

Design Issues in Next-Generation Merchant Switch Fabrics

François Abel, *Member, IEEE*, Cyriel Minkenbergh, Ilias Iliadis, *Senior Member, IEEE*, Ton Engbersen, *Senior Member, IEEE*, Mitchell Gusat, *Member, IEEE*, Ferdinand Gramsamer, and Ronald P. Luijten

Abstract—Packet-switch fabrics with widely varying characteristics are currently deployed in the domains of both communications and computer interconnection networks. For economical reasons, it would be highly desirable that a single switch fabric could accommodate the needs of a variety of heterogeneous services and applications from both domains. In this paper, we consider the current requirements, technological trends, and their implications on the design of an ASIC chipset for a merchant switch fabric. We then identify the architecture upon which such a suitable and generic switch fabric could be based, and we present the general characteristics of an implementation of this switching fabric within the bounds of current state-of-the-art technology. To our knowledge, this is the first attempt to design a chipset that can be used for both communications and computer interconnection networks.

Index Terms—Buffered crossbar, Combined Input- and Cross-point Queuing (CICQ), interconnection networks, packet switching.

I. INTRODUCTION

PACKET switch fabrics form the core of a large number of LAN/MAN/WAN routers. Numerous flavors of switching fabrics with different characteristics are currently deployed in order to satisfy the various needs in these domains. Such a variety of needs and offered solutions is also observed in the domain of computer interconnection networks, which include, for example, SANs (System Area Networks) and StANs (Storage Area Networks).

Because of price and performance requirements, high-speed switch fabrics have always based their design on one or a group of Application-Specific Integrated Circuit (ASIC) chips. These ASIC chips are integrated circuits specifically designed by the networking-equipment manufacturers to perform the required switching function. However, because of long design cycles and continuously rising chip costs, a growing number of switch manufacturers are moving away from proprietary in-house ASIC designs, and are migrating to merchant fabric silicon.

This shift has created a new and fast growing Original Equipment Manufacturer (OEM) switch fabric market. Commercial switch fabrics are marketed as semi-customized off-the-shelf

devices, usually referred to as Application-Specific Standard Products (ASSP). The concept of a switch ASSP is similar to a memory, hard drive, or graphic chipset used by the computer industry, i.e., it is a chip that is designed for a specific purpose (in our case the switching of packets) and that is sold to more than one OEM manufacturer. ASSP switches are developed by silicon vendors such as Agere, AMCC, IBM,¹ IDT, and VITESSE.

Today, ASSP switches are used in a variety of networking equipment, including i) communications networks such as carrier-class core and metro switches and routers, multiservice edge and access switches; ii) enterprise and campus networks such as Ethernet backbones, PCs and server farms; and iii) computer networks such as high-performance computing systems, system and storage area networks, I/O networks, and blade servers. One of the drawbacks of such a wide spectrum of switch fabrics is that it leads to lower volumes in many of the equipment market segments.

Chip fixed costs, which keep doubling with each process generation, have reached such high levels in the past few years that they have caused a decline in the use of ASICs. Recently however, the current lack of growth in most of the above markets has prompted ASSP vendors to experience the same trend. The chip fixed costs include the chip design and the silicon mask sets. The mask set for a chip in the advanced 90 nm process costs on the order of 1.2 M\$, and there is a rule of thumb that the chip design costs amount to about ten times the mask costs. Given that a total of four to five different ASSP chips are typically required to build a high-speed switch fabric with QoS support [1]—two for the switch fabric core and two to three to implement the ingress and egress port cards devices—market segments have to be large to return a profit or they must exhibit a rapid market growth. Moreover, the recent slump in the telecommunications industry has also shown the risk of being tied to single market revenues.

These issues motivated our investigation of the feasibility of designing an ASSP chipset family that is generic and flexible enough to be used in several of the above-mentioned market segments. Our objective is to derive an appropriate architecture and investigate the design tradeoffs of such a generic switch fabric that meets this challenge. To this end, we proceed as follows. First, we compile a comprehensive list of the current requirements, needs and technological trends that we perceive as critical in the computer and communications network domains. Then we derive their implications on the switch design and identify the possible tradeoffs within the space given by

Manuscript received November 15, 2002; revised December 8, 2005 and June 6, 2007; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor F. Neri.

F. Abel, C. Minkenbergh, I. Iliadis, T. Engbersen, M. Gusat, and R. P. Luijten are with IBM Research, Zurich Research Laboratory, CH-8803 Rüschlikon, Switzerland (e-mail: fab@zurich.ibm.com).

F. Gramsamer was with the IBM Zurich Research Laboratory. He is now with bv Software Services Corporation, 6006 Lucerne, Switzerland.

Digital Object Identifier 10.1109/TNET.2007.909727

¹September, 2003, Applied Micro Circuit Corporation (AMCC) acquired certain assets and licensed certain IP associated with the IBM Switch Fabric IC chipset product line.

these requirements, trends, and combinations thereof. Clearly, these implications reflect the limits set on system costs, power consumption and scalability, and determine the framework for a practical implementation. Next we examine the potential of existing switch architectures according to the implications obtained, so as to identify the preferred architecture for our universal chipset concept. This study reveals that the most promising emerging architecture is the Combined Input- and Crosspoint-Queued (CICQ) architecture.

The outline of the paper is as follows. In Section II, we present the principal needs, requirements, and technological trends related to the design of a merchant switch fabric. Subsequently, in Section III, a comprehensive list of implications regarding the design of such a high-speed switch is derived. This leads to a requirements/trends/implications framework, which is used throughout the paper. In Section IV, we examine and evaluate various switching architectures using this framework developed. The candidate architecture for building a generic family of ASSP switching devices is identified. In Section V we address the challenges regarding the implementation of a switch fabric chipset based on this architecture. By considering state-of-the-art technology and applying the framework developed, we derive the characteristics of such a group of integrated circuits.

II. REQUIREMENTS FOR A GENERIC SWITCH FABRIC AND TECHNOLOGY TRENDS

A carrier class switch has different requirements and needs than a multiservice edge or an enterprise LAN switch. The same gap exists between a storage-area network and a blade server switch. Clearly, a switch design that integrates specialized features to support both communications and computer requirements will not be economically competitive with a commoditized switch. A similar disadvantage also exists in the performance comparison with high-range specialized switches such as those used in backbone routers or high-performance computing (HPC) systems. Therefore, we turn our attention to the mid-range class of electronic switches, i.e., switches that deliver between 200 Gb/s and 2 Tb/s of aggregate throughput.

Before designing such an advanced switch fabric, it is crucial that system architects consider all physical and engineering aspects, which are continuously evolving owing to various trends. When combined, these aspects can have major repercussions on the system cost, power consumption, and practical implementation. For example, in recent proposals of combined input- and output-queued (CIOQ) switches with limited speed-up of a factor of S , the bandwidth carried across the switch core is multiplied by S . This affects the interconnection technology, which has now become the scarce resource owing to the major cost and power outlay aspects associated with it [2].

We therefore proceed by first listing the main needs and requirements that a switching fabric should fulfill in order to be generic and useful in practice, and then pointing out the main technological trends in switch fabric design.

A. Generic Needs and Requirements

[R1] *Support for Large Number of Ports:* Although transmission line rates have increased rapidly in the past [OC-3 (155 Mb/s) to OC-192 (10 Gb/s) and OC-768 (40 Gb/s)], it appears that the main granularity at the switch level for the

next few years will be OC-192 [3]. Also, coarse and dense wavelength-division multiplexing (C/DWDM) vastly increases the number of channels available on a single fiber, but not the speed of a single channel. The same applies in the context of LANs and SANs, where connectivity is sought at the lowest possible cost.

[R2] *Low Cost Per Port:* There is an increasingly strong demand from customers to keep the cost of networking low, especially in LAN/MAN and StAN networks. The market also expects increased performance at constant or reduced cost.

[R3] *Low Power Consumption:* From one switch generation to the next, the market expects increased performance at constant or less power dissipation. We note that the power-consumption and power-dissipation issues have grown to the level where they have become a first order of design win/loss in the OEM switch-fabric market. This market concern directly translates into drastic power-consumption requirements per card (on the order of 150 W to allow forced air cooling) and per chip (on the order of 25 W to avoid hot-spots). Moreover, in Telco environments, customers demand high-end routers to be packaged as NEBS-compliant racks (see [R11]), which are subject to a fixed power-dissipation limit (typically 2 KW/rack).

[R4] *Generic Quality of Service (QoS) Support:* The typical QoS characteristics for data networks are bandwidth, delay, delay jitter, fairness, and loss guarantees. To achieve these guarantees, different networks use different QoS models. As IP only provides best-effort service and cannot deliver service guarantees on a per-flow basis or service differentiation among traffic aggregates, the Internet Engineering Task Force (IETF) has proposed two standard mechanisms to provide QoS in IP networks: Integrated Services (IntServ) and Differentiated Services (DiffServ). Frame Relay and ATM offer services based on a virtual circuit framework, whereas IEEE 802.1D builds on a strict prioritization scheme. Guaranteed QoS is less of a need in computer networks, where fairness and separation of resources into classes remain two of the primary requirements. A general-purpose switch should support each of these models.

[R5] *Low Latency:* Latency is the key metric of performance in computer networks because it directly impacts processor idle time, remote memory and I/O access time. A SAN or I/O network typically requires a latency of 4 to 10 μ s. Although latency is not the primary concern in communication networks, several applications—mostly high-growth ones, such as media-streaming, voice-over-IP, and online gaming—are quite sensitive to delay and jitter.

