An Intra-Chip Free-Space Optical Interconnect

Jing Xue, Alok Garg, Berkehan Ciftcioglu, Shang Wang, Jianyun Hu, Ioannis Savidis, †Manish Jain,
Michael Huang, Hui Wu, Eby G. Friedman, †Gary W. Wicks, †Duncan Moore

University of Rochester
{xue, garg, ciftciog, wangsh, jihu, iosavid, huang, hwu, friedman}@ece.rochester.edu
†{jain, wicks, moore}@optics.rochester.edu

Abstract

Continued device scaling enables microprocessors and other systems-on-chip (SoCs) to increase their performance, functionality, and hence, complexity. Simultaneously, relentless scaling, if uncompensated, degrades the performance and signal integrity of on-chip metal interconnects. These systems have therefore become increasingly communications-limited. The communications-centric nature of future high performance computing devices demands a fundamental change in intra- and inter-chip interconnect technologies.

Optical interconnect is a promising long term solution. However, while significant progress in optical *signaling* has been made in recent years, applying conventional packet-switching interconnect architecture to optical networks require repeated E/O and O/E conversions that significantly diminish the advantages of optical signaling. In this paper, we propose to leverage a suite of newly-developed or emerging devices, circuits, and optics technologies to build a fully distributed interconnect architecture based on free-space optics. With a complexity-effective communication support layer to manage occasional packet collisions, the interconnect avoids packet relay altogether, offers an ultra-low transmission latency and scalable bandwidth, and provides fresh opportunities for coherency substrate designs and optimizations.

1 Introduction

Continued device scaling enables microprocessors and other systems-on-chip (SoC) to increase their performance, functionality, and complexity, which is evident in the recent technology trend toward multi-core systems [1]. Simultaneously, uncompensated scaling degrades wire performance and signal integrity. Conventional copper interconnects are facing significant challenges to meet the increasingly stringent design requirements on bandwidth, delay, power, and noise, especially for on-chip global interconnects in those multi-core SoCs with a standard bus architecture.

Optical interconnects have fundamental advantages compared to metal interconnects, particularly in delay and potential bandwidth [2, 3], and significant progress in the technology has been made in recent years [4]. While *signaling* issues have received a lot of attention [5], networking issues in the general-purpose domain remain under-explored. Networking issues can not be neglected as conventional packet-switched interconnects are ill-suited for optics: Without major breakthroughs, storing packets optically remains impractical. Hence packet switching would require repeated opto-electronic (O/E) and electro-optic (E/O) conversions that significantly diminish the advantages of optical signaling. Furthermore, on-chip interconnect poses different constraints and challenges from off-chip interconnect, and offers a new set of opportunities. Hence architecting on-chip interconnect for future microprocessors requires novel solutions and deserves more attention.

In this paper, we propose to leverage a suite of newly-developed or emerging device, circuits, and optics technologies to build an interconnect architecture void of packet switching:

- Signaling: VCSELs (vertical cavity surface emitting lasers) provide light emission without the need of external laser sources and pre-routing those laser sources all over the chip. VCSELs, modulators (if externally-modulated), photodetectors (PDs) and collimating microlens can be implemented in GaAs technologies, and 3-D integrated with the silicon chip – the latter also includes the transmitter and receiver electronics.
- Propagation medium: Free-space optics using micro-mirrors provides an economic medium allowing speed-of-light propagation with low loss.
- Networking: Direct communications through dedicated VCSELs, PDs, and mirrors (in small-scale systems) or via phase array beamsteering (in large-scale systems) allows a quasi-crossbar structure that avoids packet switching altogether, offers ultra-low communication latency in the common case, and provides scalable bandwidth thanks to the fully distributed nature of the interconnect.

The rest of the paper is organized as follows: Section 2 discusses the background of on-chip optical interconnect; Section 3 introduces our free-space optical interconnect and the array of enabling technologies; Section 4 and 5 discuss the architectural design issues and optimizations; Section 6 and 7 present the details of the experimental setup and the quantitative analysis respectively; Section 8 discusses related work; and Section 9 concludes.

2 Challenges for On-Chip Optical Interconnect

First, it is worth noting that on-chip electrical interconnects have made tremendous progress in recent years, driven by continuous transistor scaling, reverse scaling of top metal layers, and the adoption of low-k inter-layer dielectric. The band-

width density is projected to reach 100 Gbps/μm with 20-ps/mm delay by 2016 [6]. Assisted by advanced signal processing techniques such as equalization, echo/crosstalk cancellation, and error correction coding, the performance of electrical interconnects is expected to continue advancing at a steady pace. Therefore, on-chip optical interconnects can only justify the replacement of its electrical counterpart by offering significantly higher aggregated bandwidth with lower power dissipation and without significant overhead in chip area.

Current optical interconnect research efforts focus on using planar optical waveguides, which will be integrated onto the same chip as the electronics. This *in-plane waveguide* approach, however, presents some significant challenges. Pure optical switching and storage devices in silicon technologies remain far from practical. Without these capabilities, routing and flow control in a packet-switched network, as typically envisioned for an on-chip optical interconnect system, require repeated optoelectronic (O/E) and electro-optic (E/O) conversion, which can significantly increase signal delay, circuit complexity, and energy consumption. Simultaneously, efficient silicon electro-optic modulators remain challenging due to the inherently poor nonlinear optical properties of silicon (*e.g.*, lack of Pockel effect), and silicon electro-optic modulators have to rely on other weaker physical mechanisms such as the plasma dispersion effect (refractive index change induced by free carriers) [7]. Hence the modulator design requires a long optical length, which results in large device size, typically in centimeters for a Mach-Zehnder interferometer (MZI) device [8]. Resonant devices such as micro-ring resonators can effectively slow the light and hence reduce the required device size [9]. These high-Q resonators, however, also reduce the bandwidth of the modulator by the same factor. There is therefore a fundamental trade-off between the modulation efficiency and delay.

Further, there is a fundamental bandwidth density challenge for the in-plane waveguided approach: the mode diameter of optical waveguides, which determines the minimum distance required between optical waveguides to avoid crosstalk, is significantly larger than the electrical interconnect pitch at deep sub-micrometer technology nodes, and will deteriorate as CMOS technologies scale [6]. Wavelength division multiplexing (WDM), proven to be effective in long distance fiber-optic communication systems, has been proposed to solve the problem and achieve the bandwidth-density goal. It is, however, not practical for intra-chip optical interconnects due to the significant area and power overhead required for wavelength multiplexing/demultiplexing. For example, the micro-ring based implementation needs resistive thermal bias [10] to stabilize its wavelength, which adds significant amount of static power dissipation [11].

Another challenge facing the in-plane waveguide approach is the optical loss and crosstalk from the large number of waveguide crossings [12], which severely limit the topology of the interconnect system [11] and hence the total aggregated system bandwidth. Placing waveguides onto a dedicated optics plane with multiple levels would require multiple silicon-on-insulator (SOI) layers, increasing the process complexity, and the performance gain is not scalable.

We therefore conclude that (a) it is critical to achieve the highest possible data rate in each optic channel at a fixed wavelength in an on-chip optical interconnect system in order to replace the electrical interconnects; (b) in-plane optical waveguides may not be the best solution to achieve the bandwidth goal; and (c) transistors and photonic devices have different physics, follow different scaling rules, and probably should be treated differently in the integration process.

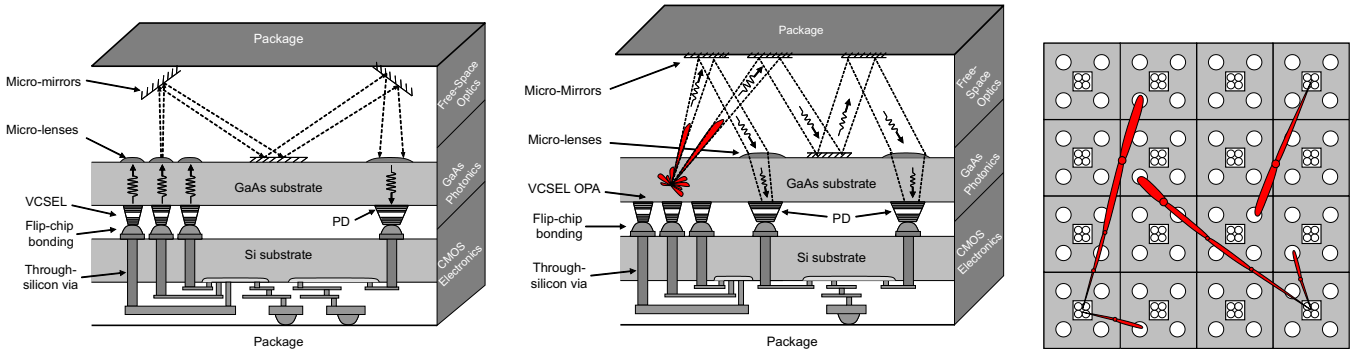
3 Overview

To address the challenges of building high-performance on-chip optical interconnects, we seek to use free-space optics and supporting device, circuit, and architecture techniques to create a high performance, complexity-effective interconnect system. We envision a system where a free-space optical communication layer, consisting of arrays of lasers, photodetectors, and micro-optics devices such as micro-mirrors and micro-lenses, is superimposed on top of the CMOS electronics layer via 3-D chip integration. This *free-space optical interconnect* (FSOI) system provides all-to-all direct communication links between processor cores, regardless of their topological distance. As shown in Figure 1, in a particular link, digital data streams modulate an array of lasers; each modulated light beam emitted by a laser is collimated by a micro-lens, guided by a series of micro-mirrors, focused by another micro-lens, and then detected by a photodetector (PD); the received electrical signals are finally converted to digital data. Note that the optical links are running at multiples of the core clock speed.

