POLITECNICO DI TORINO
Repository ISTITUZIONALE

Asynchronous vs Synchronous Input-Queued Switches

Original
Asynchronous vs Synchronous Input-Queued Switches / Bianco, Andrea; Cuda, Davide; Giaccone, Paolo. - In: COMPUTER COMMUNICATIONS. - ISSN 0140-3664. - STAMPA. - 53:(2014), pp. 52-61. [10.1016/j.comcom.2014.07.007]

Availability:
This version is available at: 11583/2556158 since:

Publisher:
Elsevier

Published
DOI:10.1016/j.comcom.2014.07.007

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)

12 October 2024

Asynchronous vs Synchronous Input-Queued Switches

Andrea Bianco, Davide Cuda, Paolo Giaccone
andrea.bianco@polito.it, davide.cuda@polito.it, paolo.giaccone@polito.it
Dipartimento di Elettronica e Telecomunicazioni
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, tailored 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 wasted 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 with different queueing architectures (one queue per input and one queue for input-output pairs) and show that, despite their simplicity, their performance is comparable or even better than those of synchronous switches. We highlight the peculiar role of the variation coefficient of the packet length. Finally, for synchronous switches we evaluate the actual bandwidth overhead due to packet segmentation, by considering a large set of traffic traces covering the period 2008-2013. We show that an impressive amount of bandwidth (up to 30%) can be lost due to segmentation, even if the internal cell size is optimally chosen.

These results demonstrate the potential interest of the asynchronous approach in the design of high-performance switches.

1. Introduction

A vast technical literature exists on input-queued (IQ) switches, which are considered as the reference architectures 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 speedup of the switching fabric) for a larger complexity in switch control and scheduling algorithms. Classical

results in this field mostly refer to a synchronous slotted operation of the entire switch; as a consequence, incoming variable-size Ethernet or IP packets must be segmented at switch inputs in fixed-size data units (denoted as cells), which are transferred to outputs, where they are re-assembled as variable-size packets. Beyond the efficiency costs of this segmentation/reassembly process, the real implementation of a fully synchronous large packet switch is not a trivial task due to the need of introducing complex segmentation/reassembly engines. Furthermore, 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), forced several manufacturers to have independent clocking domains in different subsystems of the switch, leading to an asynchronous operation. Consider that on a 1 Gbps line, each bit lasts 1 ns, corresponding to 20 cm in space on the line. Hence, the time alignment is lost for two bits traveling over paths differing more than 20 cm in length.

In this paper[1] we provide the following contributions. In Sec. 3 we analytically evaluate the maximum throughput for asynchronous IQ switches with one queue for each input, as a function of the packet size distribution. Theorem 1 extends the well known "58% throughout" result obtained by Karol et al. in [2] for synchronous switches and Bernoulli i.i.d. arrivals. Sec. 4 compares the throughput for synchronous and asynchronous IQ switches with Virtual Output Queueing, i.e. with one queue for each input-output pair. In Sec. 5 we investigate the overhead due the packet segmentation in synchronous switches. By considering a large set of real traffic traces, we show that the overhead for synchronous operations can be responsible for performance losses around 5% on average, with a maximum close to 30% in some cases.

In summary, we show that the asynchronous switch operation does not introduce significant performance detriment and, even if the architecture is simpler to implement, it may lead to higher throughput in some scenarios or to comparable performance with much simpler schedulers with respect to synchronous operation. When the asynchronous operation leads to performance losses, these losses are limited, and are smaller than the losses due to the above-mentioned segmentation/reassembly overheads.

2. 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 external switch rate. Packets arrive at switch input ports where they are stored and processed. Since no speedup is available, input queues are needed 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

[1] A preliminary version of this work appeared in [1].

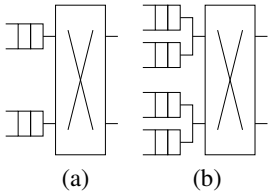

Figure 1: Input queued switching architectures: (a) with a single FIFO per input, (b) with Virtual Output Queuing (VOQ)

reassembly purposes. A scheduler solves output contention between head-of-line (HoL) packets by choosing a set of packets that can be transferred satisfying two constraints: at most one packet can be transferred i) from each input and ii) to each output at the same time. A feasible configuration of the switching fabric is referred to 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. In Sec. 2.3 we will describe in details the traffic models adopted in our work.

We consider two queueing architectures, depicted in Fig. 1. The first one corresponds to an input-queued (IQ) switch with a single FIFO queue per input and a total of N queues. The scheduling decision is relatively simple and can be taken independently at each output. Its main drawback is that it suffers from the HoL blocking problem [2] that limits the maximum achievable throughput. The second architecture is an IQ switches with VOQ (Virtual Output Queueing), i.e. with one FIFO queue for each input-output pair. This architecture avoids the throughput degradation due to HoL blocking, even if at the cost of managing N^2 queues. To obtain high throughput, the scheduler must coordinate the packet transfers between inputs and outputs to avoid conflicts, thus increasing scheduler complexity.

2.1. Synchronous (SYN) switching

In SYN switches, all data transfers across the switching fabric occur at the same time and last exactly one "timeslot". The timeslot duration is equal to the (fixed size) packet transmission time on the fastest link. In the case of variable-size packets, as in the Internet, packets are chopped into fixed sized packets (named *cells*). Cells are individually switched across the switching fabric and then reassembled at the outputs to obtain the original 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, as discussed in Sec. 5.

Schedulers for SYN switches can work in either "cell mode" or "packet mode" [3]. A *cell-mode* (CM) scheduler takes independent decisions at each time slot, without considering the packet each 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 may be originated by a packet segmentation process. Indeed, PM schedulers force the transfer of all the cells belonging to the same packet in consecutive timeslots. As a consequence, no packet interleaving occurs at the outputs.

2.2. Asynchronous (ASY) switching

In ASY switches, packet transfer through the switching fabric occurs asynchronously and packet interleaving is not allowed at the output, similarly to PM scheduling for SYN switches. The ASY model well captures the behavior of asynchronous and variable-length optical packet switching systems [4, 5] because in the optical domain packet alignment may be difficult to achieve [6]. Furthermore, the ASY model approximates electronic packet switching when the minimum unit at which the transfer and the control occurs is much smaller (e.g., word or byte) than the packet/cell. For simplicity, we consider an abstract "pure" ASY model, in which the minimum transfer unit tends to zero. Thus, we assume that the size of each packet is a continuous random variable. At a given time, at most one packet can finish its transmission across the switching fabric or can arrive to an empty port.

The asynchronous behavior limits the degree of freedoms in choosing the matching, which evolves "slowly" with the time. Hence, some queues can suffer from temporary starvation, which may increase the average delay experienced by packets. It is possible to bound the speed at which the matching evolves as follows:

Property 1. *In an ASY switch, at any time the maximum number of newly added edges in a new matching is two.*

Proof. Assume that the transmission of a packet from input i to output j ends at time t^-. This event happens asynchronously with respect to any other event at any input/output, because of the continuous support of the packet size distribution.

