A Class of Highly Scalable Optical Crossbar-Connected Interconnection Networks (SOCNs) for Parallel Computing Systems

Brian Webb, *Member*, *IEEE*, and Ahmed Louri, *Senior Member*, *IEEE*

Abstract—A class of highly scalable interconnect topologies called the Scalable Optical Crossbar-Connected Interconnection Networks (SOCNs) is proposed. This proposed class of networks combines the use of tunable Vertical Cavity Surface Emitting Lasers (VCSEL's), Wavelength Division Multiplexing (WDM) and a scalable, hierarchical network architecture to implement large-scale optical crossbar based networks. A free-space and optical waveguide-based crossbar interconnect utilizing tunable VCSEL arrays is proposed for interconnecting processor elements within a local cluster. A similar WDM optical crossbar using optical fibers is proposed for implementing intercluster crossbar links. The combination of the two technologies produces large-scale optical fan-out switches that could be used to implement relatively low cost, large scale, high bandwidth, low latency, fully connected crossbar clusters supporting up to hundreds of processors. An extension of the crossbar network architecture is also proposed that implements a hybrid network architecture that is much more scalable. This could be used to connect thousands of processors in a multiprocessor configuration while maintaining a low latency and high bandwidth. Such an architecture could be very suitable for constructing relatively inexpensive, highly scalable, high bandwidth, and fault-tolerant interconnects for large-scale, massively parallel computer systems. This paper presents a thorough analysis of two example topologies, including a comparison of the two topologies to other popular networks. In addition, an overview of a proposed optical implementation and power budget is presented, along with analysis of proposed media access control protocols and corresponding optical implementation.

Index Terms—Optical interconnections, wavelength division multiplexing, parallel architectures, networks, multiprocessor interconnection, crossbars, hypercubes, scalability.

1 Introduction

In order to deliver high speed parallel computer systems at a reasonable cost, parallel computer manufactures are designing modern high speed parallel computer systems around state-of-the-art high speed commercial-off-the-shelf processors. Future high performance parallel computers will utilize commercial-off-the-shelf processors that will require many Gigabytes/second (GBs/s) and low latency connections to their local memories. In order to make such a system scalable for a wide range of applications, the interconnection network must support communications between remote processors at bandwidths similar to the bandwidth to local memories [1], [2]. In addition, the latency for communications to remote processors must be similar to the latency to local memories. This implies a requirement in future high speed parallel computer systems for low latency interprocessor communications that support bandwidths in the range of GBs/s.

To this end, we propose a class of all-optical, highly scalable, hierarchical [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15] network architectures, referred to as the Scalable Optical Crossbar-Connected Interconnection Networks (SOCNs). A SOCN class network is a two-level hierarchical network that is based on a fully connected wavelength-division multiplexed (WDM) crossbar interconnect. The building blocks of a SOCN class network are multiprocessor clusters. Each cluster contains some number of processors (n) that are connected via an intracluster wavelength division multiplexed crossbar interconnect. These clusters are then connected together via similar intercluster WDM crossbar interconnects to form larger multicluster configurations. The SOCN class of networks combines a two-level hierarchical network with wavelength-division multiplexing to implement a highly scalable class of networks. Two example SOCN class network topologies are presented along with a design for an all-optical, high bandwidth, low latency WDM optical crossbar. A media access control (MAC) protocol is also presented for mediating access to the shared WDM optical crossbar interconnects.

2 Scalable Optical Crossbar-Connected Interconnection Networks (SOCNs)

The objective of this paper is to present a design for a class of network architectures called the Scalable Optical Crossbar Networks (SOCNs). The SOCN class of network architectures is designed to provide a multiprocessor interconnection network that is sufficiently scalable in bandwidth, latency, and cost to support parallel computer

- B. Webb is with Science Applications International Corporation, 101 N. Wilmot Road, Ste. 400, Tucson, AZ 85711.
E-mail: webbb@aries.tucson.saic.com.
- A. Louri is with the Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ 85721. E-mail: louri@ece.arizona.edu.

Manuscript received 23 June 1999; revised 12 Oct. 1999; accepted 21 Dec. 1999.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number 110113.

1045-9219/00/$10.00 © 2000 IEEE

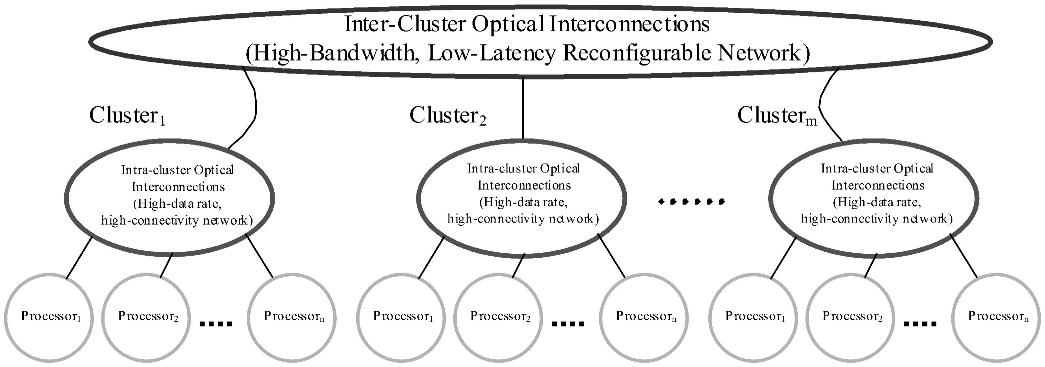
Inter-Cluster Optical Interconnections
(High-Bandwidth, Low-Latency Reconfigurable Network)
Cluster$_1$
Cluster$_2$
Cluster$_m$
Intra-cluster Optical Interconnections (High-data rate, high-connectivity network)
Intra-cluster Optical Interconnections (High-data rate, high-connectivity network)
Intra-cluster Optical Interconnections (High-data rate, high-connectivity network)
Processor$_1$
Processor$_2$
Processor$_n$
Processor$_1$
Processor$_2$
Processor$_n$
Processor$_1$
Processor$_2$
Processor$_n$