[R6] *Multicast Support:* Multicast is an important technique for an efficient delivery of packets to multiple destinations. Multicast traffic is widely used in LANs, and with the emergence of new services providing content-oriented distribution, media-streaming and “push”-oriented applications, this traffic is expected to increase in the WAN in the near future. Moreover, providing system support for collective communication in parallel interconnects and SANs will considerably reduce communication latency; see [4, ch. 5].

[R7] *Efficiency:* A commercial Internet infrastructure or a server farm for on-line transaction processing should perform well under high loads. Whereas raw bandwidth is “cheap” at the optical fiber level, operations performed at packet level, such as sorting, classification, shaping/policing, and routing, add cost at the fiber end-points. This is likely to emphasize the need for high

link utilization and high switch throughput to achieve efficient data transport.

[R8] *Support for a Variety of Heterogeneous Protocols*: A generic switch must be able to deal with numerous legacy technologies, especially in the LAN/WAN areas [5].

[R9] *Support of a Variety of Evolving Traffic Patterns*: The global Internet is an immense “moving target” because of the network’s great heterogeneity and rapid change. The heterogeneity ranges from individual links that carry the network’s traffic to the protocols that inter-operate over the links, the “mix” of different applications used at a site, and the levels of congestion seen on different links [6]. The same unpredictable traffic patterns also predominate in computing systems where workloads are application-driven.

[R10] *Reliability*: Global e-commerce and mission-critical Internet services require maximum availability (typically 99.999% uptime in a Telco environment) and a minimum of network outage. This also implies that loss of critical data will not be tolerated and that continuous operation must be guaranteed during repairs.

[R11] *Network Equipment Building System Compliance*: Network Equipment Building System (NEBS) compliance comprises a set of stringent physical (e.g., space planning, temperature, humidity, fire resistance, etc.) and electrical (e.g., EMI, power fault, bonding and grounding, etc.) requirements, originally developed for telephony equipment. Nowadays, NEBS compliance is a prerequisite for networking and computing equipment in general to ensure reliable operation (also under adverse conditions), safety, compatibility, and freedom of interference.² The maximum power consumption per board and per chip are typical consequences derived from these rules.

[R12] *Inherent Scalability and Evolutionary Migration Capabilities*: The ability to scale the line rate and to expand the port count is key for a generic chipset that targets a wide range of application markets with varying size and performance requirements. Furthermore, these migrations must follow an evolutionary path to preserve as much of the existing investment as possible.

[R13] *Lossless Switch Fabric*: Lossless operation might be required to avoid unnecessary packet retransmissions, which would waste bandwidth and increase latency. In particular, storage systems and parallel interconnects need reliable and strictly ordered delivery, as well as lossless switch operation [7].

Note that we do not address the aspects related to routers that are not of direct relevance for the hardware design of the core switching fabric as this would exceed the scope of this paper. Such aspects typically include routing protocols, network management, software reliability and stability. An explicit set of these requirements, recommendations, and options can be found in [8].

B. Technological Trends

In addition to the above requirements, high-speed electronic switch design should also take the following technological trends into account:

²Guidance and requirements such as “GR-63-CORE: NEBS Physical Protection” and “GR-1089-CORE: Electromagnetic Compatibility & Electrical Safety,” are typically specified by the Bellcore/Telcordia organization (<http://www.telcordia.com>).

[T1] *Link Speed Rate is Increasing*: The line rate has evolved exponentially to OC-192 currently, and will evolve to OC-768 in the near future. Regardless of whether traffic is ATM, IP, or SAN, the minimum packet size remains in the range of 32 to 64 B. Consequently, the transmission duration of the minimum-sized packet has shrunk from micro- to nanoseconds, and on a given length of optical fiber, cable or backplane, a larger number of packets are on the fly.

[T2] *Density of Line Cards is Not Increasing*: Contrary to the wavelength density, the port density in terms of ports per line card (LC) is not increasing significantly, for two reasons. First, availability requires that single points of failure be avoided as much as possible. Putting multiple network processors (NP) and switch ports on one line card would cause service interruption for several ports if one component on the card fails. Second, the increase in density offered by CMOS technology is invested into a richer set of features required at each port: firewall, traffic filters, security policies, VLANs (all at lower possible cost).

[T3] *The Maximum Switched Bandwidth Supported by a Single Switch Card is Limited by Card and Backplane Connector Technology*: The bandwidth growth over the card edge is a few percent annually because of the rapidly decreasing channel quality at multigigabit speeds (losses, dispersion, crosstalk), which limits the data rate and the density at which signal pairs can be packed. This growth is much smaller than that of the CMOS technology density. In the past, there was ample connectivity density to support the processing capabilities of CMOS technology. Today this has changed and interconnection bandwidth has become the scarce resource. As a result, the maximum aggregate throughput that can be implemented on a single switch card is limited by card and backplane connector technology rather than by CMOS density. Assuming a serial interconnect bit-rate of 2.5 to 3.2 Gb/s combined with the latest connector technology (VHDM-HSD) yields a density of about 120 Gb/s/in. With reasonable card sizes (≤ 50 cm card edge), this results in a maximum card throughput of about 1 Tb/s (bidirectional), although CMOS technology would allow us to achieve higher throughput on a single card. Consequently, larger switch cores (i.e., multi-Tb/s) have to be split over multiple cards. Here optical interconnect technology does not yet provide an improvement (in terms of the space and power needed), although it may be needed to cover longer distances between cards in multiple racks.

[T4] *Copper Cable Interconnects Remain a Strong Alternative to Optical Fibers in the Very Short Reach (1 to 15 m)*: “Optics” has replaced “copper” in long distance communications, and it is generally accepted that optics will eventually also be used for intra-system interconnects. This transition to optics is temporary postponed by the deployment of more advanced signal processing techniques (*pre-distortion, equalization*) that compensate for the signal attenuation at high frequencies. These signal processing techniques add design complexity and electrical power to the system but they avoid the costly move to optical interconnects (typically more expensive than copper cabling), for intra-system connections. Moreover, even if optical interconnects starts to exhibit good connector densities (Mini-MT), optical-electrical converters do not yet offer high integration density and low power consumption. For the backplane class of interconnects, it is considered that optics will not become a viable alternative to printed circuit boards before

individual line speed exceed 10 Gb/s [9]. On the other hand, the practical bandwidth * length product of the cable class of low-power interconnects is estimated at roughly 5 Gb/s over 8 m of standard coaxial cable [9], or 2.5 Gb/s over 20 m [10]. As a result, copper currently remains the best cost, power and density compromise to interconnect backplanes, racks and shelves on short distances (up to 5–10 m). However, crosstalk, weight and physical space remain challenging aspects of copper cabling.

[T5] *The Proportion of the Switch Power and Pins Used for Transporting Signals Across Chip Boundaries is Increasing:* Optical links continue to increase in speed much faster than electronic components do. Therefore, higher per-pin signal rates are needed to scale aggregate bandwidth proportionally, which in turn implies more pins and more complex analog macros, resulting in increased power consumption. At the system level, we also notice that a substantial amount of power is spent in moving rather than switching data.

[T6] *Integrated High-Speed, High-Density and Low-Power Serializers/Deserializers (SERDES) are Becoming Available in ASIC Libraries:* In response to the increasing system component count, cost and power consumption incurred by the use of external transceivers, ASIC vendors offer integrated SERDES cores and other high-performance I/O functions directly on-chip. Typical 1.25 to 3.2 Gb/s serial line interconnects are common, while 6.5 Gb/s to 12 Gb/s are the state of the art commercially available. Higher-performance interconnects are being developed, with experiments achieving 20 Gb/s under laboratory conditions [11].

[T7] *Moore's Law Paradox in the Latest CMOS Generations:* Although Moore's law doubles the performance of VLSIs every 18 to 24 months at constant cost, this performance improvement is mostly due to the increased density rather than the increased clock speed. With the ability to provide tens of millions of gates on a single die, gate density can be considered as arguably "free" in current CMOS technologies. As the typical gate delay of such gates is on the order of few tens of picoseconds, they can be operated at high clock speed. However, making the gates continuously smaller and more numerous has shifted the bottleneck to the chip wiring, turning it into the major limiting factor to the increase of the clock speed. The reason for this limitation is the increased resistance, capacitance and power consumption when the cross-section of the interconnect wires is reduced. From one CMOS generation to the next, switch-chip clock speeds can be increased by only 5% to 15%. Therefore, to exploit Moore's law, more parallelism is required, typically at constant clock speeds. This in turn results in more levels of pipelining in the control path and a higher degree of parallelism in the data path [12].

[T8] *Fixed Chip Costs Dilemma:* Fixed chip costs include the mask sets and the chip design costs. These costs typically double with each process generation, and they currently amounts to 10~15 M\$ per chip design in advanced 90 nm process. Given that advanced switch fabrics typically build on a family of ASIC devices, some fabric vendors can no longer afford the leading-edge CMOS processes because switch market volumes are too small to return a profit.

