On the Stability of Isolated and Interconnected Input-Queued Switches under Multiclass Traffic

On the Stability of Isolated and Interconnected Input-Queueing Switches Under Multiclass Traffic

Marco Ajmone Marsan, *Fellow, IEEE*,
Emilio Leonardi, *Member, IEEE*, Marco Mellia, *Member, IEEE*,
and Fabio Neri, *Member, IEEE*

***Abstract*—In this correspondence, we discuss the stability of scheduling algorithms for input-queueing (IQ) and combined input/output queueing (CIOQ) packet switches. First, we show that a wide class of IQ schedulers operating on multiple traffic classes can achieve 100% throughput. Then, we address the problem of the maximum throughput achievable in a network of interconnected IQ switches and CIOQ switches loaded by multiclass traffic, and we devise some simple scheduling policies that guarantee 100% throughput. Both the Lyapunov function methodology and the fluid modeling approach are used to obtain our results.**

***Index Terms*—Data network, network stability, performance evaluation, switching architectures.**

I. Introduction and Previous Work

A major issue in the design of input-queueing (IQ) switches based upon bufferless crossbar is that the access to the switching fabric must be controlled by some form of scheduling algorithm which operates on a (possibly partial) knowledge of the state of input queues. This means that control information must be exchanged among line cards, either through an additional data path or through the switching fabric itself, and that intelligence must be devoted to the scheduling algorithm, either at a centralized scheduler, or at line cards in a distributed manner.

We refer in this correspondence to the case of fixed-size data units, called "cells" from the asynchronous transfer mode (ATM) jargon, possibly obtained by segmenting variable-size packets (for example, Internet Protocol (IP) datagrams), and to a synchronous switch operation, according to which input/output connections are changed synchronously at every cell time (called "slot") for all ports.

The problem faced by scheduling algorithms for IQ switches with virtual output queues (VOQs) can be formalized as a maximum size or maximum weight matching on the bipartite graph in which nodes represent input and output ports, and edges represent cells to be switched. Edges may be associated with weights related to the state of input queues.

In order to achieve good scalability in terms of switch size and port data rate, it is essential to reduce the computational complexity of the scheduling algorithm. This objective has been often pursued by introducing a moderate speedup with respect to the data rate of input/output lines [2] in the switching fabric, as well as in the input and output memories. In this case, buffering is required at outputs as well as inputs, and the term "combined input/output queueing" (CIOQ) is used.

Pure IQ switches (i.e., switches with no speedup), whose scheduling policy implements a maximum weight matching (MWM) at each slot, were proved in [3], [4] to achieve the same throughput performance of output queueing (OQ) switches, under a wide class of traffic patterns, when considered in isolation, and dealing with a single class of traffic. This result holds provided that edge weights are proportional to the length of the corresponding VOQ (LQF policy), or to the age of the head-of-the-line cell (OCF policy) in the corresponding VOQ, or, finally, to the sum of all cells stored in the corresponding input and output ports (LPF policy) [5]. To the best of our knowledge, instead, no general result exists about the performance of pure IQ switches dealing with multiple traffic classes; only heuristic scheduling algorithms supporting multiple traffic classes were proposed in the literature [6]–[9], and their performance was assessed by simulation for a limited number of traffic patterns.

CIOQ switches with speedup equal to 2 were proved able to exactly emulate OQ switches implementing any monotonic work-conserving queueing discipline [2]. The scheduling algorithm considered in [2], however, is very complex. A wide class of low-complexity scheduling policies, among which maximal size matching algorithms, have been proved, in [4] and [10] to achieve the same performance of OQ switches in terms of throughput, with speedup equal to 2.

Finally, in [11] it was shown that a specific network of IQ switches implementing an MWM scheduling policy can exhibit an unstable behavior also when switches are not overloaded. This new, counterintuitive result, opened new perspectives in the research on IQ and CIOQ switches, reducing the value of most of the results obtained for switches in isolation. In [11], the authors also proposed a policy named LIN that, if implemented in each switch of the network, leads to 100% throughput under any admissible traffic pattern when each traffic flow in the network is leaky-bucket compliant. The LIN policy, however, is based on a prescheduling of cell transmissions at each switch of the network, thus relying on an exact knowledge of the traffic pattern at each switch (which calls for a large signaling bandwidth), and leading to excessive computational complexity when the traffic load approaches 1.

In this correspondence, we perform a theoretical investigation of the performance achievable by IQ switch architectures dealing with multiple traffic classes. We also focus on the performance achievable by a network of IQ and CIOQ switches. Our results are obtained by applying both the Lyapunov function and the fluid model methodologies. The interested reader can refer to [12] for a presentation of the basic theoretical results that form the background necessary to our analysis.

We first show that the extension of schedulers for IQ switches to multiclass traffic leads to surprising results. For example, we show that no IQ scheduler can achieve 100% throughput in a two-traffic-class environment, if strict priority is given to cells of one class with respect to cells of the other class. We then define a large class of scheduling policies that allow a pure IQ switch to achieve 100% throughput under multiclass traffic.

We then analyze the performance of a network of interconnected IQ switches, trying to provide a better understanding of the instability phenomena first presented in [11], which can occur in networks of IQ or CIOQ switches, even when each switch implements efficient scheduling policies.

The long-term objective of this study is the design of scheduling policies that guarantee good performance also when switches are interconnected in a network offering multiple service classes. In general, the implementation of optimal scheduling policies designed for a network of switches is rather complex, and requires a coordination among different switches, as already pointed out in [11]. However, generalizing to the context of networks of IQ switches the result obtained in [13] for networks of interacting queues, we show that the deployment of quite a simple policy, that requires a minimum amount of information to be exchanged only among neighboring nodes, guarantees 100% throughput

Manuscript received April 3, 2002; revised February 27, 2004. This work was supported in part by Lucent Technologies-Bell Labs under Contract 575/2000 and by the Italian Ministry for Education, University and Research, within the TANGO project. The material in this correspondence was presented in part at INFOCOM 2002, New York, June 2002.

The authors are with the Dipartimento di Elettronica, Politecnico di Torino, 10129 Torino, Italy (e-mail: ajmone@polito.it; leonardi@polito.it; mellia@polito.it; neri@polito.it).

Communicated by L. Tassiulas, Associate Editor for Communication Networks.

Digital Object Identifier 10.1109/TIT.2004.842562

in a network of pure IQ switches. Moreover, we show that a class of simple scheduling policies based on local information guarantees 100% throughput in a network of pure CIOQ switches with speedup equal to 2.