Fig. 1. The SOCN cluster-hierarchical optical interconnect model for scalable parallel computers.

systems containing from a few processors to thousands of processors. In order to scale a multiprocessor computer system to such large system sizes, the bandwidth of the system should increase linearly or near linearly with system size, the latency should remain nearly constant, and the cost of the system should increase proportional to the number of processors.

It is proving somewhat difficult for standard (flat) interconnection network topologies to scale to large numbers of processors while satisfying these cost and performance constraints, so in recent years, there has been a strong interest in clustered hierarchical networks [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15]. Hierarchical networks take a modular approach to network design, where local clusters of processors are connected using an intracluster network. These clusters of processors are then connected together via one or more layers of intercluster networks. Although multiple levels of hierarchy are possible in a hierarchical network, it has been observed that two levels provide a very practical choice.

An advantage of hierarchical networks is that they provide a modular network topology that is more amenable to scalability than traditional "flat" network designs. Another advantage is that each layer can be constructed with technology that is most appropriate to the requirements of that level. Also, recent studies of the communication profiles of many parallel processing applications have shown that processors engage in data transfers more frequently with nearby neighbors than with distant ones (spatial and temporal locality of reference) [16], [17], [18], [11]. Hierarchical network topologies inherently support this property by grouping nearby processors into the same cluster.

The SOCN architecture is based on a clustered-hierarchical optical network model as shown in Fig. 1. An SOCN system is a clustered-hierarchical architecture with two levels. The first (intracluster) level consists of clusters of a relatively small number of processors. The relatively small number of processors in a cluster and the relatively close spacing between processors lends itself well to a highly connected configuration, so the intracluster network level in the SOCN consists of a low latency, high bandwidth wavelength-division multiplexed all-optical crossbar interconnect that connects each processor in a cluster to every other processor in the same cluster. This local optical crossbar provides the high connectivity required to support algorithms that contain a high degree of locality of reference. Moreover, the local optical crossbar is highly amenable to low cost, high bandwidth free-space optical implementation (more on the optical implementation is given in the implementation section). Each crossbar-connected cluster is modular in nature and forms the basic building block of the larger hierarchical network.

The second level in the hierarchical network is the intercluster interconnects. In order to produce a highly scalable intercluster interconnect with a bandwidth and latency similar to the bandwidth and latency of the intracluster interconnects, a similar optical crossbar interconnect is proposed for connecting clusters. Because the intracluster optical crossbar utilizes wavelength division multiplexing (WDM), a similar intercluster optical crossbar can be constructed that extends to remote clusters using a single optical fiber send/receive pair. In this fashion, all processors in one cluster can be directly connected to all processors in another cluster via a WDM optical crossbar connection over a single intercluster optical fiber pair.

A SOCN class network topology is constructed by connecting multiple (c) crossbar-connected clusters together via multiple intercluster WDM optical crossbar interconnects. The exact topology of these intercluster interconnects (i.e., which cluster is connected to which other cluster, and how the connections are established) defines the second level of the hierarchical network, and determines the overall performance and scalability of the system. If each cluster in a network is connected to every other cluster via a single WDM fiber based optical crossbar, a large-scale fully connected crossbar network can be created that provides a direct connection from every processor in the network to every other processor in the network. Such a network is completely uniform, in that each processor has direct access to every other processor in the network with the same type of low latency and high bandwidth all-optical connection, irrespective of where the processors are in the network. The latency and bandwidth between processors in the farthest separated remote clusters are identical to the latency and bandwidth between processors in the same cluster, creating a low latency, high bandwidth, fully-connected, all-optical network of potentially hundreds of processors.

If a fully connected network is not required, a less fully connected network can be constructed using fewer intercluster connections by interconnecting the clusters using a different second-level intercluster connection scheme. This flexibility to rearrange the intercluster connection pattern to match the requirements of the target system is the primary benefit of the two-level hierarchical network model.

The SOCN class of networks exploits the excellent communication properties of optics [19], [20], [21], [22], [23] to develop interconnection topologies that span both medium-scale and large-scale high performance parallel computers. SOCN architectures can be implemented that can support high network data rates as well as low latency communication that are essential for efficient coordination and data movement on large-scale parallel systems. To this end, two intercluster topologies are presented below. These topologies could potentially provide a medium-scale, fully connected parallel system, and a large-scale highly connected parallel system, respectively.

