Design of Switches with Reconfiguration Latency

Design of switches with reconfiguration latency

Valentina Alaria, Andrea Bianco, Paolo Giaccone, Emilio Leonardi, Fabio Neri
Dipartimento di Elettronica, Politecnico di Torino, Italy

***Abstract*— Optical switching fabrics (OSF) are considered to be appealing solutions for the design of high speed packet switches, due to their excellent scalability in terms of bandwidth and power consumption. Candidate technologies are MEMS, bubble switches, broadcast-and-select networks with tunable devices. All of them suffer a reconfiguration latency each time the input/output connections are changed, due to technological constraints; unfortunately, this latency is not negligible with respect to the packet transmission time, and can adversely affect performance, especially delay and throughput.**

When scheduling the transmission of packets across an OSF, the multi-hop approach was shown to be a promising way to control the tradeoff between delay and throughput. In this case, the OSF is configured just once in a while, on a time scale much larger than the packet transmission time, and packets may be recirculated across the ports to provide full or partial connectivity among ports. Previous works have investigated this approach when a physical ring topology is used for the interconnection.

Here, we extend the multi-hop approach to multidimensional regular topologies, which offer a better tradeoff between throughput and delay. We discuss not only the scheduling problem for these topologies, but also the design of routing. We investigate performance by simple analytical models and show the design tradeoff among throughput, speedup and delays.

I. Introduction

[1] All-optical switches are considered a very appealing solution for the design of ultra-high speed networks. Their main advantage is the avoidance of optical-to-electronic conversions, which is a technological issue limiting the performance in current switches to hundreds of Gbps. Unfortunately, all-optical switches are practically infeasible for the lack of simple "optical memories" able to mimic the buffers used in electronic switches to solve temporary congestion.

Hybrid optical/electronic switching architectures are today the most promising approach to design routers able to reach aggregate bandwidths up to 100 Tbps [1]. In these designs, the switching fabric is fully optical and is typically located in a different rack with respect to the switch line-cards. Packets arrive at the router through optical links, and, after the optical/electronic conversion, they are processed and buffered in the line-card; after an electronic/optical conversion, packets are sent over optical fibers to the optical switching fabric. Note that the use of optical switching fabrics may be convenient also to reduce power consumption, since in optics power consumption is largely independent from the transmission rate.

Optical switching fabrics (OSF) may be based on several different technologies, such as MEMS [2], bubble switches [3], broadcast and select networks with tunable devices [4], etc. However, most of these technologies share a common feature, i.e., whenever the OSF configuration (input/output ports connections) is changed, a *reconfiguration latency* is required before communication takes place. At least the ports involved in such reconfiguration must refrain from transmission; in many technologies, all the ports of the switch are blocked during the reconfiguration, and we make this assumption in this paper. The reconfiguration latency, due to technological constraints like mechanical inertial effects in MEMS, or tuning times of tunable devices in broadcast and select networks, is usually not negligible with respect to the packet transmission times (which are in the order of few ns at very high line rates), and can adversely affect the switch performance.

As a consequence, the scheduling algorithm, whose task is to select the switching configuration of the optical device, should take into consideration reconfiguration latency constraints so as to minimize the number of reconfigurations required to efficiently transfer a given traffic pattern. To the best of our knowledge, only few works have been proposed that consider the additional constraints due to reconfiguration latency when defining the scheduling problem (see [5], [6], [7], [8]). All these works assume that, when input i is connected to output j, only packets stored at input port i and destined to output port j can be transferred through the switching fabric, i.e. all the packets cross the switching fabric only once. In other words, when N packets are present at one input and destined to different N outputs, at least N switching fabric reconfigurations are required to allow the full connectivity between all inputs and outputs to be obtained, and to transfer N packets in sequence.

Scheduling algorithms must carefully balance two main performance objectives: throughput and delay. This balance becomes fundamental when reconfiguration latency is not negligible. Indeed, to obtain high throughput, the scheduling should keep for long time the same switching configuration, so as to reduce the negative effect of inactivity periods due to the reconfiguration overhead; however, low delays imply to change quickly the switching configuration, so as to allow the full connectivity between all ports to be obtained in a short time interval.

Consider the following scenario, with an optical switch with $N = 1024$ ports, with a reconfiguration latency $T = 1$ ms; assume that the link speed is 10 Gbps, and that internally the switch operates on fixed-size data-unit of 64 bytes (a convenient format to transfer minimum size TCP/IP packets); thus, the data-unit transmission time is $\delta = 51.2$ ns. Assume that, on an empty switch, N packets arrive, each at a different input, all destined for the same output. Even when keeping

[1]This work was (partially) supported by the EU FP6 Network of Excellence e-Photon/ONe (through WP4)

1-4244-0355-3/06/$20.00 (c) 2006 IEEE

each switching matrix configuration for the minimum time required to transmit a single data-unit, thus obtaining a very low throughput efficiency, the k-th packet will be transferred at time $k(T+\delta)$ after arrival, because of the k reconfigurations needed before the connectivity to the desired output is provided to the input where packets are waiting. Hence, the worst delay for the N packets is $N(T+\delta) \approx NT \approx 1s$, which is obviously unacceptable for any realistic application; thus, if we do not take a different approach, this result may compromise the hopes towards the use of optical devices in routers in the future.

