

Asynchronous vs synchronous input-queued switches

Original

Asynchronous vs synchronous input-queued switches / Bianco, Andrea; Cuda, Davide; Giaccone, Paolo; Neri, Fabio. - STAMPA. - (2010). (IEEE GLOBECOM 2010 (Next Generation Networking Symposium) Miami, FL December 2010) [10.1109/GLOCOM.2010.5683642].

Availability:

This version is available at: 11583/2375044 since:

Publisher:

IEEE

Published

DOI:10.1109/GLOCOM.2010.5683642

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

Asynchronous vs Synchronous Input-Queued Switches

Andrea Bianco, Davide Cuda, Paolo Giaccone, Fabio Neri
Dipartimento di Elettronica, Politecnico di Torino (Italy)

Abstract—Input-queued (IQ) switches are one of the reference architectures for the design of high-speed packet switches. Classical results in this field refer to the scenario in which the whole switch transfers the packets in a synchronous fashion, in phase with a sequence of fixed-size timeslots, selected to transport a minimum-size packet. However, for switches with large number of ports and high bandwidth, maintaining an accurate global synchronization and transferring all the packets in a synchronous fashion is becoming more and more challenging. Furthermore, variable size packets (as in the traffic present in the Internet) require rather complex segmentation and reassembly processes and some switching capacity is lost due to partial filling of timeslots. Thus, we consider a switch able to natively transfer packets in an asynchronous fashion thanks to a simple and distributed packet scheduler. We investigate the performance of asynchronous IQ switches and show that, despite their simplicity, their performance are comparable or even better than those of synchronous switches. These partly unexpected results highlight the great potentiality of the asynchronous approach for the design of high-performance switches.

I. INTRODUCTION

A vast technical literature exists on input-queued (IQ) switches, which are considered to be a winning choice to achieve high-end performance due to their limited technological requirements. Basically, IQ switches trade a lower internal data transfer capacity (i.e., very limited speed-ups of the switching fabric) for a larger complexity in switch control and scheduling algorithms. Classical results in this field mostly refer to a synchronous and slotted operation of the entire switch, so that incoming variable-size Ethernet or IP packets must be segmented at switch inputs in fixed-size data units, which are transferred to outputs, where they are re-assembled in variable-size legacy packets. Beyond the complexity/efficiency costs of this segmentation/reassembly process, in the real world the implementation of a fully synchronous large packet switch is not a trivial task. Indeed, the difficulty in keeping under control the alignment of the clock reference signals in different parts of the (often multi-rack) switch, and the different propagation delays in boards, backplanes and interconnection ribbons (often in presence of high-parallelism buses)¹, forced several manufacturers to have independent clocking domains in different subsystems of the switch, leading to an asynchronous operation.

Unfortunately the technical literature has largely neglected this situation, and concentrated the attention on slotted, cell based, switches. Several unfunded beliefs circulate on the

inefficiency of asynchronous IQ switches. In this paper we collect known results (that are referenced in the text when needed) and derive new results, showing that the asynchronous IQ switch operation does not introduce significant detriment to performance. In some scenarios, we show that the asynchronous operation can lead to higher throughput, or to simpler scheduling. Even when the asynchronous operation leads to performance losses in comparison with synchronous cell switching, these losses are limited, and may be smaller than the losses due to the segmentation/reassembly overheads mentioned above.

II. SYSTEM MODEL

We assume that packets are switched across a $N \times N$ bufferless non-blocking switching fabric, e.g. a crossbar. Furthermore, no speedup is available, i.e. the transfer rate at the inputs and at the outputs of the switching fabric is the same as the external line rate of the switch. Packets arrive at the input ports of the switch, where they are processed. Since no speedup is available, input queues are present to cope with output contentions, i.e., when several packets from different inputs are directed to the same output. Queues at the outputs are not needed, unless for reassembly purposes. A scheduler solves output contention between head-of-line (HoL) packets by choosing a set of packets that can be transferred satisfying the physical constraints of the switching fabric: at most one packet can be transferred from each input and to each output at the same time. A feasible configuration of the switching fabric is referred as a “matching” in the bipartite graph whose left-side nodes correspond to the inputs and the right-side nodes correspond to the outputs. We assume that packets have variable size.

