{\lambda_i}, t_{1,i}^{\lambda_i+1} - 1]$  (for a  $\lambda_i \in [A_i - 1]$ ) is strictly greater than  $p_{i,\text{OPT}}^{t_{2,i}^{\lambda_i}}$  (which equals the number of extra packets that  $i$  has at  $t_{2,i}^{\lambda_i}$ , since  $i$  is an established queue at this time step). But this means that:

- The queue  $i$  has overflowed in the LQD buffer as a dominating queue, in at least one time step of the period  $[t_{2,i}^{\lambda_i}, t_{1,i}^{\lambda_i+1} - 1]$  (let us denote as  $t'$  the first time step in the period  $[t_{2,i}^{\lambda_i}, t_{1,i}^{\lambda_i+1} - 1]$  when  $i$  overflows as a dominating queue in the LQD buffer) *and*
- The queue  $i$  is an established queue in at least one time step of the period  $[t' + 1, t_{1,i}^{\lambda_i+1} - 1]$ .

But the second argument, above, implies that  $t_{2,i}^{\lambda_i+1} < t_{1,i}^{\lambda_i+1}$  which cannot happen for the strict total order defined before. This completes the proof for the first statement of the lemma.

Working in the same way, we prove the second statement, i.e., that the number of extra packets that  $i \in [N]$  has at time step  $t_{2,i}^{\lambda_i}$ , upper-bounds the number of transmitted extra packets of  $i$  in the period  $[t_{2,i}^{\lambda_i}, T]$ .  $\square$

### A.5 Proof of Lemma 8

By the definition of time step  $t_{1,i}^{\lambda_i}$ , we have that  $c$  will never be preempted from the LQD buffer, i.e., it will be transmitted by the LQD buffer at a time step of the period  $[t_{1,i}^{\lambda_i}, t_{2,i}^{\lambda_i} - 1]$  since  $i$  becomes inactive in the LQD buffer at  $t_{2,i}^{\lambda_i}$ .

Also, the packet  $c$  will stay a common packet at each time step until its transmission by the LQD buffer. This is because  $i$  cannot become a free queue at any time step of the period  $[t_{1,i}^{\lambda_i}, t_{2,i}^{\lambda_i} - 1]$ , by Lemma 5. Since the free packets are kept only by free queues and because any LQD packet can be either a free packet or a common packet at the same time step, the proof follows.  $\square$

### A.6 Proof of Lemma 9

First of all, it cannot be  $|F^t| = 0$ , because at least one extra packet of  $i$  exists in the OPT buffer at  $t$ ; therefore at least one free packet must exist in the LQD buffer at  $t$  due to Lemma 4. Also, it has to be  $|F^t| < N = 3$  since  $i$  is a dominating queue at  $t$  and, hence, not a free queue at  $t$ .

Assume for contradiction that  $|F^t| = 1$ . The number of extra packets of  $i$  at  $t$  is at least  $p_{i,\text{LQD}}^t + 1$ , by Definition 7. Therefore, by Lemmas 1 and 4, we have  $f(t) \geq (p_{i,\text{LQD}}^t + 1) + |F^t| = p_{i,\text{LQD}}^t + 2$ . But a single free queue at  $t$  cannot have  $p_{i,\text{LQD}}^t + 2$  free packets, since  $i$  overflows at  $t$  in the LQD buffer and due to Observation 1. Hence, it cannot be  $|F^t| = 1$  and, therefore, it has to be  $|F^t| = 2$ .  $\square$

### A.7 Proof of Lemma 11

If a dominating queue overflows in the LQD buffer at  $t$ , the lemma follows from Lemmas 4 and 10. If no dominating queue overflows at  $t$ , then let  $t'$  be the latest time step before  $t > t'$  when a dominating queue overflowed in the LQD buffer; hence we have  $e(t') \geq e(t)$  since no new extra packets are obtained in the period  $[t' + 1, t]$ , according to our hypothesis that no dominating queue overflows in the LQD buffer in  $[t' + 1, t]$ .