Without packet switching, this design eliminates the intermediate routing and buffering delays and makes the signal propagation delay approach the ultimate lower bound, *i.e.*, the speed of light. These links can operate at a much higher speed than core logic, making it easy to provide high throughput and low serialization latency. On the energy efficiency front, bypassing packet relaying clearly keeps energy cost low. As compared to waveguided optical interconnect, FSOI also avoids the loss and cross-talk associated with waveguide crossings. The aggregated bandwidth of such an all-to-all system can be significantly larger than both an electrical and waveguided optical interconnect. In the future, by utilizing the beamsteering capability of an optical phase array (OPA) of lasers, the number of lasers and photodetectors in each node can be constant, providing crucial scalability.

3.1 Lasers and Photodetectors

The lasers used in this FSOI system are vertical-cavity surface-emitting lasers (VCSELs) [13]. A VCSEL is a nanoscale heterostructure, consisting of an InGaAs quantum well active region, a resonant cavity constructed with a top and bottom dielectric mirrors (distributed Bragg reflectors), and a pn junction structure for carrier injection. They are fabricated on a GaAs substrate using molecular beam epitaxy (MBE) or metal-organic chemical vapor deposition (MOCVD). A VCSEL is typically a mesa structure with several microns in diameter and height. A large 2-D array with millions of VCSELs can be fabricated on the same GaAs chip.



(a) Side view (mirror-guided only) (b) Side view (with phase array beamforming) (c) Top view

Figure 1. Illustration of the overall interconnect structure and 3-D integrated chip stack. (a) and (b) also show two different optics configuration. In the top view (c), the VCSEL arrays are in the center and the photodetectors are on the periphery within each core.

The light can be emitted from the top of the VCSEL mesa. Alternatively, at the optical wavelength of 980-nm and shorter when the GaAs substrate is transparent, the VCSELs can also be made to emit from the back side of the GaAs substrate (substrate emitting). A VCSEL's optical output can be directly modulated by its current, and the modulation speed can reach tens of Gbps [14, 15].

The photodetectors can be either integrated on the CMOS chip as silicon p-i-n photodiodes [16], or fabricated on the same GaAs chip with the VCSELs as resonant cavity photodiodes [17, 18]. In the latter case, an InGaAs active region is enhanced by the resonant cavity similar to a VCSEL, and the devices offer a larger bandwidth and is well suited for this FSOI system.

3.2 Micro-lenses and Micro-mirrors

In the free-space optical interconnect, passive micro-optics devices such as micro-lenses and micro-mirrors collimate, guide, and focus the light beams in free space. Collimating and focusing allow smaller size VCSELs and PDs to be used, which reduces their parasitic capacitance and improve their bandwidth. Micro-lenses can be fabricated either on top of VCSELs when the latter are top emitting [19, 20], or on the backside of the GaAs substrate for substrate-emitting VCSELs [21, 22].

Micro-mirrors will be fabricated on silicon or polymer by micro-molding techniques [23, 24]. Looking forward, nanoscale photonic crystal devices are very promising to further reduce the feature size of these components [25].

3.3 3-D Integration

In this FSOI system, 3-D integration technologies are applied to electrically connect the free space and photonics layers with the electronics layer, forming an electro-optical system-in-package (SiP). For example, the GaAs chip is flip-chip bonded to the back side of the silicon chip, and connected to the transceiver circuits there using through-silicon-vias. Note that the silicon chip is flip-chip bonded to the package in a normal fashion. In general, such electro-optical SiP reduces the latency and power consumption of the global signaling through optical interconnect, while permitting the microprocessors to be implemented using standard CMOS technologies. Significant work has explored merging various analog, digital, and memory technologies in a 3-D stack. Adding an optical layer to the 3-D stack is the next logical step to improve overall system performance.

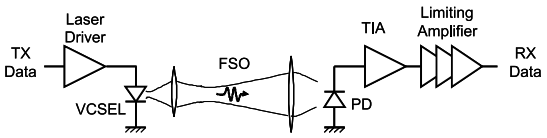


Figure 2. Intra-chip FSOI link calculation.

4 Architectural Design

4.1 Overall Interconnect Structure

As illustrated in Figure 1, in an FSOI link, a single light beam is analogous to a single wire and similarly, an array of VCSELs can form essentially a multi-bit bus which we call a *lane*. An interesting feature of using free-space optics is that signaling is not confined to fixed, prearranged waveguides and the optical path can change relatively easily. For instance, we can use a group of VCSELs to form a phase-array [26] – essentially a single tunable-direction laser as shown in Figure 1(b). This feature makes an all-to-all network topology much easier to implement.

For small- and medium-scaled chip-multiprocessors, fixed-direction lasers should be used for simplicity: each outgoing lane can be implemented by a dedicated array of VCSELs. In a system with N processors, each having a total of k bits in all lanes, $N * (N-1) * k$ VCSELs are needed for transmission. Note that even though the number scales with N^2, the actual hardware requirement is far from overwhelming. For a rough sense of scale, for $N = 16$, $k = 9$ (our default configuration for evaluation), we need approximately 2000 VCSELs. Existing VCSELs are about $20\mu m$x$20\mu m$ in dimension [14, 15]. Assuming, conservatively, $30\mu m$ spacing, 2000 VCSELs occupy a total area of about $5mm^2$. Note that on the receiving side, we do not use dedicated receivers. Instead, multiple light beams from different nodes share the same receiver. We do not try to arbitrate the shared receivers but simply allow packet collisions to happen. As will be discussed in more detail later, at the expense of having packet collisions, this strategy simplifies a number of other design issues.

4.2 Optical Links

To facilitate the architectural evaluation, a single-bit FSOI link is constructed (Figure 2) and the link performance is es-

timated for the most challenging scenario: communication across the chip diagonally. Note that the transceiver here is based on a conventional architecture, and is likely to be simplified for lower power dissipation. Since the whole chip is synchronous (*e.g.*, using optical clock distribution), no clock recovery circuit is needed.

The optical wavelength is chosen as 980 nm, which is a good compromise between VCSEL and PD performance. The serialized transmitted data is fed to the laser driver driving a VCSEL with a 5-μm optical aperture. The light from the back-emitting VCSEL is collimated through a microlens the backside of the 430-μm thick GaAs substrate. Using a device simulator, DAVINCI, and 2007 ITRS device parameter the performance and energy parameters of the optical link are calculated and detailed in Table 1. Note that the power dissipation of the serializer in the transmitter and deserializer in the receiver is much smaller compared to that of the laser driver and TIA, and hence is not included in the estimate. In addition to device scaling, the other reason our transmitter is much less power hungry than a commercial SERDES is that both the load and signal swing is much smaller (the VCSEL exhibits a resistance of over 100 Ω vs. typical 25 Ω when output matched; the VCSEL voltage swing is about 100 mV instead of several hundred mVs).

Free-space optics	
Propagation distance	2 cm
Optical wavelength	980 nm
Microlens aperture	transmitter 90 μm, receiver 190 μm
Optical path loss	2.6 dB
Transmitter	
VCSEL	aperture 5 μm, resistance 235 Ω, capacitance 90 fF, I_{th} 0.14 mA, extinction ratio 11:1
Driver bandwidth	43 GHz
Cycle-to-cycle jitter	1.7 ps
Active transmission	VCSEL (I_{bias} 0.48 mA, 2V, 0.96 mW), laser driver (6.3 mW)
Standby	0.43 mW (VCSEL biased below threshold and laser driver off)
Receiver	
PD	responsivity 0.5 A/W, capacitance 100 fF
Limiting amplifier	bandwidth 36 GHz, gain 15000 V/A
Total power	4.2 mW
Signal-to-noise ratio	7.5 dB
Bit-error-rate (BER)	10^{-10}

Table 1. Optical link parameters.

4.3 Network Design

4.3.1 Tradeoff to Allow Collision

With mirror-guided or phase array-based beamsteering, (dynamic) optical communication channels are built directly between communicating nodes within the network in a totally distributed fashion, without arbitration. An important consequence is that packets destined for the same receiver at the same time will collide. Such collisions require detection, retransmission, and extra bandwidth margin to prevent them from becoming a significant issue. However, for this one disadvantage, our design allows a number of other significant advantages (and later we will show that no significant overprovisioning is necessary):

- Compared to a conventional crossbar design, we do not need a centralized arbitration system. This makes the design scalable and reduces unnecessary arbitration latency for the common cases.
- Compared to a packet-switched interconnect, this design
  1. Avoids relaying and thus repeated O/E and E/O conversions in an optical network;
  2. Guarantees the absence of network deadlocks[1];
  3. Provides point-to-point message ordering in a straightforward fashion and thus allows simplification in coherence protocol designs;
  4. Reduces the circuit needs for each node to just drivers, receivers, and their control circuit. Significant amount of logic specific to packet relaying and switching is avoided (*e.g.*, virtual channel allocation, switch allocators, and credit management for flow control).
- The design allows errors and collisions to be handled by the same mechanism essentially requiring no extra support than that needed to handle errors, which is necessary in any system. Furthermore, once we accept collisions (with a probability on the orders of about 10^{-2}), the bit error rates of the signaling chain can be relaxed significantly (from 10^{-10} to, say, 10^{-5}) without any tangible impact on performance. This provides important engineering margins for practical implementations and further opportunities for energy optimization on the entire signaling chain.