II. Preliminary Definitions and Notations

A. Queueing Systems

Consider a system of J discrete-time queues (of infinite size) represented by row vector Q, whose jth component, $0 \leq j < J$, is a descriptor associated with the jth queue in the system. The system of queues handles $N \geq J$ classes of customers. Each customer arrives at the network from the outside, receives service at a number of queues, and leaves the network. Customers change class every time they move through the network. We suppose that each class k of customers, $0 \leq k < N$, universally identifies a queue in the system at which all class-k customers are enqueued, i.e., all customers of class k are enqueued at the same queue. Let $L(k) = j$ be the system location function that associates each class k of customers with the queue j at which class k customers are enqueued. $L^{-1}(j)$ is the counter-image of j through function $L(k)$. In general, $L^{-1}(j)$ returns a set of customer classes. When $N = J$, each customer class is in one-to-one correspondence with a queue.

Let

$$X_n = (x_n^{(0)}, x_n^{(1)}, \ldots, x_n^{(N-1)})$$

be the row vector whose kth component $x_n^{(k)}$, $0 \leq k < N$, represents the number of customers of class k in the system at time n. We say that the set of customers of the same class forms a virtual queue in the system of queues; thus, in the correspondence we indicate the set of customers of class k with the term "virtual queue k." We suppose that the service times required by customers of all classes are deterministic and equal to one unit of time. We consider only nonpreemptive atomic service policies, i.e., service policies that serve customers in an atomic fashion, never interrupting the service of the customer that is currently in service.

The evolution of the number of queued customers is described by

$$x_{n+1}^{(k)} = x_n^{(k)} + e_n^{(k)} - d_n^{(k)}$$

where $e_n^{(k)}$ represents the number of class-k customers that entered virtual queue k (and thus physical queue $L(k)$) in time interval $(n, n+1]$, and $d_n^{(k)}$ represents the number of customers departed from virtual queue k in time interval $(n, n+1]$.

$$E_n = (e_n^{(0)}, e_n^{(1)}, \ldots, e_n^{(N-1)})$$

is the vector of entrances in the virtual queues, and

$$D_n = (d_n^{(0)}, d_n^{(1)}, \ldots, d_n^{(N-1)})$$

is the vector of departures from the virtual queues. With this notation, the system evolution equation can be written as

$$X_{n+1} = X_n + E_n - D_n. \tag{1}$$

The entrance vector is sum of two terms: the vector

$$A_n = (a_n^{(0)}, a_n^{(1)}, \ldots, a_n^{(N-1)})$$

representing the customers arrived at the system from outside, and the vector

$$T_n = (t_n^{(0)}, t_n^{(1)}, \ldots, t_n^{(N-1)})$$

of recirculating customers; $t_n^{(k)}$ is the number of customers departed from some virtual queue and entered into virtual queue k in time interval $(n, n+1]$. Note that when customers do not traverse more than one queue (as it is typically the case for an IQ switch in isolation), vector T_n is null for all n, and $E_n = A_n$.

The $N \times N$ matrix $R_n = [r_n^{(k,l)}]$ is the *routing matrix*, whose element $r_n^{(k,l)}$ represents the fraction of customers departing from virtual queue k in time interval $(n, n+1]$ that enter virtual queue l.

We assume that the system of queues forms an *open network*, i.e.,[1]

$$\Gamma = I + E[R_n] + E[R_n]^2 + E[R_n]^3 + \cdots = (I - E[R_n])^{-1}$$

exists and is finite, i.e., $I - E[R_n]$ is invertible for all n. We further assume that the routing matrix is time invariant, i.e., $E[R_n] = R$ does not depend on the time instant. We also impose that R satisfies the strong law of large numbers

$$\lim_{n\to\infty} \frac{\sum_{i=0}^{n-1} R_i}{n} = R \text{ with probability 1.}$$

Note that $T_n = D_n R_n$. The law of evolution of virtual queues can thus be rewritten as

$$X_{n+1} = X_n + A_n - D_n(I - R_n). \tag{2}$$

Let us consider the external arrivals process

$$A_n = (a_n^{(0)}, a_n^{(1)}, \ldots, a_n^{(N-1)});$$

we suppose that arrival processes are stationary, i.e.,

$$E[A_n] = \Lambda = (\lambda^{(0)}, \lambda^{(1)}, \ldots, \lambda^{(N-1)})$$

does not depend on the time interval $(n, n+1]$. Moreover, we suppose that arrival processes at each virtual queue satisfy the strong law of large numbers, i.e.,

$$\lim_{n\to\infty} \frac{\sum_{i=0}^{n-1} A_i}{n} = \Lambda \text{ with probability 1.}$$

The workload W_n provided at each virtual queue by customers that in time interval $(n, n+1]$ entered the system of queues is given on average by $E[W_n] = W = \Lambda(I - R)^{-1}$.

To simplify our notation, we define the following matrix associated with queue length vectors.[2]

Definition 1: Given vector $X \in \mathbb{R}^N$, the $N \times N$ diagonal matrix $U[X]$ is such that $U^{(j,j)}[X]$ is equal to 1 if the jth component of X, $x^{(j)}$, is nonnull, and it is equal to 0 otherwise. $U^{(i,j)}[X] = 0$ when $i \neq j$. □

B. Stability Definitions for a System of Queues

Several definitions of stability for a network of queues can be found in the technical literature. We recall here two of them.

Definition 2: A system of queues achieves *100% throughput* if

$$\lim_{n\to\infty} \frac{X_n}{n} = \lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} (E_i - D_i) = 0 \text{ with probability 1}$$

where X_n is the queue length vector at time n. □

[1] $E[X]$ denotes the expectation of random quantity X. I denotes the identity matrix, whose elements are equal to 1 on the diagonal, and null everywhere else.

[2] In this correspondence, $\mathbb{N}$ denotes the set of nonnegative integers, $\mathbb{R}$ denotes the set of real numbers, and $\mathbb{R}^+$ denotes the set of nonnegative real numbers.

A system that achieves 100% throughput is also called *rate stable*.

Definition 3: A system of queues is *strongly stable* if

$$\lim_{n\to\infty} \sup E[\|X_n\|] < \infty$$

where the operator $\|\cdot\|$ represents any possible norm over $\mathbb{R}^N$. □

III. One Switch in Isolation With Multiclass Traffic

A. Notation

We consider IQ or CIOQ cell-based switches with P input ports and P output ports, all running at the same cell rate (and we call them $P\times P$ IQS or CIOQS). The switching fabric is assumed to be nonblocking and memoryless, i.e., cells are only stored at switch inputs and outputs.

At each input, cells are stored according to a multiclass virtual output queueing (MCVOQ) policy: one separate queue is maintained at each input for each output and for each traffic class. We do not model possible output queues since they never become unstable under admissible traffic patterns.

We suppose that cells belonging to C different traffic classes arrive at input (and output) ports. Thus, the total number of input queues in each switch is $N = CP^2$. With respect to the definitions of Section II, we underline the difference between *traffic* classes and *customer* classes in the network of queues: we map cells belonging to a given traffic class onto different customer classes which depend on the VOQ at which cells are enqueued. According to the definitions of Section II, we have a single traffic class when $J = N$ (the number of VOQs equals the number of customer classes).