To overcome this problem, we exploit a *multi-hop* approach, which was proposed in [9] and derived from the same approach studied in the context of WDM/TDM networks [10]. The main idea of multi-hop scheduling, better explained in Section II, is to configure the switching matrix once in a while, on a time scale significantly larger than packet transmission time, and to re-circulate packets among ports, i.e. a packet at input port i may reach its destination port j via successive transmissions through one (or more) intermediate ports. Thus, we exploit the fact that input port i and output port i reside always in the same line-card, and a packet arrived at output i can be reconsidered for retransmission across the switching fabric at negligible extra cost. Note that this architectural assumption is fairly common; e.g., see the single-stage switch described in [1]. In the same scenario previously considered, the worst case delay for a multi-hop approach can be simply $T + N\delta \approx T = 1$ ms, a much smaller value than the one obtained with the traditional single-hop approach; this delay can be acceptable for practical implementations.

Clearly, sending packets in multi-hop increases the overall load of the switching fabric; we show in this paper that we can deal with this issue, and that significant benefits in terms of delay can be obtained.

II. The multi-hop approach

We assume that the switch is built around a single optical switching fabric, running on fixed size packets, on a time slotted base. This switching fabric behaves as a buffer-less crossbar, i.e. at each time no more than one packet can be sent from an input port and can be received at an output port. A feasible switching configuration is equivalent to a *matching* in a bipartite graph, in which left-most nodes represent the input ports and right-most nodes the output ports; an edge connects left node i to right node j if input port i is connected to output port j. A matching in a generic directed graph is a set of edges which do not share the same start point or the same end point.

We consider a switch with N ports, each running at the same line rate; all the packets arriving at the same input port and directed to the same output port belong to the same flow. Input queues are used to solve contentions among packets contending for the same output, and are organized according to the Virtual Output Queue (VOQ) buffering scheme, with one FIFO queue for each flow, to achieve high throughput [11].

A centralized scheduling algorithm is in charge to select the sequence of appropriate switching configurations (matchings) that allow an efficient transfer, through the switching fabric, of packets residing at the input cards. Since, at each reconfiguration, a penalty in terms of latency has to be paid, to achieve high throughput the same matching must be held for a duration which is at least comparable with the reconfiguration latency.

To transfer all the packets according to a classical single-hop approach, full connectivity among switching ports is necessary (i.e., each input port has to be connected to every output port); as a consequence, the scheduling algorithm must cycle among at least N switching configurations. By doing so, however, the access delay can increase to unbearable values, as already shown in the introduction. On the contrary, according to the multi-hop approach, only a partial connectivity may be sufficient to guarantee the transfer of any packet through the switching fabric. Through a reduced set of switching configurations, input port i is directly connected by the scheduler only to a subset of other ports to which it can directly transmit packets. Packets directed to port j which is not connected to the port i, reach the destination in a multi-hop fashion, i.e. through some intermediate ports.

More formally, the multi-hop approach can be modeled in the following way. A connected *virtual interconnection topology* is overlaid to the set of switch ports; each node of the topology corresponds to a switch port. Let σ be the correspondence between the topology nodes and the switch ports; let $\sigma(i)$ be the node associated to port i.

Consider now two generic nodes $\sigma(i)$ and $\sigma(j)$. If $\sigma(i)$ and $\sigma(j)$ are adjacent (i.e., it exists an edge in the virtual topology between them), then port i will be directly connected to port j by a proper matching chosen by the scheduler; indeed, the scheduling process is induced by the adopted topology. If node $\sigma(i)$ and $\sigma(j)$ are not adjacent, port i and port j will not be directly connected; however packets will flow from port i to port j though a set of intermediate ports which correspond to a path of the topology connecting node $\sigma(i)$ and node $\sigma(j)$.

Note that more than one path may connect two generic nodes, but we assume that a deterministic *routing algorithm* chooses only one of the possible shortest paths, to prevent mis-sequenced delivery of packets belonging to the same flow.

Depending on the chosen virtual topology and routing scheme, the scheduler selects the matching to transfer the packets from input to output ports. Let η be the efficiency of the switching fabric, defined as the percentage of time in which the switching fabric is available for packet transfer; it results

$$\eta = \frac{P}{P+T}$$

being P the average holding time of matchings and T the reconfiguration latency. We denote with the term "epoch" a time interval comprising a matching holding time followed by reconfiguration time. As already stated, to achieve an high efficiency from the switching fabric, the same matching must be held for a duration which is larger than the reconfiguration latency.