A. Switch architecture

The first architecture is an input-queued (IQ) switch with a single FIFO queue per input. This is an architecture quite common in real implementations for its simplicity: N queues are present in the whole switch and the scheduling decision is relatively simple and can be distributed among the outputs. Its main drawback is that it suffers from the HoL blocking problem that limits the maximum achievable throughput. Finally, we consider IQ switches with VOQ (Virtual Output Queueing), i.e. with one FIFO queue for each input-output pair. This architecture is also commonly implemented, because it avoids the throughput degradation due to HoL blocking, even if at the cost of managing N^2 queues. To obtain high throughput,

¹Consider for example that on a 1 Gbps line, each bit lasts 1 ns, corresponding to 20 cm in space used on the line. Hence, the time alignment is lost for two bits traveling over paths differing 20 cm in length.

scheduling decision requires coordination between inputs and outputs, thus increasing scheduler complexity.

B. Synchronous (SYN) Switching

In SYN switches, all packet transfers across the switching fabric occur at the same time and last exactly for a timeslot. The timeslot duration is simply defined for networks in which all the packets have the same size. In the case of variable-size packets, as in the Internet traffic, the packets should be chopped into fixed sized packets (named *cells*), whose duration is the timeslot. These cells are individually switched across the switching fabric and then reassembled at the outputs to obtain the whole packet, ready to be sent to the output interface. The timeslot duration (or, equivalently, the cell size) requires careful design to minimize the throughput loss due to cell granularity; we can show by simple evaluation on two real packet traffic traces [1], [2] captured on the network of FastWeb, one of the largest Italian ISP, that around 10% of the bandwidth is lost even if the cell size is optimized for a single trace, due to partial filling of slots and extra-overheads.

In SYN switches, a *cell-mode* (CM) scheduler is not aware of the packet to which each individual cell belongs to. Thus, at the output of the switching fabric packet interleaving may occur and some reassembly queues are needed at the outputs. Furthermore, partial losses of the packet content may occur. On the contrary, *packet-mode* (PM) schedulers [3] take into account that the cells are originated by packets; indeed, PM schedulers force to transfer all the cells belonging to the same packet in consecutive timeslots. As a consequence, no packet interleaving is allowed at the outputs. Note that any cell-based scheduler can easily support PM; for example, in the case of VOQ-IQ switches, the scheduler removes from the matching computation all the edges that are currently transferring the cells of a packet.

C. Asynchronous (ASY) Switching

In ASY switches, the initial time at which a packet is transferred across the switching fabric occurs independently of the other ports. When the packet has been completely transmitted to an output, a new matching can be computed between the inputs and outputs that are currently free. The scheduling decision is similar to PM in SYN switches, because packet interleaving is not allowed. However, packet transfer through the switching fabric occurs asynchronously.

Under our assumptions, the scheduling decision is very simple to be implemented, because at most one packet can finish its transmission across the switching fabric at a given time. When the packet has been completely transmitted from an input to an output, the scheduling decision can be taken at the corresponding output, independently of all the other outputs.

As a drawback, due to the asynchronous nature of packet transmissions, when a packet has been fully transmitted, the input (output) can be matched to a different output (input) only if there exist at least another non-busy output (input). This fact limits the degree of freedoms in changing the

matching, especially for high load. Hence, VOQs can suffer from temporary starvation, which increases the average delay experienced by packets.

D. Methodology

In the following sections we consider switches with different queuing systems and discuss their performance. We present some theoretical results, validated by simulation.

Inputs arrival processes generate packets according to two states: during ON-state the input generates a single packet, whereas during OFF-state the input is idling. Both ON and OFF periods are i.i.d.. Let L be random variable corresponding to the packet length (i.e., ON-period), measured in bits/packet; let μ_L be the average packet length $E[L] = \mu_L$, and α be the variation coefficient of L . The packet length distribution for the ASY (SYN) switch is exponential (geometric) for $\alpha = 1$, hypo-exponential i.e. gamma (hypo-geometric) for $\alpha < 1$ and hyper-exponential (hyper-geometric) for $\alpha > 1$. It can be shown that for any real distribution of packet size, $\alpha \leq 2.32$.