4.3.2 Collision Handling

Collision detection Since we use the simple on-off keying (OOK) signaling, when multiple light beams from different source nodes collide at the same receiver node, the received light pulse becomes the logical "OR" of the multiple underlying pulses. The detection of the collision is simple, thanks to the synchrony of the entire interconnect. In the packet header, we encode both the sender node ID (PID) and its complement ($\overline{PID}$). When more than one packet arrives at the same receiver array, then at least one bit of the IDs (say PID_i) would differ. Because of the effective "OR" operation, the received PID_i and $\overline{PID_i}$ would both be 1, indicating a collision.

Structuring We take a few straightforward structuring steps to reduce the probability of collision.

1. Multiple receivers: It is beneficial to have a few receivers at each node so that it can receive multiple packets at the same time, reducing the probability of a collision. The effect can be better understood with some simple theoretical analysis. Using a simplified transmission model assuming equal probability of transmission and random destination, the probability of a collision per cycle in any node can be described as

$$1 - [(1 - \frac{p}{N-1})^n + C_n^1 \frac{p}{N-1}(1 - \frac{p}{N-1})^{n-1}]^R,$$

[1]Note that *fetch deadlock* is an independent issue that is not caused by the interconnect design itself. It has to be either prevented with multiple virtual networks, which is very resource intensive, or probabilistically avoided using NACKs [27]. We use the latter approach in all configurations.

where N is the number of nodes; p is the transmission probability of a node; R is the number of receivers (evenly divided among the $N-1$ potential transmitters); and $n = \frac{N-1}{R}$ is the number of nodes sharing the same receiver.

Numerical results are shown visually in Figure 3. It is worth noting that the simplifying assumptions do not distort the reality significantly. As can be seen from the plot, experimental results (details of the experimental setup is discussed later in Section 6) agree well with the theoretical calculations.

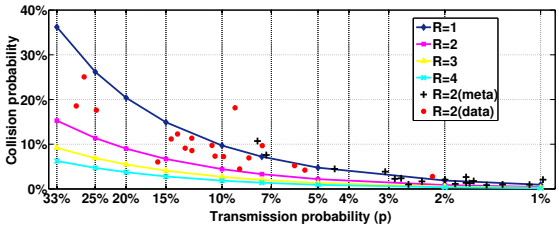


Figure 3. Collision probability (normalized to packet transmission probability) as a function of transmission probability p and the number of receivers per node (R). The result has an extremely weak dependency on the number of nodes in a system (N) as long as it is not too small. The plot shown is drawn with $N = 16$. To see that this simplified theoretical analysis is meaningful, we show experimental data points using two receivers (R=2). We separate the channels ("meta" and "data" channels as explained later).

To a first order approximation, collision frequency is inversely proportional to the number of receivers. Therefore, having a few (*e.g.*, 2-3) receivers per node is a good option. Further increasing the number will lead to diminishing returns.

2. Slotting and lane separation: In a non-arbitrated shared medium, when a packet takes multiple cycles to transmit, it is well known that "slotting" reduces collision probability [28]. For instance, suppose data packets take 5 processor cycles to transmit, then they can only start at the beginning of a 5-cycle slot. In our system, we define two packet lengths, one for *meta* packets (*e.g.*, requests and acknowledgments) and one for data packets (which is about 5 times the former). Each type will thus have a different slot length. In that case, slotting only reduces the the chance of collision between two packets of the same length (and thus the same slot length). Furthermore, the different packet lengths (especially because one is much longer than the other) also make the retransmission difficult to manage. One option to deal with both problems is to separate the packets into their own lanes and manage each lane differently.

3. Bandwidth allocation: Given a fixed bandwidth, we need to determine how to allocate the bandwidth between the two lanes for optimal performance. Even though a precise analytical expression between bandwidth allocation and performance is difficult to obtain, some approximate analysis can still be derived: each packet has an expected total latency of $L + P_c * L_r$, where L, P_c, and L_r are basic transmission latency, probability of collision, and collision resolution latency, respectively. L, P_c, and L_r are inversely proportional to the bandwidth allocated to a lane[2]. The overall latency can be expressed as $\frac{C_1}{B_M} + \frac{C_2}{B_M^2} + \frac{C_3}{1-B_M} + \frac{C_4}{(1-B_M)^2}$, where B_M is the portion of total bandwidth allocated to the meta packets, the constants ($C_{1..4}$) are a function of statistics related to application behavior and parameters that can be calculated analytically. The former includes the composition of packets (requests, data replies, forwarded requests, memory fetches, etc) and the percentage of meta and data packets that are on the critical path. The latter includes the average number of expected retries in a back-off algorithm. In our setup, the optimal latency value occurs at $B_M = 0.285$: about 30% of the bandwidth should be allocated to transmit meta packets. In reality, the allocation also needs to take into account considerations such as a packet should take an integer number of processor cycles for overall design simplicity. In our system, we use 3 VCSELs for the meta lane and 6 for the data lane, with a serialization latency of 2 (processor) cycles for a (72-bit) meta packet and 5 cycles for a (360-bit) data packet. Because we are using 2 separate receivers to reduce collisions, the receiving bandwidth is twice the transmitting bandwidth. For comparison, we use a baseline mesh network where the meta and data packets have a serialization latency of 1 and 5 cycles, respectively.

Confirmation Because a packet can get corrupted due to collision, some mechanism is needed to infer or to explicitly communicate the transmission status. For instance, a requester can time out and retry. However, solely relying on timeouts is not enough as certain packets (*e.g.*, acknowledgments) generate no response and the transmitter thus has no basis to infer whether the transmission was successful.

A simple hardware mechanism can be devised to confirm uncorrupted transmissions. We dedicate a set of VCSELs just to transmit a beam for confirmation. Upon receiving an uncorrupted packet, the receiver node activates the confirmation VCSEL and sends the confirmation to the sender. Note that by design, the confirmation beam will never collide with one another: when a packet is received in cycle n, the confirmation is sent after a fixed delay (in our case cycle $n+2$, after a cycle for any delay in decoding and error-checking). Since at any cycle n, only one packet (per lane) will be transmitted by any node, only one confirmation (per lane) will be received by that node in cycle $n+2$.

Retransmission Once packets are involved in a collision, the senders randomize their subsequent retries. In a straightforward way, the packet is retransmitted in a random slot within a window of W slots after the detection of the collision. The chance of further collision depends on W. The larger it is, the

[2] P_c is not exactly inversely proportional to bandwidth. Once transmitted, the probability of collision for 2-receiver designs is $1-(1-\frac{P_t}{N-1})^{\frac{N-2}{2}})$, where P_t is the transmission probability and N is the number of nodes. This approximately evaluates to $\frac{1}{2}\frac{1}{P_t} - \frac{1}{8}\frac{1}{P_t^2} + ...$ and can be treated as inversely proportional to P_t for a wide range of P_t.

smaller the probability of secondary collisions, but the longer the average delay in retransmission. Furthermore, as the retry continues, other packets may arrive and make collisions even more likely, greatly increasing the delay and energy waste. If we simply retry using the same window size, in the pathological case when too many packets arrive in a concentrated period, they can reach a critical mass such that it is more likely to receive a new packet from a different node to join the set than to have one successfully transmitted and leave the competition. This leads to a virtual live lock that we have to guard against.

Thus, we adopt an exponential back-off heuristic and set the window size to grow as the number of retries increases. Specifically, the window size for the r^{th} retry W_r is set to $W \times B^{r-1}$, where B is the base of the exponential function. While doubling the window size is a classic approach [29], we believe setting B to 2 is an over-correction, since the pathological case is a very remote possibility. Note that B need not be an integer. To estimate the optimal values of W and B without blindly relying on expensive simulations, we use a simplified analytical model of the network to derive the expression of the average collision resolution delay given W and B, and taking into account the confirmation laser delay (2 cycles). Although the calculation does not lead to a simple closed-form expression, numerical computation using packet transmission probability measured in our system leads to the results shown in Figure 4, where we varied B from 1 to 2 and W from 2 to 16.

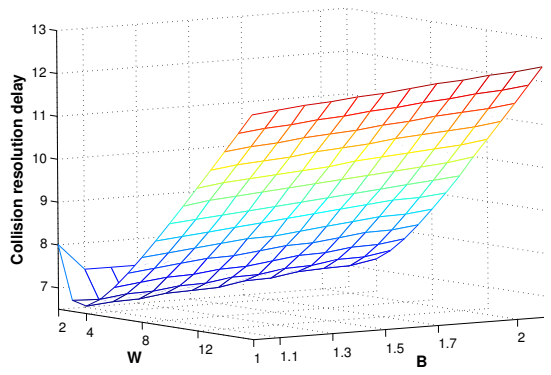


Figure 4. Average collision resolution delay for meta packets as a function of starting window and back-off speed. While retransmission is attempted, other nodes continue regular transmission. This "background" transmission has an insubstantial impact and is assumed to be 1% in the plot.

The minimum collision resolution delay occurs at $W = 3, B = 1.1$. We selected a few data points on the curve and verified that the theoretical computation agrees with execution-driven simulation rather well. (For instance, for $W = 3, B = 1.1$, the computed delay is 6.78 cycles and the simulated result is 6.57 cycles.) The graph clearly showed that $B = 1.1$ produces a decidedly lower resolution delay in the common case than when $B = 2$. This does not come at the expense of unacceptable delay in the pathological case. For example, in a 64-node system, when all other nodes send one packet to a particular node at the same time, it takes an average of about 26 retries (for a total of 416 cycles) to get one packet to come through. In contrast, with a fixed window size of 3, it would take 8.2×10^{10} number of retries. Setting B to 2, shortens this to about 5 retries (199 cycles).

