An Optical Packet-Switched Interconnect for Supercomputer Applications*

Roe Hemenway and Richard R. Grzybowski

Corning Incorporated, Science and Technology, Corning, NY 14831
{hemenwaybr,grzybowsrr}@corning.com

Cyriel Minkenberg and Ronald Luijten

IBM Research, Zurich Research Laboratory, Säumerstrasse 4, CH-8803 Rüschlikon, Switzerland
{sil,lui}@zurich.ibm.com

We describe a low-latency, high-throughput scalable optical interconnect switch for high-performance computer systems that features a broadcast-and-select architecture based on wavelength- and space-division multiplexing. Its electronic control architecture is optimized for low latency and high utilization. Our demonstration system will support 64 nodes with a line rate of 40 Gb/s per node and operate on fixed-length packets with a duration of 51.2 ns. This paper addresses the key system-level requirements and challenges for such applications.

OCIS codes: 060.1810, 250.5950.

1. Introduction

High-performance computing systems (HPCS) interconnect hundreds of high-performance microprocessors by relying on high-sustained-bandwidth, low-latency links to provide timely access to distributed memory and the intermediate results of distant CPUs [1]. For the largest systems, this interconnect has become a key contributor to sustained performance, comparable in importance to microprocessor and software design. In a recent report, the Accelerated Strategic Computing Initiative (ASCI) [1] acknowledges an emerging system imbalance as microprocessor performance outstrips interconnect performance, and identifies opportunities for revolutionary changes to HPCS design based on maturing optical technology. Under a contract related to this program, Corning and IBM are jointly developing a 64-node HPCS interconnect demonstrator with the optical data paths operated at 40 Gb/s, scalable to 2048 nodes. The Optical Shared MemOry Supercomputer Interconnect System (OSMOSIS) project aims at solving the technical challenges and at accelerating the cost reduction of all-optical packet

* This research is supported in part by the University of California under subcontract number B527064.
switches for HPCS.

Optically switched interconnects potentially offer greater scalability, higher throughput, lower latency and reduced power dissipation compared to electrically mediated systems of comparable capacity. The challenges they pose include optimized integration with the end nodes, a control system and efficient scheduling algorithm that perform well given the switch’s dynamic characteristics, an efficient structure of the data packet, a high-performance control channel, and good channel coding for error management.

Our architecture implements a synchronous broadcast-and-select (B&S) data switch using semiconductor optical amplifiers (SOAs) combining 8 wavelengths on 8 fibers to achieve 64-way distribution (Fig. 1). For improved performance, two receivers are provided per output. The switching is achieved with a fast 8:1 fiber selection stage followed by a fast 8:1 wavelength selection stage at each output port. Multiple wide-dynamic-range receivers feature fast clock recovery and packet-oriented error correction. A programmable centralized arbitration unit reconfigures the optical switch via a separate optical control channel synchronously with the arrival of fixed-length optical packets. The arbiter enables high-efficiency packet-level switching without aggregation or prior bandwidth reservation, and achieves a high maximum throughput. It features a novel pipelined implementation of a bipartite graph matching that avoids the latency penalty associated with traditional pipelining methods.

![Fig. 1. A B&S switch with combined space- and wavelength-division switching.](image)

**Paper Organization:** In Sec. 2 we describe the key requirements and outline the challenges they pose. Sec. 3 describes the data-path architecture of the switch, including the optical B&S network, and presents salient features of the optical technologies and components used in the OSMOSIS implementation. Sec. 4 introduces the control-path architecture, including descriptions of the arbitration algorithm and the control protocols, such as for request-grant arbitration, reliable delivery, flow control, and consistency. We present some initial test results in Sec. 5, and conclude in Sec. 6.