III. IMPLICATIONS ON THE DESIGN OF MERCHANT SWITCH FABRICS

Next, we derive the main implications that emerge from the requirements, the observed trends, or combinations thereof, and

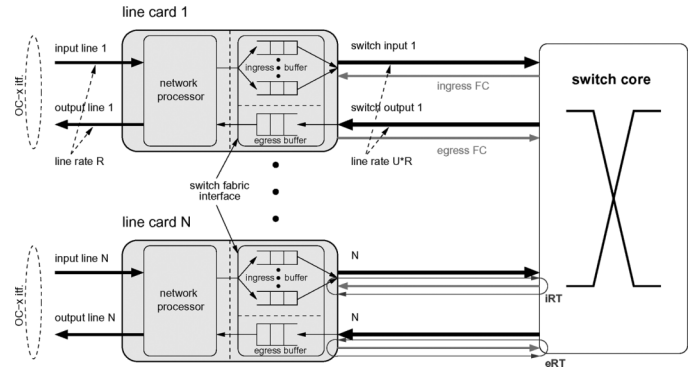


Fig. 1. Generic switch fabric model.

then elaborate on the directions regarding the design of our merchant switch fabric. We note that the focus is on packet switches for use in the distributed class of routers, where distributed forwarding engines are physically part of the line card. It has been shown that this approach is typically less expensive than the parallel class architecture, where separate banks of forwarding engines are maintained, isolated from the line cards [13].

In the following discussion we refer to the Switch Fabric (SF) as the combination of a switching core (SC) and a switch fabric interface, such as a line card of a host channel adapter (see Fig. 1).

A. Physical System Size

[I1] *Multi-Rack Packaging:* The increase in port count [R1] combined with the trends in line card density [T2] and switch-card density [T3] translates into an increase in physical space occupied by the line cards. Compliance with NEBS physical-packaging requirements [R11] imposes strict limitations on the number of cards that fit into a rack. Consequently, the switch fabric can no longer be built in a compact, single-rack fashion, and a multi-rack solution is necessary. This implies larger systems, cables and/or optical fibers replacing backplanes, and more transmission power within the SF. The multi-rack packaging entails significant physical difficulties, giving rise to implications [I1.a] and [I1.b]:

[I1.a] *Multi-Rack Interconnection:* Rack-spacing requirements and other spatial limitations (line card racks might not even be in the same room as the switch core rack) determine how far the racks have to be apart—this can easily be tens of meters. With regard to the power trend in data transport [T5], we consider two cases, which require different interconnection approaches, but should both be supported by the SF. First, if the rack spacing exceeds 15 m, there is currently no way to avoid the additional cost and power incurred by the two sets of optical conversion needed per interconnect. The electro-optical converters can be placed directly on the line and switch cards or on specific optical blades. Second, if part or all of the SF can be contained within a 15 m distance, then the latest advances in electronic cabling [T4] should be exploited to achieve low cost [R2] and power per port [R3]. With multi-rack interconnection accounting for a significant part of the overall system cost [2] and power consumption, the capability to drive the cables directly from the SF chips yields significant cost and power savings.

[11.b] *Multi-Rack Clocking and Synchronization:* The clocking of large synchronous systems is complex and becomes even more challenging at higher clock speeds [T1]. As a consequence, we will favor a plesiochronous mode of operation (small frequency differences and slowly-varying phases of the distributed clocks) to a synchronous one. Ideally, the multi-rack system should use asynchronous clocks.

B. Switch Capacity

In many switch application domains, the maximum switching capacity (or aggregate full-duplex throughput) is used as the guide of scalability. This switching capacity is the result of the number of ports (N) \times port rate (R) and can be increased by scaling either or both terms of this product. Depending on the primary metric of scalability [R12], port rate or port count, different implications for the switch fabric result:

[12.a] *Switch Degree:* Most of the switches currently used in communications areas solely base the scalability of their aggregate throughput on the port-rate scalability. This approach commonly consists of a centralized single-stage switch with a limited number of ports (16 to 32) operated at the highest possible rate. If the system has to handle larger numbers of slower external links, these are multiplexed onto a single higher link of the single-stage switch. For any given circuit technology, there is a limit to the applicability of this technique, but within its range of applicability it offers the most cost-effective way to scale to larger switch sizes [2]. Therefore, whenever the switch size does not exceed a few tens of ports, the market expects single-stage-based switches because they are efficient, fit into a single rack, and are easier to understand and manage.

Our merchant switch fabric must account for this market request by providing efficient and cost-effective operation in single-stage mode. Next, considering that the main granularity of the transmission line rates at the switch level for the next few years will be OC-192 (10 Gb/s), our switch fabric should provide 64 ports to meet the current hot-spot demand in the 160 \sim 320 Gb/s range [3], and to offer a migration path to the next fabric generation having a capacity of 640 Gb/s and higher.

[12.b] *Multi-Stage Topology:* Computer networks typically scale by increasing their port count rather than their port rate. As these networks can grow up to thousands of ports, we need to turn to multi-stage topologies. Here, we have a choice between *direct* (such as Mesh, Torus and Hypercube networks) and *indirect* (such as Beneš, Clos and Fat-Tree) topologies; see [4, ch. 1]. Although direct networks scale very well to large node counts using small degree switches, the number of hops and therefore the worst-case latency between remote nodes grow quickly. Given our challenging 4 \sim 10 μ s latency target and the recommendation for a switch degree of 64 [12.a], we therefore recommend an indirect multi-stage interconnection topology, namely, a Beneš or a Fat-Tree, which have the advantages of being non-blocking, offering high path diversity and short paths to neighboring nodes. A fat-tree network with S levels can scale to $M = N(N/2)^{S-1}$ nodes using $(N(S-1) + N/2)(N/2)^{S-2}$ switches, where N is the (even) switch degree. With $N = 64$, we can scale our network to 2 K nodes with just two levels of switches, i.e., there are at most three hops between two nodes. Moreover, adding one more stage will scale this network to 64 K nodes.

C. Board and Chip Design

[13.a] *Minimize the ASSPs Chip Designs:* Because of the costs levels of advanced CMOS processes, special attention must be paid to switch architectures that limit the number of chip designs.

[13.b] *Avoid Intermediate Drivers:* The power trend [T5] and requirement [R3] also apply at the board and chip levels. Therefore, we will banish the use of intermediate devices as much as possible by driving backplanes and cables directly from the switch chips [T6].

[13.c] *Distributed Architecture:* Although the use of on-chip serial high-speed technology helps reduce the overall power consumption and dissipation, this approach tends to shift part of the power issue of the system onto the switch chips. Assuming a SERDES with a typical power consumption of 125 mW (at 2.5 Gb/s), and five I/O pins per duplex channel (2 \times 2 differential signals + 1 overhead for power supply specific to the analog circuitry), 1 Tb/s of aggregate bandwidth requires more than 2500 chip pins and 60 to 70 W, which is not feasible at acceptable cost today. Therefore, to overcome this I/O limitation, we will select an architecture that can expand its data path over multiple chips operated in parallel. For example, four stacked chips, each of 256 Gb/s of aggregate bandwidth and operated in parallel, would provide a cost-effective package solution (640 I/O pins \approx 1000 pin package) and an acceptable power consumption of 16 W (for the I/Os only).

D. Switch-Fabric-Internal Round Trip

The increased physical distance as explained in [I1] combined with the link speed rate increase [T1] causes that:

[14] *The Switch-Fabric-Internal Round Trip (RT) is Significantly Increased:* We define RT to be the sum of both the number of packets on the fly over either backplane or cable (RT_{cable}) and the number of packets residing in the SERDES logic³ (RT_{logic}). A large RT used to be an issue only from node-to-node within a network, but has now become important also inside the switch fabric. Table I shows the evolution of the RT values for four switch-fabric generations (from OC-12 to OC-768), assuming a minimum packet length of 64 B. Note that the RT has shifted from fractional values over backplanes to tens of packets being on the fly over cable and optical fibers. The increased RT results in an increase of the amount of buffering required to ensure lossless operation [R13] and high link utilization [R7]. Note that the amount of on-chip buffering is limited by the CMOS technology deployed. Also, external memory solutions become increasingly expensive as they require a large number of pins and commensurate high power at speeds such as OC-768. As a result, the switch-fabric-internal RT has become a major design parameter for SF architectures.

E. General-Purpose Switching Fabric

[15.a] *Switching Paradigm:* The switching of a small cell size can be an attractive and efficient option to accommodate a large number of protocols [R8]. The cell-switching paradigm is well

³This accounts for the logic at the receiver and transmitter, such as clock recovery, data deskewing and alignment, arbitration and pipelining stages in both the data and the flow-control paths.

TABLE I
ROUND TRIP EVOLUTION

Line rate	OC-12	OC-48	OC-192	OC-768
Distance	1 m	1 m	6 m	30 m
Interconnect type	backplane	backplane	cable	fiber
Packet duration	819.2 ns	204.8 ns	51.2 ns	12.8 ns
Round trip	<< 1 Pkt.	~ 1 Pkt.	12 Pkt.	50 Pkt.

understood in the literature and provides the following advantages: i) it can be deployed with simple and therefore fast hardware; ii) it guarantees low transit delay and minimized delay variance (jitter) per node; and iii) it can recover from errors by adding error-correcting codes. However, the link speed rate increase at constant fixed cell size leads to reduced cell transmission duration [T1]. Eventually, the cell size must be increased to enforce a cell transmission duration that is longer than the minimum scheduling duration, which requires a system that supports the switching of a fixed but programmable cell-size. We refer to such a cell of programmable length as a “packet”.

Moreover, the advantages provided by packet switching must always be balanced against the overhead entailed by the fragmentation of variable length data into packets. This overhead includes the segmentation and reassembly (SAR) at the switch fabric edge, and the bandwidth taken up by the packet header. Both must be compensated for with additional bandwidth and/or speedup, which translate into a significant increase in power consumption [T5] and costs [I7.d].