We assume stationary traffic and we consider only periodic scheduling, in which a precomputed, fixed periodic

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2006 proceedings.

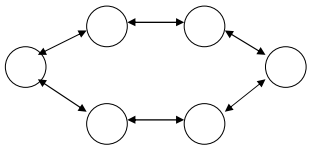

Fig. 1. Ring topology with 6 nodes

scheme [12], [13] is adopted with *constant* holding time P. Given the overlaid virtual topology, the scheduler computes a minimal set of matchings, called *covering matchings*, which covers all the edges of the topology. Let M_c be the resulting number of matchings for a particular topology; note that M_c is equal to the maximum between the in-degrees and out-degrees of all the nodes of the topology, thanks to the Birkhoff von Neumann theorem [14]. These M_c matchings are sequentially selected to configure the switching fabric according to the periodic scheme. We define as *frame* a time horizon of length $M_c(P+T)$ in which a complete scheduling cycle is performed. Note that a fixed periodic frame scheduling allows an easier implementation at high speed and can be designed to support efficiently uniform traffic.

Two types of internal bandwidth speedup S are allowed. In the case of *temporal speedup*, the switching fabric runs S times faster than the line rate and during each epoch up to SP packets are served at each port; note that the frame duration remains the same. In the case of *spatial speedup*, S switching fabrics run in parallel (*spatial speedup*), configured with different covering matchings; thanks to this, the frame duration is reduced by a factor S; in addition, when $S = M_c$ there are enough switching planes to cover all the topology without reconfiguration and the reconfiguration latency is null: $T = 0$. Finally, note that temporal and spatial speedup can be also combined together.

In summary, to design an efficient multi-hop scheduler the following issues should be considered:

- definition of the virtual interconnection *topology* and its mapping to the switch ports;
- definition of a suitable *routing* strategy of packets on the virtual topology;
- definition of the *frame scheduling* plan.

Of course all the three previous issues are not independent. The definition of the virtual interconnection topology has a direct impact both on the definition of the scheduling plan and on the definition of the packet routing strategy.

A. Multi-hop for Manhattan topologies

Many interconnection topologies can be mapped onto the switching ports. Previous work [9] has considered ring topologies, as the one depicted in Fig. 1. We present here an example of different topology, and in the following sections we generalize to any regular topology. We consider a bidirectional regular square grid topology depicted in Fig. 2, known in the literature with the name of Manhattan Street topology, overlaid to a 16×16 switch. Each input/output port corresponds to a node of the Manhattan topology, according to the following bijective mapping: node (i, j), located in row i and column j, with $0 \leq i, j \leq 3$, corresponds to port $k = 4 \times i + j$, $0 \leq k \leq 15$.

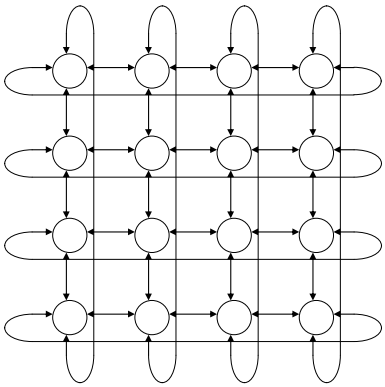

Fig. 2. Bidirectional Manhattan Street topology with 16 nodes

Given that we rely on a regular topology with node degree 4, port $k = (i, j)$ can directly (i.e. in single-hop) reach four ports: $k_1 = (i, |j + 1|_4)$, $k_2 = (i, |j - 1|_4)$, $k_3 = (|i + 1|_4, j)$, $k_4 = (|i - 1|_4, j)$; [2] all the other destinations must be reached in a multi-hop fashion. The scheduling frame is partitioned in four fixed epochs: in the first scheduling epoch every node (i, j) is connected to $(i, |j+1|_4)$ for a time equal to P and we say that the direction followed in the topology is "down"; in the second scheduling epoch, every node (i, j) is connected to $(i, |j - 1|_4)$, for a time equal to P (following "up" direction); in the third scheduling epoch every node (i, j) is connected to $(|i+1|_4, j)$, for a time equal to P (following "right" direction); in the forth scheduling epoch every node (i, j) is connected to $(|i - 1|_4, j)$ for a time equal to P (following "left" direction). In this case the frame duration is $4P + 4T$; each scheduling epoch, lasting $P + T$, is associated with a specific direction and, hence, with a matching.

This example can be extended to multidimensional Manhattan topologies of generic dimension c, with degree $2c$ at each node; in this case, each side of the corresponding hypercube is $\sqrt[c]{N}$ nodes and the frame duration is $2c(P + T)$. Note that a bidirectional ring topology is obtained by setting $c = 1$.