Let partition the inputs and the outputs depending on their matching condition. Define I_U the set of unmatched inputs at time t and O_U the set of unmatched outputs at time t. Define as indices $i \in I_U$ and $j \in O_U$. The set of candidate edges that can be added to the current matching is a subset of set Ω_{ij} defined as

$$\Omega_{ij} = \{i \to j\} \cup \{k \to j \text{ for any } k \in I_U\} \cup \{i \to k \text{ for any } k \in O_U\}$$

since the queues corresponding to the edges in Ω_{ij} may be empty. Fig. 2 shows an example of such sets. Note that the edges between unmatched inputs and outputs (except for the edge $i \to j$), i.e. the edges between any input in $I_U \setminus \{i\}$ and any output in $O_U \setminus \{j\}$, cannot exist, otherwise they would have been already matched just before t. Now the new matching computed by the outputs on Ω_{ij} can include only 0, 1 or 2 edges. □

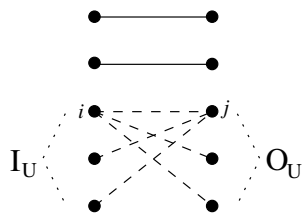

Figure 2: Bipartite graph representing an ASY switch, in which the inputs 1,2 are currently matched to outputs 1,2 respectively, and transmission of a packet from 3 to 3 has just ended. The dashed edges show the set Ω_{33} of all possible candidate edges for a new matching.

However, even if the matching evolves "slowly" according to the previous property, we will see in Sec. 4.3 that the matching is able to adapt very quickly to the queues state, being anyhow optimal most of the time.

2.3. Methodology and traffic models

We will discuss in the following sections the performance of the two presented switching architectures. The theoretical models will be validated by event-driven simulations implemented through OMNeT++ [7]. To better highlight the impact of the traffic and of the scheduling decisions, in the following results for SYN switches we do not consider the throughput degradation due to segmentation, i.e. we assume that no bandwidth is wasted due to partial cell filling and to additional overhead. Since we neglect segmentation, throughput evaluated in SYN switches will be an upper bound on the actual achievable throughput. Note that the effect of segmentation depends completely from the specific packet size distribution and it is cannot evaluated a priori; in Sec. 5 we will evaluate this effect a posteriori for some specific traffic traces observed in the Internet and show that it is not negligible at all.

In our simulations we evaluated both average delays and throughput. We run the simulations until delay statistics (after removing the transient period) reached a relative error less than 3%, measured for a 95% confidence interval, regardless the accuracy of the throughput statistics. The estimation of the confidence interval width was obtained with the batch means approach [8]. At the end of each simulation run, we evaluated also the accuracy of the throughput statistics. In all the reported throughput results, we observed a relative error less than 0.4%, lower than the one achieved by the average delays. In the graphs, we will not report the confidence intervals because they are at most the size of the symbol used for the points of the curves.

In our studies, packet arrivals and lengths are described by renewal processes, in the discrete-time domain for SYN switches and in the continuous-time domain for ASY switches. Both traffic models have been chosen for their analytical tractability. Specifically, both the packet durations and the inter-arrival times between consecutive packets are assumed to be i.i.d. random variables. Inter-arrival times are geometrically distributed for SYN switches and (analogously)

exponentially distributed for ASY switches, and their average is set as to obtain the required input load. These choices allow a fair comparison between the two time domains. Regarding the packet duration, let L be a random variable corresponding to the packet length, measured in bit/packet. Let m_L be the average packet length $E[L] = m_L$, and α be the variation coefficient of L, i.e. $\alpha = \sqrt{\mathrm{Var}(L)}/m_L$. The packet length distribution for the ASY (SYN) switch is assumed to be exponential (geometric) for $\alpha = 1$, hypo-exponential i.e. gamma (hypo-geometric) for $\alpha < 1$ and hyper-exponential (hyper-geometric) for $\alpha > 1$.

Finally, let $\hat{\lambda}_{ij}$ be the packet arrival rate from input i to output j measured in packets/s. If μ is the link capacity, measured in bit/s, the corresponding normalized arrival rate is $\lambda_{ij} = \hat{\lambda}_{ij} m_L/\mu$; we define the corresponding traffic matrix as $\Lambda = [\lambda_{ij}]$. The traffic is said to be *admissible* if neither an input nor an output is overloaded and the following conditions hold:

$$\sum_{i=1}^{N} \lambda_{ij} < 1 \qquad \sum_{j=1}^{N} \lambda_{ij} < 1$$

If we define the normalized load $\rho \geq 0$ as

$$\rho = \max_{k=1,\ldots,N} \Big\{ \sum_{i=1}^{N} \lambda_{ik}, \sum_{j=1}^{N} \lambda_{kj} \Big\}$$

then matrix Λ is admissible if $\rho < 1$. The traffic is said to be *uniform* if $\lambda_{ij} = \rho/N$, for any i and j.

Finally, the switch is said to be *in saturation* when all the queues are always non-empty. This scenario can be obtained under uniform traffic and fully loaded input queues: $\lambda_{ij} = 1/N$, which implies $\rho = 1$.

2.4. Realistic values for the variation coefficient of the packet length

In the next sections we will highlight the crucial role of α in the switch performance. We start by answering to some preliminary questions that naturally arise on α: i) what is the possible range of values, ii) what are the typical values in realistic networks, iii) is it possible to highlight any predictable temporal trend that could be exploited in the switch design phase?

To answer the first question, since any real distribution of packet size is always discrete, with a minimum byte granularity, by using standard probability theory it can be shown [9] that:

Property 2. *Assume that the packet size is arbitrarily distributed between a minimum $l_{\min}$ and maximum $l_{\max}$. The corresponding variation coefficient α is always bounded by*

$$0 \leq \alpha \leq \frac{l_{\max} - l_{\min}}{2\sqrt{l_{\min} l_{\max}}} \tag{1}$$

Note that Property 2 holds *independently* of the specific distribution and permits to bound any realistic α by considering the minimum and the maximum

Link location	Number of traces	Period	Number of IP packets	Measured α
FastWeb POP1	1	09/2006	$31 \cdot 10^6$	0.48
FastWeb POP2	1	07/2006	$7 \cdot 10^6$	1.35
Chicago OC192 A	42	04/2008-07/2013	$22 \cdot 10^9$	
Chicago OC192 B	44	03/2008-07/2013	$112 \cdot 10^9$	0.993
San Jose OC192 A	65	07/2008-07/2013	$137 \cdot 10^9$	(average)
San Jose OC192 B	67	07/2008-07/2013	$109 \cdot 10^9$	

Table 1: Overview of the traces considered to measure the variation coefficient of the packet size

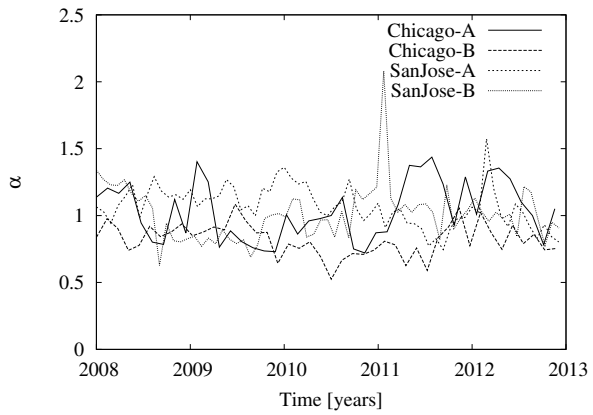

Figure 3: Temporal evolution of α observed in Chicago and San Jose OC192 links

transmission unit (MTU) allowed in the network. In the case of networks based on standard Ethernet, $l_{\min} = 64$ and $l_{\max} = 1518$ bytes, thus $\alpha < 2.34$. In the case of jumbo frames adopted in Gigabit Ethernet, $l_{\max} = 9018$ bytes and $\alpha < 5.9$.

