RZ 3440 (# 93686) 07/29/02
Electrical Engineering 8 pages

Research Report

A Stability Study of CIOQ Switches with Finite Buffers and Non-Negligible Round-Trip Time

M. Gusat, F. Abel, F. Gramsamer, R. Luijten, C. Minkenberg, and M. Verhappen

IBM Research
Zurich Research Laboratory
8803 Rüschlikon
Switzerland

A Stability Study of CIOQ Switches with Finite Buffers and Non-Negligible Round-Trip Time

M. Gusat, F. Abel, F. Gramsamer, R. Luijten, C. Minkenberg, and M. Verhappen
IBM Research, Zurich Research Laboratory, 8803 Rüschlikon Switzerland
gusat@ieee.org

Abstract—**We propose a systematic method to determine the lower bound for internal buffering of practical CIOQ switching systems. To this end we introduce a deterministic traffic scenario that stresses the global stability of finite output queues. We demonstrate its usefulness by dimensioning the buffer capacity of the CIOQ under such traffic patterns. Compliance with this property maximizes the performance achievable with finite buffers.**

Keywords: switching, backpressure, admissible, stability, RTT

I. INTRODUCTION

Most of recently proposed switching fabrics belong to the class of combined input-output queued (CIOQ) architectures [4-17]. As shown in Fig. 1, CIOQs involve a degree of internal speedup and output queue buffering. The two main classes of CIOQs are centralized with limited speedup [9-17] and distributed with full speedup [4-8]. While the speedup required for ideal output-queue (OQ) emulation [10] has been the subject of numerous studies, e.g., [2,11-17], the topic of buffering has received little attention so far. Whereas the majority of theoretical papers assume infinite buffers, practical CIOQ implementations have a limited amount of internal buffering. Independently of their physical implementation, OQ buffers can be logically managed as shared (SM) [4-8] or as partitioned/dedicated memory architectures, e.g. CICQ [23,24].

Investigations of CIOQ buffering requirements, such as [3], are less numerous, and assume negligible round-trip times (RTT). Our contribution is that we investigate the switch core behavior under increasingly large RTTs.

A question worth asking is: Can one derive a lower bound for the buffering capacity, and can one find a benchmark and a metric for assessing the global queuing capacity required by a CIOQ system? As we will show in the following, the answer is positive if we augment the definition of work-conservation with global stability.

Definition (Work-Conservation Property): A work-conserving **(WC)** switch will serve any output for which at least one packet is present in the system.

However, the WC definition is too strict to be practical in assessing the properties of CIOQ switches with finite buffers; formally; no such system can be strictly WC [1]. The consequences deriving from this result are that neither is perfect OQ emulation feasible nor is an absolutely robust and traffic-agnostic CIOQ switch physically possible. If strict work conservation is unachievable, then it follows that the notion of strict work conservation has no practical value in sizing the buffers. Therefore we propose the notion of *absolute global stability* (AGS) of a CIOQ core; this sets a tight upper limit to the global queue buildup. Based on this property we derive the lower bound for the internal buffer requirements. To derive a pragmatical instrument for buffer dimensioning, we propose a new benchmark scenario, called sweeping hotspot. From the outset we make the assumption that a switching core must be lossless under *any* traffic pattern, and its throughput (T_{put}) should emulate as closely as possible that of an ideal OQ switch. In [21] is shown that traffic patterns and behavior for the Internet are not predictable, motivating the assumption that the switch fabric should be traffic independent.

The remainder of the paper is organized as follows. In Section II we present the background and set the framework to this paper. In Section III we prove that a lower bound for the buffering capacity of CIOQ switching systems exists. To this end we will first outline our method and introduce the sweeping hotspot traffic scenario. We then prove that the absolute global stability theorem places a tight bound on the number of packets admitted to the switch without violating work-conservation. As a result, we prove that the output buffer size for both SM and CICQ must scale as $O(\mathrm{RTT}*N^2)$. The degree of AGS is proposed as the quantitative metric to differentiate between various CIOQ implementations. We conclude with future work.

II. BACKGROUND AND FRAMEWORK

A. Background

In addition to the study of the internal buffering capacity performed in [3], we credit [1,2,22] as predecessing investigations in the same direction as our current work, namely, analysis of stability and work-conservation. However, a number of basic assumptions distinguish our approach from previous studies. First, from [22] we borrow and extend the notions of admissible and inadmissible traffic [15,22]. However, as the traffic pattern proposed here is deterministic, basic calculus provides us with exact results. This contrasts with [22], where the use of ergodic traffic called for the use of stochastic methods. Second, whereas for the investigation of PPS [2] assumes infinite OQ buffers and that "no state information is communicated from the core to the inputs," we apply a different set of hypotheses for the global stability study of a single-plane CIOQ. As any practical switch contains a finite OQ capacity, OQ state information must be communicated periodically to the IQ schedulers in order to prevent OQ over-

flow. In CIOQs with centralized arbitration, such state information is implicitly included in the scheduling/arbitration algorithm [9-12]. In CIOQs with distributed scheduling, the state information can be conveyed, e.g., as backpressure (BP). A more sophisticated flow-control scheme prevents both overflow *and* underflow.