Therefore, the preferred switching paradigm should be able to switch very short packets and/or variable-length packets.

[I5.b] *Switch Virtualization*: The power [R3] and cost [R2] constraints inevitably drive the need for increased network efficiency and consolidation. This leads to network virtualization, a new trend that dynamically divides the available bandwidth into multiple, independent channels. Network virtualization arrives at the switch with a superposition of many different protocols [R8], packet formats, and traffic patterns [R9], all on a single physical interconnect. To support such a mixture of protocols and traffic distributions, the SF must be multi-standard-compliant at the border interfaces (e.g., Network Processing Forum-Streaming Interface, CSIX, Infiniband, RapidIO), and its performance cannot be tied to any particular traffic or protocol.

F. Generic QoS Support

The attempt to support different QoS requirements [R4] such as specified by ATM, IP-DiffServ, IP-IntServ, MPLS, IEEE 802.1D, and Infiniband, is one of the most challenging goals of this chipset. The behavior of the resulting QoS model must remain generic and adaptable to any of the technologies listed above.

[I6.a] *Traffic Classification and Selective Queueing*: Differential treatment of traffic categories is achieved through selective queueing and scheduling. This immediately entails the difficult problem of defining the appropriate number of queues to be implemented within the fabric. The answer to this question partially depends on the traffic management policy used at the ingress of the switch fabric:

- 1) The fabric is operated with an advanced traffic manager that enforces the QoS guarantees at the ingress of the

switch fabric. In that case, it is sufficient to provision guarantees to no more than three distinct and generic classes of service (CoS): the guaranteed-delay (GD) class, the guaranteed-bandwidth (GB) class, and the best-effort (BE) class [14]. Note however that a switch fabric providing selective feedback control (backpressure) per destination will only need to support two CoS classes: one for guaranteed delay requirements and one for non-delay critical traffic [15].

- 2) The fabric does not enforce QoS at the ingress side and performs a limited level of traffic management. This applies to mechanisms such as in IETF DiffServ [16] and IEEE 802.1D [5], which are based on aggregation per traffic type or per hop behavior (PHB), and for which more queues are required (typically eight) to maintain the service or priority level of differentiation among different classes.
- 3) The fabric does no traffic management. This is mainly the case in computer interconnects, where QoS is less useful if not inexistent because of the self-throttling behavior of the network. In such networks, a minimum of two queues is typically required: one for the short control messages, the other for the long data structures. However, as these queues are used as virtual lanes/channels, providing more queues will increase the network throughput to some extent by providing the necessary separation of resources that avoids deadlocks and the building of saturation trees [17].

Therefore, given that the predominant IP QoS mechanisms adopt the differentiated service approach [18] and that Ethernet is the most wide-spread LAN access technology, we recommend to provide a set of eight queues in the switch to primarily accommodate the needs of these QoS models.

[I6.b] *Service Scheduling*: When multiple queues corresponding to the same class or priority require service, a second level of scheduling mechanism is required among them. This mechanism should exhibit the following desirable properties of the ideal fluid Generalized Processor Sharing (GPS) algorithm [19], assumed here as reference model: (i) minimum bandwidth guarantee; (ii) protection and isolation of the sources from each other; (iii) fair bandwidth allocation; and (iv) deterministic bound on the worst-case end-to-end delay. Property (iv) is usually the most challenging one to achieve, especially for stringent delay guarantees. Although Packet Fair Queueing (PFQ) algorithms such as [20] have been proposed to approximate the GPS fluid service discipline in high-speed networks, their complexity makes them less suitable for implementation in high-speed fabrics. Therefore, mechanisms providing most of the features mentioned above but with lower implementation complexity—such as Weighted Round-Robin (WRR)—should be selected.

G. Implementation Cost and Complexity

The discussion in Sections III-A and III-B shows that power is becoming a critical design consideration. To achieve acceptable cost-per-port performance [R2], power must be considered in conjunction with the following secondary aspects, which can also impact the final system cost:

[I7.a] *Chip Packaging and ASIC Die Size*: The knee of the VLSI-cost-versus-die-size curve currently lies in the range of 250 mm². The knee of the chip-cost-versus-package curve today

lies in the range of approx. 1000 I/O signals per chip package (total ≈ 1500 pins package).

[17.b] *VLSI I/O Technology*: Because of system's power consumption and dissipation [I3.b], switches have become critically dependent on serial high-speed and high-density I/O technology. These links must be available in the ASIC technology library used by the switching chips and must have a low power consumption.

[17.c] *Standard Versus Custom Design*: The use of standard cell-design methodologies and processes significantly reduces the VLSI design time.

[17.d] *External Link Speed-Up*: External link speed-up affects system cost and power consumption, and therefore should be avoided as much as possible.

IV. THE PREFERRED ARCHITECTURE

In this section, we review a spectrum of switching architectures and identify the architecture that emerges based on our overall system design requirements, technology trends and implications.

Most current single-stage switch architectures use virtual output queueing (VOQ) at the ingress of the fabric [21]. The key feature differentiating such architectures is whether the scheduling of the ingress VOQs is *centralized* or *distributed* (Fig. 2).

A. Centralized Scheduling

The centralized approach typically uses N input buffers organized by destination (VOQ) combined with an $N \times N$ bufferless crossbar. In this centralized scheduling category, we can further distinguish between approaches without speedup, which are purely input-queued, and those with limited speedup, which are CIOQ [Fig. 2(a)]. Because the core is bufferless, this speedup applies to the ingress-buffer read access, the egress-buffer write access, the arbitration process, and the entire switch core, including links. Both approaches require a centralized arbitration algorithm to resolve input and output contention. The purely input-queued approach requires a bipartite-graph-matching algorithm, such as PIM or iSLIP [22]. For the CIOQ approach, researchers have proposed more complex arbitration algorithms to achieve exact output-queueing emulation, which produces better QoS support and performance, particularly under nonuniform traffic.

Design consequence [I1] implies that multiple racks are required and that at least some—if not all—of the line cards are at a significant distance from the switch core. Given such an arrangement, we consider two possible options for the physical placement of the VOQ:

- 1) The VOQs are placed as close as possible to the crossbar, either on the same card as the scheduler or preferably on other cards in the same rack because of the backplane connector limitations [T3]. In this case, N extra chips with VOQ, buffering, label lookup, and flow control are required, thus inefficiently duplicating functions already implemented in the ingress line cards. This addition of chips is also in conflict with [I3.a] and [I3.b] as it leads to higher design costs and power consumption.
- 2) The VOQs remain on the line cards. In this case, the scheduler receives requests from the N^2 VOQs, performs

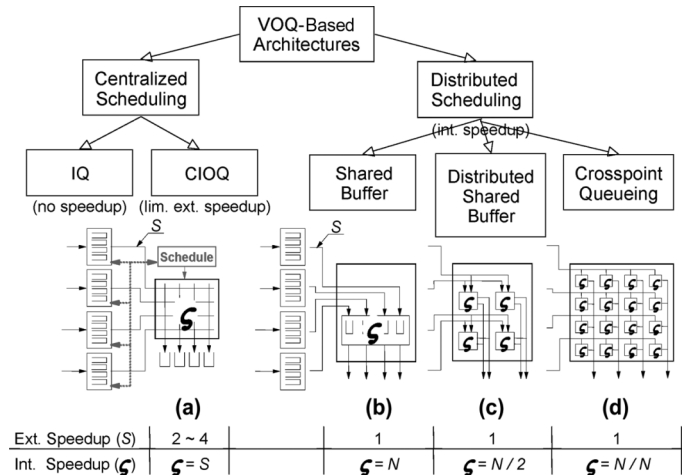


Fig. 2. Switch architectures with VOQ.

the calculation, and returns the corresponding grants. However, the matching computation is sub-optimal because there is a delay between the scheduling decision and its effect because of the long round trip [23]. Furthermore, the information transferred between the line cards and the scheduler results in additional bandwidth and power claimed on long cables because of [T5].

Whatever the placement of the VOQ is, the spatial distribution of the line cards entails a violation of many of the engineering recommendations that have emerged from the discussion in Section III. Another drawback of the centralized approach is the limited scalability of its centralized arbitration. First, it requires a high degree of connectivity and tight coupling (synchronization) between the arbitration unit and all ingress line cards to exchange requests and grants (this is in conflict with [I1.b]). Second, the arbitration algorithm must converge within a few packet cycles, which becomes extremely challenging at high line rates and large port counts because of its the wiring density [T7]. Third, significant latency is added, also at low traffic loads, because requests and grants must always be exchanged between an ingress line card and the central scheduler, before a packet can be transmitted. Moreover, QoS and multicast support turn out to be complex [24] and not trivial to implement without significantly speeding up the fabric by a factor of $S = 2$ to 4 [25]–[27].

The main implication for this speedup is that the chip and card bandwidth bottleneck problem is aggravated by a factor of S because the entire fabric, including the links to the line cards, must run S times faster. Because of [I1.a, I1.b, I3.b, I7.d], this is prohibitively expensive in terms of both hardware and power. Roughly speaking, because of [T3], the physical size (number of cards) of a switch core that is larger than 1 Tb/s must grow by a factor of S .

Note that most of these drawbacks are exacerbated by multi-rack implementation and that they are ways to limit their effects when the fabric is self-contained in a single rack. This is the reason why the centralized CIOQ remains such a popular and successful architecture in the current hot-spot market of 16 to 32 port switches.