4.4 Protocol Considerations

The delivery-order property of the interconnect can impact the complexity of the coherence protocol [27]. Our system does not rely on relaying and thus makes it easy to enforce point-to-point message ordering. We delay the transmission of another message about a cache line until a previous message about that line has been confirmed. This serialization reduces the number of transient states the coherence protocol has to handle. We summarize the remaining transient states in the protocol in Table 2.

5 Optimizations

While a basic design described above can already support the coherency substrate and provide low-latency communication, a perhaps more interesting aspect of using optical interconnect is to explore new communication or protocol opportunities. Below, we describe a few optimizations that we have explored in the proposed interconnect architecture.

5.1 Replacing Acknowledgments with Confirmations

The presence of the confirmation laser provides a unique opportunity to design messaging and protocol support to expedite certain transactions. For brevity, we only describe one simple and effective use which is to eliminate the explicit acknowledgments of an invalidation request. This approach reduces the traffic and, as we will show in Section 7.3, significantly reduces the probability of collisions. The acknowledgments are needed to determine write completion, and help ensure *write atomicity* and determine when memory barriers can finish in a relaxed consistency model [27].

To eliminate the need for acknowledgment, we use the confirmation signal as a *commitment* of carrying out the invalidation [27]. This commitment logically serializes the invalidation before any subsequent externally visible transaction. For instance, in a sequentially consistent system, any load (to the invalidated cache line) following that externally visible transaction need to reflect the effect of the invalidation and replay if it is speculatively executed out of order. For practical implementation, we freeze the retirement of any memory instructions until we have applied all pending invalidations in the input packet queue. If an in-flight load's address matches any such invalidation, the load and all subsequent instructions are replayed [30].

5.2 Ameliorating data packet collisions

Since data packets are longer than meta packets, the chance of collision is higher. Furthermore, their collisions also cause more damage – more bits need to be retransmitted and it takes longer to resolve the collision as well, adding more delay to the effective latency. On the other hand, data packets also have unique properties that can be leveraged in managing collisions: they are often the result of earlier requests. This has two implications. First, the receiver has some control over the

timing of their arrival and can use that control to reduce the probability of a collision to begin with. Second, the receiver also may have a general idea which nodes are involved in the collision and can play a role in the subsequent retransmission period.

Request spacing When a request results in a data packet reply, the most likely slot into which the reply falls can be calculated. The overall latency includes queuing delays for both the request and the reply, the collision resolution time for the request, and the memory access latency. All these components can be analyzed as independent discreet random variables. Figure 5 shows an example of the distribution of the overall latency of a read-miss request averaged over all application runs in our environment for illustration. Note that different requests have a different distribution.

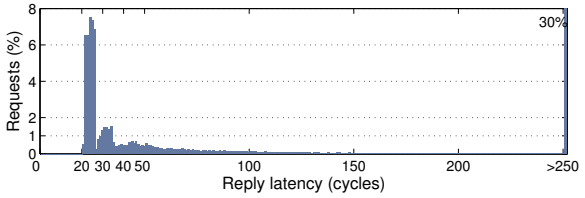


Figure 5. Probability distribution of the overall latency of a request resulting in a data reply.

As we can see, the probability is heavily concentrated in a few choices. Accordingly, we can reserve slots on the receiver. If a slot is already reserved, a request gets delayed to minimize the chance of collision. A writeback is a special case. It is generated without the receiver's prior request and thus out of the control of the receiving node. One tradeoff is to split a writeback into a meta packet request for writeback, followed by a notification from the receiving node granting the writeback, and finally followed by the actual writeback. This reduces data packet collision probability at the expense of increased latency (of an arguably non-critical operation) and increased traffic in the meta packet lane (and thus meta packet collision probability).

Hints in collision resolution When packets collide, each sender retries with the exponential back-off algorithm that tries to balance the wait time and the probability of secondary collisions (Section 4.3.2). However, the design of the algorithm assumes no coordination among the senders. Indeed, the senders do not even know the packet is involved in a collision until cycles after the fact nor do they know the identities of the other parties involved.

In the case of the data packet lane, the receiver knows of the collision early, immediately after receiving the header that encodes PID and $\overline{PID}$. It can thus send a no-collision notification to the sender before the slot is over. The absence of this notification is an indication that a collision has occurred. Moreover, even though in a collision the PID and $\overline{PID}$ are corrupted due to the collision and only indicate a super-set of potential transmitters[3], the receiver has the benefit of additional knowledge of the potential candidates – those nodes that are expected to send a data packet reply. Based on this knowledge, the receiver can select one transmitting node as the winner for the right to re-transmit immediately in the next slot. This selection is beamed back through a notification signal (via the confirmation laser) to the winner only. All other nodes that have not received this notification will avoid the next slot and start the re-transmission with back-off process from the slot after the next. This way, the winning node suffers a minimal extra delay and the remaining nodes will have less retransmission contention. Note that, this whole process is probabilistic in that the receiving node does not need to be 100% certain about the identity of the senders and the notification it sends is only considered a hint.

Finally, we note that packet collisions are ultimately infrequent. So a scheduling-based approach that avoid all possible collisions does not seem beneficial, unless the scheduling overhead is extremely low.

[3] Clearly, for small-scale networks, one could use a bit vector encoding of PID and thus allow the receiver to definitively identify the colliding parties all the time.

6 Evaluation Environment

We evaluated our optical interconnect proposal on an execution-driven chip multiprocessor (CMP) simulator. The base configuration is a 16-way CMP with private L1s and distributed shared L2. The following describes the details of various components involved in the simulator.

Shared-memory coherency substrate The simulator takes DEC alpha binaries and emulates system calls needed for parallel workload, such as for thread creation. It also supports synchronization instructions ldl_l and stl_c (load-linked and store-conditional) and combining tree barriers [27]. The simulator models an MESI-style directory-based protocol with a detailed and faithful model of both stable and transient states and queuing of requests. Table 2 shows the state transitions both for L1 and the directory controllers.

Processor microarchitecture For the processor microarchitecture, we strive to faithfully model the DEC alpha 21264 [31]. Our code is an extensively adapted version of SimpleScalar [32] 3.0. Changes include faithful modeling of the memory barriers, load-store and load-load replays, scheduling replays, etc. All memory transactions are modeled using an event-driven framework accounting for latency, bandwidth constraints, bank queuing, and other contentions. Miss status holding registers (MSHR) are also faithfully modeled. We also implemented non-blocking memory controllers to faithfully simulate accesses to the memory. Memory is address-interleaved. Every controller serves the addresses mapped to one of the four quadrants in the 4x4 mesh and uses a separate router to connect to the cores. Further details of the memory controller is shown in Table 3.

Communication substrate For the proposed optical interconnect, we modeled timing, confirmation, collision, queuing, and overflows in detail. For the 16-way CMP, we modeled a dedicated laser array. For the scaled up system (64-way), we modeled a phase array based transmitter system and one cycle delay in re-setting the phase controller register. For the conventional packet-switched interconnect, we incorporated PopNet [33] network simulator and extended it to model routers other than the canonical 4-stage routers. Details of the system

L1 cache controller transitions								
State	Read	Write	Repl	Data	ExcAck	Inv	Dwg	Retry
I	Req(Sh)/I.S^D	Req(Ex)/I.M^D	error	error	error	InvAck/I	DwgAck/I	error
S	do read/S	Req(Upg)/S.M^A	evict/I	error	error	InvAck/I	error	error
E	do read/E	do write/M	evict/I	error	error	InvAck/I	DwgAck/S	error
M	do read/M	do write/M	evict/I	error	error	InvAck(D)/I	DwgAck(D)/S	error
I.S^D	z	z	z	save & read/S or E	error	InvAck/I.S^D	DwgAck/I.S^D	Req(Sh)
I.M^D	z	z	z	save & write/M	error	InvAck/I.M^D	DwgAck/I.M^D	Req(Ex)
S.M^A	z	z	z	error	do write/M	InvAck/I.M^D	error	Req(Upg)
L2 directory controller transitions								
State	Req(Sh)	Req(Ex)	Req(Upg)	WriteBack	InvAck	DwgAck	MemAck	Repl
DI	Req(Mem)/DI.DS^D	Req(Mem)/DI.DM^D	Req(Mem)/DI.DM^D	error	error	error	error	error
DV	Data(E)/DM	Data (M)/DM	error	error	error	error	error	evict/DI
DS	Data(S)/DS	Inv/DS.DM_D^A	Inv/DS.DM^A	error	error	error	error	Inv/DS.DI^A
DM	Dwg/DM.DS^D	Inv/DM.DM^D	Inv/DM.DM^D	save/DV	error	error	error	Inv/DM.DI^D
DI.DS^D	z	z	z (Req(Ex))	error	error	error	repl & fwd/DM	z
DI.DM^D	z	z	z (Req(Ex))	error	error	error	repl & fwd/DM	z
DS.DI^A	z	z	z (Req(Ex))	error	evict/DI	error	error	z
DS.DM_D^A	z	z	z (Req(Ex))	error	Data(M)/DM	error	error	z
DS.DM^A	z	z	z (Req(Ex))	error	ExcAck/DM	error	error	z
DM.DI^D	z	z	z (Req(Ex))	save/DS.DI^A	save & evict/DI	error	error	z
DM.DS^D	z	z	z (Req(Ex))	save/DM.DS^A	error	save & fwd/DM	error	z
DM.DM^D	z	z	z (Req(Ex))	save/DM.DM^A	save & fwd/DM	error	error	z
DM.DS^A	z	z	z (Req(Ex))	error	error	Data(E)/DM	error	z
DM.DM^A	z	z	z (Req(Ex))	error	Data(M)/DM	error	error	z

Table 2. Cache controller transitions for L1 and L2 cache. The rows are the current state, the columns are the events/requests, and each entry contains an <action/next state> pair. Impossible cases are marked "error" and "z" means the event cannot currently be processed, and in some cases, the incoming request will be reinterpreted as a different one due to race. M, E, S, and I are stable states of L1 cache controller and DM, DS, DV (Valid with no sharers), and DI are stable states of L2 directory controller. Transient states are denoted by the pair of previous and next stable state. Transient states waiting for a data reply are superscripted with D and those waiting for just an acknowledgment are superscripted with A. All request events (Req) are followed by request type *i.e.*, (Sh: read in shared mode, Ex: read in exclusive mode, Upg: upgrade request, Dwg: downgrade request, and Mem: memory access request).

configuration is shown in Table 3.