By Lemma 4, at any time step when at least one extra packet exists in the OPT buffer, at least one free packet has to exist in the LQD buffer; therefore it is  $|F^{t'}| \geq 1$ . But it cannot be  $|F^{t'}| = 2$ , since a free queue has to become dominating at a time step of the period  $[t' + 1, t]$  so that it holds  $|F^t| = 1$ . To see that, note that in order for a free queue at any time step  $t_f \in [T - 1]$  to become dominating at  $t_f + 1$ , this queue has to overflow in the LQD buffer at  $t_f + 1$ . This contradicts our assumption that no dominating queue overflows in  $[t' + 1, t]$ . Therefore, it has to be  $|F^{t'}| = 1$ . By Lemmas 4 and 10 and since  $e(t') \geq e(t)$ , the proof is complete.  $\square$

### A.8 Proof of Lemma 14

The lemma holds for each common packet of  $i$  that is assigned a connection at  $t_{1,i}$ , since  $i$  becomes an established queue at  $t_{2,i}$  and therefore it is inactive in the LQD buffer at  $t_{2,i}$ .

Now, for the free packets that are assigned connections at  $t_{1,i}$ , it suffices to show that we only assign connections at  $t_{1,i}$  to free packets each of which is located at a queue position at most equal to  $p_{i,\text{LQD}}^{t_{1,i}}$  at this time step. But, by

Lemmas 4 and 9, the number of free packets in the LQD buffer at  $t_{1,i}$  is at least equal to the number of extra packets of  $i$  at  $t_{1,i}$ , plus 2. Therefore, we do not need to assign connections to the at most two free packets that may be located at a queue position equal to  $p_{i,\text{LQD}}^{t_{1,i}} + 1$  at  $t_{1,i}$  (by Observation 1). It follows that the maximum queue position at  $t_{1,i}$  of any free packet which is assigned a connection at  $t_{1,i}$  is  $p_{i,\text{LQD}}^{t_{1,i}}$ , which completes the proof.  $\square$

### A.9 Proof of Lemma 16

The extra packet  $e$  that is transmitted at  $t \in [T]$  belongs to a queue  $i \in [N]$  that was either primary dominating or secondary dominating at the latest time step before  $t$ , when  $i$  overflowed in the LQD buffer, due to Lemma 2 and Observation 3. Denoting this time step as  $t_{1,i} = t_{1,i}^{\lambda_i}$ , for some  $\lambda_i \in [A_i]$ , we shall distinguish between two cases.

Before doing so, recall, that (see Fact 1) if an extra packet becomes non-extra, then we transfer all its connections and associations to the inferior extra packet of it, which is located at the smaller queue position of the same queue, if such an inferior extra packet exists.

Let us start with the case that  $i$  was secondary dominating at  $t_{1,i} = t_{1,i}^{\lambda_i}$ . Each extra packet of  $i$  is assigned two connections at  $t_{1,i}$ , as we describe in the analysis of Case 2. Due to Lemmas 6 and 7, the number of transmitted extra packets of  $i$  in the period  $[t_{2,i}^{\lambda_i}, t_{1,i}^{\lambda_i+1} - 1]$  is at most equal to the number of extra packets that  $i$  has at  $t_{1,i}$ . By the argument of the previous paragraph, it follows that for each transmitted extra packet  $e$  of  $i$ , it has to be  $\text{con}(e, t) = 2$  at its transmission time step  $t \in [t_{2,i}^{\lambda_i}, t_{1,i}^{\lambda_i+1} - 1]$ .

Assume, now, that  $i$  was primary dominating at  $t_{1,i} = t_{1,i}^{\lambda_i}$  (i.e., Case 1). At  $t_{1,i} = t_{1,i}^{\lambda_i}$  we assigned two connections to each of the  $l_i^{t_{1,i}}$  lower extra packets of  $i$  and one connection to each of the upper  $u_i^{t_{1,i}}$  extra packets of  $i$ . At  $t_{2,i} = t_{2,i}^{\lambda_i}$  we assigned one connection to each of the  $p_{i,\text{OPT}}^{t_{2,i}} - l_i^{t_{1,i}}$  extra packets of  $i$  (if  $p_{i,\text{OPT}}^{t_{2,i}} > l_i^{t_{1,i}}$ ) or none connection (if  $p_{i,\text{OPT}}^{t_{2,i}} \leq l_i^{t_{1,i}}$ ). Hence, by Lemma 6 and the argument of the second paragraph of this proof, we have that each extra packet of  $i$  at  $t_{2,i}$  has two connections or associations at  $t_{2,i}$  (after the connections at time step  $t_{2,i}$  take place). Therefore, by Lemma 7, we have that for each extra packet  $e$  transmitted by  $i$  at a time step  $t \in [t_{2,i}^{\lambda_i}, t_{1,i}^{\lambda_i+1} - 1]$ , it is  $\text{con}(e, t) = 2$ .  $\square$