### 2. Requirements and Challenges

HPCS interconnects must be designed to handle the arrival, buffering, routing, forwarding, reception,
and error management of high-data-rate packet streams in scalable local networks. HPCS switches are required to deliver an extremely low bit-error rate, very low latency, a high data rate, and extreme scalability. To achieve a balanced system design wherein no resource generally constrains system performance, HPCS systems require internode bandwidths (between CPU clusters) comparable to intra-node bandwidths. A rule of thumb for HPCS is to achieve a ratio of 0.1 Gb/s interconnect bandwidth per GFLOP of computer performance. Moreover, they must be efficient for bursty traffic, with demands varying from many small data transfers to fewer large dataset transfers. In comparison, conventional telecom-optimized packet switches operate on comparably smooth and predictable flows with much-relieved latency constraints.

The requirements for our system include 64 input and output ports operating at 40 Gb/s with 75% available for user payload, a $10^{-21}$ bit-error rate using hardware forward error correction (FEC), less than 1 µs latency measured application-application in the scaled system, an optical data path, and a system design capable of scaling to higher line rates and port counts. Scalability to 2048 nodes is achieved by deploying 96 64×64 switches in a two-level (three-stage) Fat Tree topology (Fig. 3). The requirement for 75% user data necessitates a low-overhead packet structure and low guard times for the optical switching overhead. The overhead comprises the packet header, FEC, preamble, switching time, dead time, and other inefficiencies.

Our optical switch offers numerous opportunities for a new HPCS switching paradigm. First, the switch core operates on entire packets and does not perform bit-level operations on the data-channel content; it needs to switch only once in an entire packet duration, not at data-payload-bit speeds. This reduces both switching power and rate, and improves scaling of the switch core. Second, by provisioning a transparent data path, multiple transmissions can utilize the data path simultaneously, enabling an increase of the internode bit rate, either by operating a single channel at a higher rate or by provisioning multiple closely-spaced WDM channels in one “band” without increasing the size of the switch core. Third, by separating the control and data paths, each can be upgraded independently.

Meeting such requirements means overcoming significant technical challenges in the optical data path, in the control path, and in the scheduling and control system. Optical-path challenges start with the need to mutually synchronize data-packet transmissions to produce efficient just-in-time switching among them. This is achieved by a combination of centralized clock distribution implemented over a synchronous control channel and known optical path delays to the common resources of the switch core. The optical path lengths must be matched to a fraction of a packet time through all amplification, broadcast and switch elements. This is achieved via low-skew optical hardware based largely on planar lightwave circuits (PLCs), and discrete components with fixed-length fiber pigtails. The switching of the SOAs coincides with the arrival of the data packets to be received, so that the synchronous switch control system operates at the packet rate to set switch positions. Path- and polarization-dependent loss (PDL), which can induce optical power excursions that vary significantly from packet to packet, are managed by
limiting the PDL of the components, equalizing transmit power and fitting receivers with a wide dynamic range. Although the switch is synchronously operated, that is, all nodes operate at the same nominal bit and packet rate, the end nodes receive contiguously arriving packets exhibiting random relative bit phases. Thus the receivers must acquire the bit phase almost instantly on each successive packet. Because the packets are independent of one another, error-correction techniques that operate efficiently on individual packets must be implemented. Since FEC is more efficient with long data blocks, this must be traded off against the need for short packets with low overhead in an HPCS interconnect. Finally, the actual switching time and guard times needed to account for all timing uncertainties must result in a switch system that still achieves a high utilization for user payload.

Challenges in the control path include the need to conduct resource requests and manage reliable delivery at a rate compatible with the packet transmission times. The scheduler must not only be fast but also efficient. Ideally, it is able to process packet transmission requests for many packets without the need to buffer (and therefore delay) packets at the ingress, and optimally resolve transmission requests for the fully loaded 64-port switch operated at its maximum packet rate.

To reduce the arbitration rate, the B&S-based designs presented in [2-5] adopt aggregation approaches in which multiple packets are first assembled into containers (bursts, superframes) and then switched as an aggregate. They employ frame sizes of about 1 kB at 10 Gb/s, resulting in a time slot duration of over 800 ns. While this aggregation approach minimizes the effect of long switching times and overhead, it is not suitable for HPCS because transmission must wait until containers are full to achieve high efficiency, introducing unacceptably high latency with respect to a 1-µs requirement. Transmitting partially-filled containers reduces latency at the expense of user bandwidth and reduces throughput. In contrast, the time slot in our design is only 51.2 ns, about 16 times shorter. To cope efficiently with such relatively short packets, we require fast switching, low switching overhead, and a deeply pipelined control architecture. This is especially important at the data rates of 40 Gb/s and higher as employed in this work. In a totally different approach, such as [6], contention is resolved via deflection, which obviates the need for complex scheduling but results in less predictable latency.