While [1] assumes a CIOQ with finite OQs, for the purpose of that study "backpressure is applied instantaneously in case an OQ is completely full". In fact, many existing CIOQs assume fractional RTTs [5], or disregard the RTT altogether. However, we argue for the opposite trend. In current CIOQ systems, the distance between IQs and OQs spans tens to hundreds of feet, whereas the cycle time has shrunk from microseconds per packet to a few nanoseconds. One can no longer neglect the transport latencies, which affect both the datapath and the control path. Furthermore, scheduling performance, scalability in number of chips and power budget per system favor the support of RTT inside the CIOQ core. In our study the sum of all logical and physical delays are lumped into the RTT. BP is characterized by the fundamental time constant of a closed-loop feedback control system, τ = RTT, which marks the delay between the *issuance* of a BP command and its *effect* becoming visible at the same location where it was issued. We assume arbitrarily large RTT values *within* the switching fabric, normalized to packet cycles, in the range of tens to hundreds.

B. Framework

A typical CIOQ system is shown in Fig. 1. The CIOQ system under study contains an $N \times N$ switch core organized as single stage and single plane. We start by developing the global stability conditions of a CIOQ with full output speedup $S_o = N$; then we generalize the results to CIOQs with any speedup S. We seek to derive stability conditions without constraining the CIOQ speedup or scheduling. The system contains a total of N^2 VOQ input queues [9-17] and N OQs; its physical and logical OQ architecture could be implemented as shared-memory [4,8], distributed [23,24] or mixed, i.e., physically distributed and logically shared [5,6]. Time-slotted operation with fixed sized packets is assumed.

An implication of the results of [1] asserts that "more buffer space at the output is always better"; the authors in [5] advocate for 2 to 4 times more buffering capacity than their actual implementation provides. The main benefit of a buffered switch resides in its capability to remove the *immediate dependency* between packet arrivals and departures; provided there is sufficient internal queuing capacity, the downstream departure processes can be arbitrarily decoupled from

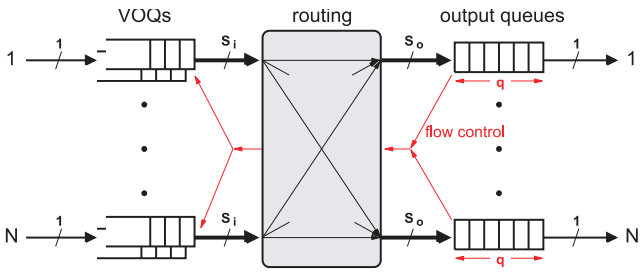

Figure 1. CIOQ system with finite buffers.

preceding and current upstream arrival processes. As a consequence, flow control, scheduling, and, essentially, the QoS service levels can be tuned to a range of throughputs and delay metrics. In short, internal buffering provides scheduling freedom, at the high expense of fast memory, i.e., low density and high power.

By definition, a WC switch must be inherently lossless. For better clarity without significantly reducing generality, we initially assume a shared-memory CIOQ architecture with On/Off backpressure, which is reactive and stateless. Here the goal of BP is to achieve lossless operation.

A difference between our work and previous studies is that [2] focuses on the absolute stability, which at the limit is equivalent to the work-conservation property of any *single* output (Definition 1, condition 3). We extend the method to a global CIOQ switching core across *all* its output ports. Intuitively defined, a globally stable CIOQ with finite buffers must achieve the same aggregate *throughput* as an ideal OQ switch with infinite buffers, if both are offered the same traffic patterns. To this end we must derive an absolute bound for the global queue buildup during a relevant, if possible, deterministic traffic scenario. The question is how to build such a scenario. This will be shown below.

III. Global Stability: Method and Traffic Scenario

In this section we analyze under which conditions a CIOQ with finite buffers and distributed scheduling can emulate the ideal OQ switch with infinite buffers, when the traffic pattern exhibits correlated spatial burstiness. Our focus is on both the individual and the aggregate output behavior. In a globally stable CIOQ the arrival processes to any O_j must be independent of those of other outputs. Decoupling can be achieved by partitioning the buffers into dedicated output queues, as in a class of CICQs, whereas in shared-memory CIOQs [4-6], it is achievable by providing sufficient speedup S and capacity for *all* the outputs. We prove here that either case elicits a global internal capacity $O(\text{RTT}*S^2)$. More specifically, we determine the global stability conditions that would permit neither overflow nor starvation of any output. Global stability under admissible traffic can be defined as the condition that, when the aggregate drain rate of all the output queues equals that of an unconstrained ideal OQ switch, the global queuing capacity available within a CIOQ core will accommodate exactly the maximum queue buildup.

A. Method

Let $\lambda_{ij}(t)$ be the instantaneous intensity of the packet-arrival process from input I_i to output queue O_j. Let $\Lambda(t)$ be the global matrix of intensities of arrival processes, as

$$\Lambda(t) = \| \lambda_{ij} \| \qquad (1)$$

With internal output speedup $S_o=N$,[1] the sum per column in (1) is the total rate of arrival processes for O_j, $\lambda_j(t) = \Sigma_i \lambda_{ij}(t)$. However, in a buffered CIOQ we are interested in the total amount of work arriving at output O_j within a specific time interval

[1] Speed up will be generalized later to values other than N.

$$a_j(t_{\text{start}}, t_{\text{end}}) = \sum_{i=1}^{N} \int_{t_{\text{start}}}^{t_{\text{end}}} \lambda ij(t)dt. \quad (2)$$