Note that the bound provided in (1) corresponds to a worst-case distribution very unlikely to happen in reality. Indeed, the actual values of α that can be observed in a real network are usually smaller, as shown in our experimental analysis.

To derive values of α from real networks, we start from two Internet traffic traces captured on high-speed core routers located in two POPs in FastWeb network [10, 11], one of the largest Italian ISP. The two traces are quite different: the first one consists mainly of multicast IP-TV traffic distributed to the users, whereas the second one of data and VoIP traffic generated by the users. Through the analysis of the two Internet traffic traces, we obtained $\alpha = 0.48$ and $\alpha = 1.35$ respectively. The smallest value, $\alpha = 0.48$, is due to traffic mainly comprising IP-TV multicast packets, with many packets of the same size.

For a more comprehensive investigation of the values for α observed in real

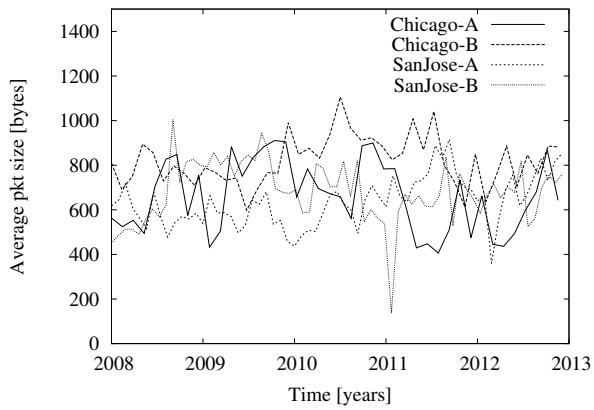

Figure 4: Temporal evolution of the average packet size m_L observed in Chicago and San Jose OC192 links

networks, we consider the publicly available statistics, provided by Caida [12] and obtained by passive measurements, for 4 OC192 links located in Chicago and San Jose (USA). These statistics, described in Table 1, refer to hour-long traces recorded once every month, during a period of 5 years, and to almost 400 billions packets, most of them IPv4. Among all the 218 traces captured in Chicago and San Jose links, we computed an average α equal to 0.993, with the corresponding 95% confidence interval equal to $[0.967, 1.020]$. Fig. 3 shows that the temporal evolution of α for the 5 years does not highlight any predictable temporal trend. In conclusion, in these OC192 links the observed values of α are very close to 1 most of the time, with (rare) maximum values close to the upper bounds computed above. Note that the volume of statistical data in this scenario is much larger than the one considered in the two FastWeb POPs. Unfortunately, we were unable to recover the kind of transported traffic, due to the lack of details on packet-level traces.

Surprisingly, despite the natural intuition that packet sizes should observe some temporal patterns, due to the popularity of particular applications, Fig. 4 shows that also the packet size does not show any particular trend. All these observations corroborate our final claim that α is not easily predictable.

3. Input-queued switches with single FIFO queue

We consider a switch with a single FIFO queue per input, with no internal speedup, controlled by a scheduler performing random choices at outputs.

In a SYN switch, at each timeslot each output chooses one cell at random among the cells at the head of the queues (referred as head-of-line (HoL) cells) directed to it. It is well know [2] that the maximum throughput, under uniform traffic and Bernoulli i.i.d. arrivals, is limited to $2 - \sqrt{2} \approx 58\%$ due to the HoL blocking problem. For correlated arrivals (bursts) of fixed size cells, [13] showed

that the throughput varies between 0.5 and 0.58, depending on the degree of burstyness in the traffic.

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. We prove that:

Theorem 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 $\alpha \neq 1$*

$$T_{ASY} = \frac{\sqrt{2\alpha^2 + 2} - 2}{\alpha^2 - 1} \tag{2}$$

Proof. The maximum throughput can be estimated in saturation conditions ($\rho = 1$) by considering a system of *fictitious queues* fed *only* by to the N HoL packets, waiting or being in service in one of the physical input queues. In such fictitious system, we neglect all the packets that are enqueued in the physical queues but not at HoL. There exist N fictitious queues, one for each output. Let $X_j(t)$ be the size of fictitious queue j at time t, corresponding to the number of HoL packets directed to output j. By construction, $X_j(t) \in \{0, 1, \ldots, N\}$ and

$$\sum_{j=1}^{N} X_j(t) = N \tag{3}$$

for any time t, since the HoL packets are always N. Note that $X_j(t)$ does not represent the occupancy of any physical queue. Whenever input i ends the transmission across the switching fabric of a packet directed to output j, fictitious queue j finishes to serve a HoL packet. Then, since the switch is in saturation, a new packet reaches the HoL of input i queue, and a new HoL packet arrives at the fictitious queue corresponding to the packet destination output. Note that the queueing network of the fictitious queues is closed, because at each service corresponds a new arrival. In summary, the arrival and departure events in the fictitious system correspond to the end of transmissions in the physical queues of the IQ switch. The service duration of a HoL packet in the fictitious system corresponds only to its transmission time, and does not comprise any other constant contribution.

Since the traffic is uniform and the scheduler operates randomly, we can consider a generic output. Let $X(t)$ be the corresponding fictitious queue size (equivalently, the number of HoL packets directed to such output). Thanks to (3), $E[X(t)] = 1$. The dynamics of $X(t)$ can be described by the queue 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. Thanks to the given assumptions, the arrivals at the M/G/1 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 [14], for $N \to \infty$, the arrival process becomes Poisson at rate λ. Note that, very similarly to our scenario, [2] showed that in a SYN switch the occupancy of the fictitious queues

follows the dynamics of a discrete time $M/D/1$ queue where the number of HoL packets arriving during a generic timeslot follows a Poisson distribution, given that $N \to \infty$.

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

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

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

$$(\alpha^2 - 1)\rho^2 + 4\rho - 2 = 0 \tag{4}$$

By solving (4), for $\alpha = 1$, the maximum throughput is $\rho = 0.5$. For $\alpha \neq 1$, we get (2). □

3.1. Related work on ASY switches

The throughput of single-FIFO ASY switches was also studied in [15, 16], in the case of Poisson or long-range-dependent arrivals process, for exponential packet lengths and under a generic traffic matrix. All these results assume exponentially distributed packet sizes and can be seen as a special case of Theorem 1 when $\alpha = 1$, which instead holds for a generic packet length distribution. Notably, [17] has extended the results in [2] to a combined input-output queued switch with backpressure, running asynchronously and fed by fixed-size cells. For the particular case in which the output queues are not available, the considered architecture degenerates into an IQ with single FIFO queues. [17] showed that an ASY switch achieves the same throughput of a SYN switch, with negligible differences in terms of delays. This result is coherent with our findings, for the specific case $\alpha = 0$. The methodology adopted in [17] holds for a generic packet size distribution (even if it was applied only to the fixed-size case). However, differently from Theorem 1, [17] does not show the explicit relation between α and the throughput.