Finally, the control architecture must accommodate scaling via a Fat Tree topology to 2048 ports without significant performance loss.

3. Data-Path Architecture and Technology

The interconnect architecture implements a WDM B&S design that combines the transmissions from the 64 nodes via eight wavelengths onto eight separate fibers as shown in Fig. 1. The major elements of the switch are shown in Fig. 2. The data path comprises the electronic host-channel adapters (HCAs) and the optical B&S network. An HCA originates and terminates packet transmissions. The set of eight “WDM, amplify and split” modules each feature an integrated 8×1 wavelength combiner followed by a high-power optical amplifier and integrated broadband splitters to implement the broadcast function. To
serve redundant receivers, discussed below, our 64 port design implements a $1 \times 128$ split function. Each of the 128 optical switch modules (OSMs) selects one of the 64 possible packets in transmission in each time slot. Multiple OSMs can simultaneously select the same packet to achieve a multicast or broadcast functionality at no additional hardware cost.

**HCA:** The HCA originates and terminates data-packet transmissions across the switch core. Each HCA comprises an ingress and an egress section. The ingress section receives traffic from the attached computing node (or the previous stage in multi-stage network) and temporarily stores every incoming packet in an electronic memory that is organized in a virtual output-queuing (VOQ) fashion until a grant to send has been received (see Sec. 4 for a description of the control path). The egress section receives traffic from the optical switching network, temporarily stores it if necessary, and forwards it when the downstream receiver (either a computing node or the next stage in a multi-stage network) is ready. The high-speed digital functions of the HCA, implemented largely in high-performance FPGA technology, perform error correction, packet framing, queuing, reliable delivery, and handling of the control channel protocols. The HCA interfaces to the high-speed host bus and its memory subsystem as well as distributes the system clock to critical subsystems and manages optical control channel communications. The HCA further comprises an optical network interface card (ONIC) that contains the optical data and control channel interfaces.

![Fig. 2. Major elements of the optical interconnect switch include the HCA with embedded ONIC, the amplifying multiplexer and broadcast splitters, the optical switch modules, and the centralized arbiter.](image)

**ONIC:** The ONIC performs serialization, deserialization, and OE and EO conversions. A distributed feedback (DFB) laser is coupled to a 40-Gb/s electro-absorption modulator (EAM). We selected this combination because of its compact size, low drive requirements, bit-rate scalability and performance. On
each HCA, multiple receivers are provisioned, each of which implements a high-speed PIN photodiode receiver with wide-dynamic-range transimpedance and limiter amplifier functions. A clock recovery and decision ASIC circuit, optimized for fast clock acquisition of optical packets exhibiting more than 5 dB packet–packet power variation, completes the ONIC functions.

**Multiple Receivers:** A distinguishing feature of the OSMOSIS architecture is the presence of multiple receivers per output port. This allows more than one packet to be transferred to the same port simultaneously, which helps resolve contention faster and therefore improves latency. The preferred configuration provides two receivers per output, because simulations have shown that the additional cost of having more than two receivers is not justified, as they would offer only minor additional performance improvements. The presence of two receivers can be exploited by changing the arbiter to match up to two inputs to one output, instead of just one, which requires modifications to the matching algorithm.

**B&S:** The Multiplex, Amplify and Split WDM broadcast stage combines the eight WDM channels assigned to all eight independent broadcast networks. A high-power erbium-doped fiber amplifier (EDFA), capable of more than 20 dBm output power, less than 7 dB noise figure and fitted with an automatic gain control (AGC) function, boosts the signal power to overcome subsequent broadcast-splitting losses. The broadcast 1×128 splitter is achieved in two stages: 1×8 followed by 1×16 for equipment modularity using conventional PLC technology. To control data-path skew, the entire optical data path is controlled to a small fraction of the packet length.