Next, let $\mu_j(t)$ be the instantaneous intensity of the departure process from the output queue O_j; $\boldsymbol{\mu}(t)$ is the vector of departure rates

$$\mu(t) = \| \mu_{j(t)} \|, \quad j \in [1, N]. \quad (3)$$

The output service rate $\mu_j(t)$ can be independently constrained, i.e., the service rate of an output O_j can be reduced or blocked by a condition external to our system (e.g., a blocking condition downstream). In a lossless CIOQ, if such an output is temporarily blocked while the arrival processes $\lambda_j(t)$ are still active, no packets should be dropped; instead, backpressure mode is activated. Accordingly, a(ny) constraint applied to $\mu(t)$ will eventually backpressure the arrival processes, in a feedback loop. Thus, $\mu(t)$ is considered an independent variable that influences $\Lambda(t)$. In general, the aggregate rate of arrival processes for any output O_j,

$$\lambda j(t) = \Sigma_i \lambda_{ij}(t) \in [0, N \cdot \Sigma \lambda_{\max,i}]. \quad (4)$$

Under ideal traffic, the total rate of arrival processes for any output O_j is bounded

$$0 \le \lambda j\,(t) \le 1, \quad (5)$$

while under admissible traffic, the sum of any column of the global matrix of intensities of arrival processes, $\lambda_j(t) < 1, (\forall)t$. We assume a bimodal distribution for the arrival processes; thus, either $\lambda_{ij}(t) = \lambda_{\max} = 1$, or $\lambda_{ij}(t) = \lambda_{\min} = 0$. If $\lambda_j(t) = 0$, e.g., because no traffic is available in the IQs, then O_j is idling by necessity. If, at the other extreme, $\lambda_j(t) > \mu_j(t)$ for a given time interval, then O_j becomes congested and its OQ will backlog, which eventually will cause backpressure. This will occur despite a departure service with maximum rate $\mu_{\max} = 1$.

Definition *We denote as inadmissible traffic any situation when the aggregate arrival intensity exceeds the available aggregate departure service rate, $\lambda_j(t) > \mu_j(t) = \mu_{\max}$.*

During inadmissible traffic, even the ideal OQ switch experiences loss of throughput and queue buildup; this will be discussed later. On the other hand, whereas the ideal OQ switch has maximum throughput under admissible traffic, a real CIOQ may still experience loss of throughput and non-work-conserving behavior. One cause is that, if not dimensioned according to global stability, admissible traffic may induce an OQ to overflow, or, starve. This could, for example, result from premature backpressuring of inputs, which otherwise could provide the OQs with new arrivals. Thus, not all the traffic offered will materialize in throughput.

B. Traffic Scenario

For the study of global stability we employ a deterministic form of admissible traffic, denoted as *sweeping hotspot.* As the name suggests, this is a deterministic traffic pattern whereby all input traffic sources synchronously target one output after another. If the *k*-th output is currently hotspotted, then

$$\sum_{i=1}^{N} \lambda_{ij}(t) = \begin{cases} N \cdot \lambda_{\max}, & j = k, \\ 0, & j \ne k. \end{cases} \quad (6)$$

Persistent contention of two, or more, arrival processes at a single O_j will eventually result in congestion and backlog the corresponding OQ. We define the *congestion rate* at the single hotspot O_j as

$$\lambda_{\text{cong},j}(t) \triangleq \lambda_j(t) - \mu_j(t) = \Sigma_i \lambda_{ij}(t) - \mu_j(t). \quad (7)$$

Assume that O_j is just being hotspotted, starting at time t_j. We define the *congestion epoch.*

$$t_{\text{CE}} = \varepsilon + \tau\,, \quad (8)$$

where ε is the BP *activation* delay offset, an initial reaction time until backpressure mode is activated[2], and, as mentioned, τ is the round-trip time (RTT) after which the effects of back-pressure activation are felt. Correspondingly, under maximum degree N:1 of hotspot congestion starting at $t_0 = 0$, the corresponding OQ_j will receive a monotonically increasing amount of work,

$$a_{\text{cong},j}(0, t_{\text{CE}}) = \sum_{i=1}^{N} \int_{0}^{t_{\text{CE}}} \lambda_{\text{cong},j}(t)dt, \quad (9)$$

which, if the traffic pattern does not cease, will also cause the hotspotting IQs upstream to backlog once the BP is activated. The congestion graph rooted on the congested output O_j, and buildup of OQ_j concatenated with its respective ingress VOQs —all fully backlogged, is denoted *saturation tree* [20].

The sweeping hotspot will periodically (minor cycle) shift the hotspot target from one output to another one; and then repeat the sequence in the next *major cycle*. A round-robin sweeping sequence is shown in Fig. 2.

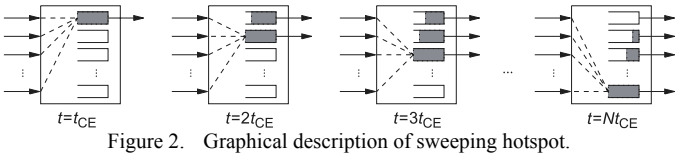

Figure 2. Graphical description of sweeping hotspot.

After a certain number of congestion epochs which depends on the CIOQ speedup, this admissible traffic pattern converges to a steady state (see Fig. 4). This particular property of the proposed benchmark for global stability enables us to test the convergence (absolute bound) of the queue buildup during back-to-back congestion periods. We will show that a CIOQ with finite buffers can emulate the ideal behavior and achieve the same steady state.

C. Absolute Global Stability

Here we prove that during admissible congestion, the global queue buildup of a CIOQ core with finite buffers is strictly upper-bounded. Moreover, no OQ underflows, i.e., it is work-conserving, provided that the global queuing capacity available in the CIOQ core equals this absolute bound.

[2]The reasoning behind this will be elaborated later.