The switch in isolation can be modeled as a system comprising N virtual queues. Let $q^{(k)}, k = CPi+Cj+l$ be the virtual queue at input i storing cells of class l directed to output j, with $i, j = 0, 1, 2, \ldots, P-1$ and $l = 0, 1, 2, \ldots, C-1$.

We define three functions referring to VOQ $q^{(k)}$:

- $I(k)$: returns the index of the input card in which the VOQ is located;
- $O(k)$: returns the index of the output card to which VOQ cells are directed;
- $C(k)$: returns the index of the traffic class associated with the VOQ.

We consider a synchronous operation, in which the switch configuration can be changed at slot boundaries. We call internal time slot the time necessary to transmit a cell from an input toward an output. We call instead external time slot the duration of a cell on input and output lines. The difference between external and internal time slots is due to the switch speedup, and to possibly different cell formats (e.g., due to additional internal header fields).

At each internal time slot, the switch scheduler selects cells to be transferred from input queues to output queues. The set of cells to be transferred during an internal time slot must satisfy two constraints: i) at most one cell can be extracted from the MCVOQ structure at each input, and ii) at most one cell can be transferred toward each output, thus resulting in a correlation among servers activities at different queues.

We define a norm function that will be helpful in the sequel:

Definition 4: Given a vector $Z \in \mathbb{R}^N$, $Z = (z^{(k)}, k = CPi+Cj+l, i, j = 0, 1, \ldots, P-1, l = 0, 1, \ldots, C-1)$, the norm $\|Z\|_{IO}$ is defined as

$$\|Z\|_{IO} = \max_{j=0,\ldots,P-1} \left\{ \sum_{k\in I^{-1}(j)} |z^{(k)}|, \sum_{k\in O^{-1}(j)} |z^{(k)}| \right\}. \quad \square$$

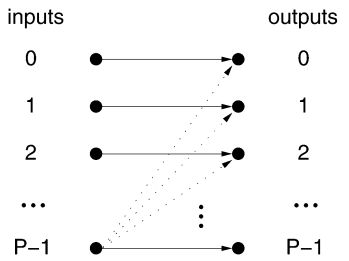

Fig. 1. Scenario in which any CIOQ switch with speedup smaller than $2-1/P$ implementing a strict priority discipline cannot achieve 100% throughput.

The constraint on the set of cells transferred through the switch can be formalized in the following manner.

Definition 5: At each time slot, the scheduler of an IQS selects for transfer from queues $Q = (q^{(k)})$ a set of cells denoted by vector $D \in \mathbb{N}^N$, $D = (d^{(k)} \in \{0,1\}, k = CPi+Cj+l, i, j = 0, 1, \ldots, P-1, l = 0, 1, \ldots, C-1)$ so that $\|D\|_{IO} \le 1$. Set D is said to be a set of noncontending cells, or a switching vector. □

In order not to overload any input and output switch port, the total average arrival rates in cells/(external slot) must be less than 1 for all input and output ports; in this case, we say that the traffic pattern is *admissible*.

Definition 6: The traffic pattern loading an (isolated) IQS is admissible if and only if $\|W\|_{IO} = \|E\|_{IO} = \|\Lambda\|_{IO} < 1$, where E is the stationary average of E_n. □

Note that any admissible traffic pattern can be sustained in an output buffered switch architecture with infinite queues.

B. Results for a Switch in Isolation

In [3] and [4], it has been proved, using two different approaches, that IQ switches subject to a single traffic class can achieve 100% throughput under a wide class of arrival processes.

In this section, we extend the discussion to IQ switches operating on multiple traffic classes. We first show that the extension of schedulers for IQ switches to the multiclass case leads to the surprising result that no IQ scheduler can achieve 100% thoughput with two (or more) traffic classes when strict priority is given to cells of one class. We then define a wide class of scheduling policies which allow the switch to achieve 100% throughput in a multiclass environment.

Let us consider a multiclass CIOQS operating according to a strict priority discipline.

Theorem 1: $2-1/P$ is the minimum speedup S required to achieve 100% throughput in a $P \times P$ CIOQ switch handling multiclass cells according to a strict priority rule.

Proof:

Necessity. Let us consider the traffic pattern described in Fig. 1, in which flows $i \to i$, with $0 \le i < P$, have strict higher priority with respect to flows $P-1 \to i$, with $0 \le i < P-1$. Suppose that the high priority arrival process at input i, with $0 \le i < P-1$, is Bernoulli, with probability $p = (P-1)/P$. Let us further suppose that high priority cells arrivals at input $i, 0 \le i < P-1$, are correlated in such a way that in each slot either no high-priority cells arrive at the switch, or $P-1$ high-priority cells arrive at the switch, one at each input i, with $0 \le i < P-1$. Finally, high-priority cells arrive at input $P-1$ with rate $q = 1/P-\epsilon$, but they can arrive only when no other higher priority cells arrive at other inputs. Low-priority cell arrivals are described by independent Bernoulli processes, with probability $q = 1/P-\epsilon$. It is immediate to verify that, for every small $\epsilon > 0$, the traffic pattern loading the switch is admissible.

We notice that, under these assumptions, high-priority and low-priority cells are never transferred at the same time. All $i \to i$ cells, with $0 \le i < P-1$, are transferred together, while $(P-1) \to (P-1)$ cells traverse the switch alone. Thus, in order to guarantee the full transfer of all the cells arriving at the switch, it must be $(P-1)q+p+q \le S$, i.e., $S \ge 2 - 1/P$.

Sufficiency. It was proved in [2] that a CIOQS with speedup $2-1/P$ can exactly emulate an OQ switch operating on different traffic classes with a strict priority discipline (in the sense that cells can depart from the two systems at the same time). The proof follows immediately. □

Speedup $2 - 1/P$ is sufficient, as proved in [2], to guarantee 100% throughput (and to guarantee strong stability) under any admissible traffic pattern for quite a large class of multiclass service disciplines. However, since the implementation cost of the scheme proposed in [2] can be significantly large, both in terms of internal bandwidth due to the required speedup, and in terms of algorithmic complexity of the scheduler, the identification of *simpler* multiclass schedulers that allow an IQS to achieve good performance is fundamental.

In the rest of this section we focus on the definition of a wide class of IQ schedulers that achieve 100% throughput in a multiclass traffic environment.