3.2. Simulation results

Fig. 5 compares the maximum throughput for ASY and SYN switches as a function of the variation coefficient of packet size. 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.

When $\alpha \to 0$, i.e. fixed packet sizes, the maximum throughput for an ASY switch is $\sqrt{2} - 2 \approx 58\%$ as in a SYN architecture, coherently with [17]. Even if the arrivals in an ASY switch are time-continuous, this result is not surprising, because in overload: (1) the queues mask the effect of the exact packet arrival times, and (2) during any busy period for an output, the packets are served periodically exactly every packet transmission time, mimicking a synchronous service process. When $\alpha \to \infty$, the maximum throughput goes to zero. This

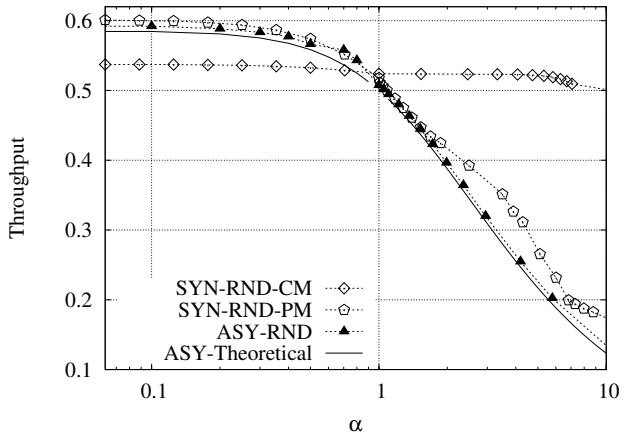

Figure 5: Maximum throughput for single-FIFO ASY and SYN switches under uniform traffic for a 100×100 switch. In Ethernet networks, $\alpha < 2.34$ but measured values vary between 0.48 and 1.35 according to Table 1.

theoretical result shows that the throughput degradation due to ASY mode can be very large, as expected. However, this happens only when α is very large: only for $\alpha > 2$, the throughput remains smaller than 30%. Recalling from Sec. 2.3 that in standard Ethernet networks $\alpha < 2.34$ and thanks to (2), the throughput for ASY switch would be always larger than 0.358. For the measured values of α considered in the traces of Table 1, the throughput will be 0.56 (for $\alpha = 0.48$), 0.50 (for $\alpha = 0.993$) and 0.46 (for $\alpha = 1.35$).

Performance of SYN switch with CM scheduler are almost constant with α. On the other hand, ASY-RND and SYN-RND-PM behave similarly, presenting the same throughput degradation as α increases. When comparing ASY switch with SYN-RND-CM, in the worst case the throughput reduction is 13% in realistic cases (for $\alpha = 1.35$).

In summary, these results show that i) the results from our analytical formula (2) are accurate, ii) depending on the traffic conditions and on the scheduler, an ASY switch can perform better or worse than a SYN switch. For realistic values of α, the throughput degradation due to the ASY behavior, if any, is marginal.

4. Input-queued switch with VOQ

We now consider an input-queued (IQ) switch with no speedup and Virtual Output Queueing (VOQ) architecture, i.e. one FIFO queue for each input-output pair (see (b) in Fig. 1). In a SYN switch, the scheduler transfers a non-conflicting set of HoL cells by computing a matching between the inputs and the outputs.

We will provide a simulation study to compare the performance of scheduling algorithms for SYN switches and ASY switches.

4.1. Related work

Assume that 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 [18] 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 segmentation process.

Many extensions/variations of the MWM have been proposed to achieve the maximum throughput in a SYN switch operating in CM [19, 20]. 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 intuitive idea 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 [21], where it was shown that, under regenerative traffic, PM-MWM is throughput optimal.

Furthermore, [21] showed that, when the traffic is non-regenerative, PM-MWM may not be optimal from the throughput point of view. Indeed, it is possible to devise counterexamples in which the traffic, even if admissible, always forces the selection of the same matching, preventing all the other queues from being served. These counterexamples require a strong correlation among the arrivals at different inputs and, even if not realistic, show the limitations of PM schedulers. To deal with non-regenerative traffic, [21] proposes to freeze the matching after a fixed number s_f of timeslots and wait until one of the corresponding queues empties. After this event, the scheduler computes again the matching on the whole set of queues. This process introduces some throughput loss: Given the maximum packet size $l_{\max}$ in timeslots, then the period in which the matching is kept frozen is $l_{\max}-1$ and the maximum bandwidth loss is equal to $(l_{\max}-1)/(s_f+l_{\max}-1)$. It is sufficient to set $s_f = \lceil (1-\epsilon)l_{\max}/\epsilon - 1/\epsilon \rceil + 1$, to experience a bandwidth loss ϵ that can be compensated by a switching speedup equal to $(1+\epsilon)$ [21]. Finally, [22] discusses in details the asynchronous implementation of the classical iSLIP [23] scheduling algorithm. It highlights also some malicious traffic patterns, non-regenerative according to the definitions in [21], that may cause starvation problems.

In general, an input queued switch can be modeled as a special case of a generic constrained queueing network with asynchronous behavior. In [24] the optimal policy in terms of throughput in generic networks under Poisson arrivals and random packet size has been defined. For the ASY switch, the optimal policy proposed in [24] degenerates in computing the MWM at time t_n (when the weight of the MWM is w_n) and keeping such matching for a time equal to w_n^r, for some $r < 1$. Then the policy waits until all the outputs end their current

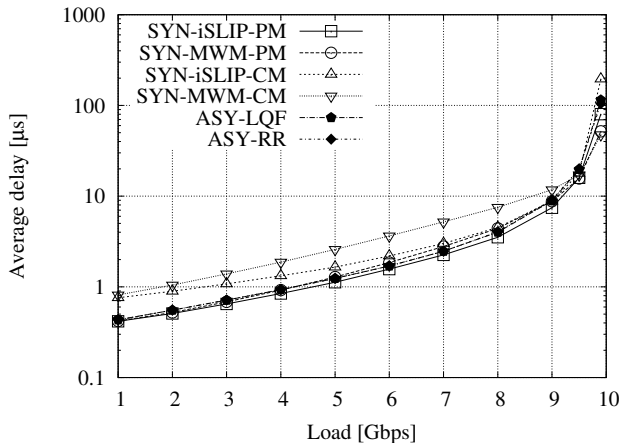

Figure 6: Average packet delay under uniform traffic for a trimodal packet size distribution

packet transmissions, and then recomputes the MWM, very similarly to what has been proposed in [21]. A wide generalization of the asynchronous queueing scenario considered by [24] are the stochastic processing networks, for which an extension of the MWM (called, maximum pressure policy) has been shown to be throughput optimal [25].