**OSM:** The OSM implements a two-stage selection function: first a fiber-select SOA is activated to select the fiber that contains the wavelength-multiplexed packet to be received. Regardless of the fiber selected, the WDM multiplex is combined in an 8×1 PLC to a common port. This stream is then demultiplexed and passed through a second stage in which the WDM channel of interest is selected. Regardless of the color selected, the stream in then re-multiplexed onto a common output port and delivered to the broadband receiver residing at the egress node ONIC. Synchronously with packet transmission, the centralized arbiter resolves transmission requests delivered to it via an optical control channel, and sets the state of the SOA selector gates just as the packets arrive at each OSM. At the egress node HCA, receiver electronics extract the user data, correct errors or request retransmission of uncorrectable packets, forward the packets to the next level of the Fat Tree (if it is an intermediate node in the network) or reassemble packets and deliver those packets to the terminal host node (if it is a terminal node in the network).

**Multi-Stage Scalability:** One of the main objectives of the OSMOSIS project is providing a solution that scales to thousands of nodes. This cannot be achieved economically in a single-stage configuration due to the quadratic complexity involved in the control path (arbitration) and the physical bulk of the data path (number of SOAs, splitters, couplers, etc.). A more economical way to scale to much larger networks is with a multi-stage network, such as a Fat Tree. With the 64-port basic OSMOSIS switch, a 2048-node network with full bisectional bandwidth can be constructed with a two-level Fat Tree network as shown in
Fig. 3. Even larger networks can be achieved in a two-level network by configuring it with less than full bisectional bandwidth or by deploying more levels. Although having an end-to-end all-optical data path throughout the multi-stage network is very attractive, this is not practical in a packet-switching context, because the arbitration must then arbitrate and configure all switches in the network simultaneously. Therefore, there will be electronic buffers in between stages to decouple the arbitration. Obviously, the main disadvantage of this approach is significant additional cost and latency, owing to the required O/E/O conversions and buffering.

Multi-stage scaling involves many additional challenges, including link-level flow control, routing, and congestion control/management. Many performance characteristics, such as the latency and throughput, are harder to control in a multi-stage network. These issues are outside the scope of this paper.

4. Control-Path Architecture and Protocols

The OSMOSIS control structure involves the HCAs and the arbiter. Every HCA has a dedicated, bidirectional (full-duplex) Control Channel (CC) to the arbiter to exchange requests, grants, credits, and acknowledgments. Additionally, there is a Switch Command Channel (SCC, i.e., the configuration channel) between the arbiter and the crossbar to reconfigure the SOAs according to the optimal crossbar setting computed by the arbiter. In this section, we describe the control functions of the HCA and the arbiter, the CC protocols, and our approach to reliable delivery.

**HCA:** Every HCA comprises an ingress (TX) and an egress (RX) interface. The ingress interface is connected to an input port of the switch fabric and contains electronic memory to store packets as they wait to be transmitted across the switch core. The logical queuing structure is arranged in a virtual output queuing (VOQ) fashion, so called because there is one separate queue per output. This arrangement is...
known to eliminate *head-of-line blocking*, which limits throughput to 58.6% of full utilization.

The HCAs issue requests to the arbiter, which performs bookkeeping for the requests of all HCAs, computes a matching between input and output ports in every time slot based on the currently pending requests, and issues corresponding grants to the HCAs. Upon receipt of a grant, the HCA dequeues a packet from the VOQ granted and transmits it to the switch core, which will route it to the correct destination according to the configuration applied by the arbiter via the SCC.

The egress interface is connected to an output port of the switch fabric. The OSMOSIS architecture implements two OSMs per egress interface. This means that multiple packets can be received by the same HCA in one time slot. The egress interface comprises an electronic output buffer, in which it stores arriving packets before delivering them to the local node. This buffer temporarily stores packets that cannot immediately be accepted by the local computing node or the next stage in a Fat Tree.