Global Stability Theorem. *If Q is the total internal buffer capacity of an N×N CIOQ with effective RTT = $\tau + \varepsilon$, and, $Q_{\max,\,\text{open}}$ is the maximum of the global queue buildup with open outputs, then the system is globally stable if and only if*

$$Q_{\max,\,\text{open}} \le Q.$$

Assuming that $S = N$, for a shared-memory CIOQ, $Q_{\text{SM\_max, open}} = N(N-1)(\lambda_{\max} - \mu_{\max}/2)(\varepsilon + \tau)$, and for CICQ, $Q_{\text{CICQ\_max, open}} = N(N-1)\tau$.

Proof. We are interested in the effects on the global queue buildup function (sum of all backlogged OQs) when shifting a synchronized hotspot target as soon as O_j is backpressured. Globally synchronized shifting arrival processes are readily obtained, even with loosely correlated inputs; it is sufficient that all input VOQ schedulers make their *first transmission to the same* O_j and, upon receipt of the backpressure signal for OQ_j, they choose the *next hotspot target in the same sequence* —whether incremental or random walk. After the first iteration of N minor cycles, the major sweep cycle will repeat starting from the OQ signalled as available; clearly, the oldest queue will be the first to disable its BP, BP_j = Off, when j is the first queue that was congested during the preceding major sweep cycle. Global synchronization is effectively achieved by the backpressure signal, assuming that the backpressure mode is fair.

Other assumptions are that the switch starts empty, and that it employs instant and per-output discriminative backpressure mode to prevent OQ overflow. In the context of CIOQs with non-negligible RTTs, "instant backpressure" denotes that the backpressure mode is activated as soon as conflicting arrival processes are detected at any output—at most after an activation delay ε. This delay ensures that the OQ will not unnecessarily starve, by activating BP only after at least RTT*$\mu_{\max}$ packets have arrived. Thus, the effective RTT is composed of $\tau = \sum Transport\_lags$, plus the activation delay ε. Hence, "instant" activation of backpressure is not equivalent to instant effect, i.e., the arrival processes at O_j may continue with maximum intensity for up to another τ after the activation delay ε. Also, for reasons of fairness and work-conservation, backpressure must discriminate between congested and non-congested outputs; otherwise a single backlogged OQ will head-of-line (HOL)-block the remaining $N-1$ outputs as well, thus eliminating the benefits of the VOQ in the IQ.

During the congestion epoch focused on O_j the following amount of work arrives

$$a_{\text{cong},j}(t_j, t_j + t_{\text{CE}}) = \sum_{i=1}^{N} \int_{tj}^{t_j + t_{\text{CE}}} \lambda_{ij}(t)dt = \sum_{i=1}^{N} \int_{0}^{t_{\text{CE}}} \lambda_{\max,i}(t)dt = N \cdot \lambda_{\max,i} \cdot (\tau + \varepsilon). \quad (10)$$

However, not all of this work will be backlogged. Here we assume that the output departure processes are not constrained, $\mu_j(t) = \mu_{\max}$. Queue buildup $q_j(t)$ of OQ_j is proportional to the amount of work produced by the congestion rate $\lambda_{\text{cong},j}(t)$ within this epoch

$$q_{\text{cong},j}(t_j, t_j + t_{\text{CE}}) = \int_{tj}^{t_j + t_{\text{CE}}} \lambda_{\text{cong},j}(t)dt. \quad (11)$$

If we insert (7) into (11), with $\lambda_{\max} = \mu_{\max,j} = 1$,

$$q_{\text{cong},j}(t_j, t_j + t_{\text{CE}}) = (N-1) \cdot \lambda_{\max,i} \cdot (\tau + \varepsilon), \quad (12)$$

i.e., $q_{\text{cong},j}(t_j, t_j + t_{\text{CE}}) < a_{\text{cong},j}(t_j, t_j + t_{\text{CE}})$.

Next, we observe that the maximum of the local queue buildup is reached when hotspotting arrivals shift from O_j to O_{j+1}; on the traffic scenario timeline, $q_j(t)$ is a monotonically increasing function from t_j up to this moment; the rate of increase is $(N-1)\lambda_{\max}$. From now on, until the next congestion epoch, OQ_j enters its *drain epoch*, monotonically decreasing with rate $\mu_j(t) = \mu_{\max}$.

An interesting question is what is the behavior of the *global queue buildup* $Q(t)$ during a number of contiguous congestion epochs, before the sweep converges to steady state and becomes cyclic. Two lemmas are needed to conclude the proof of AGS theorem.

Monotonicity Lemma. In the first $N-1$ congestion epochs, the global queue buildup $Q(t)$ is a monotonically increasing function.

Proof. We will prove that in the restricted time domain of the initial $N-1$ congestion epochs (see Fig. 3), the global queue buildup $Q(t)$ continues to consume OQ buffering capacity, i.e., monotonicity. This sets an absolute lower bound on the capacity required for ensuring that *all* outputs are work-conserving under traffic of maximum contention, i.e., the finite buffer CIOQ emulates an ideal OQ switch. We express Q(t) as the difference between two cumulative functions,

$$Q(t) = A(0,\, t_{N-1}) - D(0,\, t_{N-1}), \quad (13)$$

where $A(0,t)$ is the total amount of work that has arrived in the switch, and $D(0,t)$ is the total number of departures since the initial time 0. Both functions are readily calculated by integration over $N-1$ congestion epochs.