Idle OFF-periods are geometrically distributed for the SYN switch, and exponentially distributed for the ASY switch, and their average is set to obtain the required average input load ρ . Let λ_{ij} be the packet arrival rate from input i to output j , measured in packets/s; the traffic matrix is defined as $\Lambda = [\lambda_{ij}]$. Let c be the link capacity, measured in bit/s. The traffic is said to be admissible is neither an input or an output is overloaded:

$$\sum_{i=1}^N \lambda_{ij} \mu_L \leq c \quad \sum_{j=1}^N \lambda_{ij} \mu_L \leq c$$

We will consider always *admissible traffic* in the following. The traffic is said to be uniform if $\lambda_{ij} = \rho/\mu_L$ for any i, j . The switch is said to be in saturation whenever $\rho = 1$.

Note that for SYN switches we assume, optimistically, that no bandwidth is wasted due to partial cell filling and to additional overhead.

III. INPUT-QUEUED SWITCHES WITH SINGLE QUEUE

We consider a SYN switch with a single queue per input and controlled by a random scheduler: among a set of cells at the head of the queues (referred as head-of-line (HoL) cells) directed to the same output, i.e., a set of cells creating output contention, the output scheduler chooses one cell at random. [4] showed that the maximum throughput, under uniform traffic and Bernoulli i.i.d. arrivals, is given by $2 - \sqrt{2} \approx 58\%$; because of the HoL blocking inherent to the queuing structure, this architecture is not able to achieve 100% throughput.

In an ASY switch, when an output finishes to serve a packet, the output scheduler chooses one packet at random among the HoL packets directed to it; if no packet is available, the output scheduler waits for the first HoL packet directed to it. The throughput of such architecture was studied in [5], [6], in the case of Poisson or long-range-dependent arrivals process, for exponential packet lengths and under a generic traffic matrix. The adopted methodology is derived from [4], and consists of mapping the switch dynamics into a particular

closed queueing network. We will now extend such approach to generally distributed packet lengths.

The maximum throughput can be estimated in saturation by considering the system of *virtual queues* corresponding to the HoL packets, waiting or being in service. Such virtual system is built on N queues, one for each output, and N jobs, one for each possible HoL packet. By construction, the size of virtual queue j corresponds to the number of HoL packets directed to output j . Whenever an input ends the transmission across the switching fabric of a packet directed to output j , virtual queue j finishes to serve a job. Since the switch is in saturation, a new packet, behind the HoL packet just served, reaches the HoL, and a new job arrives at the virtual queue corresponding to its destination output. Note that the queueing network of the virtual queues is closed, with N jobs, because at each service corresponds a new arrival. In summary, the arrival and departure processes in the virtual system correspond to ends of transmissions of the real switch system. A bijective relation exists always between any of the HoL packets and the jobs; the service duration of a job in the virtual system corresponds to the transmission time of the corresponding packet.

Since the traffic is uniform, we can consider a generic output and let X be the corresponding virtual queue size (i.e., the number of HoL packets directed to this output). By definition $X \in [0, N]$ and $E[X] = 1$ because the total number of HoL packets is N . The dynamics of X can be described by the occupancy of a continuous time $M/G/1$ queue in which the service time is equal to the packet length L , which is a random variable. Since traffic is uniformly distributed among outputs, the arrivals at the queue are given by the superposition of N independent and identically distributed renewal processes, each with rate λ/N . Now, thanks to the superposition limit theorem [7], for $N \rightarrow \infty$, the arrival process becomes Poisson at rate λ . Note that, very similarly, [4] showed that in a SYN switch X follows the dynamics of a discrete time $M/D/1$ queue where the number of jobs arriving during a generic timeslot follows a Poisson distribution, given that $N \rightarrow \infty$.