**Arbiter:** HPCS requirements impose stringent demands on the control of the system. To ensure that messages are delivered with minimum latency, we must be able to reconfigure the optical routing fabric in every time slot, contrary to many existing switches with optical data paths. Similar optical B&S switches, presented in [3] and [5], employ burst-switching aggregation techniques to reduce the arbitration rate. However, our stringent latency requirement implies that we cannot apply such techniques because assembling individual messages into large containers introduces too much latency.

Reconfiguring the switch entails computing a one-to-one matching between the inputs and outputs in every time slot. To achieve good performance in terms of the delay-throughput characteristics, the matching should be as complete as possible, which requires a centralized bipartite graph-matching algorithm. As the optimum solution to this problem is known to be infeasible in fast hardware, many heuristic approaches have been studied in recent years, giving rise to a class of practical solutions based on iterative, round-robin-based algorithms, such as i-SLIP [7] and DRRM [8]. However, these algorithms must perform a number of iterations (depending on the number of ports $N$) to converge and ensure good performance characteristics. Therefore, arbitrating for a large number of ports $N$ in a short time remains a major challenge. It has been shown that the aforementioned iterative algorithms on average converge in $\log_2(N)$ iterations. In the OSMOSIS demonstrator, this implies that the arbiter should complete about six iterations on a $64 \times 64$ request matrix in one time slot. The targeted packet size is 256 B at a line rate of 40 Gb/s, resulting in a time slot of 51.2 ns. Considering the additional requirement of an FPGA implementation, it is very challenging to achieve up to six iterations within one time slot.

We have addressed this challenge with a novel pipelined implementation that we refer to as FLPPR: Fast Low-latency Parallel Pipelined aRbitration [9]. It is related to existing techniques for parallel matching, such as PMM [10,11], in that multiple matchings to be issued at subsequent time slots are computed in parallel. However, our approach avoids the latency penalty associated with existing approaches, in which, by design, every request enters at the beginning of the pipeline, so that the
corresponding grant must traverse all pipeline stages before being issued. The FLPPR scheme, on the other hand, allows requests to enter at any stage in the pipeline, so that the minimum pipeline latency is equivalent to just a single pipeline stage. Moreover, this minimum latency is independent of the number of pipeline stages. Every stage of the pipeline performs one iteration of the DRRM [8] algorithm, which we prefer because it is highly amenable to a distributed implementation.

In addition to lower latency, FLPPR simultaneously achieves high maximum throughput, and can provide up to 20% higher throughput with non-uniform traffic. Performance results with uniform as well as non-uniform traffic, as well as several FLPPR variants ranging from basic to optimized, have been reported in [9]. Fig. 4 compares the mean latency (VOQ waiting time expressed in time slots) as a function of utilization of PMM [11] and FLPPR [9] for a 64×64 switch with $K = 6$ pipeline stages and uniform uncorrelated arrivals. We observe that even the basic FLPPR implementation exhibits significantly lower delay than PMM throughout the load range and that the optimized FLPPR implementation achieves a further reduction in latency.

**Fig. 4.** Mean latency vs. utilization with $N = 64$ and uniform uncorrelated arrivals. Left: basic FLPPR, different values of $K$. Right: comparison of PMM with basic and optimized FLPPR with $K = 6$.

**Control-channel protocol:** The physical implementation and packaging aspects of OSMOSIS, or indeed large packet switches in general [12,13], lead to increased physical distribution and thus a large physical distance between the HCAs and the switch core, to increased latency due to pipelining, and to a shrinking packet cycle. All of this adds up to a significant *round trip* (RT) between the ICA and the arbiter, i.e., the bandwidth-delay product expressed as a multiple of the minimum packet duration is significantly larger than one. This requires that data packets and control messages be pipelined on the data and control paths, respectively. Moreover, this RT has a significant impact on two aspects of the switch design. First, it implies that the arbiter has delayed knowledge of the current VOQ state. Therefore, the control protocol between HCAs and arbiter must be carefully designed to avoid performance degradation with long RTs [14]. Second, this RT is a direct contributor to the arbitration latency, which is defined as
the time that elapses between the issuance of a request and the reception of the corresponding grant. The minimum arbitration latency comprises the RT on the CC and the time needed to compute a matching.