3 Interconnection Structure and Properties of the Optical Crossbar-Connected Cluster Network (OC^3N)

In the first proposed topology, which we will refer to as the Optical Crossbar-Connected Cluster Network (OC^3N), the intercluster crossbar connection scheme extends the crossbar connection paradigm and directly connects each processor in a cluster to every other processor in a remote cluster. The unique nature of optics makes this connection scheme possible and cost effective. Wavelength division multiplexing is utilized to produce local crossbar connections that connect the processors within a cluster. These local crossbars can be extended over a single optical fiber using wavelength division multiplexing over the fiber to provide crossbar interconnects to processors in remote clusters.

The local crossbar interconnect is created by placing a single tunable optical source (i.e., tunable VCSEL or multiwavelength array of VCSEL's, etc.) and a single optical receiver at each processor. Each processor **p** is assigned a single wavelength channel that it receives on λ_p. The optical signals from all the processors in a cluster are routed to a free-space optical crossbar that routes the signals from each processor to the appropriate destination processor depending on the wavelength of each of the signals. For example, if processor **p** wishes to transmit to processor **3**, it simply transmits on wavelength λ_3. The free-space optical crossbar will route the signal from processor **p** to processor **3** based on the wavelength of the signal λ_3.

It would seem that by simply adding additional wavelengths to a such a wavelength division multiplexed system, one could build a network of any size. Ultimately, though, such a system is limited to some fixed maximum number of wavelengths due to the fixed tuning range of the tunable optical source and/or the various optical parameters of the system. Wavelength reuse [24], [25] can be used to support larger system sizes by adding additional tunable optical sources at each processor. If the signals from each of these additional optical sources are kept optically isolated from each other, then the number of optical channels is multiplied by the number of tunable optical sources at each processor. This ability to reuse wavelengths extends the scalability of the system far beyond the number of available wavelengths.

In a SOCN class network, a single tunable optical source at each processor is used to provide a crossbar interconnect between processors on the same cluster. Additional tunable optical sources are used to provide crossbar interconnects between clusters, which implements the second layer of the two layer hierarchical network. The signal from each corresponding tunable optical source at each processor is multiplexed onto a single optical fiber that connects to the remote cluster. At the remote cluster, the signal is demultiplexed to the appropriate destination processor using a free-space demultiplexer similar to the free-space optical crossbar used by local intracluster interconnects. Using this WDM scheme, each processor in the local cluster is directly connected to every other processor in a remote cluster, providing a completely connected crossbar interconnect from all the processors in one cluster to all the processors in another cluster. If each cluster in the network is connected to every other cluster via a single optical fiber pair (send and receive) in such a fashion, a fully connected crossbar interconnect is created for the entire network.

As an example, Fig. 2 shows an $OC^3N(n = 4, c = 4)$ fully connected crossbar network consisting of $N = n \times c = 16$ processors. The figure depicts a system with $c = 4$ clusters, and each cluster contains $n = 4$ processors. In general, if there are n processors contained in each cluster and there are c clusters in the network then there are $N = n \times c$ processors in the system. In this configuration, each cluster is connected to every other cluster via a crossbar interconnect that directly connects each processor in each cluster to every processor in every other cluster. Since each processor is directly connected to every other processor in the network then the diameter of the network is identically one, which provides the lowest possible latency for interprocessor communications between any two processors in the network. In addition, the bisection width of the OC^3N is the same as a crossbar containing $N = n \times c$ processors, which is:

$$B_C = N^2/4 = (n \times c)^2/4, \tag{1}$$

so the bisection width increases as the square of the number of processors in the network ($O(N^2)$), implying very high interprocessor bandwidths for any network size.

One disadvantage of crossbar networks is that they typically require a very high node degree, which limits their scalability. Since each processor in an OC^3N is directly connected to every other processor in the network, the degree of each processor seems to be $D = N - 1$, however, each WDM link carries n distinct physical channels, which allows up to n simultaneous data transfers, so the actual physical degree of each processor is:

$$D_C = c = \frac{N}{n}. \tag{2}$$

Equation (2) highlights the scalability benefits of wavelength reuse. Without wavelength reuse, N wavelengths

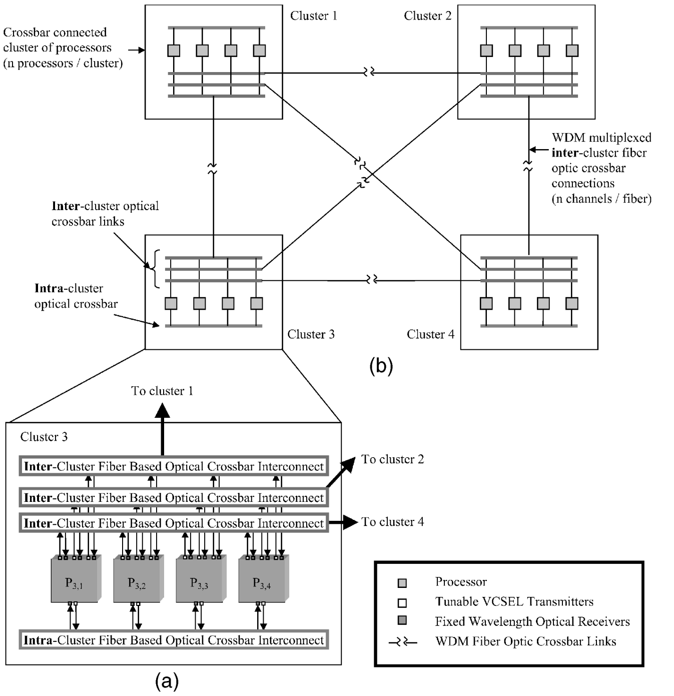
Cluster 1
Cluster 2
Crossbar connected cluster of processors (n processors / cluster)
WDM multiplexed inter-cluster fiber optic crossbar connections (n channels / fiber)
Inter-cluster optical crossbar links
Intra-cluster optical crossbar
Cluster 3
Cluster 4
(b)
To cluster 1
Cluster 3
Inter-Cluster Fiber Based Optical Crossbar Interconnect
Inter-Cluster Fiber Based Optical Crossbar Interconnect
Inter-Cluster Fiber Based Optical Crossbar Interconnect
To cluster 2
To cluster 4
P3,1
P3,2
P3,3
P3,4
Intra-Cluster Fiber Based Optical Crossbar Interconnect
Processor
Tunable VCSEL Transmitters
Fixed Wavelength Optical Receivers
WDM Fiber Optic Crossbar Links
(a)