Recently, [26] has proposed a distributed randomized policy, based on the knowledge of the input queues occupancies, able to achieve the maximum throughput under Poisson arrivals, in an ASY switch. Note that, differently from our work, [26] does not consider the effect on performance of the variance of the packet size.

4.2. Traffic scenarios

While comparing the performance of ASY and SYN switches, we have used the following traffic scenarios:

- *Uniform (UNI) traffic*: $\lambda_{ij} = \rho/N$, for all i, j; this is the most classic traffic scenario in the literature used as basic benchmark;
- *Bidiagonal (BID) traffic*: $\lambda_{ii} = 2\rho/3$, $\lambda_{i|i+1|_N} = \rho/3$, for any $0 \leq i \leq N-1$, being $|x|_N$ equal to x modulus N; this traffic is well known in the literature for SYN switches, since it highlights performance losses due to non-optimal scheduling algorithms.
- *Logdiagonal (LOG) traffic*: $\lambda_{ij} = 2^{|j-i|_N}/c$, for any $0 \leq i, j \leq N-1$, being c an appropriate normalization constant; also this traffic highlights performance losses due to non-optimal scheduling algorithms.

Packet sizes were chosen according to the following distributions:

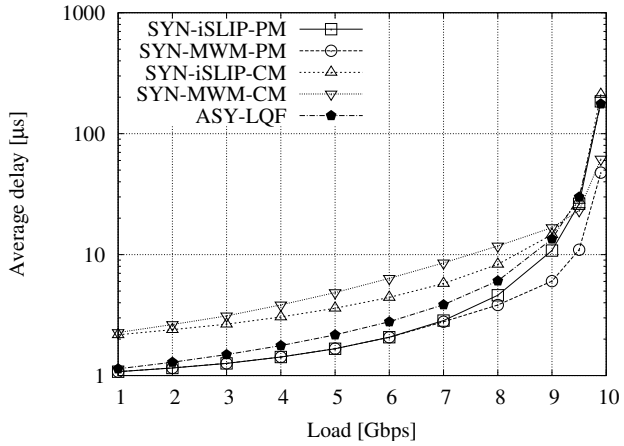

Figure 7: Average packet delay under uniform traffic for large packets ($l_1 = 1500$ bytes)

- *Fixed packet size*: $L = l_1$, being l_1 a constant value. For small packets, we chose $l_1 = 40$ bytes corresponding to the minimum IP packet size, after removing the MAC header. For large packets, we chose $l_1 = 1500$ bytes corresponding to the maximum packet size seen on a Ethernet network.

- *Trimodal packet size*: $P\{L = l_i\} = p_i$ for $i = 1, 2, 3$, being $\{l_i\}$ the set of packet sizes and $\{p_i\}$ the corresponding probabilities. We approximated the distribution observed in the FastWeb POP2 trace with the following parameters: $\{l_i\} = \{40, 576, 1500\}$ bytes and $\{p_i\} = \{0.62, 0.22, 0.16\}$, to match both the variation coefficient ($\alpha = 1.35$) and the three peaks present in the empirical distribution.

4.3. Simulation results

In the case of SYN switches, we considered iSLIP [23] and MWM, both 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.

We report the results for a 16×16 switch; for larger switches we observed similar results. Port rate is set equal to 10 Gbit/s. In the case of SYN switches, the timeslot is equal to the minimum packet size, 40 bytes (32 ns). The queue size is equal to 400,000 bytes. 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.

Fig. 6 shows the average packet delay under uniform traffic and trimodal packet size distribution. All the algorithms behave similarly, achieving the max-

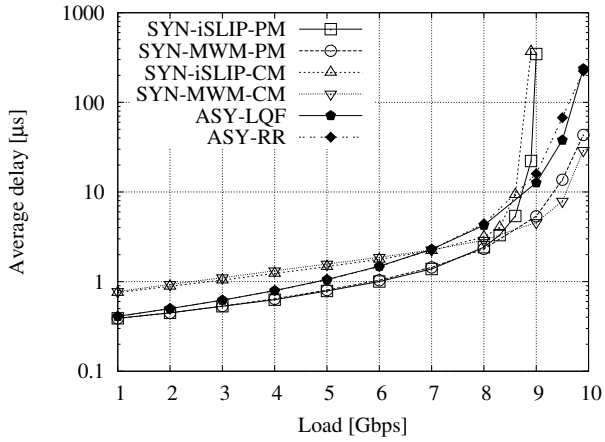

Figure 8: Average packet delay under bidiagonal traffic for the trimodal packet size distribution with $N = 16$.

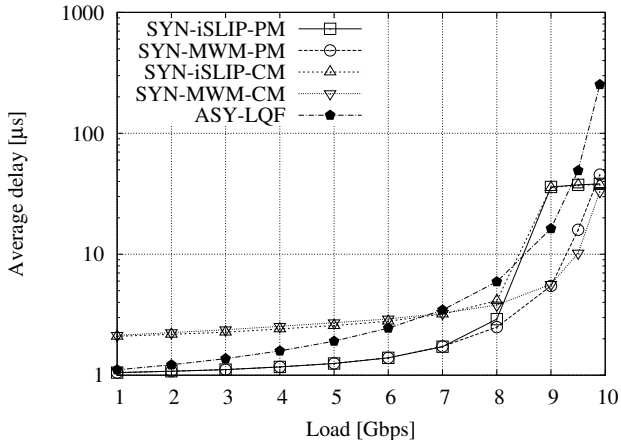

Figure 9: Average packet delay under bidiagonal traffic for large packets ($l_1 = 1500$ bytes)

imum throughput. In SYN switches, CM shows slightly larger delays due to the packet interleaving at each output, as discussed in [3]. To highlight the effect of packet interleaving, in Fig. 7 we show the delays obtained with a traffic scenario with only large packets. 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 behave similarly under uniform traffic, because the packet interleaving is negligible with respect to the packet delay.

Fig. 8 shows the performance achieved under bidiagonal traffic and trimodal

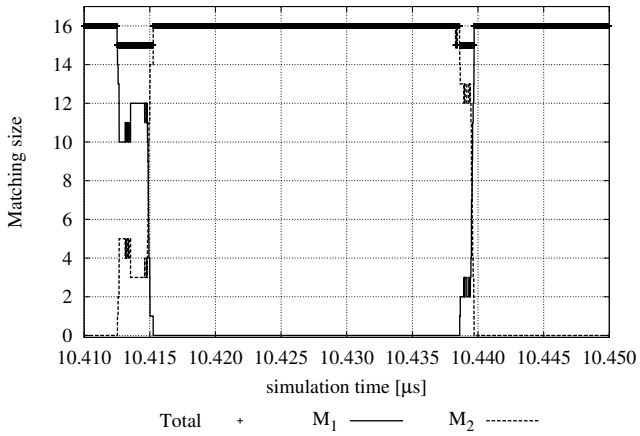

Figure 10: Number of edges in M (denoted as "Total"), in $M \cap M_1$ (denoted as "M_1") and in $M \cap M_2$ (denoted as "M_2") for an ASY switch under bidiagonal traffic and trimodal packet size distribution

Table 2: Maximum throughput achieved by SYN and ASY switches with greedy schedulers for normalized input load $\rho = 0.99$

System	SYN		ASY	
Scheduler	iSLIP-CM	iSLIP-PM	LQF	RR
Uniform traffic	0.99	0.99	0.99	0.99
Bidiagonal traffic	0.88	0.89	0.99	0.99
Logdiagonal traffic	0.84	0.87	0.99	0.98

packet size distribution. This traffic scenario is very critical to be scheduled because of the limited degrees of freedoms in choosing the matchings. 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 that "mix" M_1 with M_2, leading to a throughput degradation. For this reason, this traffic pattern is considered a challenging scenario to assess the performance of non-optimal algorithms for SYN switches.