Definition 7: Let $F(X)$ be a regular function[3]

$$F \in C^1[\mathbb{R}^{+N} \to \mathbb{R}^{+N}].$$

An IQS adopts an $F(X)$-*max-scalar* scheduling policy if the selection of the switching vector in each slot is implemented according to the following rule:

$$D_n = \arg\left(\max_{D_i \in \mathcal{D}_{X_n}} F(X_n) D_i^T\right) \tag{3}$$

where X_n is the vector of queue lengths, and $\mathcal{D}_{X_n}$ denotes the set of all possible switching vectors at time n. □

Theorem 2:

Let $F(X)$ be a regular function $F \in C^1[\mathbb{R}^{+N} \to \mathbb{R}^{+N}]$ such that

1) $F(X)$ defines a conservative field, i.e.,
$$\oint_\Gamma F(X) d\Gamma(X)^T = 0 \tag{4}$$
for each regular closed line Γ in $\mathbb{R}^{+N}$;
2) $F(X)$ grows to infinity when X grows to infinity; formally, there exists a finite $s > 0$ such that
$$\liminf_{\|X\| \to \infty} \frac{\|F(X)\|}{\|X\|} \ge s; \tag{5}$$
3) all null elements of X remain null
$$U[X]F(X) = F(X). \tag{6}$$

Then an IQ switch adopting the $F(X)$-max-scalar policy is strongly stable under any admissible independent and identically distributed (i.i.d.) traffic pattern. □

Due to lack of space, the proof of this theorem is omitted. The interested reader can refer to [1], [12]. Note that condition (5) of Theorem 2, while permitting to associate different finite weights with different traffic classes, prevents strict priorities among traffic classes, which would require infinite weight ratios. Using fluid models [14], the previous theorem can be extended as follows to more general traffic processes, by relaxing the stability conditions.

Theorem 3: An IQ switch adopting the $F(X)$-max-scalar policy satisfying the conditions of Theorem 2, and such that $F(\alpha X) = \alpha F(X)$ for all scalars α, achieves 100% throughput under any admissible traffic pattern satisfying the strong law of large numbers. □

We omit also the proof of this theorem for lack of space (see [1], [12]); however, we notice that Theorem 4, whose proof is reported in Section IV, provides a more general result, from which the statement of this theorem can be directly derived as a particular case.

It can be easily verified that, for each symmetric co-positive diagonal matrix W, $F(X) = XW$ (note that $F(X)$ is now a function in $C^\infty[\mathbb{R}^{+N} \to \mathbb{R}^{+N}]$) satisfies properties (4) and (5). Note that $F(X)$ can be seen as the gradient of function $\mathcal{L}(X) = \frac{1}{2} XWX^T$. To meet also constraint (6), we require W to be diagonal, and state the following result.

Corollary 1: Let W be a diagonal co-positive matrix, and let $F(X)$ be a function in $C^\infty[\mathbb{R}^{+N} \to \mathbb{R}^{+N}]$ defined as $F(X) = XW$. Then, a multiclass switch implementing the $F(X)$-max-scalar policy is strongly stable under any admissible traffic pattern if the number of arrivals at VOQs in each slot forms an i.i.d. sequence. The switch is rate stable, under any admissible traffic pattern, if the sequences of arrivals at the VOQs satisfy the strong law of large numbers. □

IV. Networks of IQS

A. Notation

We consider in this section a network of K IQS. Switch k, $0 \le k < K$, has P_k input ports and P_k output ports, all at the same cell rate. Each switch handles C classes of traffic, and performs an MCVOQ at inputs. Thus, there are CP_k^2 different VOQs at switch k.

The network of switches can thus be modeled as a system Q containing $N = \sum_k CP_k^2$ virtual queues. We restrict our study to the case $P_k = P \ \forall\, k$, so that $N = CP^2K$. Let $S(n)$ be the function that returns the switch on which VOQ n is located; let $I(n)$ be the function that returns the index of the input card at switch $S(n)$ on which the VOQ is located; let $O(n)$ be the function that returns the index of the output card at switch $S(n)$ to which VOQ cells are directed; let, finally, $C(n)$ be the function that returns the index of the traffic class associated with queue n. The queue at input $I(n)$ of switch $S(n)$ storing cells of class $C(n)$ directed to output $O(n)$ is called $q^{(n)}$.

We adapt as follows the concept of $\|Z\|_{IO}$ to the case of a network of switches handling multiclass traffic.

Definition 8: Given a vector $Z \in \mathbb{R}^N$, $Z = \{z^{(n)}, n = CP^2k + CPi + Cj + l, 0 \le k < K,\ i,j = 0,1,\ldots,P-1,\ l = 0,1,\ldots,C-1\}$, the norm $\|Z\|_{IO}$ is defined as

$$\|Z\|_{IO} = \max_{\substack{k=0,\ldots,K-1 \\ i=0,\ldots,P-1}} \left\{ \sum\nolimits_{n \in S^{-1}(k) \cap O^{-1}(i)} |z^{(n)}|, \sum\nolimits_{n \in S^{-1}(k) \cap I^{-1}(i)} |z^{(n)}| \right\}. \tag{7}$$

□

At each time slot, a set of noncontending cells departs from the VOQs of each switch. More formally, we say that, at each time slot, the departure vector $D \in \{0,1\}^N$ satisfies

$$\|D\|_{IO} \le 1.$$

Definition 9: The traffic pattern loading a network of IQSs is admissible if and only if

$$\|W\|_{IO} = \|E\|_{IO} = \|\Lambda(I-R)^{-1}\|_{IO} \le 1$$

where $R = E[R_n]$ is the $N \times N$ average routing matrix defined in Section II-A. □

[3] C^n denotes the set of continuous functions with continuous ith derivative, $1 \le i \le n$.

Note that an admissible traffic pattern can be transferred without losses in a network of output buffered switches.

B. Main Result for a Network of IQS

In [11], it was shown that a particular network of IQS exhibits an unstable behavior under admissible traffic patterns, even when the switches implement a policy that would guarantee the stability of each switch in isolation under the same load. In this subsection, we formalize and generalize such result by providing through the fluid models theory a general definition of the stability region of a network of IQS.

Let us introduce our main result.

Theorem 4: An open network of multiclass IQS implementing the $F(X)$-max-scalar policy is rate stable under each admissible traffic pattern such that arrival sequences at VOQs satisfy the strong law of large numbers if

- $G(X) = F(X)[(I-R)^{-1}]^T$ defines a conservative field;
- $F(X)$ satisfies conditions (5) and (6);
- $F(\alpha X) = \alpha F(X)$ for all scalars α.