Many routing algorithms on a Manhattan network can be devised. In our work we consider the following routing scheme, called "Privileged Directions Routing" (PDR), described for a bi-dimensional Manhattan network for simplicity, but that can be easily extended to multidimensional networks. Among all the possible shortest paths from a node (i, j) to a node (k, l), consider the path through node (i, l), following (possibly) first the row direction and then (possibly) the column direction. Fig. 3 shows the minimum distance paths from the central node of a 5×5 Manhattan topology reaching all other nodes. Note that the PDR scheme has the following properties: (i) unique routing path between any pairs of nodes, (ii) easy computation

[2] We denote with $|\cdot|_n$ the modulo-n operator, i.e., the remainder of the division by n.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2006 proceedings.

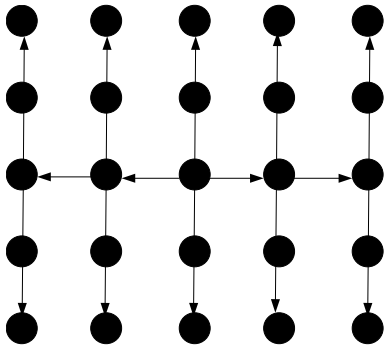

Fig. 3. Routing paths, according to PDR, for the central node of a 5×5 Manhattan network corresponding to a 25×25 switch; at most 2 directions are needed to reach any destination node.

and (iii) the load across all edges is balanced under uniform traffic.

Packets could be stored according to a classical VOQ scheme: in the example of Fig. 3, the "right" direction allows the central node to reach 10 destinations nodes; hence, during the matching corresponding to the "right" direction the packets present in 10 VOQs are served. However, to allow fair and easier access to the switching fabric, we adopt a FIFO selection among all the packets served in the same input and following the same direction; this is equivalent to consider just $2c$ queues per input, one for each possible direction, instead of N VOQs at each input port, with evident benefits for the scalability of the queueing system to large switches.

III. Theoretical performance analysis

In general, a given topology affects significantly the maximum throughput reachable in the switch architecture. Equivalently, when an internal speedup is allowed, the topology affects the minimum required speedup to achieve 100% throughput. Here we consider only some examples of regular topologies, derived from the most common families of interconnection networks: rings, Manhattan (derived from torus networks), Shuffle (derived from butterflies) [16] and Kautz [17] (derived from De Bruijn graphs [18]).

Let Λ be the $N \times N$ traffic matrix whose element λ_{ij} corresponds to the average traffic load from input port i to output port j. We assume that the switch is fed by a uniform traffic, i.e. $\lambda_{ij} = \lambda/N$, $\forall i, j$. In Sec. III-C we will discuss how to design switches fed by non-uniform traffic.

We assume also that routing in the topology is able to distribute the traffic uniformly among all links; when the topology is symmetric, this assumption is usually met. Then, the average traffic offered to a port is due to the traffic entering the port from outside and the traffic traversing that port to reach its final destination. Given a topology mapping σ, let d_{ij}^{σ} be the length of the path (in terms of number of hops/edges) along which traffic from port i to port j is routed; d_{ij}^{σ} can be seen as the distance between ports i and j (or equivalently, between nodes $\sigma(i)$ and $\sigma(j)$) under the particular σ chosen. The total traffic flowing on the topology is:

$$\rho_{tot} = \sum_{i,j} \lambda_{ij} d_{ij}^{\sigma} = \sum_{i,j} \frac{\lambda}{N} d_{ij}^{\sigma}$$

and the overall load offered to a port is:

$$\rho = \frac{\rho_{tot}}{N} = \frac{\sum_{i,j} \lambda d_{ij}^{\sigma}}{N^2} = \lambda E[d]$$

being $E[d]$ the average overall nodal distance according to the selected routing strategy. The diameter d_{max} of the topology is defined as the maximum distance among any pair of nodes: $d_{max} = \max_{i,j}\{d_{ij}\}$.

As a consequence, an upper bound $\hat{\lambda}$ to the maximum throughput is given by the traffic load at which the port load equals the port capacity. Since during an epoch of duration $(P+T)$, the port is able to serve the traffic for a duration P, then the port capacity is given by:

$$\mu = \frac{P}{P+T}$$

By imposing $\rho < \mu$, the maximum offered load to a port is:

$$\hat{\lambda} = \frac{P}{P+T} \frac{1}{E[d]} \tag{1}$$

The minimum speedup necessary to achieve 100% throughput is $1/\hat{\lambda}$; in other words, the maximum throughput achievable can be bounded by $\min\{1.0, \hat{\lambda}S\}$ being S the adopted speedup.