Now we can exploit the known result for the $M/G/1$ queue:

$$E[X] = \rho + \frac{\lambda^2 E[L^2]}{2(1-\rho)} = \rho + \frac{\rho^2(1+\alpha^2)}{2(1-\rho)}$$

Since $E[X] = 1$, we obtain:

$$(\alpha^2 - 1)\rho^2 + 4\rho - 2 = 0 \quad (1)$$

For $\alpha = 1$, corresponding to the exponential distribution of the packet sizes, the maximum throughput is $\rho = 0.5$; this has been already shown in [5], [6] but also in [8] for correlated arrivals (bursts) of fixed size cells. Solving (1) for $\alpha \neq 1$, we can compute the maximum throughput and prove this new result:

Claim 1: Under uniform ON-OFF traffic, a single-FIFO ASY switch achieves a maximum throughput T_{ASY} equal to 0.5 for $\alpha = 1$; for $\alpha \neq 1$

$$T_{ASY} = \frac{\sqrt{2\alpha^2 + 2} - 2}{\alpha^2 - 1} \quad (2)$$

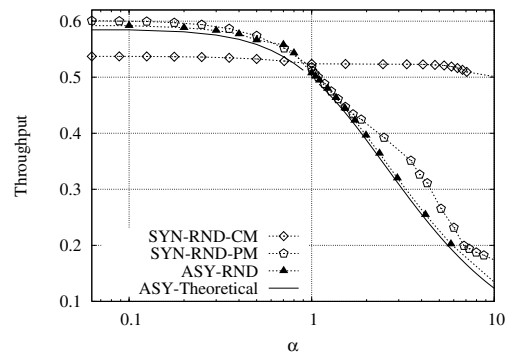


Fig. 1. Maximum throughput for single-FIFO ASY and SYN switches under uniform traffic for a 100×100 switch. In real networks $\alpha < 2.32$.

A. Simulation Results

Fig. 1 shows the maximum throughput in function of α . In the case of ASY switches, we report the results obtained by considering a random output scheduler (ASY-RND) and the theoretical curve obtained by (2), which appears to be very accurate. In the case of SYN switches, we considered two random schedulers (SYN-RND-CM, SYN-RND-PM) operating in CM and PM respectively.

In the case $\alpha \rightarrow 0$, i.e. all the packet sizes are the same, the maximum throughput for a ASY switch is $\sqrt{2} - 2 \approx 58\%$ as in a SYN architecture. This is not surprising, since even if the arrivals in a ASY switch are time-continuous, the queueing effect tends to synchronize the services among all the outputs and, after a transient period, the system behaves like a SYN switch in saturation. When $\alpha \rightarrow \infty$, the maximum throughput goes to zero. This theoretical result shows that the throughput degradation due to ASY mode can be very large, as expected, but this happens only when α is very large: only for $\alpha > 2$, the throughput remains smaller than 30%. For any realistic values of α , the estimated throughput is larger than 0.4.

Performance of the SYN switch in CM are almost constant with α . On the other hand, ASY-RND and SYN-RND-PM behave almost similarly, presenting the same throughput degradation as α increases.

These results show that, depending on the traffic conditions, an ASY switch can be better or worse than a SYN switch, and, in the worst case, the throughput degradation due to the ASY behavior is limited.

IV. INPUT-QUEUED SWITCH WITH VOQ

We now consider an input-queued (IQ) switch with one FIFO queue for each input-output pair.

In a SYN switch, the scheduler transfers a non-conflicting set of HoL cells by computing a matching between the inputs and the outputs. Each VOQ is associated with a weight equal to the number of enqueued cells. The maximum weight matching (MWM) algorithm chooses, among all possible matchings, the one with the maximum weight. It is well known [9] that MWM is able to achieve 100% throughput under any admissible Bernoulli i.i.d. traffic. This result has been notably extended to any admissible traffic process in which the cumulative number

of cells arrived follows the strong law of large numbers; this means that MWM is optimal also when the traffic is correlated, as in the case of cell arrivals due to the packetization process.