According to Fig. 8, for SYN switches, MWM achieves 100% throughput and outperforms iSLIP, which achieves a throughput less than 0.9 in CM and PM, as shown in Table 2. 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 fixed

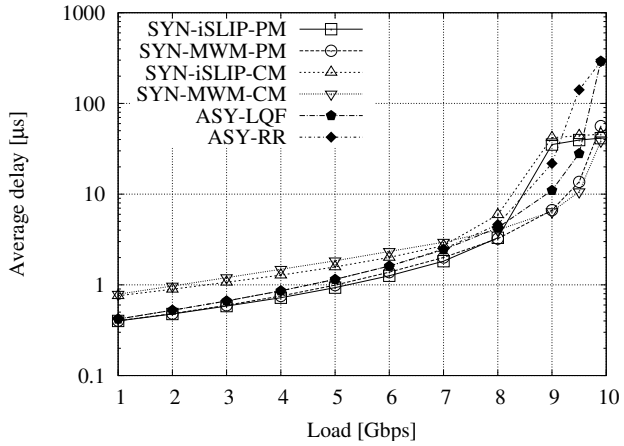

Figure 11: Average packet delay under logdiagonal traffic for the trimodal packet size distribution

size. For example, Fig. 9 shows the delays for large packets.

The good performance of ASY-LQF and ASY-RR can be explained in details as follows. Fig. 10 shows the total number of edges in the matching M (i.e., the instantaneous overall throughput), and the corresponding number of edges shared with M_1 (i.e. $|M \cap M_1|$) and with M_2 (i.e. $|M \cap M_2|$). We know that to achieve 100% throughput the matching size should be N, and this can happen only when the number of edges belonging either to M_1 or to M_2 is also N. This must always occur, except for a negligible time during which the matching can be an hybrid between M_1 and M_2. As shown in Fig. 10, most of the time the switch is configured according to one of the two optimal maximum matchings, and transitions between matchings are fast, even if transition speed is limited by Property 1, shown in Sec. 2.2. Indeed, under bidiagonal traffic, the output has a very limited degree of freedom in choosing the input to be matched to. In ASY switches the output tends to serve a queue exhaustively, since it is able to change the input to which it is matched only when another output becomes free. Now all the queues served by the current matching tend to become empty, whereas the ones served by the other one tends to grow. This fact induces a "chain" reaction among the ports that permit to change quickly the matching from M_1 to M_2.

Fig. 11 shows the delays under logdiagonal traffic. The qualitative behavior is similar to the one of bidiagonal traffic: iSLIP still only achieves 0.84 and 0.87 throughput in CM and PM respectively, as shown in Tab. 2, whereas ASY-LQF and ASY-RR achieve almost the maximum throughput.

Fig. 12 investigates the effect of the variation coefficient of packet size 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. ASY switching always outperform SYN switching, for any $\alpha \leq 5$.

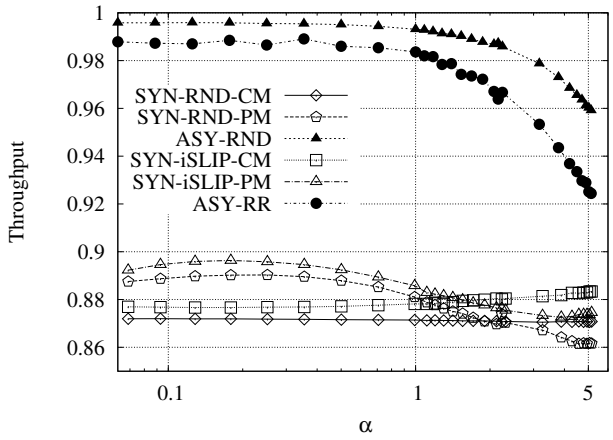

Figure 12: Maximum throughput under bidiagonal traffic vs packet length variation coefficient, with $N = 16$.

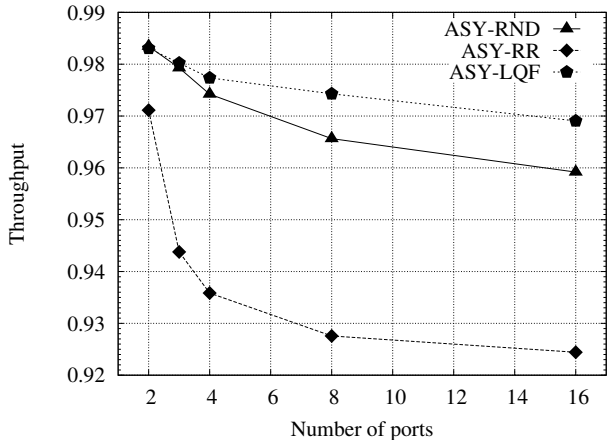

Figure 13: Maximum throughput vs. the number of ports N for an ASY VOQ switch under bidiagonal traffic and $\alpha = 5$

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 packet segmentation (see Sect. 2.1). Looking at the detailed behaviors of the different algorithms, consistently with Fig. 5 obtained with a single FIFO per input, throughput decreases for larger α.

Finally, in Fig. 13 we investigate the behavior of ASY switches as a function of their number of ports N, in the case of a very large packet-length variance ($\alpha = 5$). This value is taken more than twice than the maximum observable in standard Ethernet networks, according to Property 2, to highlight extreme

starvation problems. The throughput reduction is still limited (less than 5-8%) and is more relevant for larger switches, since the temporary starvation due to long packet can delay the transfer of packets from the other $N-1$ ports.

5. Segmentation overhead in synchronous switching

In SYN switches, when packets are segmented into fixed-size cells, some bandwidth is wasted mainly due to two effects: i) unfilled cells, ii) additional control information needed on each cell. Regarding the first effect, the last cell of a packet may be only partially filled due to rounding effects. In the worst case, a packet slightly larger than a cell generates two cells. As a consequence, almost 50% of the bandwidth can be "wasted" due to the segmentation process. Furthermore, each cell should carry some control information to correctly reassemble the packet at the outputs. Examples of such information are: the sequence number, the last-cell flag, the packet identifier, or the payload size. Each cell could carry also some control information to correctly route the cell inside the switching fabric, such as the router port or interface. On the contrary, in ASY switches, when transferring a packet across the switching fabric, the packet simply carries control information to route the packet.