Here we are interested in the monotonicity of their difference; i.e., either prove the positive sign of the first derivative or show that $Q(t_{j+1}) > Q(t_j)$ *for all* $t_{j+1} > t_j$. $A(0,t)$ is a monotonically increasing function of time, with constant rate of $a_{\text{cong},j}$, i.e., $N\lambda_{\max}$. While also monotonically increasing on each subdomain, i.e., distinct congestion epochs, the departure function $D(0,t)$ is not continuous[3] across congestion epochs; its rate, denoted $M(t)$, increases in discrete steps with every new epoch. $M(t)$ represents the aggregate departure rate from the switch as the sum of service rates $\mu_j(t)$ of currently active outputs. For $M(t)$ we observe

1st congestion epoch: $j = 1 \Rightarrow M(1) = \mu_{\max,j}\, t_{\text{CE}}$
2nd congestion epoch: $j = 2 \Rightarrow M(2) = 2 \cdot \mu_{\max,j}\, t_{\text{CE}}$
:
jth congestion epoch: $M(j) = j \cdot \mu_{\max,j}\, t_{\text{CE}}$.

[3]Therefore, we can neither use the sign of first derivative to prove monotonicity nor the second derivative to locate the maximum of $Q(t)$.

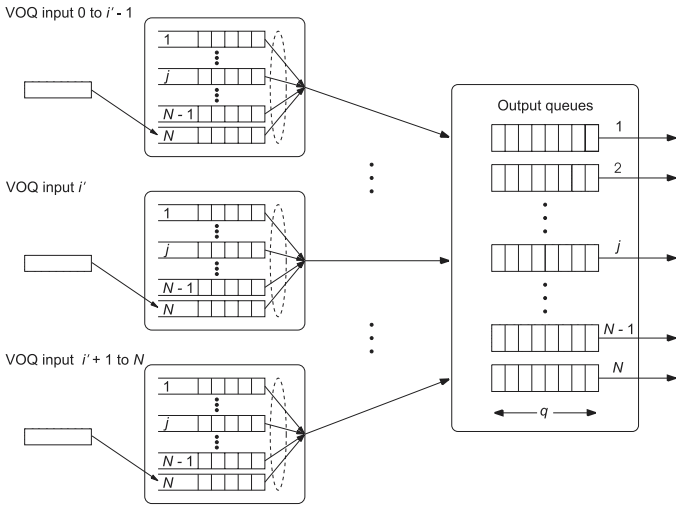

Figure 3. Sweeping hotspot snapshot: All OQs, except O_N, backlogged after $N-1$ congestion epochs.

Thus, after $N-1$ congestion epochs, there were a total of

$$A(0, t_{N-1}) = N(N-1)\,\lambda_{\max}\,t_{CE} \tag{14}$$

arrivals, whereas the total number of departures $D(0,t_{N-1})$ is sum of the arithmetic progression

$$D(0,t_{N-1}) = \sum_{j=1}^{N-1} j\cdot\mu_{\max}\cdot t_{CE} = \frac{N(N-1)}{2}\cdot\mu_{\max}\cdot t_{CE} \tag{15}$$

In general,

$$A(0,t_{j-1}) = N(j-1)\lambda_{\max}\cdot t_{CE}, \quad \text{and} \tag{16}$$

$$D(0,t_{j-1}) = \frac{j(j-1)}{2}\mu_{\max,j}\cdot t_{CE}. \tag{17}$$

The corresponding global queue buildup after $N-1$ congestion epochs $Q(t_{N-1})$ will be

$$Q(t_{N-1}) = N(N-1)(\lambda_{\max} - \mu_{\max,j}/2)\,t_{CE}. \tag{18}$$

If, for simplification. we assume $\lambda_{\max} = \mu_{\max,j} = 1$ in (18), then,

$$Q(t_{N-1}) = N(N-1)\,t_{CE}/2. \tag{19}$$

After simple calculations, we obtain

$$Q(t_{N-2}) = (N-2)(N+1)t_{CE}/2.$$

Thus, if $\delta(t_N)$ denotes the difference function,

$$\delta(t_{N-1}) = Q(t_{N-1}) - Q(t_{N-2}) = 1 > 0$$
$$\delta(t_{N-2}) = Q(t_{N-2}) - Q(t_{N-3}) = 2 > 0.$$
$$\vdots$$

Straightforward polynomial manipulations yield the general function form

$$\delta(t_j) = N-j, \qquad 0 < j \le N-1. \tag{20}$$

Therefore, the following inequality series holds:

$$Q(t_{N-1}) > Q(t_{N-2}) > \ldots > Q(t_{N-j}) > \ldots > Q(1) > 0,$$

$$(\forall)\, j \le N-1.$$

Consequently, ***$Q(t_{j+1}) > Q(t_j)$ for all $t_{j+1} > t_j$.***

□ qed. Monotonicity Lemma.

Convergence Lemma. The function $Q(t)$ converges to absolute steady-state value $Q_{\max,\,\text{open}}$.

Proof. If expressed in total number of packets to be buffered in a shared-memory CIOQ, i.e., the cumulative function of queue buildup

$$Q(t) = N(N-1)(\lambda_{\max} - \mu_{\max,j}/2)(\varepsilon + \tau). \tag{21}$$

Whereas $Q(t)$ reaches its maximum after the first congestion epoch when the difference between aggregate input and output is maximum, the cumulative function of queue buildup, $Q(t)$, peaks after precisely $N-1$ congestion epochs. From this moment on, as the sweeping hotspot visits O_N and subsequently continues with a new cycle, the system converges to steady state when, assuming that $\lambda_{\max} = \mu_{\max} = 1$,

$$Q(t) = Q_{\max,\,\text{open}} = N(N-1)\,t_{CE}/2 = \text{constant}, \tag{22}$$

and $\qquad \delta(t_j) = 0, \quad \forall\ j > -1.$

□ qed. Convergence Lemma.