B. Distributed Scheduling

The distributed approach eliminates the need for centralized input arbitration by using a limited number of output buffers, typically integrating them in the switch core [Fig. 2(b), (c), and (d)]. In other words, this method creates a CIOQ architecture based on a buffered switch core with an internal speedup of ζ . This also eliminates the need for a speedup of the external links to compensate the scheduler inefficiencies. This approach requires link-level flow control between the input and output buffers to ensure losslessness [R13] and prevent output buffer monopolization (hogging). Typically, such architectures use on/off or credit-based flow control. Such a combination of VOQ and output buffering has been shown to achieve excellent performance [28].

The main drawback of distributed CIOQ is the complexity of implementing output queueing with speedup. The traditional implementation uses a shared-memory switch core, where all inputs and outputs simultaneously access a common memory [28], [29], thus requiring an internal read and write speedup of $\zeta = N$ [Fig. 2(b)]. However, with current technology this implementation does not scale to multiterabits per second. Moreover, for architectures that use VOQ at the ingress to reduce head-of-line blocking, the performance advantage of a shared memory is no longer compelling.

This situation calls for a more distributed output buffer implementation, such as can be achieved by partitioning the shared memory into dedicated buffers for groups of inputs or outputs [Fig. 2(c)]. Taking this approach to the extreme leads to the well-known classic buffered crossbar architecture, which provides a dedicated buffer, with read and write speedup of $\zeta = 1$, for every input-output combination [Fig. 2(d)]. Several groups have recently proposed to combine such a buffered crossbar with VOQ ingress buffers—the *combined-input-crosspoint-queueing (CICQ)* architecture—and have demonstrated performance very close to ideal output queueing [30]–[33]. The performance depends on the contention-resolution mechanisms for the VOQs and the buffered crossbar outputs, for example, RR_RR⁴ [30], OCF_OCF [32], and LQF_RR [33].

In a CIOQ architecture, contention resolution is distributed over both inputs and outputs: N independent input schedulers resolve input contention, whereas N independent output schedulers resolve contention among packets in the output queues for transmission at the output ports. This results in a simpler, more distributed implementation, because $2N$ schedulers of $O(N)$ complexity are required instead of one centralized $O(N^2)$ scheduler. This exploits latest trends in CMOS technology [T7], i.e., greater density (more parallelism) rather than increased clock speeds (faster logic). Finally, the decoupling of the arrival and departure processes in the switch core relaxes the overall system-synchronization and clocking constraints [I1.b].

C. Multi-Stage Fabric

At this point we are left with two candidates, namely, the crossbar-based CIOQ and the CICQ. With our insight that multi-stage switch fabrics are also required [I2.b], we turn our attention to the use of these two architectures as building blocks for

a multi-stage topology. We use the term of switch element to refer to a particular switch of the multi-stage fabric.

The bufferless nature of the crossbar used in the CIOQ architecture gives rise to the important question of the buffer placement. To avoid intermediate chips and to reduce latency, it is desirable to interconnect the crossbars between themselves and to implement buffers only at the perimeter of the multi-stage fabric. Matching algorithms for multi-stage bufferless networks exists [34], but are not practical for fabrics with thousands of ports. Therefore, buffers are required between stages, and each stage requires a dedicated scheduler. Another reason for output queues is the typical use of speedup to compensate the scheduling inefficiency. Therefore, a CIOQ is not an efficient way to build multi-stage topologies because of the number of parts required per stage: a crossbar, a scheduler, an input and an output buffer adapter. Output buffers of stage x can in principle be combined with input buffers of stage $x + 1$, but this implies that the full speedup must be carried over all distances between stages.

A CICQ does not suffer from all these drawbacks. The buffered crossbar that it implements has its own internal scheduler per output port and does not require output queues. Considering the complexity and cost tradeoffs of the two alternatives, we retain the buffered crossbar as a building block for our multi-stage switch, and conclude that CICQ emerges as the most promising architecture to build a new family of ASIC devices for next-generation merchant switch fabrics.

V. PRACTICAL VLSI DESIGN AND IMPLEMENTATION

In the preceding section, we have identified the CICQ architecture as the most promising topology with regard to the requirements/trends/implications framework for our generic chipset. However, there is a potential showstopper for this architecture, which is the VLSI implementation of the buffered crossbar with a large number of ports. Practical realization of buffered crossbar switches exist, but they are mostly limited to small switch degrees (< 16) because of their quadratic memory requirement $O(N^2)$.

Therefore, according to current state-of-the-art technology we aim to i) prove that this architecture has become feasible for larger switches because of the most recent advances in CMOS integration density; ii) find the largest possible single-stage switch fabric that can be built with current technology; and iii) derive, from this results, the suitable switch element for later use in multi-stage Beneš and fat-tree topologies.

The proof of concept and the maximum switch size can be found in [35], where we evaluated the design and performance of a 4 Tb/s version (64×64 with 64 Gb/s/port) of this CICQ switch element. In this section, we intend to summarize the chief implementations challenges and design tradeoffs made to achieve such a high switch degree and port rate in spite of the stringent requirements imposed by the merchant switch fabric market. As our primary concern was the single-chip implementation of the N^2 crosspoint memories and their control, we only considered the RR arbitration policy for the N crosspoint schedulers. Similarly, we did not address variable packet scheduling and multi-stage support.

A. CICQ Architecture

A CICQ architecture is shown in Fig. 3. It consists of a buffered crossbar switch combined with a VOQ arrangement at

⁴These XX_YY notations indicate the combination of scheduler disciplines, with XX denoting the VOQ arbitration and YY denoting the buffered crossbar output arbitration. RR is round-robin, OCF is oldest-cell first, and LQF is longest-queue first.

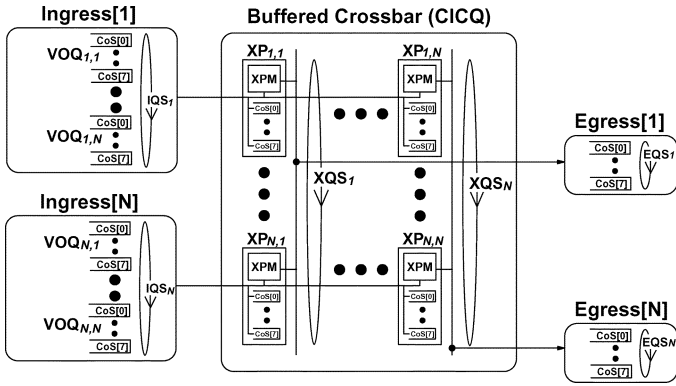


Fig. 3. CICQ architecture with classes of service.

the ingress. In this distributed architecture, both ingress cards and the buffered crossbar maintain a set of dedicated output queues (per input/output pair), for which the read operation is managed by a specific scheduler. Accordingly, this queueing arrangement provides the GPS *protection* property [I6.b] by isolating traffic sources from each other, and is particularly suitable to support RR, WRR or fair queueing algorithms.

A buffered crossbar of size $N \times N$ consists of N^2 crosspoints ($XP_{i,o}$), and N crosspoint schedulers (XQS_o)—one per output port—that select which packets are to be transmitted next. We further partition the crosspoint into a data section ($XPM_{i,o}$), which provides the buffer space to store the incoming packets, and a control section ($XPC_{i,o}$), which implements a queueing structure for the differential treatment of the traffic classes. The control section has much smaller memory requirements than the data section because it only stores packet descriptors, i.e., pointers to the packets stored in the corresponding $XPM_{i,o}$. Following recommendation [I6.a], the control section implements a structure of eight queues per crosspoint to sort the incoming packets per class of traffic. (Note that the addition of CoS queues to the CICQ architecture significantly increases the complexity of the crosspoint scheduler, which now has to perform service scheduling [I6.b] in addition to the contention resolution.)

The dedicated queues do not allow any buffer sharing, and the total buffering requirement is large $O(N^2)$. However, this scheme is inherently free of the *buffer-hogging* effect, which can seriously degrade performance in shared-memory switches that do not provide efficient congestion-control mechanisms. In particular, it provides a better performance under bursty traffic conditions than the shared buffer architecture does [36]. Unlike a bufferless crossbar, multiple line cards can simultaneously transmit packets destined to a particular output port in a time slot. The contending packets are first stored in the corresponding crosspoint queues, and then transmitted to the output by a dedicated crosspoint scheduler. Note that the N crosspoint schedulers are operated independently of each other, which is the key feature that enables variable-length packet switching.

B. Maximum Switch Degree

The VLSI implementation of a buffered crossbar can be limited by i) the chip pin count; ii) the I/O power consumption; and iii) the die size area needed to implement the data section memories of the N^2 crosspoints. We address these three factors in reverse order of their importance:

- i) Implication [I7.a] recommends the use of a chip package with no more than 1000 I/O pins. Let us assume that 80% of the I/O pins are used to implement the SERDES channels, and that a SERDES consists of five I/O pins [I3.c]. The resulting maximum module size is a bufferer crossbar with 160 ports.
- ii) Requirement [R3] specifies that the chip power consumption should not exceed 25 W. Let us assume that 75% of the power is solely used by the high-speed I/Os and that the typical power consumption of these I/Os is 125 mW per duplex channel [I3.c]. The resulting module is a buffered crossbar with a maximum of 150 channels or ports.
- iii) From our feasibility study [35], we learned that approx. 1.25 MB of on-chip memory is achievable in the latest 0.11- μm CMOS technology if a matrix organization of multiple small memories and a standard die size of 200 mm^2 (as a consequence of [I7.a]) are used. The resulting number of ports depends on the minimum packet size (e.g., 64 B) and on the required number of packets per crosspoint (XP).
 - $N = 71 \rightarrow 260 \text{ B}$ ($4 \times 64 \text{ B}$) per XP
 - $N = 64 \rightarrow 320 \text{ B}$ ($5 \times 64 \text{ B}$) per XP
 - $N = 48 \rightarrow 568 \text{ B}$ ($8 \times 64 \text{ B}$) per XP

As expected, the size of the die area that implements the crosspoint memories is the limiting factor for the VLSI implementation. For a buffered crossbar, this memory integration density is not optimum because of the many small memory macros that need to be instantiated, translating into a significant overhead incurred by the control logic of each memory.