Our CC protocol incorporates incremental VOQ state updates to cope with the physical distribution of the system, incremental credit flow control to prevent overflow of the egress buffers, per-packet acknowledgments to achieve the required error rate, and a census protocol to ensure consistency between the HCAs and the arbiter. The CC protocol comprises a number of sub-protocols that share the same physical channel and are encapsulated in the control messages. These sub-protocols are:

*Arbitration*: The long RT has an impact on the design of the request-grant protocol between the VOQs on the HCAs and the arbiter [14]. To cope with a long RT without performance loss, the arbiter maintains pending-request counters to reflect the state of all VOQs. The HCA issues a request for arbitration for every newly arriving packet. The arbiter increments the corresponding counter upon receipt of a request, computes a matching based on the current values of all counters, and issues grants according to the current matching. When the arbiter has matched a specific VOQ, it immediately decrements the corresponding counter to reflect that a request has been granted. In this protocol, requests and grants are incremental, rather than absolute as traditionally used.

*Consistency*: An incremental request-grant protocol as described above is not inherently robust to errors. Therefore, we employ a census protocol [15] to ensure that the actual VOQ state at the line cards and the state maintained by the arbiter are consistent within the bounds given by the RT. From time to time, the HCA triggers a census to check whether the arbiter's image of the state of its VOQs is consistent with the actual state, taking into account requests and grants that are in flight. To this end, the HCA injects a control message containing a census count. According to specific rules, the arbiter returns a reply comprising an updated census count, which the HCA, upon receipt, checks for consistency. The check result not only indicates whether the state is consistent, but also indicates the magnitude of the inconsistency if it fails. In that case, the HCA initiates corrective action.

*Local flow control*: As the HCA comprises egress as well as ingress buffers and the egress buffer’s arrival rate can exceed its departure rate, local flow control is necessary to ensure that no packets are lost owing to an egress buffer overflow. To this end, we employ an incremental credit flow control scheme between the egress buffers and the arbiter. The egress buffers release credits to the arbiter via the CC, and the arbiter performs the credit bookkeeping. It will only issue grants for outputs that have credits available. Upon issuing a grant, the arbiter decrements the credit counter of the corresponding output. This incremental credit flow control scheme is also protected against errors by a census mechanism.

*Reliable delivery*: To ensure correctness, we employ a hop-by-hop reliable delivery scheme. For every correctly received packet, the egress HCA returns an acknowledgment via the CC. The arbiter relays the acknowledgment to the corresponding ingress HCA, which then safely dequeues the acknowledged packet. This protocol is described in more detail below.

*Clock distribution*: A very important aspect of the OSMOSIS design is clock distribution and
synchronization. We take advantage of the presence of the centralized arbiter to distribute the clock and synchronize all HCAs using the physical layer of the CCs.

The physical implementation of a CC consists of two (one upstream, one downstream) 8b/10b-coded optical InfiniBand links running at 2.5 Gb/s, which implies that the length of a control message is 128 bits. This is sufficient to ensure that every control message can carry one complete sub-message for each of the sub-protocols. Each control message also comprises a CRC to detect errors. With respect to CC reliability, note that census mechanisms protect both arbitration and the flow-control sub-protocols, and that the reliable delivery and the census sub-protocols themselves are inherently robust against errors.

**Reliable Delivery:** To meet the objective of an error rate better than $10^{-21}$, we have adopted a two-tier approach to reliable delivery. As the raw bit-error rate of the optical path is designed to be near $10^{-19}$ we will employ a forward error-correcting (FEC) code on the header and data part of the packet to reduce the bit-error rate to an estimated $10^{-17}$. To reduce the error rate observed by the application running on the nodes even further, we employ a window-based hop-by-hop retransmission scheme that performs a given number of retries for every corrupted packet. Together, these schemes can meet the extremely low error-rate requirement.