Fig. 2. A structural organization of the Optical Crossbar-Connected Cluster Network (OC^3N). A 16 processor $(n = 4, c = 4)$ fully connected network is shown. (a) Every processor is connected to all the processors in the same cluster via a local WDM optical crossbar. (b) Every processor is also connected to every processor in all other clusters via a WDM fiber-based optical crossbar.

would be required to support a network of N processors. Wavelength reuse divides the number of wavelengths required by the number of optical channels per processor. For a system containing N processors, with each processor having a physical degree of D_C, the number of wavelengths required on each interconnect is $n = \frac{N}{D_C}$.

We will assume that a processor can only receive from a single transmitter at any given time, so there will never be a situation where two processors on a cluster are communicating on the same wavelength to the same cluster. This implies that a single shared medium can be used to connect pairs of clusters. Therefore, the degree of each cluster (i.e., the number of physical links connected to a cluster) is $c - 1$, and a fully connected network of $N = n \times c$ processors can be constructed using only:

$$L_C = c(c-1)/2 = (N^2/n^2 - N/n)/2 \tag{3}$$

intercluster connections. A traditional crossbar, on the other hand, would require $(N^2 - N)/2$ connections.

As an example, if an OC^3N based system is constructed with $n = 16$ processors per cluster and $(c - 1) = 15$ WDM links per cluster, then a system containing $N = n \times c = 256$ processors could be constructed. Each processor is directly connected to every other processor, so there is no network imposed interprocessor latency, and each processor has a full bandwidth connection to every other processor. The total number of intercluster links required is $c(c-1)/2 = 120$. Notice that if one uses a traditional crossbar network to provide full connectivity among 256 processors, the total number of required links would be $(N^2 - N)/2 = 32,640$. This numerical example shows that an OC^3N architecture can be used to connect a reasonably large number of processors in a fully connected crossbar network with relatively low interconnection complexity and part count. Additionally, an OC^3N can be scaled up in size by either increasing the cluster node degree c, changing the number of processors per cluster n, or by changing the intercluster network topology.

3.1 Granularity of Size Scaling of the OC^3N

An important factor for scalability of a network architecture is the granularity of size scaling, which is a measure of the increments in the number of processors that are required to scale a network of N processors to a larger size network. Two extreme examples include a bus or ring-based network and a hypercube network. A bus or ring based network can be increased in size by simply adding a single processor. The hypercube network, on the other hand, requires a doubling of the size of the network to maintain a hypercube topology.

For an OC^3N fully connected crossbar network containing a fixed number of processors per cluster n, we can increase the network size by adding another cluster to the network $(c_2 = c_1 + 1)$. This increases the size of the network by n, but requires adding another intercluster link to each cluster in the network, increasing the intercluster node degree by one. In this case, which we will refer to as the fixed-n case, the granularity of size scaling is the cluster size n.

$$N_{2,c} = n \times c_2 = n \times (c_1 + 1) = N_1 + n. \tag{4}$$

If we instead fix the number of clusters c, then we can increase the network size by adding another processor to each cluster in the network $(n_2 = n_1 + 1)$. This increases the size of the network by c and does not affect the cluster node degree of the network.

$$N_{2,n} = n_2 \times c = (n_1 + 1) \times c = N_1 + c. \tag{5}$$

The additional processors utilize an additional frequency over the existing fiber connections, so this is the easiest method to increase the size of the system. Increasing the number of processors per cluster increases the system size without effecting the interconnect hardware. In this case, which we will refer to as the fixed-c case, the granularity of size scaling is the number of clusters c. In addition, for the fixed-c case, the bandwidth of the network increases linearly with the addition of processors because the additional processors are more fully utilizing the high bandwidth inherent in wavelength division multiplexing. Therefore, in this case, the SOCN architecture is readily scalable. The primary limitation is the number of unique frequencies supported by the optical hardware. Progress in this area is moving rapidly and various tuning ranges and tuning times are already available [26].

As a third alternative, the size of the network can also be increased by leaving n and c fixed and changing the topology of the intercluster network. This produces a configuration that is a hybrid of crossbar connections interconnected via a higher level interconnection pattern. An example of such, a network is presented in the next section.