16-way CMP, private L1, distributed shared L2	
Processor core	
Fetch/Decode/Commit	4 / 4 / 4
ROB	64
Functional units	INT 1+1 mul/div, FP 2+1 mul/div
Issue Q / Reg. (int,fp)	(16, 16) / (64, 64)
LSQ(LQ,SQ)	32 (16,16) 2 search ports
Branch predictor	Bimodal + Gshare
- Gshare	8K entries, 13 bit history
- Bimodal/Meta/BTB	4K/8K/4K (4-way) entries
Br. mispred. penalty	at least 7 cycles
Process specifications	Feature size: 45nm, Frequency: 3.3 GHz, V_{dd}: 1 V
Memory hierarchy	
L1 D cache (private)	8KB [34], 2-way, 32B line, 2 cycles, 4 ports (duplicate tags for coherence controller)
L1 I cache (private)	32KB, 2-way, 64B line, 1 cycle
L2 cache (shared)	1MB, 8-way, 16 banks, 64B line, 15 cycles, 2 ports
Memory bus	1.1 GHz, 4 links, 64-bit link width, access latency 200 CPU cycles
Prefetch logic	stream prefetcher [35, 36]
Network packets	Flit size: 64-bit, data packet: 5 flits, meta packet: 1 flit
Wired interconnect	4x4 mesh + 4 nodes for memory controller, 4 virtual channels, 5x12 flit input buffer, 1 cycle link latency, 4 cycle routing delay, 64-bit link width
Optical interconnect (each node)	
VCSEL frequency	40 GHz, 12-bit per CPU cycle
Packet transmitter	16x6 VCSELs for data lane, 16x3 VCSELs for meta lane
Packet receiver	2 receiver arrays x 6 PDs (each includes photo detector and trans-impedance amplifier) for data lane, 2x3 for meta lane.
Confirmation	2 sets of (16 VCSELs + 1 receiver) for each lane
Input buffer	40 flits for data channel, 8 flits for meta channel
Output buffer	16 flits

Table 3. Baseline system configuration.

Power The simulator includes both switching and leakage power models. Switching power of the processor core, coherence controller, memory subsystems, and interconnect buffers is modeled by extending Wattch [37]. Leakage power is temperature-dependent and computed based on predictive SPICE circuit simulations for 45nm technology using BSIM3 [38]. We used HotSpot [39] to model dynamic temperature variation across the chip. The floorplan is derived from that of Alpha 21364. We base device parameters on the 2004 ITRS projection of 45nm CMOS technology file. Power consumption modeling of the optical links is described in Section 4.2. Conventional interconnect power consumption is modeled using Orion [40].

Applications Evaluation is performed using a suite of parallel applications including SPLASH2 benchmark suite [34], two applications from PARSEC benchmark suite [41], a program to solve electromagnetic problem in 3 dimensions (*em3d*) [42], a parallel genetic link-age analysis program (*ilink*) [43], a program to iteratively solve partial differential equations (*jacobi*), a 3-dimensional particle simulator (*mp3d*), a shallow water benchmark from the National Center for Atmospheric Research to solve difference equations on a two-dimensional grid for weather prediction (*shallow*), and a branch-and-bound based implementation of the non-polynomial (NP) traveling salesman problem (*tspuo*) and a version that is optimized to address false sharing (*tspo*). We follow the recommendation in [34] to scale down the L1 cache to mimic realistic cache miss rate.

7 Evaluation

The proposed intra-chip free-space optical interconnect has many different design tradeoffs compared with a conventional

wire-based interconnect or newer proposals of optical versions. Some of these tradeoffs can not be easily expressed in quantitative terms, and are discussed in the architectural design and later in the related work section. Here, we attempt to demonstrate that the proposed design offers ultra-low latency, excellent scalability, and superior energy efficiency. We also show that accepting collisions does not necessitate drastic bandwidth over-provisioning. We start our evaluation with the performance analysis of the proposed interconnect.

7.1 Performance Analysis

We model a number of conventional interconnect configurations for comparison. To normalize performance, we use a baseline system with canonical 4-cycle routers. Note that while the principles of conventional routers and even newer designs with shorter pipelines are well understood, practical designs require careful consideration of flow control, deadlock avoidance, QoS, and load-balancing and are by no means simple and easy to implement. For instance, the router in Alpha 21364 has hundreds of packet buffers and occupies a chip area equal to 20% of the combined area of the core and 128KB of L1 caches. The processing by the router itself adds 7 cycles of latency [44]. Nevertheless, we provide comparison with conventional interconnects with aggressive latency assumptions.

Figure 6 shows the average latency of transferring a packet in our free-space optical interconnect and in the baseline mesh interconnect. Latency in the optical interconnect is further broken down into queuing delay, intentionally scheduled delay to minimize collision, the actual network delay, and collision resolution delay. Clearly, even with the overhead of collision and its prevention, the overall delay is still very low.

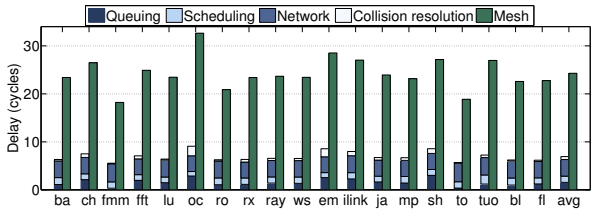


Figure 6. Total packet latency in the free-space optical interconnect (left) and the conventional mesh (right). The total latency in the optical interconnect is broken down into 4 components (queuing delay, scheduling delay, network latency, and collision resolution delay).

We show the application speedup in Figure 7. We use the ultimate execution time[4] of the applications to compute speedups against the baseline using a conventional mesh interconnect. For relative comparison, we model a number of conventional configurations: L_0, L_{r1}, and L_{r2}. In L_0, the transmission latency is idealized to 0 and only the throughput is modeled. In other words, the only delay a packet experiences is the serialization delay (1 cycle for meta packets and 5 cycles for data packets) and any queuing delay at the source node. L_0 is essentially an idealized interconnect. L_{r1} and L_{r2} represent the cases where each hop consumes 1 cycle for link traversal and 1 or 2 cycles respectively for router processing.

[4]For applications too long to finish, we measure the same workload, *e.g.*, between a fixed number of barrier instances.

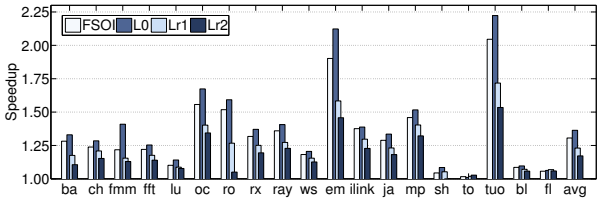


Figure 7. Speedups of free-space optical interconnect (FSOI) and various configurations of conventional mesh relative to the baseline.

While the performance gain varies from application to application, our design tracks the ideal L_0 configuration well, achieving a geometric mean of 1.31 speedup versus the ideal's 1.36. It also outperforms the aggressive L_{r1} (1.23) and L_{r2} (1.17) configurations. The performance benefit is especially obvious for those applications that are sensitive to interconnect performance. For instance, *em3d* and *tspuo* run about 20% faster on FSOI than on a mesh with single-cycle routers.

Although a mesh interconnect is scalable in terms of aggregate bandwidth provided, latency worsens as the network scales up. In comparison, our design offers a direct-communication system that is scalable while maintaining low latency. In another experiment, we increase the system size to 64 nodes and for the mesh interconnect, we double the input/output buffer size. All other parameters remain the same. The simulation results are shown in Figure 8 (for latency) and Figure 9 (for performance comparison).

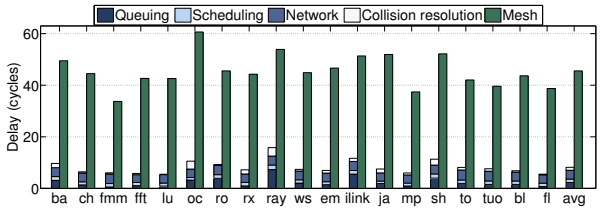


Figure 8. Total packet latency in the free-space optical interconnect and the conventional mesh on a 64-way CMP.