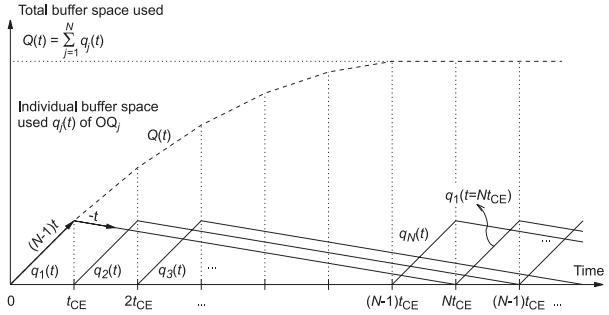

Figure 4. Convergence to steady-state limit value

As shown in Fig. 4, the aggregate drain rate increases proportionally to the number of backlogged outputs during the transient warm-up phase. The process repeats for each new congestion epoch, until convergence to steady state, i.e., until all outputs are active. At this instance the rate of global queue buildup becomes zero, while the aggregate throughput is 1.

We calculate for a shared-memory CIOQ,

$$Q_{\text{SM\_max, open}} = N(N-1)(\lambda_{\max} - \mu_{\max}/2)(\varepsilon + \tau), \tag{23}$$

At this point we compute the BP activation delay, ε, as follows. In order not to underflow an OQ with fully open output, the On/Off threshold is set according to:

$$\varepsilon * \lambda_{\text{cong},j} = \tau * \mu_{\max} \tag{24}$$

$$\boldsymbol{\varepsilon * (N-1) * \lambda_{\max} = \tau * \mu_{\max}}. \tag{25}$$

Assuming $\lambda_{\max} = \mu_{\max} = 1$, the activation delay required before *BP = Off* is

$$\varepsilon = \tau/(N-1). \tag{26}$$

Introducing the BP activation delay into Eqs. (23), we obtain the absolute lower bounds of global stability for the two classes of CIOQs:

$$Q_{\text{SM\_max, open}} = \tau N^2/2 \tag{27}$$

$$Q_{\text{CICQ\_max, open}} = \tau N (N-1). \tag{28}$$

□ (A)GS Theorem

The strong form of the global stability condition above denotes the *absolute* global stability. That is, the end of the drain epoch for the first (oldest) output hotspotted, i.e., OQ_1, coincides with the termination of the N^{th} congestion epoch and with the start of new major sweep cycle, or

$$T_{\text{drain\_AGS}} = N t_{\text{CE}}, \tag{29}$$

which prevents the underflow of the oldest queue, i.e., the loss of the work-conservation property. Absolute global stability under sweeping hotspot traffic ensures that after an output was activated, it will never unnecessarily idle. Also, the global capacity according to Eqs. (23) is sufficient to prevent that an output can be starved due to other congested outputs; this is equivalent to non-hogging.

The *necessity* proof of AGS theorem is simple. If the total amount of internal buffering is less than $Q_{\text{max, open}}$, the CIOQ will block *before* converging to steady state; i.e., one or more outputs will starve. Hence, compliance with the AGS property allows the exact emulation of an ideal OQ switch under admissible congestion traffic.

Generalization of the absolute global stability theorem (for any S). It can straightforwardly be shown that the above analysis also holds for the more general case of an $N \times N$ switch fabric with output speedup S. This is done by introducing S, instead of N, in (7)–(29). More specifically, if in (13)–(15) we replace N with S, and assume that $\lambda_{\max} = \mu_{\max} = 1$ and $\varepsilon = \tau/(S-1)$, then, for example, the total internal buffering capacity required for a shared-memory CIOQ is

$$Q_{\text{SM\_max, open}} = \tau S^2/2. \tag{30}$$

As a result, the required number of buffers depends on the RTT and speed-up S rather than the number of ports N. This result holds independently of the scheduling algorithm. □

Derivatives that follow from global stability:

Lemma 1. The load offered under admissible multiple-hotspot congestion traffic makes the same forward progress in a globally stable CIOQ switch core with finite buffers as it would in an ideal OQ switch.

Lemma 2. Separation Principle. An absolute, globally stable CIOQ switch core provides strict independence of any {*in*,*out*}*-tuple*. Therefore, a stable switch of speedup $S = N$ is resilient to N distinct saturation trees, i.e., it is non-blocking. Such a switch can support any admissible traffic pattern without loss of throughput, independently of its spatial and temporal distributions.

D. Discussion

In order to compare our results with [3], we will reverse an initial assumption, i.e., the one that all outputs are always fully *open*. This was required to establish the absolute lower bound of queue buildup during admissible congestion; in fact, $\mu_j(t) = \mu_{\max}$, $\forall\ j, t$, is optimistic over arbitrary time intervals. In a "closed" scenario, any output O_j may be arbitrarily constrained to a service rate $\mu_j(t) \in [0,1]$. A conservative variation of the sweeping hotspot scenario assumes that $\mu_j(t) = 0$ *just after*[4] the onset of a congestion epoch. In this case the local maximum of $Q(t)$ is reached only after N congestion epochs (after which outputs must open), during which no departures occurred:

$$Q_{\text{max, closed}} = N^2 \lambda_{\max} (\tau + \varepsilon'), \tag{31}$$

where ε' is the activation delay with closed outputs, calculated similar to (24)–(26). However, in this case the on/off threshold is reached sooner, after

$$\varepsilon' = \tau / N. \tag{32}$$

If $\lambda_{\max} = 1$, then

$$Q_{\text{max, closed}} = N (N+1) \tau. \tag{33}$$

The difference between our study and the worst case from [3, Section 3.2] is that we do not assume that the congested outputs must be externally blocked. By comparison, the absolute global stability shows that a shared-memory CIOQ with finite buffers can emulate the ideal OQ with infinite buffers with *less than half of the buffer* capacity resulting from [3]. Indeed, if all other factors are equal,

$$N^2/2 < N(N+1). \tag{34}$$

Next, unlike [2], where O_j is investigated independently of the other outputs, we study the stability across the full set of N outputs, by considering the dependencies arising from practical constraints and/or resource sharing.