**FEC:** The link-level transmission coding scheme must correct all single-, detect all double- and most triple- and higher-order transmission bit errors. The coding scheme furthermore has to generate sufficient transitions for the receiving PLL to enable an accurate tracking of phase changes by limiting the run lengths. As maintaining low latency is one of the most important requirements for OSMOSIS, the FEC coding scheme is a compromise between low latency, requiring a short code block, and low coding overhead, requiring a long code block. To meet the system-level requirement that 75% of transmitted data must be user information, we designed the total coding overhead to be less than 10%. Because we have not found an appropriate standard code, we have developed a custom code. It is in the class of generalized non-binary cyclic Hamming codes, with a run length of 256 bits and a coding overhead of 16 bits. The primitive polynomial is chosen to be $p(x) = x^8 + x^4 + x^3 + x^2 + 1$. This code yields an overhead of approximately 6%; the remaining 4% overhead is reserved for limiting the run length.

Fig. 5 shows the transfer efficiency ($1 –$ coding overhead) versus the block code length (user bits plus coding bits) with the number of correctable bits $t$ as a parameter ($t = 1$ corresponds to 1 bit correction capability). The yellow area marks the eligible region with less than 10% overhead. We also observe that, given the low coding overhead requirement, we can choose a code that corrects at most two bit errors, and that the minimum block length is 64 bits.

---

1 The physical layer of the Switch Command Channel is identical to the control channel.
Fig. 5. Code transfer efficiency versus code length.

**Hop-by-hop retransmission:** Because the error rate after FEC still does not meet the requirement, we implement a hop-by-hop retransmission scheme between the input and the output buffers. Retransmission schemes in general require that the sender stores sent packets in a retransmission (RTX) buffer and that the receiver notifies the sender about successful (positive acknowledgment – ACK) and/or unsuccessful (negative acknowledgment – NAK) transmissions. These acknowledgments can be transmitted in-band on the data channel using either dedicated packets or by piggybacking on data traffic in the reverse direction, or out-of-band. Because in OSMOSIS we have a dedicated CC from every HCA to the arbiter, we accommodate ACKs in the control messages. A significant advantage of such a fixed-bandwidth out-of-band channel is that the ACKs can be returned immediately, which reduces the RT and therefore also the RTX buffer requirements. Additionally, there is no need to perform ACK aggregation, so that every packet is acknowledged individually.

We exclusively use ACKs. The only advantage of NAKs is that they can expedite the retransmission of corrupted packets, but by using immediate per-packet ACKs without aggregation, it is straightforward to instantly detect and retransmit the missing packets. The arbiter must route the incoming ACKs to the appropriate HCAs. As there is a one-to-one matching between inputs and outputs in every time slot, it follows that the ACKs generated upon reception of these packets also form a one-to-one matching with the inputs and outputs reversed, so there is no contention in routing the ACKs in the arbiter.

We employ a Go-Back-$N$ (GBN) retransmission policy that retransmits all unacknowledged packets sent since the last positive ACK. Although GBN may lead to unnecessary duplications compared with a Selective Retry (SR) policy that only retransmits the corrupted packets rather than the entire window, it has two key advantages over SR, namely that there is no need for a resequencing buffer at the receiver, and that the RTX buffer is a simple FIFO. As the error rate after FEC is already very low, we expect that
the extra overhead due to unnecessary retransmissions will be negligible.

As the RT is fixed and known, retransmissions can be triggered autonomously by the sender by attaching a timestamp to every packet. If the head-of-line packet of the RTX buffer is older than RT, the packet is considered lost (either the packet or the ACK has been corrupted or lost), and the contents of the RTX buffer are retransmitted. To avoid endless retransmissions when a link is broken, we impose a fixed upper limit on the number of retransmissions (e.g. three) per packet by adding a time-to-live (TTL) field to every entry in the retransmission buffer that is decremented on every retransmission. When it reaches zero, the sender gives up and drops the packet.