As expected, latency in mesh interconnect increases significantly, but the latency does increase in our network too, from 6.7 cycles in a 16-node system to 8.2 cycles in a 64-node system. However, much of this increase is due to an increase of 1.1 cycles in queuing delays on average. In certain applications (*e.g.*, *raytrace*), the increase is significant. This increase in queuing delays is not a result of interconnect scalability bottleneck, but rather a result of how the interconnect is *used* in applications with a larger number of threads. For example, locks are likely to be more heavily contested, and when they are released, more invalidations are needed, causing large temporary queuing delays. Indeed, the queuing delay of 2.3 cycles in our system is only marginally higher than the 1.9 cycles in the ideal L_0 configuration.

Understandably, the better scalability led to wider performance gaps between our optical interconnect and the non-ideal mesh configurations. The speedup of our FSOI continues to track that of the ideal L_0 configuration (with a geomet-

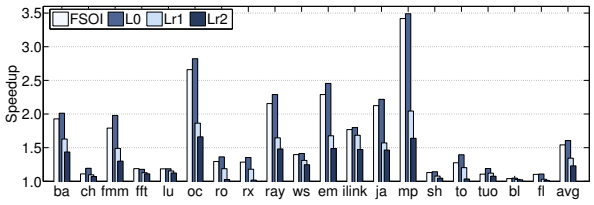


Figure 9. Speedups of free-space optical interconnect (FSOI) and ideal interconnect relative to the baseline mesh interconnect on a 64-way CMP.

ric mean of 1.55 vs 1.61), and pulls further ahead of those of L_{r1} (1.34) and L_{r2} (1.23).

In sum, the proposed interconnect offers an ultra-low communication latency and maintains a low latency as the system scales up. The system outperforms aggressively configured packet-switched interconnect and the performance gap is wider for larger-scale systems and for applications whose performance has a higher dependence on the interconnect.

7.2 Energy Consumption Analysis

We have also performed a preliminary analysis of the energy characteristics of the proposed interconnect. Figure 10 shows the total energy consumption of the system normalized to the baseline configuration using mesh. Our direct communication substrate avoids the inherent inefficiency in repeated buffering and processing in a packet-switched network. The energy spent in the interconnect itself is an order of magnitude less: about 8.9% of the energy spent in the mesh. The fast execution also saves energy overhead inside the cores. On average, our system achieves a 29.3% energy savings. The energy savings has roughly the same magnitude with execution time reduction, resulting in a relatively small difference in the average power: 151W for conventional system and 139W for our design.

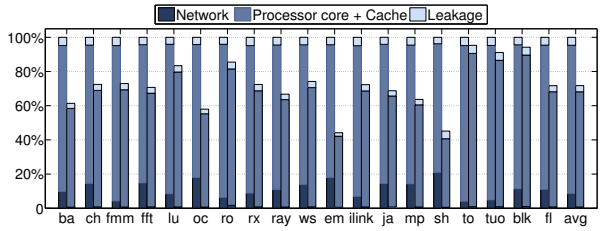


Figure 10. Energy relative to baseline mesh interconnect.

7.3 Analysis of Optimization Effectiveness

Meta packet collision reduction Our design does not rely on any arbiter to coordinate the distributed communication, making the system truly scalable. The tradeoff is the presence of occasional packet collisions. Several mechanisms are used to reduce the collision probability. The most straightforward of these mechanisms is using more receivers. We use 2 receivers per lane. Our detailed simulations show that this indeed roughly reduces collisions by half in both cases as predicted by the simplified theoretical calculation and Monte Carlo simulations. This partly validates the use of simpler analytical means to make design decisions.

Using the confirmation of successful invalidation delivery as a substitute for an explicit acknowledgment packet is a particularly effective approach to further reduce unnecessary traffic and collisions. Figure 11 shows the impact of this optimization. The figure represents each application by a pair of points. The coordinates show the packet transmission probability and the collision rate of the meta packet lane.

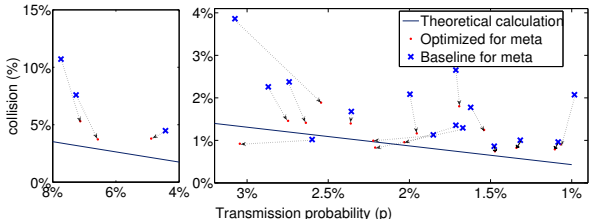


Figure 11. Change in packet transmission probability and collision rate with and without the optimization of using confirmation signal to substitute acknowledgment. For clarity, the applications are separated into two distinctive regions.

In general, as we reduce the number of packets (acknowledgments), we reduce the transmission probability and naturally the collision rate. However, if reduction of the transmission probability is the only factor in reducing collisions, the movement of the points would follow the slope of the curve which shows the theoretical collision rate given a transmission probability. Clearly, the reduction in collision is much sharper than simply due to the reduction of packets. This is because the burst of the invalidation messages sent leads to acknowledgments coming back at approximately the same time and much more likely to collide than predicted by theory assuming independent messages. Indeed, after eliminating these "quasi-synchronized" packets, the points move much closer to the theoretical predictions. Clearly, avoiding these acknowledgments is particularly helpful. Note that, because of this optimization, some applications speed up and the per-cycle transmission probability actually increases. Overall, this optimization reduces traffic by only 4.2% but eliminates about 33.6% of meta packet collisions.

Data packet collision reduction We also looked at a few ways to reduce collisions in the data lane. These techniques include probabilistically scheduling the receiver for the incoming replies, applying split transactions for writebacks to minimize unexpected data packets, and using hints to coordinate retransmissions (Section 5.2). Figure 12 shows the breakdown of the type of collisions in the data packet lane with and without these optimizations. The result shows the general effectiveness of the techniques: about 27.1% of all collisions are avoided.

Data packet collision resolution hint As discussed in Section 5.2, when a data lane collision happens we can guess the identities of the senders involved. From the simulations, we can see that based on the information of potential senders and the corrupted pattern of PID and $\overline{PID}$, we can correctly identify a colliding sender 80% of the time. Even for the rest of the time when we mis-identify the sender, it is usually harmless: If the mis-identified node is not sending any data packet at the time and it simply ignores the hint. Overall, the hints are quite accurate and on average, only 2.4% of the

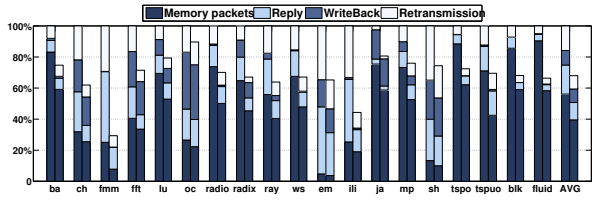


Figure 12. Breakdown of data packet collisions by type: involving memory packets (Memory packets), between replies (Reply), involving writebacks (Writeback), and involving re-transmitted packets (Retransmission). The left and the right bars show the the result without and with the optimizations, respectively. The collision rate for data packets ranges from 2.8% to 25.1%, with an average of 10.2%. After optimization, the collision rate is between 1.5% and 20.6% with an average of 7.4%.

hints cause a node to wrongly believe it is selected as a winner to re-transmit. As a result, the hint improves the collision resolution latency in general and Figure 13 shows the result.

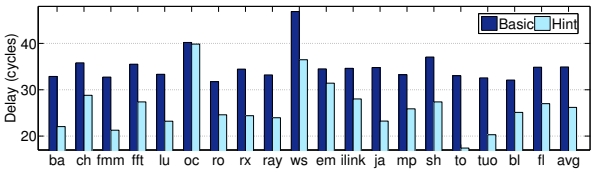


Figure 13. The average collision resolution delay when using collision resolution hint (Hint) and without this hint (Basic).

Finally, note that all these measures that reduce collisions may not lead to significant performance gain when the collision probability is low. Nevertheless, these measures lower the probability of collisions when traffic is high and thus improve the resource utilization and the performance robustness of the system.

7.4 Sensitivity Analysis

As discussed before, we need to over-provision the network capacity to avoid excessive collisions in our design. However, such over-provisioning is not unique to our design. Packet-switched interconnects also need capacity margins to avoid excessive queuing delays, increased chance of network deadlocks, etc. In our comparison so far, the aggregate bandwidth of the conventional network and of our design are comparable: the configuration in the optical network design has lower (50%) transmitting bandwidth and roughly the same receiving bandwidth as the baseline conventional mesh. To understand the sensitivity of the system performance to the communication bandwidth provided, we progressively reduce the bandwidth until it is halved. For our design, this involves reducing the number of VCSELs, rearranging them between the two lanes, and adjusting the cycle-slotting as the serialization latency for packets increases[5]. Figure 14 shows the overall performance impact. Each network's result is normalized to that of its full-bandwidth configuration. For brevity, only the average slowdown of all applications is shown.

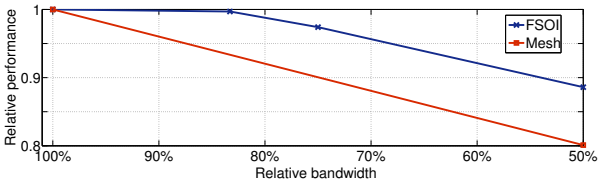


Figure 14. Performance impact due to reduction in bandwidth.

We see that both interconnects demonstrate noticeable performance sensitivity to the communication bandwidth provided. In fact, our system shows *less* sensitivity. In other words, both interconnects need to over-provision bandwidth to achieve low latency and high execution speed. The issue that higher traffic leads to higher collision rate in our proposed system is no more significant than factors such as queuing delays in a packet-relaying interconnect; it does not demand drastic over-provisioning. In the configuration space that we are likely to operate in, collisions are reasonably infrequent and accepting them is a worthwhile tradeoff. Finally, thanks to the superior energy efficiency for the integrated optical signaling chain, bandwidth provisioning is rather affordable energy-wise.