Many extensions/variations of the MWM have been proposed to achieve the maximum throughput [10], [11] in a SYN switch operating in CM. In summary, [3] showed that: i) the MWM operating in PM (PM-MWM) achieves 100% throughput under Bernoulli i.i.d. packet generation; ii) the delay performance of PM can be better or worse than cell-based schedulers depending on the variation coefficient α of the packet size distribution (this result is in contrast with the common but wrong belief that PM can only increase delays due to packet starvation); iii) non-optimal PM schedulers behave very closely to optimal schedulers (since less degrees of freedom in the matching choice require less iterations). These results were generalized in [12], where it was shown that, under regenerative traffics, PM-MWM is throughput optimal.

In an ASY switch, the scheduler has few degrees of freedom in choosing the packets, similarly to PM schedulers in SYN switches. Since packet arrivals are time-continuous, all the scheduling choices are concentrated at output ports. Whenever an output finishes to transmit a packet (this event happens asynchronously with respect to all the other outputs), only two events can occur. Either there are other queued packets (at most N) to choose from, or no packet is present and the first packet arriving at the output will be served as soon as it arrives. Note that each output operates asynchronously and independently of all other outputs, allowing fully distributed scheduling algorithms, in which the output scheduling complexity is $O(N)$. Finally, [13] discusses in details the asynchronous implementation of the classical iSLIP [14] scheduling algorithm. It highlights also that some traffic patterns may cause starvation problems.

A. Simulation Results

The simulation study aims at comparing the performance of scheduling algorithms for SYN switches and ASY switches. In the case of SYN switches, we considered iSLIP [14] and MWM, running in cell-mode (CM) and in packet-mode (PM); these algorithms are denoted as SYN-iSLIP-CM, SYN-iSLIP-PM, SYN-MWM-CM and SYN-MWM-PM. In the case of ASY switches, we considered the following algorithms running at each output: round-robin (ASY-RR), random (ASY-RND) and longest queue first (ASY-LQF). Note that ASY-LQF is similar to SYN-MWM-PM.

In addition to uniform traffic, we consider *bidiagonal traffic*, defined as $\lambda_{ii} = 2\rho/3$, $\lambda_{i|i+1|N} = \rho/3$, for any $1 \leq i \leq N$ (being $|x|_N$ equal to $[(x-1) \bmod N] + 1$). This traffic scenario is well known in the literature for SYN switches, since it highlights performance losses due to non-optimal scheduling algorithms.

Packet sizes L (in bytes) were chosen according to a trimodal distribution, approximating the one observed in the FastWeb traces [1], [2]: $P\{L = 40\} = 0.56$, $P\{L = 240\} = 0.20$, $P\{L = 1280\} = 0.24$. The considered switch is 16×16 . Port rate c is 10 Gbps. In the case of SYN switches, the

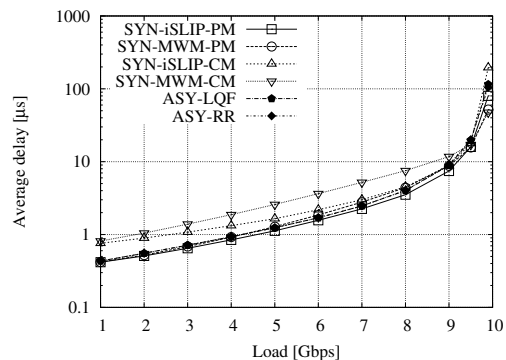


Fig. 2. Average delay under uniform traffic for VOQ switches with $N = 16$.

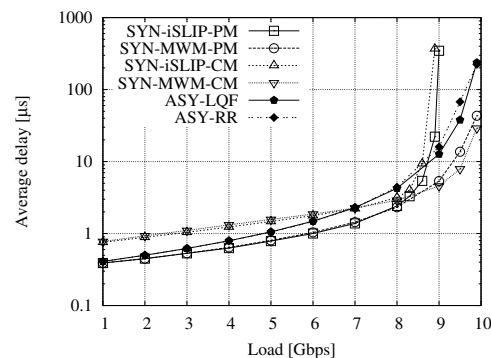


Fig. 3. Average delay under bidiagonal traffic for VOQ switches with $N = 16$.

timeslot is equal to the minimum packet size, 40 bytes (0.32 μs). The queue size is equal to 10,000 cells (400,000 bytes) for the SYN (ASY) switch. The investigated performance metrics are the average throughput and the average packet delay, versus the offered load in Gbps. Note that a load equal to 10 Gbps corresponds to a fully loaded switch for which the average delay is bounded by the finite queue size. Statistics were obtained, after removing the transient period, with an accuracy of 2% for a 95% confidence interval.