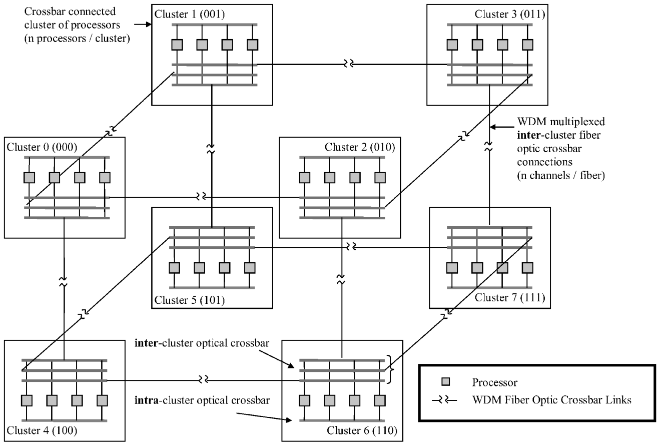
Crossbar connected cluster of processors (n processors / cluster)
Cluster 1 (001)
Cluster 3 (011)
WDM multiplexed inter-cluster fiber optic crossbar connections (n channels / fiber)
Cluster 0 (000)
Cluster 2 (010)
Cluster 5 (101)
Cluster 7 (111)
inter-cluster optical crossbar
intra-cluster optical crossbar
Cluster 4 (100)
Cluster 6 (110)
Processor
WDM Fiber Optic Crossbar Links

Fig. 3. A Structural organization of the Optical Hypercube-Connected Cluster Network (OC^2N) containing 32 processors $(n = 4, d = 3)$. Processors within the same cluster are connected via a free-space optical crossbar. Processors in different clusters are connected via fiber-based optical crossbar channels. Messages are routed via a hypercube-like topology for nonneighboring clusters. Messages are routed directly to processors in neighboring clusters using optical crossbar channels.

4 Interconnection Structure and Properties of the Optical Hypercube-Connected Cluster Network (OHC^2N)

One of the benefits of the hierarchical model is the ability to change one topological layer while not affecting the other layer(s). In a SOCN class network, if one does not require full connectivity, the intercluster links can be configured into any point-to-point topology to fit the requirements of the system. In this context, a SOCN class network can also be thought of as a hybrid network.

For example, if the intercluster crossbar links are connected in such a way as to produce a hypercube configuration, where the clusters are the nodes in the hypercube, and the intercluster links are the edges in the hypercube, then the network can be thought of as a hybrid of a hypercube network and a crossbar network. If the crossbar interconnects are ignored and each cluster is thought of as a traditional network, then the number of WDM links (m) can be thought of as the degree of the network. Therefore, any point-to-point network that can be constructed with degree $\leq m$ can be configured into a SOCN class network.

As an example, Fig. 3 depicts a d-degree binary hypercube topology overlayed over a SOCN class network, which is referred to as the Optical Hypercube-Connected Cluster Network OHC^2N. A d-degree binary hypercube contains 2^d nodes (clusters). Therefore, an OHC^2N with n processors per cluster contains:

$$N_H = n \times 2^d \tag{6}$$

total processors.

4.1 Node Degree / Link Complexity

The link complexity or node degree in a network is defined as the number of links at each node in the network. The node degree at each processor in an OHC^2N network is the same as the degree of the hypercube subnetwork (d) plus one for the local intracluster connection, so the node degree is:

$$D_H = d + 1. \tag{7}$$

The hierarchical nature of a SOCN network adds another measure of the network complexity: cluster degree. The cluster degree is the number of physical links connected to each cluster in the network. Since each physical link connected to each cluster multiplexes n channels of data, the cluster degree of an OHC^2N network is the same as the degree of the hypercube network (d). Therefore, the total number of links in an OHC^2N is:

$$L_H = \frac{1}{2}2^d d = \frac{N}{2n} log_2\left(\frac{N}{n}\right), \tag{8}$$

as opposed to the $L = \frac{N}{2} log_2(N)$ links required for a standard binary hypercube. This multiplexing greatly reduces the link complexity of the entire network, reducing implementation costs proportionately.

4.2 Network Diameter

The diameter of a network is defined as the minimum distance between the two most distant processors in the network. Since each processor in an OHC^2N cluster can communicate directly with every processor in each directly connected cluster, the diameter of a OHC^2N containing $N_H = n \times 2^d$ processors is:

$$K_H = d = log_2\left(\frac{N}{n}\right), \tag{9}$$

which is dependent only on the degree of the hypercube (the diameter and the degree of a hypercube network are the same).

4.3 Bisection Width

The bisection width of a network is defined as the minimum number of links in the network that must be broken to partition the network into two equal sized halves. The bisection width of a d-dimensional binary hypercube is 2^{d-1}, since that many links are connected between two $(d-1)$-dimensional hypercubes to form a d-dimensional hypercube. Since each link in an OHC^2N contains n channels, the bisection width of the OHC^2N is:

$$B_H = n2^{d-1} = N/2, \tag{10}$$

which increases linearly with the number of processors.

A major benefit of such a topology is that a very large number of processors can be connected with a relatively small diameter and relatively fewer intercluster connections. For example, with $n = 16$ processors per cluster and $L = 6$ fiber links per cluster, 1,024 processors can be connected with a high degree of connectivity and a high bandwidth. The diameter of such a network is 6, which implies a low network latency for such a large system, and only 192 bidirectional intercluster links are required. If a system containing the same number of processors is constructed using a pure binary hypercube topology, it would require a network diameter of 10, and 5,120 interprocessor links.