Proof: Let us write the fluid equations [4]

$$X(t) = \Lambda t - D(t)(I-R)$$

with the constraints for a max-scalar service policy

$$\dot{d}^{(k)}(t) = \sum_{\alpha} \dot{w}_\alpha(t)\pi_\alpha^{(k)}$$

$$\sum_{\alpha} \dot{w}_\alpha(t) = 1$$

$$\dot{w}_\alpha(t) = 0 \quad \text{if } \exists \alpha' : F(X(t))\Pi_{\alpha'}^T > F(X(t))\Pi_\alpha^T.$$

Note that the expression that defines the $F(X)$-max-scalar policy is equivalent to

$$\dot{D}(t) = \arg\max_{\Pi_\alpha} F(X(t))\Pi_\alpha{}^T.$$

Being the network of switches open, and being $F(X)[(I-R)^{-1}]^T$ a conservative field, $\sum_{i=0}^{\infty} R^i$ converges to the finite co-positive matrix $(I-R)^{-1}$. We define as Lyapunov function of the system

$$\mathcal{L}(X) = \int_{\Gamma_X} F(Y)[(I-R)^{-1}]^T d\Gamma_X^T(Y)$$

where Γ_X is a regular line whose endpoints are 0 and X.

Since $(I-R)^{-1} - I = \sum_{i=1}^{\infty} R^i$ is weakly co-positive,[4] it results in

$$\mathcal{L}(X) \geq \int_{\Gamma_X} F(Y) d\Gamma_X^T(Y) > 0, \qquad \forall\, X \neq 0.$$

Let us write the time derivative of $\mathcal{L}(X(t))$

$$\dot{\mathcal{L}}(X(t)) = \nabla \mathcal{L}(X(t))\dot{X}(t)^T = F(X(t))[(I-R)^{-1}]^T \dot{X}^T(t).$$

Substituting in the relation above the expression

$$\dot{X}(t) = \Lambda - \dot{D}(t)(I-R)$$

we obtain

$$\dot{\mathcal{L}}(X(t)) = F(X(t))[(I-R)^{-1}]^T[\Lambda - \dot{D}(t)(I-R)]^T.$$

Then

$$\begin{aligned}\dot{\mathcal{L}}(X(t)) &= F(X(t))[(I-R)^{-1}]^T\Lambda^T - F(X(t))\dot{D}^T(t) \\ &= F(X(t))[(I-R)^{-1}]^T\Lambda^T \\ &\quad - F(X(t))\left(\arg\max_{\Pi_\alpha} \Pi_\alpha F(X(t))^T\right)^T.\end{aligned}$$

[4]Matrix $W \in \mathbb{R}^{+\mathbb{N}} \times \mathbb{R}^{+\mathbb{N}}$ is said weakly co-positive if, for each $X \in \mathbb{R}^{+\mathbb{N}}$, XWX^T is nonnegative.

By definition of the $F(X)$-max-scalar policy, for each Λ such that $\Lambda(I-R)^{-1}$ belongs to the convex hull of Π_α, expression (8) is negative. Thus, for each traffic pattern such that $\|\Lambda(I-R)^{-1}\|_{IO} < 1$, the network of switches is rate stable. □

Similarly to the case of the single switch in isolation, it is possible to extend the result to more general functions $F(X)$ under any admissible i.i.d. traffic pattern (i.e., under a smaller class of traffic patterns with respect to the assumptions of Theorem 4), by directly applying the Lyapunov function methodology to equations describing the stochastic evolution of the system.

Theorem 5: An open network of multiclass IQSs implementing the $F(X)$-max-scalar policy is strongly stable under each i.i.d. admissible traffic pattern if

- $G(X) = F(X)[(I-R)^{-1}]^T$ defines a conservative field;
- $F(X)$ satisfies conditions (5) and (6). □

For a proof of this theorem the interested reader is, again, referred to [12].

The $F(X)$-max-scalar policy defined in this subsection represents a generalization of the policy originally defined in [13] for the context of networks of IQSs dealing with multiclass traffic.

The problem of the existence of a scheduling policy for interconnected IQS that makes the network rate stable under any admissible traffic pattern is thus related to the existence of a function $F(X)$ at each switch such that (5) and (6) are satisfied, and in addition, $F(X)[(I-R)^{-1}]^T$ defines a conservative field.

Note that $F(X) = XM(I-R)^T$, where M is a co-positive diagonal matrix, satisfies both (5) and (6), and

$$F(X)[(I-R)^{-1}]^T = XM(I-R)^T[(I-R)^{-1}]^T = XM$$

hence $F(X)[(I-R)^{-1}]^T$ defines a conservative field. The policy $F(X)$-max-scalar can be implemented in a distributed fashion by running, at each switch, a "local" MWM algorithm, for which the weight associated with VOQ $q^{(k)}$ is given by the kth component of vector

$$F(X) = XM(I-R)^T = XM - XMR^T$$

i.e.,

$$x^{(k)}M^{(k,k)} - \sum_{j=0}^{N-1} x^{(j)}M^{(j,j)}r^{(k,j)}.$$

Since $r^{(k,j)}$ is nonnull only for directly connected downstream switches, only the knowledge of VOQ's lengths at neighboring switches is required to correctly compute the weight of VOQ $q^{(k)}$. This requires some form of interaction (through signaling) among adjacent switches in the network. The policy above cannot be exactly implemented in a network due to the propagation delay between switches. However, it can be approximately implemented by acquiring periodically at each switch an approximate knowledge of the queue's state at neighboring switches. It is possible to show (see, e.g., [15], [16]) that finite delays in propagating the queue's state information may affect general performance indices, such as average packet delays, but they do not reduce the stability region of the scheduling algorithm. As a consequence, by properly choosing the frequency at which the exchange of the queue's state information among neighboring nodes takes place, it is possible to limit the required communication overhead.

In general, when $F(X)[(I-R)^{-1}]^T$ does not form a conservative field, Theorems 4 and 5 provide no insight into the stability region of the network of switches. The methodology developed in the proof of the theorem can be, however, extended to find conditions on the stability region of policies that do not define a conservative field.

Indeed, restricting our analysis to $F(X) = XM$ policies, whenever $M[(I-R)^{-1}]^T$ is not symmetric, and thus does not define a conservative field, it is always possible to find a matrix B, such that $M[(I-R)^{-1}]^T + B$ becomes symmetric; by defining as Lyapunov function of the system

$$\mathcal{L}(X) = \frac{1}{2} X\{M[(I-R)^{-1}]^T + B\}X^T$$

we get

$$\begin{aligned}\dot{\mathcal{L}}(X(t)) &= X(t)\{M[(I-R)^{-1}]^T + B\}\dot{X}(t)^T \\ &= X(t)\{M[(I-R)^{-1}]^T + B\}[\Lambda - \dot{D}(t)(I-R)]^T \\ &= X(t)\{M[(I-R)^{-1}]^T + B\}\Lambda^T \\ &\quad - X(t)M\dot{D}(t)^T - X(t)B(I-R)^T\dot{D}(t)^T \\ &= X(t)\{M[(I-R)^{-1}]^T + B\}\Lambda^T \\ &\quad - X(t)M\Pi_F^T - X(t)B(I-R)^T\Pi_F^T \\ &= X(t)M\left\{[[(I-R)^{-1}]^T + M^{-1}B]\Lambda^T \right. \\ &\qquad \left. -M^{-1}B(I-R)^T\Pi_F^T\right\} - X(t)M\Pi_F^T\end{aligned}$$

where Π_F is the switching matrix selected according to the $F(X)$-max-scalar policy. Thus, the policy is rate stable for each Λ when the term inside the braces belongs to the convex hull defined by departure vectors, i.e., when

$$\|[(I-R)^{-1}]^T + M^{-1}B\}\Lambda^T - M^{-1}B(I-R)^T\Pi_F^T\|_{IO} < 1. \quad (8)$$

Note that, since Π_F depends on $X(t)$, and in general can be any switching matrix, the above inequality must be satisfied for any switching matrix. Note, finally, that the satisfaction of the equation above represents a sufficient (but not necessary) condition for stability.