We now estimate the worst case access delay, i.e. the delay experienced by a generic packet entering an empty switch. Of course, this is only a lower bound (i.e. optimistic bound) on the average delay experienced by a packet in generic traffic conditions. For simplicity, we neglect propagation delays. As already discussed, a frame lasts $M_c(P+T)$. When a packet enters the switch, the routing process computes a path (i.e., an ordered sequence of edges) to transfer the packet from its input port to its output port. All the edges of the routing path are included, by construction, in the covering matchings inside a frame. Hence, during each frame at least one matching is employed to forward the packet along its path; if two or more adjacent edges of the path appear in the same order as in the corresponding matchings inside the frame, then they can be served during the same frame. Note that the *average* number of covering matchings necessary to serve a path is always not greater than $E[d]$, since it cannot be larger than the number of ports to traverse; on the contrary, during a single epoch a packet may be transferred across many ports in multi-hop fashion if a sequence of adjacent edges in the path is included in the matching. Let n_d be the maximum number of covering matchings necessary to switch a packet; note that n_d can be upper bounded by the diameter d_{max} of the topology. The worst case access delay W can be computed by considering a packet that has to traverse a number of edges equal to the diameter of the topology in different epochs. Two main contributions must be added: (i) the time to transfer the packet for d_{max} times across the switching fabric; (ii) the delay to

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2006 proceedings.

wait for the matchings serving the edges in the routing path. This second contribution is generally difficult to predict, since it depends on the sequence of epochs in the frame and on the routing strategy. But it can be bounded by considering the worst case in which each edge in the path is served always in different epochs and each time a new edge of the path should be served, the packet must wait for a complete frame. This bound can be estimated as $(n_d - 1)$ times the duration of a frame, each time corresponding to the service of a particular edge in the path, plus one additional frame in which the final edge of the path is served at the beginning of the last epoch. Hence, we obtain:

$$W \leq d_{max}\delta + (n_d - 1)M_c(P+T) + (M_c - 1)(P+T) = \\ = d_{max}\delta + (n_d M_c - 1)(P+T) \quad (2)$$

In all this computation we have assumed the fact that $d_{max} \ll P$, which holds always in all the curves shown in the following graphs. Note that, if some spatial speedup S is available, then W is decreased by a factor S; as already mentioned, if $S = M_c$, then $T = 0$.

A. Topologies for multi-hop

We now discuss some specific regular topologies, chosen among the most common ones used as interconnection networks.

1) Unidirection and bidirectional ring: In a bidirectional ring (RI), $M_c = 2$ since only two matchings are necessary to cover the clockwise edges and the anti-clockwise edges. The average distance is simply $N/4$, whereas $d_{max} = N/2$. Just one matching is sufficient for a packet to reach the destination port: $n_d = 1$. Similarly, for a unidirectional ring (UR) it holds: $M_c = 1$, $E[d] = N/2$, $d_{max} = N$ and $n_d = 1$; in this case, the switching configuration is kept fixed and the reconfiguration latency is null.

2) Multidimensional Manhattan: For multidimensional bidirectional Manhattan (MN) topologies, under PDR routing, the average distance can be estimated in the following way: for each dimension, two possible directions can be chosen, hence $\sqrt[c]{N}/4$ is the approximated[3] average distance traversed along the same direction; since c dimensions are allowed:

$$E[d] = c\sqrt[c]{N}/4 \quad (3)$$

It also holds: $d_{max} = 2E[d]$. Furthermore, the degree $2c$ of the topology corresponds to the number of different matchings to provide full connectivity; hence, $M_c = 2c$. The shortest path between two generic nodes can be associated with an ordered sequence of c directions (one for each dimension), corresponding to c matchings: $n_d = c$.

3) Shuffle networks: We consider a bidirectional shuffle (SF) network (also known as wrapped butterfly network). A shuffle network is a regular topology composed by $N = kp^k$ nodes, organized in k levels of nodes; each node of a level is connected to p nodes of the following level (nodes of level k are connected to nodes of level 1). We omit here the details of the topology and the routing algorithm, well known in the literature [16], [18]. Since the in/out degree for a node is always $2p$, then only $2p$ matchings are necessary to cover the topology: $M_c = 2p$. Routing in the network can follow the algorithm in [19] for which it has been shown that:

$$E[d] \approx \frac{5}{4}k$$

It also holds: $d_{max} = 2k - 1$; n_d can be upper bounded by d_{max}.

4) Kautz networks: Kautz (KN) networks [17] derive from De Bruijn graphs [18]; they are regular topologies, with number of nodes $N = p^k + p^{k-1}$ where p is the degree of the network and k is the diameter of the corresponding graph: $d_{max} = k$. Routing is simple and is described in [17]. It can be easily shown that for this topology: $M_c = p$, $E[d] = N_d = k$.

5) Single hop: Also the single-hop (SH) approach can be studied under our general framework as a particular topology. Indeed, we consider a frame scheduling approach for single-hop, adopting a sequence of N disjoint matchings given by the Birkhoff von Neumann decomposition [14] of the traffic matrix. Under uniform traffic, the optimal frame is composed by N scheduling epochs; during the k-th scheduling epoch ($0 \leq k < N$), input port i will be connected to output port $|i + k|_N$ for a duration P. In our framework, this case corresponds to a fully connected topology, in which $d_{ij} = 1$ always. The N disjoint matchings of the frame cover all the topology: $M_c = N$. Trivially, $n_d = 1$, $E[d] = 1$ and $d_{max} = 1$. Note that for single-hop, the bound given by (2) is strict.