To evaluate the bandwidth overhead due to segmentation, we can assume that b_{reas} is the amount of control information added for each cell to correctly reassemble the packet at the outputs, and b_{route} be the amount of control information for each cell devoted to the routing process. A packet of size p generates a number of cells equal to $\lceil p/c \rceil$, where $\lceil x \rceil$ is the smallest integer $\geq x$ and c is the cell payload size. The total amount of data D_{SYN} transferred across the switching fabric to switch a packet of size p is

$$D_{SYN} = \left\lceil \frac{p}{c} \right\rceil (c + b_{route} + b_{reas}) \tag{5}$$

The bandwidth fraction due to the segmentation overhead for a packet of size p is then

$$O_{SYN} = \frac{D_{SYN} - p}{D_{SYN}} = 1 - \frac{p}{D_{SYN}}$$

Fig. 14 reports the bandwidth overhead O_{SYN} for $p \in [20, 1500]$ bytes corresponding to the range of IP packets occurring in standard IP/Ethernet networks. We have neglected the control information in each cell (i.e., $b_{route} = b_{reass} = 0$) to highlight the bandwidth waste due to partial cell filling. Even in this optimistic scenario, the bandwidth overhead can be very high, especially for small packets.

In an ASY switch, we need only some control information b_{route} to route the packet across the switching fabric. A lower amount of control information than SYN switches is required, because it is not needed to add information for the reassembly process. Hence, when a packet of size p is transferred, the total amount of information D_{ASY} transferred across the switching fabric is simply

$$D_{ASY} = p + b_{route} \tag{6}$$

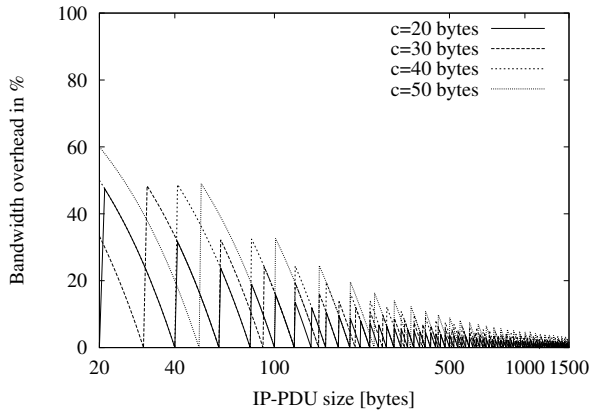

Figure 14: Bandwidth overhead due only to partial filling in SYN switches for different cell payload sizes c

Table 3: Segmentation overhead, evaluated on 218 traces for Chicago and San Jose OC192 links during the period 2008-2013. For the SYN case, the cell size has been optimally adapted at each month.

Scenario	Bandwidth segmentation overhead			
		Minimum	Average	Maximum
ASY	O_{ASY}	0.09%	0.15%	0.73%
SYN	O_{SYN}	2.7%	3.9%	7.8%

Obviously, $D_{SYN} > D_{ASY}$ and the overall bandwidth overhead is simply:

$$O_{ASY} = \frac{D_{ASY} - p}{D_{ASY}} = \frac{b_{route}}{D_{ASY}}$$

which monotonically decreases as p increases.

5.1. Optimal cell size

The curves in Fig. 14 suggests that it is possible to minimize the impact of the segmentation by choosing a proper cell size c. However, given the non-monotonic behavior of the curves, finding the optimal value of c is not immediate. To maximize the data throughput, c must be chosen as to minimize the average bandwidth overhead. If we know the distribution $F(p)$ of the packet sizes, the optimal payload cell size can be found by solving:

$$c^{\star} = \arg\min_{c} \sum_{p} D_{SYN}(p, c) F(p) \tag{7}$$

To evaluate the practical impact of segmentation, we numerically solved (7) and found the optimal $c^{\star}$ for the traffic transferred on four OC192 links (two

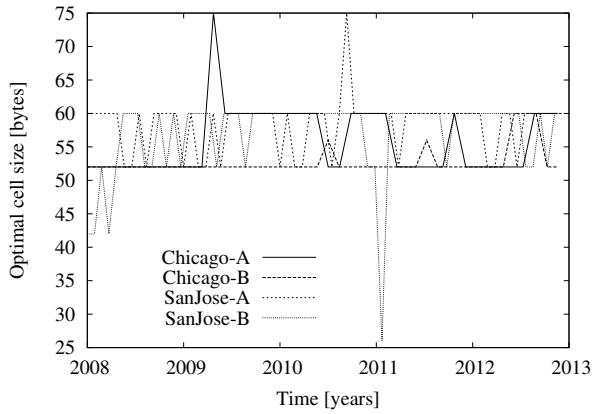

Figure 15: Temporal evolution of the optimal value of payload cell size $c^\star$ evaluated for Chicago and San Jose OC192 links

Table 4: Segmentation overhead for SYN switches, evaluated for Chicago and San Jose OC192 links during the period 2008-2013

Adopted payload cell c [bytes]	Num. of traces such that optimal $c^\star = c$	Bandwidth segmentation overhead O_{SYN}		
		Minimum	Average	Maximum
26	1	4.3%	5.0%	7.9%
42	3	4.0%	4.8%	12.7%
52	92	2.9%	4.0%	17.6%
56	2	2.9%	4.4%	20.3%
60	118	2.7%	4.1%	23.0%
75	2	3.0%	4.9%	25.9%

in Chicago and two in San Jose) during the last 5 years, exploiting the traces described in Sec. 2.4. We assume one byte of additional control information only for both SYN and ASY switches (i.e., $b_{route} = 1$ byte and $b_{reass} = 0$ byte), even if this scenario is slightly optimistic for SYN switches. Fig. 15 reports the temporal evolution of the optimal cell size for the four links. Clearly, *it is not possible to observe any trend* in the values of the optimal $c^\star$, which implies the difficulty in predicting future values of $c^\star$. The only possible deduction is that 52 and 60 bytes are the most common optimal values since 2008 till 2013. In the case $c^\star$ is chosen on the whole 5 years traces, the optimal cell size would be 60 bytes. As second observation, there are some months which appear as "outliers", since their optimal cell sizes is quite different from 52-60.

In Tab. 3 we investigate the bandwidth overhead due to segmentation for the optimal choice of $c^\star$. We report the statistics for both ASY and SYN switches evaluated on all 218 traces of the period 2008-13. In ASY switches, such overhead is very limited, being equal to $1/\hat{p}$, where $\hat{p}$ is the average packet

size in bytes. In the worst case (just one trace), $\hat{p} = 136$ bytes and we get $O_{ASY} = 7.3\ 10^{-3}$, a negligible overhead. For SYN switches, in Table 3, we solved (7) in each month and evaluated the segmentation overhead (O_{SYN}) when using the optimal cell size $c^\star$ for the corresponding month. The actual values of O_{SYN} are very large: up to 7.8% of the bandwidth is "wasted" even if the cell size is adapted every month to the traffic. This is mainly due to the partial filling of cells. Recall that, in the worst case $p = c + 1$, $D_{SYN} = 2c$ and $O_{SYN} = 1 - c/(2c) \approx 50\%$. Note that the results computed in Table 3 for SYN switches are very optimistic, since i) the cell size is adapted with the time (which is usually unfeasible) and ii) the optimal value $c^\star$ is chosen in advance, at the beginning of each month, based on the distribution of the future arrivals during the month (which is impossible).