Comparing (27) and (30) we observe that for limited speedup values, e.g., $S < N$, absolute global stability can be achieved with less internal buffering capacity. Therefore, if not considering other issues such as scalability, general work conservation, multicast and QoS, a CIOQ core with limited speedup is less expensive in terms of memory size. Also it must be observed that in Eq. (27) the value $Q_{\text{SM\_max, open}} = \tau N^2/2$, holds only for shared-memory architectures. If partitioned per output queue, an absolute globally stable CICQ with backpressure and full speedup requires *more* capacity. The following inequalities hold

$$Q_{\text{SM\_max, open}} < Q_{\text{CICQ\_max, open}} < Q_{\text{max, closed}}. \tag{35}$$

Because arrivals in a shared-memory CIOQ can readily use the queuing capacity just freed by the departures of other outputs, this architecture seems appealing. However, the relative benefit of shared-memory CIOQ vs. CICQ (ca. 50% less internal capacity for SM) is rather theoretical, as the shared-memory requires that the entire OQ capacity is fully-spedup RAM—instead of memory operating at line speed, which is sufficient for CICQ.

Finally, we observe that, whereas in (33) we derive $Q_{\text{max, closed}}$ for the first N congestion epochs, i.e., the first major cycle of the sweeping hotspot, the result is of limited practical

[4] In fact this timing is not essential; outputs could have been closed *ab initio*.

value. First, under such circumstances and with closed outputs, the ideal OQ remains *non-blocking* owing to its infinite buffers. Meanwhile its throughput is null, because the system functions as a degenerated buffered switch, i.e., as a memory. This makes the "closed output" assumption questionable. As there is no reason to stop the sweeping hotspot after the first major cycle, if this will continue past the N-th congestion epoch, the global queue buildup is unbounded in an ideal OQ switch. Unlike during the "open output" scenario, the $Q(t)$ function is *not convergent* as long as at least one departure process is constrained; the global backlog continues to increase monotonically—or, at the best, interleaved with periods of stagnation. Finally, we observe that the issue of potential output starvation upon restart is covered by the BP activation delay, according to (24)–(26) and (32).

If convergence to steady state is not achievable with constrained departures, we argue that a "closed" traffic scenario can neither be used as benchmark nor the global stability property as a metric; instead, this case ought to be treated as inadmissible traffic. Our target was to determine the inflexion point beyond which the performance gains are cancelled by the cost of over-provisioning with additional capacity.

E. Global Stability Degree

As *degree of global stability* of a CIOQ core, we introduce the normalized ratio of the global stability condition--e.g., Eq. (23) for shared-memory CIOQs--to the available internal OQ capacity of that CIOQ. This is a helpful metric to assess to which degree a CIOQ will not experience throughput loss, i.e., to which degree it is non-blocking under admissible traffic.

For example, consider a 64×64 CIOQ core with full output speedup $S = N$, packet size and memory width of 64 B, RTT = 50 packet cycles and arrival, resp. service rates $\lambda_{max} = \mu_{max} =$ 64 Gbps. To achieve a global stability degree of 1.0, if implemented in shared-memory, 6.25 MB of RAM with cycle time 0.06 ns are needed. If implemented as a CICQ with per output queue partitioned memory, 12.3 MB of RAM with a cycle time of 4 ns are needed. Clearly, the shared-memory requires a large number of interleaved memory banks, each 512-bit wide; despite the 50% advantage, the implementation cost of a globally stable SM CIOQ architecture becomes prohibitive for line rates beyond 10Gbps.

IV. Conclusions

We have derived the exact lower bound of global stability under admissible traffic, and proved that in such conditions a practical CIOQ can emulate the ideal OQ switch. For the particular case of shared-memory CIOQs, stability is achievable with buffers of less than half the size as estimated by the previous dimensioning attempts. Moreover, we have proposed the *global stability degree* as a metric to assess the potential loss of throughput under deterministic admissible traffic.

As method we have used the sweeping hotspot as a deterministic traffic benchmark to measure the stability degree. The sweeping hotspot is an important benchmark because of its unique combination of contrasting properties: locally and periodically it produces maximally contending arrival patterns of inadmissible traffic, while globally, when applied to an ideal OQ with infinite buffers, it converges to admissible traffic with steady state whereby throughput is maximal. The method is useful in sizing any CIOQ systems that must support non-negligible RTTs within the switching core.

V. Limitations and Future Work

Whereas the absolute lower bound of global stability is of practical value for a global dimensioning of CIOQ cores, the result does not provide insight into how to partition and schedule this buffering capacity. Notably missing are more specific work-conservation issues[5], i.e., does Eq. (23) hold for any other traffic scenario?