B. Topology comparison

To compare different topologies we show the performance in terms of throughput, speedup and access delay for the parameters setting of Table I. The main findings of this section still hold qualitatively for other realistic scenarios. T is given by technological constraints related to MEMS reconfiguration latencies [7], and P is set to guarantee a switching efficiency $\eta \approx 90\%$, corresponding to 10% of throughput reduction in the single-hop case. With the packet set equal to the MTU of Ethernet, the slot duration is $1.2\ \mu s$, corresponding to $T = 100$ timeslots and $P = 1000$ timeslots.

In our investigations we have compared the following topologies: single-hop (SH), unidirectional ring (UR), bidirectional ring (RI), Manhattan x-dimensional (MHx), Shuffle

Parameter	Symbol	Value
reconfiguration latency	T	$0.12\ ms$
scheduling period	P	$1.2\ ms$
I/O link rate		10 Gbps
packet size		1500 bytes
packet transmission time	δ	$1.2\ \mu s$

TABLE I
PARAMETERS CONSIDERED FOR THE SWITCHING ARCHITECTURE UNDER STUDY

[3]Precise evaluation of the average distance is possible, but the approximation here provides an upper bound good enough for our purposes.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2006 proceedings.

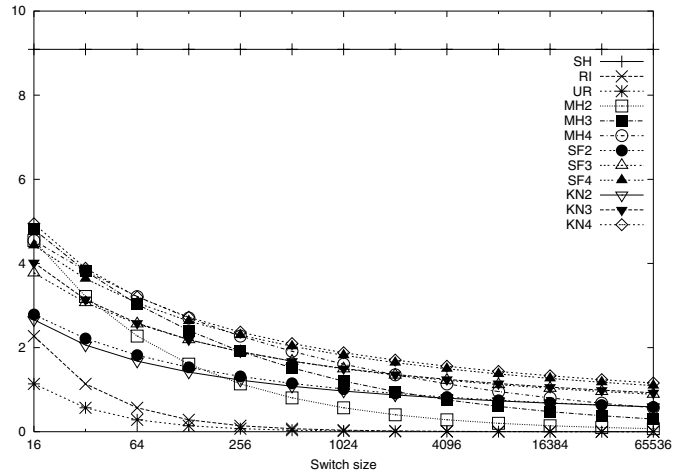

Fig. 4. Maximum throughput achievable $\hat{\lambda}$ in Gbps

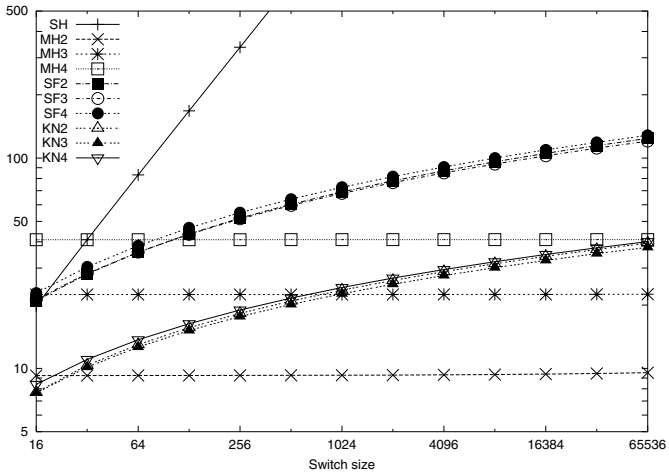

Fig. 6. Worst case access delay W in ms

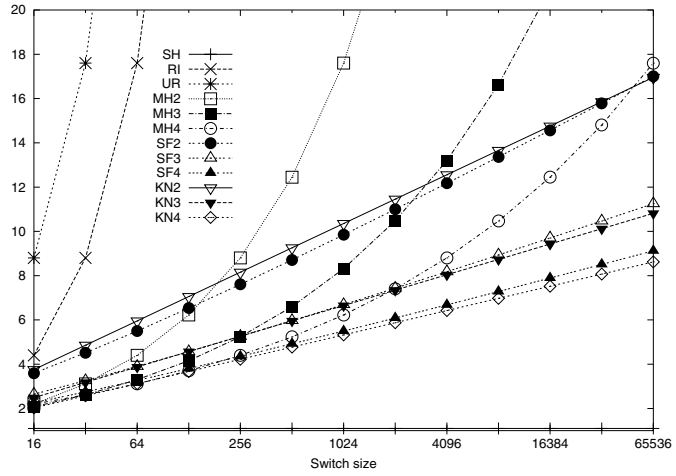

Fig. 5. Minimum speedup S necessary to achieve 100% throughput

x-dimensional (SHx) and Kautz x-dimensional (KNx), for dimensions $x = 2, 3, 4$.