4.4 Average Message Distance

The average message distance for a network is defined as the average number of links that a message should traverse through the network. This is a slightly better measure of network latency than the diameter, because it aggregates distances over the entire network rather than just looking at the maximum distance. The average message distance $\bar{l}$ can be calculated as [27]:

$$\bar{l} = \frac{1}{N-1}\sum_{i=1}^{K} iN_i, \tag{11}$$

where N_i represents the number of processors at a distance i from the reference processor, N is the total number of processors in the network, and K is the diameter of the network.

Since the OHC^2N is a hybrid of a binary hypercube and a crossbar network, the equation for the number of processors at a given distance in an OHC^2N can be derived from the equation for a binary hypercube:

$$N_{i,BHC} = \binom{K}{i}, \tag{12}$$

and since each cluster in the OHC^2N hypercube topology contains n processors, the number of processors at a distance i for an OHC^2N can be calculated as:

$$N_{i,OHC^2N} = n\binom{K}{i}, \tag{13}$$

with the addition of $(n-1)$ for $i=1$ to account for the processors within the local cluster. Substituting into (11) and computing the summation gives the equation for the average messages distance for the OHC^2N:

$$\bar{l}_{OHC^2N} = \frac{1}{N-1}\left[\frac{KN}{2} + (n-1)\right]. \tag{14}$$

Substituting in the diameter of the OHC^2N produces:

$$\bar{l}_{OHC^2N} = \frac{1}{N-1}\left[\frac{Nlog_2\left(\frac{N}{n}\right)}{2} + (n-1)\right]. \tag{15}$$

4.5 Granularity of Size Scaling of the OHC^2N

For an OHC^2N hypercube connected crossbar network containing a fixed number of processors per cluster n, we can increase the network size by increasing the size of the second level hypercube topology. Since the granularity of size scaling for an c-processor hypercube is c, it would require the addition of c clusters to increase the size of the OHC^2N in the fixed-n case ($c_2 = 2c_1$). Increasing the size of the OHC^2N in the fixed-n case would also require adding another intercluster link to each cluster in the network, increasing the intercluster node degree by one. In this case, the granularity of size scaling is:

$$N_{2,c} = nc_2 = 2nc_1 = 2n2^{d_1} = 2N_1. \tag{16}$$

If we assume, instead, the fixed-c case, then we can increase the network size by adding another processor to each cluster in the network ($n_2 = n_1 + 1$). This would increase the size of the network by c:

$$N_{2,n} = n_2 \times c = (n_1 + 1) \times c = N_1 + c, \tag{17}$$

and would not effect the cluster node degree of the network. This is very similar to the fixed-c case for the OC^2N configuration, and the granularity of size scaling in this case is also the number of clusters c. Again, this is the easiest method for scaling because it does not require the addition of any network hardware, and it more fully utilizes the inherently high bandwidth of the WDM optical links.

The two topologies presented in this paper are by no means the only two topologies that could be utilized to construct an SOCN class network. As an example, assume that an SOCN network exists that is configured in a torus configuration, and the addition of some number of processors is required. The total number of processors required in the final network may be the number supported by an OHC^2N configuration. In this case, the network could be reconfigured into an OHC^2N configuration by simply changing the routing of the intercluster links and changing the routing algorithms. This reconfigurability makes it conceivable to reconfigure an SOCN class network with a relatively arbitrary granularity of size scaling.

4.6 Fault Tolerance and Congestion Avoidance

Since the OHC^2N architecture combines the edges of a hypercube network with the edges a crossbar network, the fault tolerance and congestion avoidance schemes of both architectures can be combined into an even more powerful congestion avoidance scheme. Hypercube routers typically scan the bits of the destination address looking for a difference between the bits of the destination address and the routers address. When a difference is found, the message is routed along that dimension. If there are multiple bits that differ, the router may choose any of those dimensions along which to route the message. The number of redundant links available from a source processor along an optimum path to the destination processor is equal to the Hamming distance between the addresses of the two respective processors. If one of the links are down, or if one of the links is congested due to other traffic being routed through the connecting router, the message can be routed along one of the other dimensions.

In addition, the crossbar network connections between clusters greatly increases the routing choices of the routers. The message only must be transmitted using the wavelength of the destination processor when it is transmitted over the last link in the transmission (the link that is directly connected to the destination processor). A message can be transmitted on any channel over any other link along the routing path. This means that each router along the path of the message traversal not only has a choice of links based on the hypercube routing algorithm, but also a choice of n different channels along each of those links. The router may choose any of the n links that connect the local cluster to the remote cluster. This feature greatly increases the fault tolerance of the network as well as the link load balancing and congestion avoidance properties of the network.