Once the issue of stability has been dealt with, some more pragmatical issues arise next. (i) Is a globally stable CIOQ core feasible, and up to which size? (ii) Can we emulate its correctness properties with less expensive constructs? (iii) Can other, tighter, benchmarks be found?

References

[1] C. Minkenberg, "Work-conservingness of CIOQ packet switches with limited output buffers," IEEE Commun. Lett. (in press, 2002).

[2] D. Khotimsky and S. Krishnan, "Stability analysis of a PPS with bufferless input demultiplexors," in Proc. ICC'2001.

[3] M. Katevenis, "Buffer requirements of credit-based flow control when a minimum draining rate is Guaranteed," in Proc. HPCS '97, 4th IEEE Workshop on Architecture & Implementation of High Performance Communication Subsystems, Chalkidiki, Greece, June 1997.

[4] C. Minkenberg and T. Engbersen, "A combined input- and output-queued packet-switch system based on PRIZMA switch-on-a-chip technology," IEEE Commun. Mag., vol. 38, no. 12, pp. 70-77, Dec. 2000.

[5] M. Katevenis, D. Serpanos, and E. Spyridakis, "Switching fabrics with internal backpressure using the ATLAS I single-chip ATM switch," in Proc. GLOBECOM '97, Phoenix, AZ, Nov. 1997.

[6] F. M. Chiussi and A. Francini, "A distributed scheduling architecture for scalable packet switches," IEEE J. Sel. Areas Commun., vol. 18, no. 12, pp. 2665-2683, Dec. 2000.

[7] R. Fan, H. Suzuki, K. Yamada, and N. Matsuura, "Expandable ATOM switch architecture (XATOM) for ATM LANs," in Proc. ICC '94, vol. 1, New Orleans, LA, May 1994, pp. 402-409.

[8] K. Yun, "A terabit multi-service switch with quality of service support," in Proc. HOTI8, A Symposium on High Performance Interconnects, Aug. 2000, Stanford, CA.

[9] N. McKeown, B. Prabhakar, and M. Zhu, "Matching output queueing with combined input and output queueing," in Proc. 35th Annual Allerton Conf. Communication, Control and Computing, Monticello, Oct. 1997.

[10] I. Stoica and H. Zhang, "Exact emulation of an output queueing switch by a combined input output queueing switch," in Proc. 6th IEEE/IFIP IWQoS '98, Napa Valley, CA, May 1998, pp. 218-224.

[11] A. Charny, P. Krishna, N. Patel, and R. J. Simcoe, "Algorithms for providing bandwidth and delay guarantees in input-buffered crossbar switches with speedup," in Proc. 6th IEEE/IFIP IWQoS '98, Napa Valley CA, May 1998, pp. 225-234.

[12] P. Krishna, N. S. Patel, A. Charny, and R. J. Simcoe, "On the speedup required for work-conserving crossbar switches," IEEE J. Sel. Areas Commun., vol. 17, no. 6, pp. 1057-1066, 1999.

[13] B. Prabhakar and N. McKeown, "On the speedup required for combined input and output queued switching," Automatica, vol. 35, 1999.

[14] S.-T. Chuang, A. Goel, N. McKeown, and B. Prabhakar, "Matching output queueing with a combined input output queued switch," IEEE J. Sel. Areas Commun., vol. 17, no. 6, pp. 1030-1039, Jun. 1999.

[5] Topics such as QoS and multicast are considered orthogonal to our study.

[15] J. G. Dai and B. Prabhakar, “The throughput of data switches with and without speedup,” in Proc. INFOCOM 2000, Tel Aviv, Israel, Mar. 2000, vol. 2, pp. 556-564.

[16] Y. Oie, M. Murata, K. Kubota, and H. Miyahara, “Effect of speedup in nonblocking packet switch,” in Proc. ICC ’89, Jun. 1989, pp. 410-414.

[17] A. K. Gupta and N. D. Georganas, “Analysis of a packet switch with input and output buffers and speed constraints,” in Proc. IEEE INFOCOM ’91, Bal Harbour, FL, Apr. 1991, pp. 694-700.

[18] I. Iliadis and W. E. Denzel, “Analysis of packet switches with input and output queuing,” IEEE Trans. Commun., vol. 41, no. 5, pp. 731-740, May 1993.

[19] A. Pattavina and G. Bruzzi, “Analysis of input and output queueing for nonblocking ATM switches,” IEEE/ACM Trans. Networking, vol. 1, no.3, pp. 314-328, Jun. 1993.

[20] G. F. Pfister and V. A. Norton, “Hot spot contention and combining in multistage interconnection networks,” IEEE Trans. Computers, vol. C-34, no. 10, pp. 933-938, Oct. 1985.

[21] S. Floyd and V. Paxson, “Difficulties in simulating the Internet,” IEEE/ACM Trans. Networking, vol. 9, no.4, pp. 392-403, Aug. 2001.

[22] N. McKeown, V. T. Anantharam, and J. Walrand, “Achieving 100% throughput in an input-queued switch,” in Proc. IEEE Infocom '96, vol. 1, pp. 296-302, San Francisco, Mar.1996.

[23] T. Javidi, R. Magill, and T. Hrabik, “A high-throughput scheduling algorithm for a buffered crossbar switch fabric,” in Proc. ICC 2001, Helsinki, SF, June 2001, vol. 5, pp. 1586-1591.

[24] R. Rojas-Cessa, E. Oki, and H. Jonathan Chao, “CIXOB-k combined input-crosspoint-output buffered packet switch,” in Proc. GLOBECOM '01, vol. 4, pp. 2654-2660.