5. Initial Testing Results

Initial testing of the assembled optical layer confirms the expected performance of the data transmission path. The transmitted 20logQ factor is 18.5 dB, with an extinction ratio of greater than 9.5 dB under $2^{31} - 1$ pseudo-random binary sequence (PRBS). The fiber-select SOAs deliver more than 15 dB of gain with less than 6.5 dB noise figure and typically less than 0.6 dB polarization-dependent gain. They exhibit output saturation power above +17.5 dBm and are operated at 1 dB or less into gain compression. They achieve less than 0.4 dB Q penalty when operated at their optimum input power, yet still deliver a high Q margin over a wide input dynamic range (Fig. 6). The received optical signal-to-noise ratio exceeds 30 dB, as measured in a 0.1-nm-resolution bandwidth. The receiver has a sensitivity of less than –10 dBm measured back–back (Fig. 7, left) and recovers the bit clock phase from the received packet stream in less than 3 ns. It achieves error-free decisions when receiving packets with up to 5 dB packet-to-packet power variation (Fig. 7, right).

![Fig. 6. Signal-quality performance as a function of SOA total input power for eight 40-Gb/s channels under NRZ modulation. Below gain compression, signals are OSNR-limited; above gain compression they become limited by cross-gain modulation in the SOA. Back–back Q is 18.5 dB.](image-url)
6. Conclusions

The time-to-solution metric of high-performance computing systems is becoming increasingly dependent on the latency and throughput of their interconnection networks. As such systems approach and exceed the 1 PetaFLOP/second benchmark, they will require throughput and latency potentially unaffordable or unachievable with further scaling of today’s electrically switched paradigm. Power dissipation, chip pinouts, RF interference, and sheer part-count management represent looming bottlenecks for the current technology. Our effort does not focus solely on the data path and its performance but instead addresses all of the key system-level challenges and requirements of future HPCS by introducing a scalable optical packet switch provisioned to be fully functional at the system level. The challenges of efficiency, throughput, latency, scalability, and reliable delivery are met with a novel pipelined control architecture overlaid on a scalable optical crossbar that leverages space- and wavelength-division multiplexing in a fast-switched technology. Initial testing at 40 Gb/s per port for a modular 64-port element of the Fat Tree demonstrates that the “optical switching tax” can be acceptable and manageable, while still offering a technology with room to scale in terms of port count and bit rate.

Significant engineering work remains, however, to achieve a commercially viable optical-packet-based system solution. While the challenges of burst-mode data recovery, ultra-low bit-error rate, switch dynamics and control-system integration have been met, commercial systems will require substantial further integration of the optoelectronic (OE) subsystems. A factor of 100 reduction in part count over discrete implementations to reduce cost, power consumption, and size, can be achieved by producing functional OE devices at high levels of monolithic and hybrid integration. By selecting the optimum level of integration, by simplifying and automating the processes of OE sub-assembly and

Fig. 7. Left: Burst-mode optical receiver performance at 40 Gb/s. Right: Optical packet train shown with 3 dB power variation received without error.
testing, and by defining functional optical modules that maximize technology re-use, we expect that our optical packet-switching technology will achieve cost parity with unscaled electronic solutions while delivering substantial performance benefits and opportunities for future growth.

In future work, we shall report on the detailed performance of the 64-port interconnect, i.e., both its optical transmission and dynamic performance as well as the efficacy of the control channel, control system, queuing and scheduling algorithms, and reliable delivery protocols. We shall provide updates on the progress toward higher levels of optoelectronic integration, and report on progress in evaluating and introducing advanced technologies, such as more capable modulation formats, improved multiplexing and transport capabilities.

7. Acknowledgments

We thank our sponsors at the University of California, and acknowledge technical contributions by Michael Sauer, John Downie, Deepukumar Nair, Martin Hu, Ron Karfelt, Dave Peters, Jason Dickens, François Abel, Urs Bapst, Wolfgang Denzel, Mitch Gusat, Ilias Iliadis, Raj Krishnamurthy, Peter Müller, Venkatesh Ramaswamy, Enrico Schiattarella, Alan Benner, Henry Brandt, Craig Stunkel, Marco Galli, Thomas Kneubühler, Elmar Ottiger, and their extended teams.

8. References and Links