V. Networks of CIOQS

In this section, we investigate the stability properties of networks of CIOQS by applying the fluid models methodology. Switch k is provided with CP_k^2 different VOQs at input ports, and CP_k queues at output ports. We assume that queues at output ports implement any work-conserving service discipline. Since, under admissible traffic, all queues at output ports are rate stable, the network of queues is stable whenever queues at input ports are rate stable, i.e., instabilities may originate only at input queues. We therefore neglect output queues in the fluid models. Note, indeed, that applying fluid scaling, each rate-stable queue reduces to a fluid queue that is permanently empty and can thus be neglected without inducing any perturbation in the behavior of other fluid queues.

A. Minimum Speedup for the $F(X)$-Max-Scalar Policy

An interesting problem is, given a network of CIOQ switches and an assigned $F(X)$-max-scalar policy, to find the rate stability region of the policy. Particularly, we are interested in the minimum speedup required at switches in order to make the network stable under any admissible traffic pattern.

Although this problem cannot be exactly solved, by restricting the investigation to the particular case $F(X) = XM$, an upper bound to the required speedup is provided by the solution of the following quadratic optimization problem.

Let Π be a binary switching vector in $\mathbb{R}^{+N}$, and B be a matrix in $\mathbb{R}^{+N} \times \mathbb{R}^{+N}$ satisfying the following constraints:

$$\|\Pi\|_{IO} = 1$$
$$M[(I-R)^{-1}]^T + B \in \mathcal{S}_{\mathrm{sym}}$$

where R is the routing matrix, and $\mathcal{S}_{\mathrm{sym}}$ is the set of all symmetric matrices. Find the minimum speedup

$$\min S$$

subject to the constraint

$$\begin{aligned}S \geq \| &\left\{[(I-R)^{-1}]^T + M^{-1}B\right\}\Lambda^T \\ &- M^{-1}B(I-R)^T\Pi^T\|_{IO}, \qquad \forall\, \Lambda : \|\Lambda\|_{IO} \leq 1.\end{aligned}$$

The above optimization problem stems from (8): if all switches are stable under the traffic pattern assumed by Theorem 4, the network is also stable.

B. Networks of Switches With Per-Flow Queueing

We are now interested to find conditions for stability in a network of switches where each switch schedules cells according to local information. We restrict our investigation to switches performing per-flow queueing at inputs, i.e., storing each network flow in a separate queue.

In this case, the routing matrix R has as many rows and columns as the number of flows, and, under deterministic routing, it is a binary matrix. This means that R is subunitary; i.e., $RR^T = I_s$, where I_s is a binary diagonal matrix with unitary and null eigenvalues. Let us introduce the diagonal matrix H whose diagonal element $h^{(i,i)}$ represents the number of switches that packets stored in VOQ $q^{(i)}$ must still traverse, i.e., the residual hop count.

Definition 10: A scheduling policy is said to be "rate-fair" if, for some finite integer number w in every window of size w slots, the ratio between the service rate given at VOQ $q^{(k)}$ (with $S(k) = s$, $I(k) = i$, and $O(k) = j$) that never empties during the considered window, and the total service rate of VOQs either residing at input i of switch s, or directed to output j of switch s, is larger than $[\Lambda(I-R)^{-1}]^{(k)}/2$. □

In plain words, a rate-fair policy serves contending nonempty queues proportionally to their average arrival rates.

Note that rate-fair policies can be easily implemented in switches. For example, the policy RED-SP described in [10], and PIM [17] with queue selection probabilities at input and output arbiters proportional to the average rates, are rate-fair policies. Also i-SLIP [2] can be made rate-fair by modifying pointers update rules, which must follow a weighted round robin adapted to the average rates. However, the implementation of all rate-fair policies requires the exact knowledge of average arrival rates of individual flows.

We propose here a Rate-Fair maximal Size Matching Selection Policy (RFmSM-SP), a rate-fair maximal[5] size matching policy that does not require the knowledge of average rates. The proposed scheduling policy works as follows: in each internal time slot, the queues at which an arrival occurred in the corresponding external time slot are considered. Among them, a maximal set of noncontending queues (matching) is selected, by solving the possible contentions on output ports at random. Then, the matching is completed to maximal size by considering also queues at which no arrivals were observed. Note that, since RFmSM-SP operates on the basis of local information, it does not require any form of information exchange (signaling) among switches.

It can be easily proved that RFmSM-SP is rate-fair under any admissible traffic pattern satisfying the strong law of large numbers, by repeating the same calculations performed in [10] to prove the stability of the Random Rate-Driven (RRD) policy.

[5]A *maximal* matching is such that no input/output pairs can be added without violating admissibility constraints; by contrast, a *maximum* matching has maximum weight or size.

Theorem 6: A network of CIOQS that performs per-flow queueing at the inputs, with speedup $S \geq 2$, implementing RFmSM-SP, achieves 100% throughput under any admissible traffic pattern satisfying the strong law of large numbers.

Proof: To simplify the notation, we assume in the proof that all switches are of the same size $P \times P$, but the same arguments can be easily extended to more general cases.

In order to prove the theorem, we need to introduce the following assertions.

Proposition 1: Let $\mathbb{1}$ be a vector in $\mathbb{R}^{+\mathbb{N}}$ whose components are all equal to 1. Then

$$[(I-R)H]\mathbb{1}^T - \mathbb{1}^T = 0.$$

This property can be immediately verified by direct inspection. Indeed, the rows of $(I-R)H$ contain only two nonnull elements: the element on the diagonal is equal to the distance of the corresponding queue from the flow destination, while the other nonnull element is equal to minus the distance of the first downstream queue from the destination.

Proposition 2: For any admissible external rate vector Λ

$$\Lambda H\mathbb{1}^T = \Lambda(I-R)^{-1}\mathbb{1}^T = (\Lambda + \Lambda R + \Lambda R^2 + \cdots + \Lambda R^{\Delta})\mathbb{1}^T$$

where Δ is the network diameter.