To better estimate the overhead due to partial cell filling in more practical cases, we considered another scenario in which we optimized the cell size based on a single trace, and observed the effects on all the other traces. This is meaningful for the realistic case in which the optimal cell size has been chosen according to some past measurements and we wish to evaluate the segmentation overhead for the future. Table 4 reports, for each chosen cell size c, the total number of traces in which c was optimal (i.e., $c = c^\star$) during the period 2008-13, coherently also with Fig. 15. The last three columns report the statistics regarding the bandwidth segmentation overhead (O_{SYN}), evaluated for all 218 traces. Note that the actual bandwidth loss due to segmentation is on average ($O_{SYN} \approx 4\%$) similar to the values reported in Table 3. But, in the worst case, the loss is much higher, reaching also impressive values of 26%. To understand the reason for such large values, consider the case in which the cell size has been chosen to be the best choice for all the five years: $c^\star = 60$ bytes. In the worst case, $O_{SYN} = 23\%$ occurred for a single trace, i.e., during the hour in which the trace was collected, around 23% of the bandwidth would have been lost due to the segmentation overhead. The reason is that more than 2.2 billions of packets (i.e, around 50%) were 76 bytes packets long and most of the packets were quite small (the average packet size was $\hat{p} = 136$). For any of these 76 bytes packets, the bandwidth overhead (due to the incomplete filling) is very large: $1 - 76/(2 \times 61) = 37\%$.

As a general conclusion, even if the cell size has been optimized based on past measurements, we can expect that the loss due to segmentation process can become very large.

The non-negligible segmentation overhead for SYN switches puts a new perspective in the performance comparison between ASY and SYN switches seen so far. All the throughput results for SYN switches shown in Sec. 3 and 4 should be reduced by the segmentation overhead to provide a fair comparison with ASY switches. For the specific traces considered in this paper, this implies that in many scenarios where we showed SYN switches outperforming ASY switches, actually ASY switches are outperforming SYN switches. Since the segmentation overhead depends on the actual packet length distribution, we preferred to keep the results of the previous sections independent of the specific traffic pattern, even if this implies that the throughput results for SYN are rather optimistic.

6. Conclusions

We compared the performance of synchronous and asynchronous switches for variable-size packet arrivals, considering IQ switches with a single FIFO queue per input and IQ switches with Virtual Output Queueing. We show that asynchronous switches experience either a small performance degradation or better performance with respect to synchronous switches, when neglecting the segmentation overhead peculiar of synchronous switches. Furthermore, we also highlighted that one of the key traffic parameters affecting the performance of asynchronous switches is the variation coefficient of the packet size, which is usually small in realistic scenarios. Finally, thanks to an extensive study of real traffic in the Internet, we showed that the bandwidth overhead due to the segmentation mat be very large in synchronous switches. Because of i) the reduced complexity of the schedulers for asynchronous switches, ii) their simple architecture which avoids the need of complex segmentation and reassembly machines and simplify clock alignment circuits and iii) their negligible segmentation overheads, we expect that asynchronous architectures will lead to simpler and more efficient switching architectures in the near future.

[1] A. Bianco, D. Cuda, P. Giaccone, and F. Neri, "Asynchronous vs synchronous input-queued switches," in *IEEE GLOBECOM 2010*, Dec. 2010, pp. 1–5.

[2] M. Karol, M. Hluchyj, and S. Morgan, "Input versus output queueing on a space-division packet switch," *Communications, IEEE Transactions on*, vol. 35, no. 12, pp. 1347–1356, Dec. 1987.

[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] S. Yoo, "Energy efficiency in the future internet: The role of optical packet switching and optical-label switching," *Selected Topics in Quantum Electronics, IEEE Journal of*, vol. 17, no. 2, pp. 406–418, Mar-Apr. 2011.

[5] S. J. B. Yoo, "Optical packet and burst switching technologies for the future photonic internet," *Lightwave Technology, Journal of*, vol. 24, no. 12, pp. 4468–4492, Dec. 2006.

[6] P. Hansen, S. Danielsen, and K. Stubkjaer, "Optical packet switching without packet alignment," in *Optical Communication, 1998. 24th European Conference on*, vol. 1, Sep. 1998, pp. 591–592.

[7] "Omnet++ community site," Aug 2013, available on `http://www.omnetpp.org/`.

[8] K. Pawlikowski, "Steady-state simulation of queueing processes: survey of problems and solutions," *ACM Computing Surveys (CSUR)*, vol. 22, no. 2, pp. 123–170, 1990.

[9] J. J. A. Moors and J. Muilwijk, "An inequality for the variance of a discrete random variable," *Sankhy: The Indian Journal of Statistics*, vol. 33, no. 3/4, pp. 385–388, Dec. 1971.

[10] R. Birke, M. Mellia, M. Petracca, and D. Rossi, "Understanding VoIP from backbone measurements," in *INFOCOM*, 2007, pp. 2027–2035.

[11] K. Imran, M. Mellia, and M. Meo, "Measurements of multicast television over IP," in *LANMAN*, 2007, pp. 2027–2035.

[12] "Trace statistics for CAIDA passive OC48 and OC192 traces," Aug 2013, available on `http://www.caida.org/data/passive/trace_stats/`.

[13] 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.

[14] K. Sriram 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.

[15] S. Fuhrmann, "Performance of a packet switch with crossbar architecture," *Communications, IEEE Transactions on*, vol. 41, no. 3, pp. 486–491, Mar. 1993.

[16] 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.

[17] I. Iliadis, "Synchronous versus asynchronous operation of a packet switch with combined input and output queueing," *Performance Evaluation*, vol. 16, no. 1-3, pp. 241–250, 1992.

[18] 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.

[19] L. Tassiulas, "Linear complexity algorithms for maximum througput in radio networks and input queued switches," in *INFOCOM*, 1998, pp. 533–539.

[20] P. Giaccone, B. Prabhakar, and D. Shah, "Towards simple, high-performance schedulers for high-aggregate bandwidth switches," in *INFOCOM*, 2002.

[21] 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.

[22] G. Passas and M. Katevenis, "Asynchronous operation of bufferless crossbars," in *HPSR*, June 2007, pp. 1–6.

[23] N. McKeown, "The iSLIP scheduling algorithm for input-queued switches," *Networking, IEEE/ACM Transactions on*, vol. 7, no. 2, pp. 188–201, 1999.

[24] L. Tassiulas and P. Bhattacharya, "Allocation of interdependent resources for maximal throughput," *Stochastic Models*, vol. 16, no. 1, pp. 27–48, 2000.

[25] J. Dai and W. Lin, "Maximum pressure policies in stochastic processing networks," *Operations Research*, vol. 53, no. 2, pp. 197–218, 2005.

[26] T. Bonald and D. Cuda, "Rate optimal scheduling schemes for asynchronous input queued packet switches," *ACM SIGMETRICS Performance Evaluation Review*, vol. 40, no. 3, pp. 95–97, 2012.