C. Scaling the Module Capacity

As the raw circuit density typically doubles from one CMOS generation to the next, it will take two generations before the number of ports can be doubled while keeping the crosspoint capacity constant.

However, as the module is currently neither power- nor pin-limited, there is still room for this architecture to scale its capacity by increasing the line rate (as recommended by [I2.a]). The line rate can be increased (linearly) either by operating the XP memory faster or by increasing its width (w), or by a combination thereof. Let us take the chip module used in our 4 Tb/s demonstrator as example: it supports a port rate of 2 Gb/s with $w = 2 \text{ B}$. This combination corresponds to a XP memory access time of 8 ns, which is at least four times slower than what advanced CMOS can deliver. Consequently, there is room for a fourfold access time decrease, which permits a fourfold line-rate increase. On the other hand, higher line rates can also be achieved by increasing w up to the minimum packet size. For example, in the case of a minimum packet size of 40 B, the line rate can be increased 20 times by increasing w from 2 to 40 B.

Accordingly, this scaling technique of adding more or faster links to the module can be used as long as I/O pin and power consumption requirements are not exceeded. Note however that a larger w affects the granularity of variable packets (if supported) because more bytes are written per memory cycle.

D. Scaling the Fabric Port Speed

More links at constant link speed add pins to the module. Faster links at constant link count increase power and area

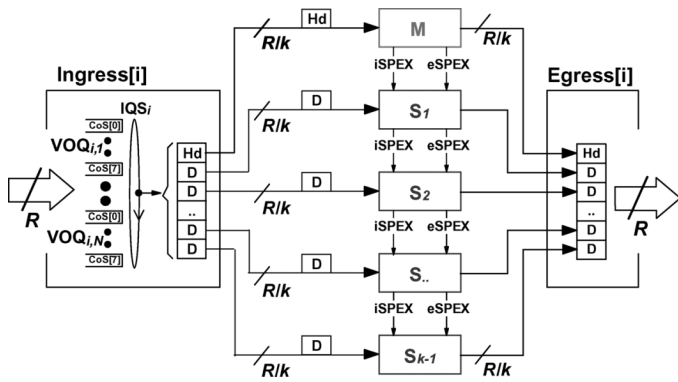


Fig. 4. Speed expansion concept.

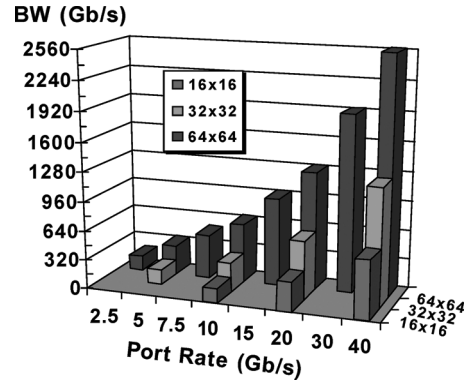
needs because more complex signal-processing techniques are required, such as equalization or multilevel signaling. Therefore, the aggregate bandwidth per module cannot exceed 320–400 Gb/s because of our stringent 1000 pin and 25 W power requirements.

To build a multiterabits switch fabric that overcomes these pin and power limits, the switch capacity must be *sliced* and distributed over multiple chips. We propose to divide the port rate (R) and buffering capacity (M) over k switching modules operated in parallel. Each module is an $N \times N$ buffered crossbar with a buffer capacity of $m = M/k$ and a port rate $r = R/k$. This technique of *bit slicing* [17] has been successfully used to scale shared-memory architectures up to 512 Gb/s [37], and is often referred to as port speed expansion (SPEX) or distributed switch architecture with centralized control.

Fig. 4 shows the concept of the SPEX arrangement when used in the single-stage mode of operation. It consists of one master chip (M) and $k - 1$ slave chips (S). Incoming packets are sliced by the ingress interface into k identical segments before being sent to the divided switches over k different links, each operating at $r = R/k$. The packet header is sent only to the master slice, whereas the other segments containing only data payload are transmitted to the slave slices.

When the master receives a segment from input port i , it decodes the XP destination from the header information, generates a storing address for the incoming packet within the decoded destination XP, and queues this storing address according to further routing and QoS information that the header carries. At the same time, control information is forwarded to all slave chips over the ingress speed expansion $iSPEX(i)$. This control information tells all slaves what kind of data segment they are currently receiving from their input port i and how they should handle it, i.e., whether it is an idle, control or unicast/multicast data segment, what the destination XP(s) within row i is (are), and what the store address within that (those) XPM(s) is (are).

Every time the master schedules a packet for transmission, the corresponding data payload segments must be retrieved from all slave chips. Therefore, a similar process takes place at the egress side of the sliced switch over N egress speed expansion interfaces (eSPEX). The control information transmitted on $eSPEX(o)$ is forwarded to the output controller o of all slaves. This control information specifies which XP to select within the column controlled by output o , and the address to read from within the selected XP.

Fig. 5. Scaling with SPEX ($r = 2.5$ Gb/s, $N = 64$).

Note that the daisy-chain implementation of the SPEX interfaces shown in Fig. 4 is for sake of clarity only. Indeed, while this slice interconnection might be acceptable for a small number of slaves (1 to 4), it translates into a significant latency penalty for a large number of slaves. In that case a broadcast interconnection (as used in [35]) is much more suitable.

Speed expansion is the most straightforward and efficient way to scale the capacity of the single-stage fabric. The method is also cost effective because a single slave design enables the deployment of a wide range of switch-fabric capacities, ranging from $N \times N$ at port rate of r when instantiating one slave, up to a $N \times N$ at port rate of kr when using k slaves.

By applying internal speed expansion, the extent of the capacity scale can be further increased for those cases where a smaller switch degree than 64 is desired. Internal SPEX applies the expansion idea within the module itself and comes at small additional complexity cost. The concept consist of combining two ports (say i and $i + 32$) or four ports (say i , $i + 16$, $i + 32$ and $i + 48$) of the same $N \times N$ module at port rate r to realize a $N/2 \times N/2$ module at port rate $2r$ or a $N/4 \times N/4$ module at port rate of $4r$. Fig. 5 shows the internal and external SPEX combinations supported by our flexible design, for $k = \{1, 2, 3, 4, 6, 8, 12, 16\}$, port rate $r = 2.5$ Gb/s and switch degree $N = 64$.

E. Scaling the Fabric Buffering

In a buffered crossbar, the size of the crosspoint buffer is the key parameter that determines the performance and the correctness of the system.

Crosspoint-buffer dimensioning has been addressed in [30], where the relation between the XP size and the performance is shown under various degrees of unbalanced traffic. To achieve 100% throughput under any traffic pattern [R7][R9]—while no input or output is oversubscribed—the crosspoint memory size must be greater than or equal to N , or the buffered crossbar should be operated with a 10%–20% speedup. They are, however, two other major parameters that impact the minimum crosspoint buffer: the presence of a switch-fabric-internal round trip and the flow-control mechanism.

a) *Flow-control mechanism*: Flow control (FC), round-trip and buffer requirements are strongly related to each other. Here we discuss the choice of FC mechanism based on the corresponding buffer size required for lossless operation without starvation [R13], and the associated bandwidth overhead [R7].

Given a communicating channel with an RT of τ packets, the two suitable candidate schemes are the on/off and the credit-based flow control. On/off FC requires a buffer memory of at least 2τ packets and a large bandwidth overhead because of its stateless protocol [17]. The credit FC scheme on the other hand requires a buffer memory of τ packets [38], and the bandwidth overhead can be contained to a desired level. We therefore propose the use of the credit FC scheme because of its lower buffer requirements. We discuss the topic of bandwidth overhead and bandwidth-induced latency in the case of limited FC channel capacity in [39].

b) Switch-fabric-internal round trip: Credit FC control inherently satisfies the losslessness property, but to satisfy the work-conservation property under any traffic pattern, we must scale the size of the crosspoint buffer proportionally to the RT, which translates into at least RT credits being available to fully utilize the link (memory size and credits are linear functions of RT). This ensures that any ingress port can transmit to any egress port at any instant and at full rate—for example, under directed traffic or in the absence of output contention, when traffic should always proceed at the maximum rate.

With [14] we have defined the switch-fabric-internal RT to be $\tau = RT_{\text{total}} = RT_{\text{cable}} + RT_{\text{logic}}$, where RT_{cable} is given by $RT_{\text{cable}} = (2RD)/(S_L P_S)$, with R the port rate in bit/s, D the (one-way) cable length in m, S_L the propagation delay over dielectric (typically assumed to be 2×10^8 m/s), and P_S the packet size in bits. Consequently, the minimum crosspoint buffer size of backplane-type interconnects ($1 \text{ m} \leq D \leq 2 \text{ m}$, $2.5 \text{ Gb/s} \leq R \leq 5 \text{ Gb/s}$) is solely determined by RT_{logic} . On the other hand, when the physical system size grows, which typically is accompanied by an increase in port rate, then the product RD starts to represent a significant part of the RT. In our design study with $R = 64 \text{ Gb/s}$, the contribution of RT_{cable} overtakes the RT_{logic} when $D \geq 24 \text{ m}$, and there are more than 4 KB in flight when $D \geq 51 \text{ m}$.