In Fig. 4 the values of $\hat{\lambda}$ in Gbps are reported for different topologies and different switch sizes, when no speedup is allowed. Fig. 5 shows the necessary speedup required, under the same conditions, to achieve 100% throughput; i.e., this graph is obtained by computing the inverse of Fig. 4.

As expected, single-hop is the most efficient from the point of view of the required speedup (and, hence, throughput), since the required speedup is simply $(P + T)/P = 1.1$, independently from the size of the switch. This is a big advantage of the single-hop approach with respect to the multi-hop, but this advantage is traded off with a much larger delay, as shown later.

On the contrary, ring topologies (unidirectional and bidirectional) require the highest speedup, which grows linearly with the switch size; for this poor scalability, we exclude it from our next investigations. We mention that [9] already discussed the design issues related to such topologies.

Furthermore, for Manhattan, Shuffle and Kautz networks, as the dimension of the topologies grows, a lower speedup (larger throughput) is achieved, thanks to the reduced average distance. Note that Shuffle and Kautz behave almost the same. It is interesting to note the performance gain obtained by increasing the number of dimensions from 2 to 3 for Manhattan, Shuffle and Kautz; a limited speedup (< 10) is required, even if the switch size is very large. Note also that for all these x-dimensional networks, M_c is very small (equal to x or $2x$) and independent from the switch size; this allows to exploit a limited spatial speedup to remove completely the effect of the reconfiguration latency. This does not hold for single-hop.

To understand how throughput is traded with delay, consider the worst case access delays of Fig. 6, in which the curves for the rings have been removed. Even for relative small switches ($N > 64$), single-hop shows unacceptable access delays (recall that W gives a strict bound on the performance of single-hop). Manhattan networks show a delay almost constant with respect to the switch size; indeed, by (2), $W \approx 2c^2(P+T) = 2.64c^2$ ms. All Shuffle topologies behave almost the same, and this is also true for Kautz topologies, for which the access delay is lower thanks to the smaller diameter.

From the previous results, the ranking of the studied topologies is somehow arbitrary, since each of them shows a different tradeoff between throughput and delay. But observe that Fig.s 4, 5 and 6 show some performance limits with different impacts in the design. Indeed, the throughput limit (or the required speedup) can be considered a loose design constraint, since technology is always pushing further speed and packaging density; the optical domain, in particular, is offering large transmission bandwidths at limited costs. On the contrary, access delay is an hard design constraint, since real-time applications running on any networks cannot accept large delays; for example, worst case or average delays larger than few tens of milliseconds can be considered unacceptable in the Internet today. Recall also that W does not take into

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2006 proceedings.

account queueing delay, which can be also much larger than W when the switch is fully loaded. Hence, delays of Fig. 6 should be considered very optimistic.

C. Performance under non-uniform traffic

When the traffic is not uniform, to improve system performance, both the virtual interconnection topology and the routing strategy should be adapted to the traffic pattern. Indeed, the goal of the virtual interconnection topology and routing strategy design is to minimize the amount of packets sent in multi-hop fashion. Intuitively, under non-uniform traffic two ports exchanging a large amount of traffic should be placed in topologically close nodes; whereas two ports exchanging a very small amount of traffic can be placed far apart.

The placement problem of the switch ports into the virtual topology, given the amount of traffic exchanged among the ports, is equivalent to the problem of node placement and wavelength assignment in WDM network [20], which has been proved to be NP-hard. As a consequence, many sub-optimal heuristics can be devised, for a particular topology, to solve the placement problem. We leave this design aspect for future works.

As an alternative solution, both the topology and the routing could be devised as time-adaptive to match the actual traffic conditions and hence to optimize performance. From practical point of view, it is very difficult to know the actual traffic matrix, mainly because traffic is not stationary with the time. One possible solution could be to devise some dynamic measurement schemes to estimate, in real time, the actual traffic matrix, and change dynamically the topology placement: but this solution introduces many other technical issues to solve, for example, the possibility of out-of-sequence delivery of packets when the topology changes. For this reason, we think that it could be more feasible to consider the traffic matrix as unknown, and to design the switch to perform well under uniform traffic. A still open issue is to understand how robust this design would be for varying traffic patterns.

IV. Conclusions

In this paper we studied the multi-hop approach to schedule the packets across a switching fabric with very large reconfiguration latency. The main idea is to send a packet from an input port to an output port across the switching fabric through (possibly) many intermediate ports, in order to reduce the need of switching reconfiguration to provide full connectivity between input and output ports.

We have investigated the multi-hop approach based on common regular topologies, and have shown the tradeoffs between throughput, speedup and access delays.