Fig. 2 shows the average packet delay under uniform traffic and trimodal packet size distribution. All the algorithms behave similarly, achieving the maximum throughput. In SYN switches, CM shows slightly larger delays due to the packet interleaving at each output, as discussed in [3]. Furthermore, in CM the queue length metrics adopted by MWM tends to interleave packets more than the simple round robin of iSLIP. Indeed, assuming equal size packets, in the case of round robin a packet can be interleaved with at most $2(N-1)$ other packets, whereas for a longest queue this value is unbounded. For small packet size, CM and PM schedulers would behave similarly under uniform traffic, because the packet interleaving is negligible with respect to the packet delay (results not shown for lack of space).

Fig. 3 shows the performance achieved under bidiagonal traffic and trimodal packet size distribution. This traffic scenario is very critical to be scheduled because of the limited degrees of freedoms in choosing the matchings: it can be

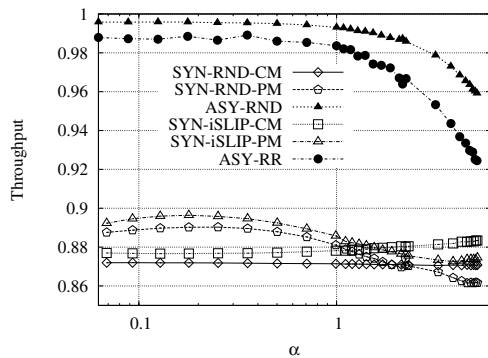


Fig. 4. Maximum throughput under bidiagonal traffic vs packet length variation coefficient, with $N = 16$.

shown that, to achieve the maximum throughput, the scheduler must cycle among only two complete matchings M_1 and M_2 , corresponding to the two non-empty diagonals of the traffic matrix. Whenever the scheduler chooses a matching different from M_1 and M_2 , the matching size is smaller than N , and a throughput loss is experienced. The greedy choice of all algorithms, except for SYN-MWM-CM and SYN-MWM-PM, lead to matchings which are a mix of M_1 with M_2 ; for this reason, this traffic pattern has been considered as a challenging scenario to assess the performance of non-optimal algorithms for SYN switches. According to Fig. 3, for SYN switches, MWM achieves 100% throughput and outperforms iSLIP, which achieves only a throughput equal to 0.88 in both CM and PM; note that we omitted all the points for load larger than 9 Gbps due to the large packet losses. On the contrary, ASY-LQF and ASY-RR are able to achieve 100% throughput, even if at the cost of large delays due to temporary starvation, but outperforming the heuristic scheduling algorithms in SYN switches. Similar performances are observed when packets have a constant size.

The good performance of ASY-LQF and ASY-RR is surprising and due to the fact that non-complete matchings are unstable in ASY switches, and tend to be progressively changed into a complete matching, which are kept until the queues empty. Furthermore, the scheduler is able to change from M_1 to M_2 (and viceversa) in a negligible time, avoiding throughput losses. Indeed, because of the limited degrees of freedom in the output scheduler, when a single queue becomes empty, a “wave” of changes in the matching is generated and it propagates across adjacent ports, driving the complete change of the matching in a very short time, achieving maximum throughput.