This property can be either verified by direct inspection, or from Proposition 1

$$H\mathbb{1}^T = (I-R)^{-1}\mathbb{1}^T.$$

Proposition 3: For each empty queue i, at any regular point t, $[\dot{X}]^{(i)}(t) = 0$, i.e., $\dot{d}^{(i)}(t) = \dot{e}^{(i)}(t)$.

This property was proved for a general network of queues in [14].

Let Q be the normalized matrix defined by

- $q^{(i,i)} = 1, \forall\, i$;
- $q^{(i,j)} = 1, \forall\, i,j$ such that i and j refer to two VOQs that are either both located at the same input port of the same switch, or located at different input ports of the same switch but leading to the same output port;
- $q^{(i,j)} = 0$ otherwise.

Note that Q is symmetrical.

Proposition 4: For each queue i such that $x^{(i)}(t) > 0$: $[\dot{D}(t)Q]^{(i)} \geq S$.

This property immediately derives from the definition of maximal matching scheduling policy, as pointed out in [4], and from the available speedup S. This property implies that, for each queue such that $x^{(i)}(t) > 0$, the aggregate service rate at the associated input $[\sum_{k \in S^{-1}(i) \cap I^{-1}(i)} \dot{d}^{(k)}(t)]$ and the aggregate service rate at the output $[\sum_{k \in S^{-1}(i) \cap O^{-1}(i)} \dot{d}^{(k)}(t)]$, counting $\dot{d}^{(i)}(t)$ only once, cannot be less than S. Thus, by the definition of rate-fair policy, we have the following.

Proposition 5: Under any admissible traffic, for each queue $q^{(i)}$ such that $x^{(i)}(t) > 0$, there exists an $\epsilon > 0$ such that

$$\dot{d}^{(i)}(t) \geq [\Lambda(I-R)^{-1}]^{(i)} + \epsilon, \qquad \text{if } S \geq 2.$$

Indeed, the definition of rate-fair policy implies that

$$\frac{\dot{d}^{(i)}(t)}{[\dot{D}(t)Q]^{(i)}} > \frac{[\Lambda(I-R)^{-1}]^{(i)}}{2}.$$

Proposition 5 is immediately obtained by combining Proposition 4 with the equation above, under the assumption $S \geq 2$.

Proposition 5 can be extended also to empty queues, i.e., to queues such that $x^{(i)}(t) = 0$. In this case, however, the assumptions of Proposition 5 must be relaxed in order to allow $\dot{d}^{(i)}(t) = [\Lambda(I-R)^{-1}]^{(i)}$. Indeed, since by Proposition 3, for each empty queue $q^{(i)}$, the departure rate $\dot{d}^{(i)}(t)$ must equal the arrival rate at the queue, and the arrival rate at the queue cannot be less than $[\Lambda(I-R)^{-1}]^{(i)}$, since it must either be equal to the departure rate of the last nonempty upstream queue traversed by the considered flow, or be equal to the external flow arrival rate if no nonempty queues are encountered by the considered flow on its way to queue $q^{(i)}$. Thus. we obtain the following.

Proposition 6: For each empty queue $q^{(i)}$ (i.e., for each queue i such that $x^{(i)}(t) = 0$), under speedup $S \geq 2$: $\dot{d}^{(i)}(t) \geq [\Lambda(I-R)^{-1}]^{(i)}$.

We are now ready to prove Theorem 6.

Let $\mathcal{L}(X) = XH\mathbb{1}^T$ be the Lyapunov function. Note that $\mathcal{L}(X) \geq 0$, for any $X \in R^{+\mathbb{N}}$; in addition $\mathcal{L}(X) = 0$ iff $X = 0$.

For any t such that $X(t)$ is derivable

$$\dot{\mathcal{L}}(X) = \dot{X}H\mathbb{1}^T.$$

Thus, for $X(t) \neq 0$, and for each t such that $X(t)$ is derivable

$$\begin{aligned}
\dot{\mathcal{L}}(X) &= [\Lambda - \dot{D}(I-R)]H\mathbb{1}^T \\
&= \Lambda H\mathbb{1}^T - \dot{D}(I-R)H\mathbb{1}^T = \text{(by Proposition 1)} \\
&= \Lambda H\mathbb{1}^T - \dot{D}\mathbb{1}^T = \text{(by Proposition 2)} \\
&= \Lambda(I-R)^{-1}\mathbb{1}^T - \dot{D}\mathbb{1}^T \leq \text{(by Propositions 5 and 6)} \\
&\leq \Lambda(I-R)^{-1}\mathbb{1}^T - \Lambda(I-R)^{-1}\mathbb{1}^T - \epsilon\mathbb{1}_{U[X]}\mathbb{1}^T \\
&< 0
\end{aligned}$$

where $\mathbb{1}^T_{U[X]}$ is a vector whose kth component is 1 if queue $x^{(k)}(t) > 0$, and 0 otherwise. Thus, $\dot{\mathcal{L}}(X) < 0$ whenever $\mathcal{L}(X) > 0$, so that the system is rate stable. □

VI. Conclusion

We considered in this correspondence input-queued packet switches under multiclass traffic. Here are our three most important results. We defined a large class of scheduling algorithms, called $F(X)$-max-scalar, that guarantee stability to a switch in isolation under admissible multiclass traffic patterns. We extended the above result to networks of interconnected switches, showing that state information must be exchanged among adjacent switches to guarantee stability. We defined a class of scheduling policies requiring no exchange of information among switches, that guarantee the stability of networks of combined input/output queued switches operating with internal speedup; we also proposed RFmSM-SP, a simple, easily implementable, scheduling policy belonging to the above class.

References

[1] E. Leonardi, M. Mellia, M.. Ajmone Marsan, and F. Neri, "On the throughput achievable by isolated and interconnected input-queueing switches under multiclass traffic," in *Proc. IEEE INFOCOM 2002*, New York, Jun. 2002, pp. 1605–1614.

[2] S. T. Chuang, A. Goel, N. McKeown, and B. Prabhakar, "Matching output queuing with combined input and output queuing," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 12, pp. 1030–1039, Dec. 1999.

[3] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand, "Achieving 100% throughput in an input-queued switch," *IEEE Trans. Commun.*, vol. 47, no. 8, pp. 1260–1272, Aug. 1999.

[4] J. G. Dai and B. Prabhakar, "The throughput of data switches with and without speedup," in *Proc. IEEE INFOCOM 2000*, Tel-Aviv, Israel, Mar. 2000, pp. 556–564.

[5] N. McKeown, "Scheduling algorithms for input-queued cell switches," Ph.D. dissertation, Univ. Calif. Berkeley, 1995.