8 Related Work

The effort to leverage optics for on-chip communication spans multiple disciplines and there is a vast body of related work, especially on the physics side. Our main focus in this paper is to address the challenge in building a scalable interconnect for general-purpose chip-multiprocessors, and doing so without relying on repeated O/E and E/O conversions or future breakthroughs that enable efficient pure-optical packet switching. In this regards, the most closely related design that we are aware of is [4].

In [4], packets do not need any buffering (and thus conversions) at switches within the Omega network because when a conflict occurs at any switch, one of the contenders is dropped. Even though this design addresses part of the challenge of optical packet switching by removing the need to buffer a packet, it still needs high-speed optical switches to decode the header of the packet in a just-in-the-time fashion in order to allow the rest of the packet to be switched correctly to the next stage. In a related design [45], a circuit-switched photonic network relies on an electrical interconnect to route special circuit setup requests. Only when an optical route is completely set up can the actual transfer take place. Clearly, only bulk transfers can amortize the delay of the setup effort. In contrast to both designs, our solution does not rely on any optical switch component.

Among the enabling technologies of our proposed design, free-space optics have been discussed in general terms in [3, 46]. There are also discussions of how free-space optics can serve as a part of the global backbone of a packet-switched interconnect [47] or as an inter-chip communication mechanism (*e.g.*, [48]). On the integration side, leveraging 3D integration to build on-chip optoelectronic circuit has also been mentioned as an elegant solution to address various integra-

[5] For easier configuration of the optical network, we use a slightly different base configuration for normalization. In this configuration, both data and meta lanes have 6 VCSELs and as a result, the serialization latency for a meta packet and a data packet is 1 and 5 cycles respectively – the same as in the mesh networks.

tion issues [49].

Many proposals exist that use a globally shared medium for the optical network and use multiple wavelengths available in an optical medium to compensate for the network topology's non-scalable nature. [50] discussed dividing the channels and using some for coherence broadcasts. [51] also uses broadcasts on the shared bus for coherence. A recent design from HP [11, 52] uses a microring-based EO modulator to allow fast token-ring arbitration to arbitrate the access to the shared medium. A separate channel broadcast is also reserved for broadcast. Such wavelength division multiplexing (WDM) schemes have been proven highly effective in long-haul fiber-optic communications and inter-chip interconnects [53, 54]. However, there are several critical challenges to adopt these WDM systems for intra-chip interconnects. First, they require a large number of wavelength multiplexing/demultiplexing devices, which can be implemented as either March-Zenhder interferometers (MZI) [55], or smaller size micro-resonator-based optical add-drop filters [56–61], Such large number of add-drop filters will consume significant amount of chip area. The situation will further deteriorate with technology scaling since the size of these photonic devices is limited by the optical wavelength. Second, each one of the add-drop filters needs to achieve very stringent spectral and loss requirements, which translates into extremely fine device geometries and little tolerance for fabrication variability [57–61]. Currently, electron-beam lithography is needed to achieve the resolution, and the the manufacturing challenges to move the process into production is even greater than integrating non-silicon components. Furthermore, the fine wavelength resolution in WDM will fundamentally translate into larger latency in device response for both modulators and add-drop filters. In addition, there is a large hidden cost of multiple external laser sources at multiple wavelengths, and each of them requires finer linewidth than a single-wavelength system. Therefore, the number of wavelengths employed in on-chip optical interconnects will likely to be limited to a small number.

9 Conclusion

While optics are believed to be a promising long-term solution to address the worsening processor interconnect problem as technology scales, significant technical challenges remain to allow scalable optical interconnect using conventional packet switching technology. In this paper, we have proposed a scalable, fully-distributed interconnect based on free-space optics. The design leverages a suite of maturing technologies to build an architecture that supports a direct communication mechanism between nodes and does not rely on any packet switching functionality and thus side-steps the challenges involved in implementing efficient optical switches. The tradeoff is the occasional packet collisions from uncoordinated packet transmissions. The negative impact of collisions is minimized by careful architecting of the interconnect and novel optimizations in the communication and coherence substrates of the multiprocessor.

Based on parameters extracted from device and circuit simulations, we have performed faithful architectural simulations with detailed modeling of the microarchitecture, the memory subsystems, the communication substrate, and the coherence substrates to study the performance and energy metrics of the design. The study shows that compared to conventional electrical interconnect, our design provides good performance (superior than even the most aggressively configured mesh interconnect), better scalability, and a far better energy efficiency. With the proposed architectural optimizations to minimize the negative consequences of collisions, the design is also shown to be rather insensitive to bandwidth capacity. Overall, we believe the proposed ideas point to promising design spaces for further exploration.

References

[1] SIA. International technology roadmap for semiconductors. Technical report, 2008.

[2] J.W. Goodman, F.J. Leonberger, et al. Optical Interconnections for VLSI Systems. *Proc. IEEE*, 72:850–866, July 1984.

[3] D.A.B. Miller. Optical interconnects to silicon. *Selected Topics in Quantum Electronics, IEEE Journal of*, 6(6):1312–1317, Nov/Dec 2000.

[4] A. Shacham and K. Bergman. Building Ultralow-Latency Interconnection Networks Using Photonic Integration. *IEEE Micro*, 27(4):6–20, July/August 2007.

[5] Y. Vlasov, W. M. J. Green, and F. Xia. High-Throughput Silicon Nanophotonic Wavelength-Insensitive Switch for On-Chip Optical Networks. *Nature Photonics*, (2):242–246, March 2008.

[6] M. Haurylau, G. Chen, J. Zhang, N.A. Nelson, D.H. Albonesi, E.G. Friedman, and P.M. Fauchet. On-chip optical interconnect roadmap: Challenges and critical directions. *IEEE J. Sel. Quantum Electronics*, (6):1699–1705, 2006.

[7] R. Soref and B. Bennett. Electrooptical effects in silicon. *Quantum Electronics, IEEE Journal of*, 23(1):123–129, Jan 1987.

[8] Ling Liao, Dean Samara-Rubio, Michael Morse, Ansheng Liu, Dexter Hodge, Doron Rubin, Ulrich Keil, and Thorkild Franck. High speed silicon mach-zehnder modulator. *Opt. Express*, 13(8):3129–3135, 2005.

[9] Qianfan Xu, Bradley Schmidt, Sameer Pradhan, and Michal Lipson. Micrometre-scale silicon electro-optic modulator. *Nature*, 435(7040):325–327, May 2005.

[10] Sasikanth Manipatruni, Rajeev K. Dokania, Bradley Schmidt, Nicolás Sherwood-Droz, Carl B. Poitras, Alyssa B. Apsel, and Michal Lipson. Wide temperature range operation of micrometer-scale silicon electro-optic modulators. *Opt. Lett.*, 33(19):2185–2187, 2008.

[11] R. Beausoleil et al. A Nanophotonic Interconnect for High-Performance Many-Core Computation. *IEEE LEOS Newsletter*, June 2008.

[12] Wim Bogaerts, Pieter Dumon, Dries Van Thourhout, and Roel Baets. Low-loss, low-cross-talk crossings for silicon-on-insulator nanophotonic waveguides. *Opt. Lett.*, 32(19):2801–2803, 2007.

[13] Rainer Michalzik and Karl Joachim Ebeling. *Vertical-Cavity Surface-Emitting Laser Devices*, chapter 3, pages 53–98. Springer, 2003.

[14] K. Yashiki, N. Suzuki, K. Fukatsu, T. Anan, H. Hatakeyama, and M. Tsuji. 1.1-um-Range Tunnel Juntion VCSELs with 27-GHz Relaxation Oscillation Frequency. In *Proc. Optical Fiber Communications Conf.*, page Paper OMK1, 2007.

[15] Y.-C. Chang, C.S. Wang, and L.A. Coldren. High-efficiency, high-speed vcsels with 35[emsp4 1/4-em space]gbit/s error-free operation. *Electronics Letters*, 43(19):1022–1023, 2007.

[16] B. Ciftcioglu, Jie Zhang, Lin Zhang, J.R. Marciante, J.D. Zuegel, R. Sobolewski, and Hui Wu. 3-ghz silicon photodiodes integrated in a 0.18-mum cmos technology. *Photonics Technology Letters, IEEE*, 20(24):2069–2071, Dec.15, 2008.

[17] A. Chin and T.Y. Chang. Enhancement of quantum efficiency in thin photodiodes throughabsorptive resonance. *J. Vac. Sci. and Techn.*, (339), 1991.

[18] G.G. Ortiz, C.P. Hains, J. Cheng., H.Q. Hou, and J.C. Zolper. Monolithic integration of in0.2ga0.8asvertical-cavity surface-emitting lasers with resonance-enhanced quantumwell photodetectors. *Elec. Lett.*, (1205), 1996.

[19] Si-Hyun Park, Yeonsang Park, Hyejin Kim, Heonsu Jeon, Seong Mo

Hwang, Jeong Kwan Lee, Seung Ho Nam, Byeong Cheon Koh, J. Y. Sohn, and D. S. Kim. Microlensed vertical-cavity surface-emitting laser for stable single fundamental mode operation. *Applied Physics Letters*, 80(2):183–185, 2002.

[20] Ki Soo Chang, Young Min Song, and Yong Tak Lee. Self-aligned microlens-integrated vertical-cavity surface-emitting lasers. *Photonics Technology Letters, IEEE*, 18(21):2203–2205, Nov.1, 2006.