In Section V-B, the case when $N = 64$ translates into a maximum crosspoint memory size of 320 B ($5 \times 64 \text{ B}$), which is far from the 4 KB required when $\tau = N$. However, remember that the port rate of 64 Gb/s is obtained by slicing the fabric 32 times, and that SPEX not only expands the port speed but also the crosspoint memory capacity. Therefore, providing $32 \times 320 = 10 \text{ KB}$, which solves the scaling issue of the XP buffer size.

To further guarantee a maximum rate for C classes of service, τ packets must be further allocated to every CoS in the XP memory. The worst-case traffic scenario that leads to this memory requirement is described in [40]. To avoid such a C -fold increase of the XPM, we propose to use the priority-elevation mechanism described in [41], which allows the crosspoint memory size to be reduced from $\tau \times C$ to $\tau + (C - 1)$ packets.

F. Scaling the Fabric Port Count

In Section V-B, we have derived a maximum switch degree of 64, based on the pin, power and die limitations. This switch size, however, does not satisfy some of the current needs for switches with large numbers of ports at rates of 1 to 10 Gb/s [R1].

As already mentioned in [I2.a], one cost-effective way to build a switching system handling larger numbers of ports is to multiplex multiple (e.g., four or eight) external links onto

a single higher-speed link. Because of the widespread of the OC- x rate multiples, we specifically focused on the physical link being shared by four external links which allows a 256×256 fabric to be built. At the system level, this translates into a switch fabric that supports 64 full-duplex physical ports, each of which can be configured as either one full-rate (e.g., OC-192) interface or four quarter (OC-48 Gb/s) interfaces. We refer to the fabric-external links operated at full rate as ports (P) and to the fabric external links operated at quarter rate as sub-ports (SP).

To maintain internal non-blocking behavior between sub-ports that share the same physical output link, the packets destined to different SPs must be written into dedicated queues to guarantee separation of resources [I6.a] and to allow CoS scheduling at the current contention point [I6.b]. The resulting queueing structure at every crosspoint is then multiplied by four, for a total of 32 queues (4 SPs times 8 CoS). Similarly, buffer sharing among SPs is not possible here because of buffer *hogging*, which would create inter-blocking situations between SPs. Note that the addition of SPs does not impact the XPM dimensioning. Although the SPs multiply the amount of logically partitioned buffers by a factor of 4, the RT at the SP level is also reduced by factor of 4, so that the total amount of buffering remains constant. At the chip level, this translates into an arrangement of 128 K queues ($64(\text{P}) \times 256(\text{SP}) \times 8(\text{CoS})$). Note that for the same queueing and scheduling complexity (128 K queues), a 512×512 or 1024×1024 switch configuration can also be realized if the number of traffic classes is reduced to 4 or 2, respectively.

The sizing of such an implementation is addressed in [35]. It is shown there that this arrangement can be implemented in 90 nm CMOS technology. However, given our chip area [I7.a] and power [R3] limits, there is no space left to implement the buffer space that stores the incoming packets. This leads to a chipset consisting of two devices: a master chip that implements the 64 output queue schedulers and the control section of the 4096 crosspoints, and a slave chip that implements the data section discussed in Section V-B. One drawback of this chipset approach is that it requires two chip designs and that the master can only handle the packet header but no payload.

Another attractive approach is to decrease the switch degree—in the range of 32 to 40 ports—while increasing the port rate to 5 or 6.25 Gb/s (Section V-C). This keeps the module capacity constant, while significantly relaxing the implementation density. As a result, a unique design is now required because both data and control sections will fit into the same device. This implies that the devices that are operated as slave slices always have their control section disabled.

Finally, because of the non-blocking behavior of the port and sub-ports, more slow ports can always be added at the switch fabric edge (e.g., 10,000 OC-3), if the added ports are operated with hierarchical link sharing in a cooperative and regulated traffic environment [42].

VI. CONCLUSION

We have addressed the development of a switch fabric based on a commercial ASIC chipset. This group of integrated circuits can be deployed in the domains of both communications and computer interconnection networks and satisfies the different requirements of a wide variety of services and applications. A

systematic method was introduced to achieve that goal. First, we provided a comprehensive list of the basic requirements and current technological trends, and then considered their implications on the switch fabric design. This framework was subsequently used to identify the architecture upon which such a suitable and generic switch fabric could be based. By considering the relevant aspects, such as power consumption and design costs, and by also addressing the interconnection resources that now also need to be considered as scarce, the Combined Input- and Crosspoint Queued (CICQ) architecture was identified as the most promising architecture. We then presented the general characteristics of an enhanced CICQ packet switch based on a group of integrated circuits using state-of-the-art CMOS technology. This ASIC chipset only consists of two devices, but enables the deployment of a wide range of switch-fabric capacities, ranging from 128 Gb/s up to 4 Tb/s of aggregate bandwidth.

Finally, we note that most research on high-speed packet switches focuses on high-level architectural issues such as buffering, queueing arrangements, and scheduling algorithms. Although these are important issues, there are other equally significant aspects that arise when actually building a system and which are not always taken into account. One of the main contributions of this paper is to identify and catalogue the important tradeoffs such as chip count, power consumption and packaging.

ACKNOWLEDGMENT

The authors extend special thanks to the Prizma Technology group of IBM La Gaude, France, who started them thinking in this direction. They also thank the switch team of the IBM Zurich Research Laboratory, Switzerland, and the logical and physical design team of IBM Böblingen Laboratory, Germany, for their contributions.

Finally, the authors are particularly indebted to Prof. Jose Duato, Technical University of Valencia, and Alan Benner, IBM Systems and Technology Group, Poughkeepsie, NY, USA, for their substantial contributions to this architecture.