[6] M. Ajmone Marsan, A. Bianco, E. Leonardi, and L. Milia, "RPA: A flexible scheduling algorithm for input buffered switches," *IEEE Trans. Commun.*, vol. 47, no. 12, pp. 1921–1933, Dec. 1999.
[7] A. Hung, G. Kesidis, and N. McKeown, "ATM input-buffered switches with the guaranteed-rate property," in *Proc. ISCC '98*, Athens, Greece, Jun. 1998, pp. 331–335.
[8] C. Cheng-Shang, C. Wen-Jyh, and H. Hsiang-Yi, "Birkhoff–von Neumann input buffered crossbar switches," in *Proc. IEEE INFOCOM 2000*, Tel Aviv, Israel, Apr. 2000, pp. 1614–1623.
[9] V. Tabatabaee, L. Georgiadis, and L. Tassiulas, "QoS provisioning and tracking fluid policies in input queueing switches," in *Proc. IEEE INFOCOM 2000*, Tel-Aviv, Israel, Apr. 2000, pp. 1624–1633.
[10] E. Leonardi, M. Mellia, F. Neri, and M. Ajmone Marsan, "On the stability of input-queued switches with speedup," *IEEE/ACM Trans. Netw.*, vol. 9, no. 1, pp. 104–118, Feb. 2001.
[11] M. Andrews and L. Zhang, "Achieving stability in networks of input-queued switches," in *Proc. IEEE INFOCOM 2001*, Anchorage, AK, Apr. 2001, pp. 1673–1679.
[12] E. Leonardi, M. Mellia, M. Ajmone Marsan, and F. Neri, "On the throughput achievable by isolated and interconnected input-queueing switches under multiclass traffic," Dipartimento di Elettronica, Politecnico di Torino, Tech. Rep. 16-02-2004, 2004.
[13] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Autom. Control*, vol. 37, no. 12, pp. 1936–1948, Dec. 1992.
[14] J. G. Dai. (1999) Stability of Fluid and Stochastic Processimg Networks. Miscellanea Publication no. 9, Centre for Mathematical Physichs and Stochastic, Denmark. [Online]. Available: http://www.maphysto.dk
[15] L. Tassiulas, "Linear complexity algorithms for maximum throughput in radio networks and input queued switches," in *Proc. IEEE INFOCOM 1998*, San Francisco, CA, Apr. 1998, pp. 533–539.
[16] P. Giaccone, B. Prabhakar, and D. Sha, "Randomized scheduling algorithms for high-aggregate bandwidth switches," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 5, pp. 546–559, May 2003.
[17] T. Anderson, S. Owicki, J. Saxe, and C. Thacker, "High speed switch scheduling for local area networks," *ACM Trans. Comp. Syst.*, pp. 319–352, Nov. 1993.

Diversity Combining for the Z-Channel

Torleiv Kløve, *Fellow, IEEE*, Paul Oprisan, and Bella Bose, *Fellow, IEEE*

***Abstract*—Corrupted packets that cause retransmission requests in automatic retransmission request (ARQ) systems can be reused. They can be combined with additional stored copies of the transmitted packet in order to obtain a single packet which is more reliable than any of the constituents. A scheme which suits the Z-channel is proposed here and the performance is analyzed under different coding assumptions.**

***Index Terms*—Asymmetric errors, automatic retransmission request (ARQ) protocols, error detection, optical communication, packet combining.**

I. Introduction

Error detection as part of a feedback error control system is a reliable alternative to feedforward error correction in asymmetric channels. This is because of simpler hardware implementation of the encoding/decoding system and, further, the lack of asymmetric error-correcting codes with better rates than the corresponding binary symmetric codes. Asymmetric errors are typical in optical communication because, upon transmission, photons may fade or decay but new photons cannot be generated [1]. Also, the most likely faults that affect address decoders and word lines in very large scale integration (VLSI) memories and stuck-faults in a serial bus cause asymmetric errors. The common channel model for this type of errors is the Z-channel.

For noisy channels repeated retransmission requests can decrease the throughput efficiency of the system. The method proposed here improves this throughput for feedback error control in asymmetric channels using diversity packet combining. The idea was first introduced by Sindhu in [2] who discussed a scheme that made use of the packets that cause retransmission requests, which are simply discarded in pure and type-I hybrid automatic retransmission request (ARQ) protocols. Such packets can be stored and combined with additional retransmissions of the packet, thus creating a single packet that is likely to be the correct version of the transmitted one.

There are two basic categories of packet combining systems: code-combining systems and diversity combining systems. In code combining systems, the packets are concatenated to form noise-corrupted codewords from increasingly longer and lower rate codes. This is the basis for type-II hybrid ARQ protocols [3]. On the other hand, in diversity-combining systems, the individual symbols from identical copies of a packet are combined to create a packet with more reliable constituent symbols. Most of the discussions on diversity combining systems are based on majority logic decoding [4], [5], or on soft channel outputs [6]. The Z-channel error characteristic provides a simple framework which can improve the performance of an ARQ system without adding much to the hardware complexity of the decoder.

The correspondence begins with introducing the proposed asymmetric error-correction scheme. Then the undetected error probability and the average number of transmissions are determined for unordered codewords under the assumption that there are at most $k-1$ retransmission requests for a given codeword. The case of unlimited number of retransmissions, that is, a codeword is retransmitted upon error detection until accepted, follows immediately. Further, some bounds are proposed for the more general case when some codewords of the asymmetric error detecting (AED) code cover others.

II. Diversity Combining Scheme and Problem Formulation

As already mentioned, instead of discarding the erroneous packets which cause retransmissions requests, they can be saved and combined with the retransmitted ones as in Fig. 1.

Packet combining consists of a bit-by-bit logic OR operation. Assuming only $1 \rightarrow 0$ errors (the $0 \rightarrow 1$ type will require complementing the words prior to the OR operation), note that any bit in error may or may not be corrected, but new errors cannot be created. An example is given in Table I. We assume that $\boldsymbol{x} = 0100111010101$ is transmitted and suffers three bit errors during the initial transmission. Assuming that the first two retransmissions yield the words shown in Table I, the codeword is recovered after these two retransmissions.

In other words, a codeword is transmitted repeatedly over a Z-channel. At the receiving end, the OR of the received copies is stored (we will call this the combined word). When the combined word becomes a codeword, this is passed on, and a new codeword is transmitted. If the passed codeword is different from the one sent, then we have an *undetected error*. This process is illustrated in Fig. 2, where the states T, RQ, and FWD represent word transmission, retransmission request, and next word transmission, respectively. We will further assume that there is a limit k on the number of transmissions of a codeword, that is, if the combined word is not a codeword after k

Manuscript received February 6, 2004; revised October 23, 2004. This work was supported by the National Science Foundation under Grant CCR-0105204 and the Norwegian Research Council.

T. Kløve is with Department of Informatics, University of Bergen, N-5020 Bergen, Norway,

P. Oprisan and B. Bose are with the School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR 97331 USA.

Communicated by M. P. Fossorier, Associate Editor for Coding Techniques.
Digital Object Identifier 10.1109/TIT.2004.842771