[21] E.M. Strzelecka, G.D. Robinson, M.G. Peters, F.H. Peters, and L.A. Coldren. Monolithic integration of vertical-cavity laser diodes with refractive gaas microlenses. *Electronics Letters*, 31(9):724–725, Apr 1995.

[22] D.A. Louderback, O. Sjolund, E.R. Hegblom, S. Nakagawa, J. Ko, and L.A. Coldren. Modulation and free-space link characteristics of monolithically integrated vertical-cavity lasers and photodetectors with microlenses. *Selected Topics in Quantum Electronics, IEEE Journal of*, 5(2):157–165, Mar/Apr 1999.

[23] S. Y. Chou, P. R. Krauss, W. Zhang, L. Guo, and L. Zhuang. Sub-10 nm imprint lithography and applications. *J. Vac. Sci. Technol. B.*, 15:2897–2904, 1997.

[24] M. D. Austin, H. Ge, W. Wu, M. Li, Z. Yu, D. Wasserman, S. A. Lyon, and S. Y. Chou. Fabrication of 5 nm linewidth and 14 nm pitch features by nanoimprint lithography. *Appl. Phys. Lett.*, 84:5299–5301, 2004.

[25] T.J. Suleski and R.D.T. Kolste. Fabrication trends for free-space microoptics. *Lightwave Technology, Journal of*, 23(2):633–646, Feb. 2005.

[26] P.F. McManamon, T.A. Dorschner, D.L. Corkum, L.J. Friedman, D.S. Hobbs, M. Holz, S. Liberman, H.Q. Nguyen, D.P. Resler, R.C. Sharp, and E.A. Watson. Optical phased array technology. *Proceedings of the IEEE*, 84(2):268–298, Feb 1996.

[27] D. E. Culler and J. P. Singh. *Parallel Computer Architecture: a Hardware/Software Approach*. Morgan Kaufmann, 1999.

[28] L. Roberts. ALOHA Packet System With and Without Slots and Capture. *ACM SIGCOMM Computer Communication Review*, 5(2):28–42, April 1975.

[29] R. M. Metcalfe and D. R. Boggs. Ethernet: Distributed Packet Switching for Local Computer Networks. *Communications of the ACM*, 26(1):90–95, January 1983.

[30] K. Yeager. The MIPS R10000 Superscalar Microprocessor. *IEEE Micro*, 16(2):28–40, April 1996.

[31] Compaq Computer Corporation. *Alpha 21264/EV6 Microprocessor Hardware Reference Manual*, September 2000. Order number: DS-0027B-TE.

[32] D. Burger and T. Austin. The SimpleScalar Tool Set, Version 2.0. Technical report 1342, Computer Sciences Department, University of Wisconsin-Madison, June 1997.

[33] PoPNet. `http://www.princeton.edu/~lshang/popnet.html`.

[34] S. Woo, M. Ohara, E. Torrie, J. Singh, and A. Gupta. The SPLASH-2 Programs: Characterization and Methodological Considerations. In *Proc. Int'l Symp. on Comp. Arch.*, pages 24–36, June 1995.

[35] S. Palacharla and R. Kessler. Evaluating Stream Buffers as a Secondary Cache Replacement. In *Proc. Int'l Symp. on Comp. Arch.*, pages 24–33, April 1994.

[36] I. Ganusov and M. Burtscher. On the Importance of Optimizing the Configuration of Stream Prefetchers. In *Proceedings of the 2005 Workshop on Memory System Performance*, pages 54–61, June 2005.

[37] D. Brooks, V. Tiwari, and M. Martonosi. Wattch: A Framework for Architectural-Level Power Analysis and Optimizations. In *Proc. Int'l Symp. on Comp. Arch.*, pages 83–94, June 2000.

[38] BSIM Design Group, `http://www-device.eecs.berkeley.edu/~bsim3/ftv322/Mod_doc/V322manu.tar.%Z`. *BSIM3v3.2.2 MOSFET Model - User's Manual*, April 1999.

[39] K. Skadron, M. Stan, W. Huang, S. Velusamy, and K. Sankaranarayanan. Temperature-Aware Microarchitecture. In *Proc. Int'l Symp. on Comp. Arch.*, pages 2–13, June 2003.

[40] H. Wang, X. Zhu, L. S. Peh, and S. Malik. Orion: A Power-Performance Simulator for Interconnection Networks. In *Proc. Int'l Symp. on Microarch.*, pages 294–305, November 2002.

[41] C. Bienia, S. Kumar, J. P. Singh, and K. Li. The PARSEC Benchmark Suite: Characterization and Architectural Implications. In *Proc. Int'l Conf. on Parallel Arch. and Compilation Techniques*, September 2008.

[42] D. Culler, A. Dusseau, S. Goldstein, A. Krishnamurthy, S. Lumetta, T. Eicken, and K. Yelick. Parallel Programming in Split-C. In *Proc. Supercomputing*, November 1993.

[43] S. Dwarkadas, A. Schaffer, R. Cottingham, A. Cox, P. Keleher, and W. Zwaenepoel. Parallelization of General Linkage Analysis Problems. *Human Heredity*, 44:127–141, 1994.

[44] S. Mukherjee, P. Bannon, S. Lang, A. Spink, and D Webb. The ALpha 21364 Network Architecture. *IEEE Micro*, 22(1):26–35, January/February 2002.

[45] A. Shacham, K. Bergman, and L. Carloni. On the Design of a Photonic Network-on-Chip. In *First Proc. Int'l Symp. on Networks-on-Chip*, pages 53–64, May 2007.

[46] A. Krishnamoorthy and D. Miller. Firehose Architectures for Free-Space Optically Interconnected VLSI Circuits. *Journal of Parallel and Distributed Computing*, 41:109–114, 1997.

[47] P. Marchand et al. Optically Augmented 3-D Computer: System Technology and Architecture. *Journal of Parallel and Distributed Computing*, 41:20–35, 1997.

[48] A. C. Walker, T. Y. Yang, J. Gourlay, J. A. B. Dines, M. G. Forbes, S. M. Prince, D. A. Baillie, D. T. Neilson, R. Williams, L. C. Wikinson, G. R. Smith, M. P. Y. Desulliez, G. S. Buller, M. R. Taghizadeh, A. Waddie, I. Underwood, C. R. Stanley, F. Pottier, B. Vogele, and W. Sibbett. Optoelectronic Systems Based on InGaAs Complementary-Metal-Oxide-Semiconductor Smart-Pixel Arrays and Free-Space Optical Interconnects. *Applied Optics*, 37(14):2822–2830, May 1998.

[49] R. Beausoleil et al. Nanoelectronic and Nanophotonic Interconnect. *Proceedings of the IEEE*, February 2008.

[50] J. Ha and T. Pinkston. SPEED DMON: Cache Coherence on an Optical Multichannel Interconnect Architecture. *Journal of Parallel and Distributed Computing*, 41:78–91, 1997.

[51] N. Kirman, M. Kirman, R. Dokania, J. Martinez, A. Apsel, M. Watkins, and D. Albonesi. Leveraging Optical Technology in Future Bus-based Chip Multiprocessors. In *Proc. Int'l Symp. on Microarch.*, pages 492–503, December 2006.

[52] D. Vantrease et al. Corona: System Implications of Emerging Nanophotonic Technology. In *Proc. Int'l Symp. on Comp. Arch.*, June 2008.

[53] E. A. De Souza, M. C. Nuss, W. H. Knox, and D. A. B. Miller. Wavelength-division multiplexing with femtosecond pulses. *Opt. Lett.*, 20(10):1166, 1995.

[54] B.E. Nelson, G.A. Keeler, D. Agarwal, N.C. Helman, and D.A.B. Miller. Wavelength division multiplexed optical interconnect using short pulses. *Selected Topics in Quantum Electronics, IEEE Journal of*, 9(2):486–491, March-April 2003.

[55] C. Gunn. Cmos photonics for high-speed interconnects. *Micro, IEEE*, 26(2):58–66, March-April 2006.

[56] B.E. Little, S.T. Chu, H.A. Haus, J. Foresi, and J.-P. Laine. Microring resonator channel dropping filters. *Lightwave Technology, Journal of*, 15(6):998–1005, Jun 1997.

[57] Qianfan Xu, Brad Schmidt, Jagat Shakya, and Michal Lipson. Cascaded silicon micro-ring modulators for wdm optical interconnection. *Opt. Express*, 14(20):9431–9435, 2006.

[58] Miloš A. Popovíc, Tymon Barwicz, Michael R. Watts, Peter T. Rakich, Luciano Socci, Erich P. Ippen, Franz X. Kärtner, and Henry I. Smith. Multistage high-order microring-resonator add-drop filters. *Opt. Lett.*, 31(17):2571–2573, 2006.

[59] T. Barwicz, M.A. Popovic, M.R. Watts, P.T. Rakich, E.P. Ippen, and H.I. Smith. Fabrication of add-drop filters based on frequency-matched microring resonators. *Lightwave Technology, Journal of*, 24(5):2207–2218, May 2006.

[60] Shijun Xiao, Maroof H. Khan, Hao Shen, and Minghao Qi. Multiple-channel silicon micro-resonator based filters for wdm applications. *Opt. Express*, 15(12):7489–7498, 2007.

[61] Shijun Xiao, Maroof H. Khan, Hao Shen, and Minghao Qi. A highly compact third-order silicon microring add-drop filter with a very large free spectral range, a flat passband and a low delay dispersion. *Opt. Express*, 15(22):14765–14771, 2007.