Main finding of our investigation is that, especially for large switches, the multi-hop approach may become the only feasible approach to exploit optical switching fabrics with reconfiguration latencies, since delay performance is unacceptable for the single-hop approach. Depending on the design constraints related to the allowed speedup and the maximum access delay, a set of topologies can be considered for the implementation. Many other design issues, which have not been discussed here, should be taken into account like path diversity for reliability and power consumption due to electro-optical conversions.

References

[1] I. Keslassy, S.T. Chuang, K. Yu, D. Miller, M. Horowitz, O. Solgaard, N. McKeown, "Scaling internet routers using optics", *ACM SIGCOMM*, 2003, Aug. 2003, pp. 189-200.

[2] P.D. Dobbelaere, K. Falta, S. Gloeckner, "Advances in integrated 2D MEMS-based solutions for optical network applications", *IEEE Optical Communications*, May 2003, pp. 16-23

[3] S. Hengstler, J.J. Uebbing, P. McGuire, "Laser-activated optical bubble switch element", *2003 IEEE/LEOS International Conference on Optical MEMS*, Aug. 2003, pp. 117-118

[4] S.L. Danielsen, C. Joergensen, B. Mikkelsen, K.E.Stubkjaer, "Optical packet switched network layer without optical buffers", *IEEE Photonics Technology Letters*, vol. 10, n. 6, Jun. 1998, pp. 896-898

[5] K. Kar, D. Stiliadis, L.T. Lakshman, T.V., L. Tassiulas, "Scheduling algorithms for optical packet fabrics", *IEEE JSAC*, vol. 21, n. 7, Sept. 2003, pp.1143-1155

[6] I. Keslassy, M. Kodialam, L.T. Lakshman, D. Stiliadis, "On guaranteed smooth scheduling for input-queued switches", *INFOCOM 2003*, vol. 2, Apr. 2003, pp. 1384-1394

[7] X. Li, M. Hamdi, "On scheduling optical packet switches with reconfiguration delay", *IEEE JSAC*, vol. 21, n. 7, Sept. 2003, pp.1156-1164

[8] B. Towles, W.J. Dally, "Guaranteed scheduling for switches with configuration overhead", *IEEE/ACM Trans. on Networking*, vol. 11, n. 5, pp. 835-847, Oct. 2003

[9] Andrea Bianco, Paolo Giaccone, Emilio Leonardi, Fabio Neri, Paola Rosa Brusin, "Multi-hop scheduling for optical switches with large reconfiguration overhead", *High Performance Switching and Routing (HPSR 2004)*, Phoenix, AZ, USA, April 2004

[10] M. Ajmone Marsan, A. Bianco, E. Leonardi, F. Neri, A. Nucci, "multi-hop Packet Scheduling in WDM/TDM Networks with Nonnegligible Transceiver Tuning Times", *IEEE Trans. on Communications*, Vol.48, n.4, pp.692-703, Apr.2000

[11] T. Anderson, S. Owicki, J. Saxe, C. Thacker, "High speed switch scheduling for local area networks", *ACM Trans. on Computer Systems*, vol. 11, n. 4, Nov. 1993, pp. 319-352

[12] T. Weller, B. Hajek, "Scheduling nonuniform traffic in a packet-switching system with small propagation delay", *IEEE/ACM Transactions on Networking*, 1997

[13] A.Bianco, M.Franceschinis, S.Ghisolfi, A.Hill, E.Leonardi, F.Neri, R.Webb, "Frame-based matching algorithms for input-queued switches", *High Performance Switching and Routing (HPSR 2002)*, Kobe, Japan, May 2002

[14] C.S. Chang, W.J. Chen, H.Yi Huang, "Birkhoff-von Neumann Input Buffered Crossbar Switches", *IEEE INFOCOM 2000*, Tel Aviv, Israel, Apr.2000, pp.1614-1623.

[15] N.F. Maxemchuck, "Routing in the manhattan street network", *IEEE Trans. on Communications*, n. 35, May 1987, pp. 503-512

[16] W.J.Dally, B.Towles, *Principles and practise of interconnection networks*, Elsevier, Morgan Kaufmann, 2004

[17] Smit, G.J.M.; Havinga, P.J.M.; Jansen, P.G., "An algorithm for generating node disjoint routes in Kautz digraphs", *IEEE Parallel Processing Symposium*, pp. 102-107, May 1991

[18] F.T.Leighton, *Introduction to parallel algorithms and architectures: arrays, trees, hypercubes*, Morgan Kaufmann, 1992

[19] M.Gerla, E.Leonardi, F.Neri, P.Palnati, "Routing in the Bidirectional Shufflenet", *IEEE/ACM Transactions on Networking*, Vol. 9, No. 1, pp. 91–103, February 2001

[20] Yeung, K.L.; Yum, T.-S.P.; "Node placement optimization in ShuffleNets", *IEEE/ACM Transactions on Networking*, vol. 6 , n. 3, June 1998, pp. 319-324

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2006 proceedings.