REFERENCES

- [1] F. M. Chiussi, J. G. Kneuer, and V. P. Kumar, "Low-cost scalable switching solution for broadband networking: The ATLANTA architecture and chipset," *IEEE Commun. Mag.*, vol. 35, no. 3, pp. 44–45, Dec. 1997.
- [2] J. Turner and N. Yamanaka, "Architectural choices in large scale ATM switches," *IEICE Trans. Commun.*, vol. E81-B, no. 2, pp. 120–137, 1998.
- [3] J. Bolaria and B. Wheeler, *A Guide To Switch Fabrics*. Mountain View, CA: The Linley Group, 2002.
- [4] J. Duato, S. Yalamanchili, and L. Ni, *Interconnection Networks, An Engineering Approach*. San Francisco, CA: Morgan Kaufmann, 2003.
- [5] *Standard for Local and Metropolitan Area Networks: Overview and Architecture*, IEEE Std 802-2001, LAN/MAN Standards Committee, IEEE Computer Society, Feb. 2002.
- [6] S. Floyd and V. Paxson, "Difficulties in simulating the Internet," *IEEE/ACM Trans. Networking*, vol. 9, no. 4, pp. 392–403, Aug. 2001.
- [7] *Infiniband Architecture*, Specification 1.0.a, Jun. 19, 2001.
- [8] F. Baker, "Requirements for IP version 4 routers," IETF RFC 1812, Jun. 1995.
- [9] C. Berger, M. Kossel, C. Menolfi, T. Morf, T. Toifl, and M. Schmatz, "High-density optical interconnects within large-scale systems," *Proc. SPIE*, vol. 4942, pp. 222–235, 2003.
- [10] W. Ng, P. Galloway, and M. Annand, "Copper cabling for multi-gigabit serial links for inter-cabinet connections," in *Proc. High Performance System Design Conf. (DesignCon2002)*, Santa Clara, CA, Jan. 2002.
- [11] P. Chiang, W. J. Dally, and M.-J. E. Lee, "A 20 Gb/s 0.13 μ m CMOS serial link," in *Proc. Hot Chips 2002, 14th Symp. High Performance Chips*, Palo Alto, CA, Aug. 2002.
- [12] R. Ronen, A. Mendelson, K. Lai, S.-L. Lu, F. Pollack, and J. P. Shen, "Coming challenges in microarchitecture and architecture," *Proc. IEEE*, vol. 89, no. 3, pp. 325–340, Mar. 2001.
- [13] H. C. C. Chan, H. M. Alnuweiri, and V. C. M. Leung, "A framework for optimizing the cost and performance of next-generation IP routers," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 6, pp. 1013–1029, Jun. 1999.
- [14] F. M. Chiussi and A. Francini, "Providing QoS guarantees in packet switches," in *Proc. GLOBECOM '99*, Rio de Janeiro, Brazil, Dec. 1999, pp. 1582–1590.
- [15] F. M. Chiussi, A. Francini, D. A. Khotimsky, and S. Krishnan, "Feedback control in a distributed scheduling architecture," in *Proc. GLOBECOM 2000*, San Francisco, CA, Nov. 2000, pp. 525–531.
- [16] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services," IETF RFC 2475, Dec. 1998.
- [17] W. J. Dally and Brian Towles, *Principles and Practices of Interconnection Networks*. San Francisco, CA: Morgan Kaufmann, 2004, pp. 239–249.
- [18] V. Fineberg, "A practical architecture for implementing end-to-end QoS in an IP network," *IEEE Commun. Mag.*, vol. 40, no. 1, pp. 122–130, Jan. 2002.
- [19] A. Parekh and R. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single-node case," *IEEE/ACM Trans. Netw.*, vol. 1, no. 3, pp. 344–357, Jun. 1993.
- [20] D. C. Stephens, J. C. R. Bennett, and H. Zhang, "Implementing scheduling algorithms in high-speed networks," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 6, pp. 1145–1158, Jun. 1999.
- [21] Y. Tamir and G. Frazier, "High performance multiqueue buffers for VLSI communication switches," in *Proc. 15th Annu. Symp. Computer Architectures*, Honolulu, HI, Jun. 1988, pp. 343–354.
- [22] N. McKeown, "The iSLIP scheduling algorithm for input-queued switches," *IEEE/ACM Trans. Netw.*, vol. 7, no. 2, pp. 188–201, Apr. 1999.
- [23] C. Minkenberg, "Performance of i-SLIP scheduling with large round-trip latency," in *Proc. IEEE Workshop on High-Performance Switching and Routing (HPSR 2003)*, Torino, Italy, Jun. 2003, pp. 49–54.
- [24] F. M. Chiussi and A. Francini, "A distributed scheduling architecture for scalable packet switches," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 12, pp. 2665–2683, Dec. 2000.
- [25] S.-T. Chuang, A. Goel, N. McKeown, and B. Prabhakar, "Matching output queueing with a combined input output queued switch," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 6, pp. 1030–1039, Jun. 1999.
- [26] P. Krishna, N. Patel, A. Charny, and R. J. Simcoe, "On the speedup required for work-conserving crossbar switches," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 6, pp. 1057–1066, Jun. 1999.
- [27] I. Stoica and H. Zhang, "Exact emulation of an output queueing switch by a combined input output queueing switch," in *Proc. 6th Int. Workshop on Quality of Service (IWQoS '98)*, Napa, CA, 1998, pp. 218–224.
- [28] C. Minkenberg and T. Engbersen, "A combined input and output queued packet-switched system based on PRIZMA switch-on-a-chip technology," *IEEE Commun. Mag.*, vol. 38, no. 12, pp. 70–77, Dec. 2000.
- [29] M. Katevenis, D. Serpanos, and E. Spyridakis, "Switching fabrics with internal backpressure using the ATLAS I single-chip ATM switch," in *Proc. GLOBECOM '97*, Phoenix, AZ, Nov. 1997, pp. 242–246.
- [30] R. Rojas-Cessa, E. Oki, and H. J. Chao, "CIXOB-k: Combined input-crosspoint-output buffered packet switch," in *Proc. GLOBECOM '01*, 2001, vol. 4, pp. 2654–2660.
- [31] D. C. Stephens and H. Zhang, "Implementing distributed packet fair queueing in a scalable switch architecture," in *Proc. IEEE INFOCOM '98*, San Francisco, CA, 1998, vol. 1, pp. 282–290.
- [32] M. Nabeshima, "Performance evaluation of a combined input- and crosspoint-queued switch," *IEICE Trans. Commun.*, vol. E83-B, no. 3, pp. 737–74, Mar. 2000.
- [33] T. Javidi, R. Magill, and T. Hrabik, "A high-throughput scheduling algorithm for a buffered crossbar switch fabric," in *Proc. ICC 2001*, Helsinki, Finland, Jun. 2001, vol. 5, pp. 1586–1591.
- [34] H. J. Chao, S. Y. Liew, and Z. Jing, "A dual-level matching algorithm for 3-stage closed-network packet switches," in *Proc. Hot Interconnects 2003*, Stanford, CA, Aug. 2003, pp. 38–43.
- [35] F. Abel, C. Minkenberg, R. P. Luijten, M. Gusat, and I. Iliadis, "A four-terabit packet switch supporting long round-trip times," *IEEE Micro*, vol. 23, pp. 10–24, Jan./Feb. 2003.
- [36] J. W. Causey and H. S. Kim, "Comparison of buffer allocation schemes in ATM switches: Complete sharing, partial sharing, and dedicated allocation," in *Proc. ICC '94*, New Orleans, LA, 1994, pp. 1164–1168.
- [37] F. Le Maut and G. Garcia, "A scalable switch fabric to multi-terabit: Architecture and challenges," in *Proc. Hot Chips 2002, 14th Symp. High Performance Chips*, Palo Alto, CA, Aug. 2002.
- [38] N. T. Kung and R. Morris, "Credit-based flow control for ATM networks," *IEEE Network*, vol. 9, no. 2, pp. 40–48, Mar./Apr. 1995.

- [39] F. Gramsamer, M. Gusat, and R. Luijten, "Flow control scheduling," *J. Microprocess. Microsyst.*, vol. 27, no. 5–6, pp. 233–241, Jun. 2003.
- [40] M. Katevenis, "Buffer requirements of credit-based flow control when a minimum draining rate is guaranteed," in *Proc. HPCS '97*, Chaldiki, Greece, 1997, pp. 168–178.
- [41] R. P. Luijten, C. Minkenberg, and M. Gusat, "Reducing memory size in buffered crossbars with large internal flow control latency," in *Proc. GLOBECOM 2003*, San Francisco, CA, Dec. 2003, vol. 7, pp. 3683–3687.
- [42] S. Floyd and V. Jacobson, "Link-sharing and resource management models for packet networks," *IEEE/ACM Trans. Netw.*, vol. 3, no. 4, pp. 365–386, Aug. 1995.



François Abel (M'01) received the B.S. degree in engineering from the École Nationale d'Ingénieurs, Belfort, France, and the M.S. degree in electrical engineering from the École Supérieure d'Ingénieurs, Marseille, France.

He is a Research Staff Member at the Systems Department of the IBM Zurich Research Laboratory, Switzerland, which he joined in 1997. His research interests include the architecture and VLSI design of high-speed, low-latency switching systems. Currently, he is responsible for the architecture and

implementation of the crossbar scheduler for the OSMOSIS demonstrator, a Corning–IBM joint project on Optical Shared Memory Supercomputer Interconnect Systems.



Cyriel Minkenberg received the M.S. and Ph.D. degrees in electrical engineering from the Eindhoven University of Technology, Eindhoven, The Netherlands, in 1996 and 2001, respectively.

Since 2001, he has been a Research Staff Member at the IBM Zurich Research Laboratory, Switzerland, where he has contributed to the design and evaluation of the IBM PowerPRS switch family. He was also responsible for the architecture and performance evaluation of the crossbar scheduler for the OSMOSIS optical supercomputer interconnect. Currently, he partic-

ipates in the standardization of congestion management for 10G Ethernet.



Ilias Iliadis (S'84–M'88–SM'99) received the B.S. degree in electrical engineering from the National Technical University of Athens, Greece, in 1983, the M.S. degree from Columbia University, New York, as a Fulbright Scholar in 1984, and the Ph.D. degree in electrical engineering in 1988, also from Columbia University.

He has been at the IBM Zurich Research Laboratory since 1988. He was responsible for the performance evaluation of IBM's PRIZMA switch chip. His research interests include performance

evaluation, optimization and control of computer communication networks and storage systems, switch architectures, and stochastic systems. He holds several patents.

Dr. Iliadis is a member of IFIP Working Group 6.3, Sigma Xi, and the Technical Chamber of Greece. He has served as a Technical Program Co-Chair for the IFIP Networking 2004 Conference.



Ton Engbersen (M'92–SM'03) has worked in the IBM Zurich Research Laboratory since 1980. He developed the PRIZMA switch architecture. PRIZMA became a family of communication switch offerings from IBM. He spent two years at the IBM T. J. Watson Research Center, Yorktown Heights, NY, where he led the initial development of MPLS. Since 1997, he has assumed several management positions and today manages the Server Technology Research group in Zurich. His interests are in I/O technology for servers, scale-out and the implications thereof.

He has been a member of the IBM Academy of Technology since 1994.



Mitchell Gusat (M'94) received the Masters degrees in electrical engineering from the University of Timisoara, Romania, and in computer engineering from the University of Toronto, Toronto, Canada.

He is a Researcher at the IBM Zurich Research Laboratory, Switzerland. His research interests include computer architecture, coherency, distributed systems, switching and scheduling. His current research focus is flow control and congestion management for interconnection networks and data-centers. He has contributed to RapidIO, InfiniBand

and Ethernet standards. He is a member of the Association for Computing Machinery.



Ferdinand Gramsamer (M'06) received the Diploma in electrical engineering from the University of Stuttgart, Germany, and the Ph.D. degree from ETH Zürich, Switzerland.

He was working as a Ph.D. student at the IBM Zurich Research Laboratory, Switzerland, from 1998 until 2003 with research interests in flow control, performance modeling, communication protocols for multiprocessor systems, and system validation. He currently works with bbv Software Services AG, Lucerne, Switzerland, and is head of testing services.



Ronald P. Luijten has managed the IBM server interconnect fabrics research team in Zurich since 1997, which is currently finishing up the OSMOSIS optical switch demonstrator in close collaboration with Corning, inc. for the U.S. Department of Energy (DOE). His research interests are in multistage fabric system performance and design, including electronic crossbars and all-optical switches. His current focus is on traffic use patterns for HPC interconnect fabrics with the Barcelona Supercomputer Center and for commercial data center fabrics with a large bank.

His team also works on standardizing congestion control within Ethernet 802.1Qau. He holds more than 20 patents in the switching area of ATM, and has co-organized five IEEE ICCCN conferences.