TABLE 1
Comparison of Size, Degree, Diameter, and Number of Links of Several Popular Networks

Network	**Size** (N)	**Degree** (D)	**Diameter** (K)	**Number of Links** (L)
$OC^3N(n,c)$	nc	$\frac{N}{n}$	1	$\frac{N}{2n}\left(\frac{N}{n}-1\right)$
$OHC^2N(n,d)$	$n2^d$	$log_2\frac{N}{n}$	$log_2\frac{N}{n}$	$\frac{N}{2n}log_2\left(\frac{N}{n}\right)$
$CB(N)$	N	$N-1$	1	$\left(N^2-N\right)/2$
$BHC(d)$	2^d	log_2N	log_2N	$\frac{N}{2}log_2N$
$CCC(d)$	$d2^d$	3	$\frac{5d-2}{2}$	$\frac{3}{2}N$
$Torus(w,d)$	w^d	$2log_wN$	$\frac{w}{2}log_wN$	$Nlog_wN$
$SBH(w,d)$	w^d	log_wN	log_wN	$\frac{N}{w}log_wN$
$SMLH(w,d)$	w^22^d	$2+log_2\frac{N}{w^2}$	$2+log_2\frac{N}{w^2}$	$\frac{N}{2}\left(\frac{4}{n}+log_2\frac{N}{w^2}\right)$

n = number of processors per cluster, c = number of clusters, d = dimensionality, w = number of processors per bus/ring/multichannel link, and N = total number of processors.

As an example, if the Hamming distance between the cluster address of the current router and the destination processor cluster address is equal to b, the router will have a choice $b \times n$ different channels with which to choose to route to. Even if all the links along all the routing dimensions for the given message are down or are congested, the message can still be routed around the failure/congestion via other links along nonoptimal paths, as long as the network has not been partitioned. In addition, if nonshortest path routing algorithms are used to further reduce network congestion, many more route choices are made available.

5 Comparison to Other Popular Networks

In this section, we present an analysis of the scalability of the SOCN architecture with respect to several scalability parameters. Bisection width is used as a measure of the bandwidth of the network, and diameter and average message distance are used as measures of the latency of the network. Common measures of the cost or complexity of an interconnection network are the node degree of the network and the number of interconnection links. The node degree and number of links in the network relates to the number of parts required to construct the network. Cost, though, is also determined by the technology, routing algorithms, and communication protocols used to construct the network. Traditionally, optical interconnects have been considered a more costly alternative to electrical interconnects, but recent advances in highly integrated, low power arrays of emitters (e.g., VCSELs and tunable VCSELs) and detectors, inexpensive polymer waveguides, and low cost microoptical components can reduce the cost and increase the scalability of high performance computer networks, and can make higher node degrees possible and also cost effective.

Both the OC^3N and OHC^2N configurations are compared with several well-known network topologies that have been shown to be implementable in optics. These network topologies include: a traditional Crossbar network (CB), the Binary Hypercube (BHC) [27], the Cube Connected Cycles (CCC) [28], the Torus [29], the Spanning Bus Hypercube (SBH) [30], and the Spanning Multichannel Linked Hypercube (SMLH) [25]. Each of these networks will be compared with respect to degree, diameter, number of links, bisection bandwidth, and average message distance. There are tradeoffs between the OC^3N and OHC^2N configuration, and other configurations of a SOCN class network might be considered for various applications, but it will be shown that the OC^3N and OHC^2N provide some distinct advantages for medium sized to very large-scale parallel computing architectures.

Various topological characteristics of the compared networks are shown in Tables 1 and 2. The notation $OC^3N(n=16,c)$ implies that the number of processors per cluster n is fixed and the number of clusters c is changed in order to vary the number of processors N. The notation $OHC^2N(n=16,d)$ implies that the number of processors per cluster n is fixed, and the dimensionality of

TABLE 2
Comparison Bisection Bandwidth and Average Message Distance for Several Popular Networks

Network	**Bisection Bandwidth** (B)	**Average Message Distance** $(\bar{l})$
$OC^3N(n,c)$	$\frac{N^2}{4}$	1
$OHC^2N(n,d)$	$\frac{N}{2}$	$\left(\frac{Nlog_2\frac{N}{n}}{2}+(n-1)\right)\left(\frac{1}{N-1}\right)$
$CB(N)$	$\frac{N^2}{4}$	1
$BHC(d)$	$\frac{N}{2}$	$\frac{log_2N}{2}\left(\frac{N}{N-1}\right)$
$CCC(d)$	$\frac{N}{2n}$	$\frac{7}{4}d-3+\frac{(d+1)}{2^{d-1}}$
$Torus(w,d)$	$2w^{log_wN-1}$	$log_wN\frac{w}{4}\left(\frac{N}{N-1}\right)$
$SBH(w,d)$	$2w^{log_wN-1}$	$\frac{log_wN(w-1)}{2}\left(\frac{N}{N-1}\right)$
$SMLH(w,d)$	N	$\frac{2w(w-1)+d2^{d-1}}{(w^2-1)+(2^d-1)}$

the hypercube d is varied. The number of processors is the only variable for a standard crossbar, so $CB(N)$ implies a crossbar containing N processors. For the binary hypercube, the dimensionality of the hypercube d varies with the size of the network. The notation $CCC(d)$ implies that the number of dimensions of the Cube Connected Cycles d varies. The notation $Torus(w, d = 3)$ implies that the dimensionality d is fixed and the size of the rings n varies with the number of processors. The notation $SBH(w = 3, d)$ implies that the size of the buses in the SBH network w remains constant while the dimensionality d changes. The notation $SMLH(w = 32, d)$ denotes that the number of multichannel links w is kept constant and the dimensionality of the hypercube d is varied.

