

On the speedup required for combined input and output queued switching

Balaji Prabhakar
LIDS, Department of EECS
MIT
Cambridge, Massachusetts
Email balaji@lids.mit.edu

Nick McKeown
Departments of EE and CS
Stanford University
Palo Alto, California
Email nickm@mekong.stanford.edu

Abstract — Architectures based on a non-blocking fabric, such as a crosspoint switch, are attractive for use in high-speed LAN switches, ATM switches and IP routers. These fabrics, coupled with memory bandwidth limitations, dictate that queues be placed at the input of the switch. But it is well known that input-queueing can lead to low throughput, and does not allow the control of latency through the switch. This is in contrast to output-queueing, which maximizes throughput, and permits the accurate control of packet latency through scheduling. We ask the question: Can a switch with combined input and output queueing be designed to *behave identically* to an output-queued switch? In this paper, we prove that if the switch uses virtual output queueing, and has an internal speedup of just four, it is possible for it to behave identically to an output queued switch, regardless of the nature of the arriving traffic. Our proof is based on a novel scheduling algorithm, known as *Most Urgent Cell First*. This result makes possible switches that perform as if they were output-queued, yet use memories that run more slowly.

I. INTRODUCTION

Many commercial switches and routers today employ output-queueing. When a packet arrives at an output-queued (OQ) switch, it is immediately placed in a queue that is dedicated to its outgoing line, where it will wait until departing from the switch. This approach is known to maximize the throughput of the switch: so long as no input or output is oversubscribed, the switch is able to support the traffic and the occupancies of queues remain bounded.

The use of a separate queue for each output means that flows of packets for different outputs are kept separate, and cannot interfere with each other. By carefully scheduling the time a packet is placed onto the outgoing line, a switch or router can control the packet's latency, and hence provide quality-of-service (QoS) guarantees. But output queueing is impractical for switches with high line rates, or with a large number of ports. The fabric and memory of an $N \times N$ switch must run N times as fast as the line rate. Unfortunately, at high line rates, memories with sufficient bandwidth are simply not available.

On the other hand, the fabric and the memory of an input queued (IQ) switch need only run as fast as the line rate. This makes input queueing very appealing for switches with fast line rates, or with a large number of ports. For a given speed of memory, it is possible to build a faster switch; or for a given speed switch, it is possible to use slower, lower-cost memory devices.

But, the main problem of IQ switching is head-of-line (HOL) blocking, whose effect on throughput can be severe. It is well-known that if each input maintains a single FIFO, then HOL blocking can limit the throughput to just 58.6%.

One method that has been proposed to reduce HOL blocking is to increase the "speedup" of a switch. A switch with a speedup of S can remove up to S packets from each input and deliver up to S packets to each output within a time slot, where a time slot is the time between packet arrivals at input ports. Hence, an OQ switch has a speedup of N while an IQ switch has a speedup of 1. For values of S between 1 and N packets need to be buffered at the inputs before switching as well as at the outputs after switching. We call this architecture a combined input and output queued (CIOQ) switch. Both analytical and simulation studies of a CIOQ switch which maintains a single FIFO at each input have been conducted for various values of the speedup. A common conclusion of these studies is with $S = 4$ or 5 one can achieve about 99% throughput when arrivals are independent and identically distributed at each input and the distribution of packet destinations is uniform across the outputs.

In practice, we are not only interested in the throughput of a switch, but also in the latency of individual packets. This is particularly important if a switch or router is to offer QoS guarantees.

II. PROBLEM FORMULATION AND RESULTS

Previous studies of CIOQ switches make no guarantees about the delay of an individual packet, but only about average delay and throughput. We are interested in the delay of individual packets. Hence our result subsumes previous work, and our approach is quite different. Rather than find values of speedup that work well on average, or with simplistic traffic models, we find the minimum speedup such that a CIOQ switch behaves *identically* to an OQ switch for *all* types of traffic. Here, "behave identically" means that, when the *same* inputs are applied to both the OQ switch and to the CIOQ switch, the corresponding output processes from the two switches are completely indistinguishable. Two processes are indistinguishable if and only if their packet sequences are identical – both in terms of packet-occurrence times and packet identities. Further, we place absolutely no restrictions on arrivals.

In other words, our formulation allows us to build a CIOQ switch that performs exactly the same as an OQ switch, using memory devices operating more slowly. Specifically, we prove that for a CIOQ switch to mimic an OQ switch it is sufficient that the speedup equals four.