Fig. 4 investigates the effect of α under bidiagonal traffic. We considered the RND and RR schedulers in ASY switches and the corresponding schedulers in SYN switches, in both CM and PM versions. The main message is that ASY switches are always outperforming SYN switches, for any $\alpha \leq 5$. It is also worth to note that ASY, for fixed sized packets ($\alpha = 0$), achieves almost the maximum throughput. The small throughput loss is, in any case, smaller than the average 10% loss due to packetization (see Sect. II-B). Looking at the de-

tailed behaviors of the different algorithms, consistently with Fig. 1, obtained with a single FIFO per input, the throughput decreases for larger α . Furthermore, the same qualitative behavior is affecting round-robin based algorithms: SYN-iSLIP-CM, SYN-iSLIP-PM and ASY-RR for VOQ switches behave similarly to SYN-CM, SYN-PM and ASY-RND for single-FIFO switch. Finally, random-based scheduling algorithms are slightly outperforming round-robin based ones.

V. CONCLUSIONS

We compared the performance of SYN and ASY switches for variable-size packet arrivals, considering IQ switches with a single FIFO queue per input and IQ switches with VOQs. In the first case, ASY performance are comparable to SYN performance with PM schedulers, and better than SYN performance with CM schedulers. Finally, for VOQ architectures, ASY switches outperform SYN switches for bidiagonal traffic, and provide better or comparable delay performance for uniform traffic, even when using random schedulers, that are much simpler than the iSLIP or MWM schedulers, normally considered for SYN switches.

REFERENCES

- [1] R. Birke, M. Mellia, M. Petracca, and D. Rossi, “Understanding voip from backbone measurements,” in *INFOCOM*, 2007, pp. 2027–2035.
- [2] K. Imran, M. Mellia, and M. Meo, “Measurements of multicast television over ip,” in *LANMAN*, 2007, pp. 2027–2035.
- [3] M. Ajmone Marsan, A. Bianco, P. Giaccone, E. Leonardi, and F. Neri, “Packet-mode scheduling in input-queued cell-based switches,” *Networking, IEEE/ACM Transactions on*, vol. 10, no. 5, pp. 666–678, 2002.
- [4] M. Karol, M. Hluchyj, and S. Morgan, “Input versus output queuing on a space-division packet switch,” *Communications, IEEE Transactions on*, vol. 35, no. 12, pp. 1347–1356, Dec 1987.
- [5] S. Fuhrmann, “Performance of a packet switch with crossbar architecture,” *Communications, IEEE Transactions on*, vol. 41, no. 3, pp. 486–491, Mar 1993.
- [6] D. Manjunath and B. Sikdar, “Variable length packet switches: Delay analysis of crossbar switches under poisson and self similar traffic,” in *INFOCOM*, 2000, pp. 1055–1064.
- [7] K. Srinam and W. Whitt, “Characterizing superposition arrival processes in packet multiplexers for voice and data,” *Selected Areas in Communications, IEEE Journal on*, vol. 4, no. 6, Sep 1986.
- [8] S.-Q. Li, “Performance of a nonblocking space-division packet switch with correlated input traffic,” *Communications, IEEE Transactions on*, vol. 4, no. 1, Jan 1992.
- [9] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand, “Achieving 100% throughput in an input-queued switch,” *Communications, IEEE Transactions on*, vol. 47, no. 8, pp. 1260–1267, Aug 1999.
- [10] L. Tassiulas, “Linear complexity algorithms for maximum throughput in radio networks and input queued switches,” in *INFOCOM*, 1998, pp. 533–539.
- [11] P. Giaccone, B. Prabhakar, and D. Shah, “Towards simple, high-performance schedulers for high-aggregate bandwidth switches,” in *INFOCOM*, 2002.
- [12] Y. Ganjali, A. Keshavarzian, and D. Shah, “Cell switching versus packet switching in input-queued switches,” *Networking, IEEE/ACM Transactions on*, vol. 13, no. 4, pp. 782–789, 2005.
- [13] G. Passas and M. Katevenis, “Asynchronous operation of bufferless crossbars,” in *HPSR*, June 2007, pp. 1–6.
- [14] N. McKeown, “The islip scheduling algorithm for input-queued switches,” *Networking, IEEE/ACM Transactions on*, vol. 7, no. 2, pp. 188–201, 1999.

Acknowledgments: This work was partially supported by the BONE project, a Network of Excellence funded by the European Commission within the 7th Framework Programme.