5.1 Network Degree

Fig. 4 shows a comparison of the node degree of various networks with respect to system size (number of processing elements). It can be seen that for medium size networks containing 128 processors or less, the two examples OC^3N networks provide a respectable cluster degree of 4 for a $OC^3N(n = 16, c)$ configuration, and 8 for a $OC^3N(n = 32, c)$ configuration. This implies that a fully connected crossbar network can be constructed for a system containing 128 processors with a node degree as low as 4. A traditional crossbar would, of course, require a node degree of 127 for the same size system.

The node degrees of the $OHC^2N(n = 16, d)$ and $OHC^2N(n = 32, d)$ configurations are very respectable for much larger system sizes. For a system containing on the order of $10,000$ processor, both the $OHC^2N(n = 16, d)$ and the $OHC^2N(n = 32, d)$ configurations would require a node degree of around 7-8, which is comparable to most of the other networks, and much better than some.

5.2 Network Diameter

Fig. 5 shows a comparison of the diameter of various networks with respect to the system size. The network diameter is a good measure of the maximum latency of the network because it is the length of the shortest path between the two most distant nodes in the network. Of course, the diameter of the OC^3N network is the best because each node is directly connected to every other node, so the diameter of the OC^3N network is identically 1.

As expected, the diameter of the various OHC^2N networks scale the same as the BHC network, with a fixed negative bias due to the number of channels in each crossbar. The $SMLH(w, d)$ networks also scale the same as the BHC network, with a larger fixed bias. For a $10,000$ processor configuration, the various OHC^2N networks are comparable or better than most of the comparison networks, although the $SMLH(w, d)$ networks are better because of their larger inherent fixed bias.

5.3 Number of Network Links

The number of links (along with the degree of the network) is a good measure of the overall cost of implementing the network. Ultimately, each link must translate into some sort of wire(s), waveguide(s), optical fiber(s), or at least some set of optical components (lenses, gratings, etc.). It should be noted that this is a comparison of the number of interprocessor/intercluster links in the network and a link could consist of multiple physical data paths. For example, an electrical interface would likely consist of multiple wires. The proposed optical implementation of a SOCN crossbar consists of an optical fiber pair (send and receive) per intercluster link.

Fig. 6 shows a plot of the number of network links with respect to the number of processors in the system. The OC^3N network compares very well for small to medium sized systems, although the number of links could become prohibitive when the number of processors gets very large. The OHC^2N network configurations show a much better scalability in the number of links for very large-scale systems. For the case of around $10,000$ processors, the $OHC^2N(n = 32, d)$ network shows greater than an order of magnitude less links than any other network architecture.

5.4 Bisection Width

The bisection width of a network is a good measure of the overall bandwidth of the network. The bisection width of a network should scale close to linearly with the number of processors for a scalable network. If the bisection width does not scale well, the interconnection network will become a bottleneck as the number of processors is increased.

Fig. 7 shows a plot of the bisection width of various network architectures with respect to the number of processors in the system. Of course, the OC^3N clearly provides the best bisection width because the number of interprocessor links in an OC^3N increases as a factor of $O(N^2)$ with respect to the number of processors. the OHC^2N configurations are very comparable to the best of the remaining networks, and are much better than some of the less scalable networks.

5.5 Average Message Distance

The average message distance within a network is a good measure of the overall network latency. The average message distance can be a better measure of network latency than the diameter of the network because the average message distance is aggregated over the entire network and provides an average latency rather than the maximum latency.

Fig. 8 shows a plot of the average message distance with respect to the number of processors in the system. Of course, the OC^3N provides the best possible average message distance of 1 because each processor is connected to every other processor. The OHC^2N network configurations displays a good average message distance for medium to very large-scale configurations, which is not as good as the average message distance of the $SMLH$ networks, but is much better than the remaining networks.

6 Optical Implementation of the SOCN

Tunable VCSELs provide a basis for designing compact all-optical crossbars for high speed multiprocessor interconnects. An overview of a compact all-optical crossbar can be seen in Fig. 9. A single tunable VCSEL and a single fixed-frequency optical receiver are integrated onto each processor in the network. This tight coupling between the optical

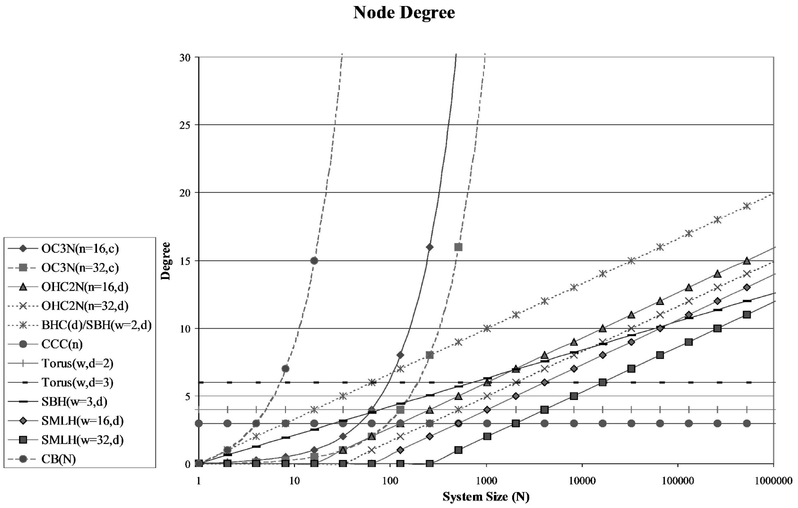

Fig. 4. Comparison of network degree with respect to system size for various networks.

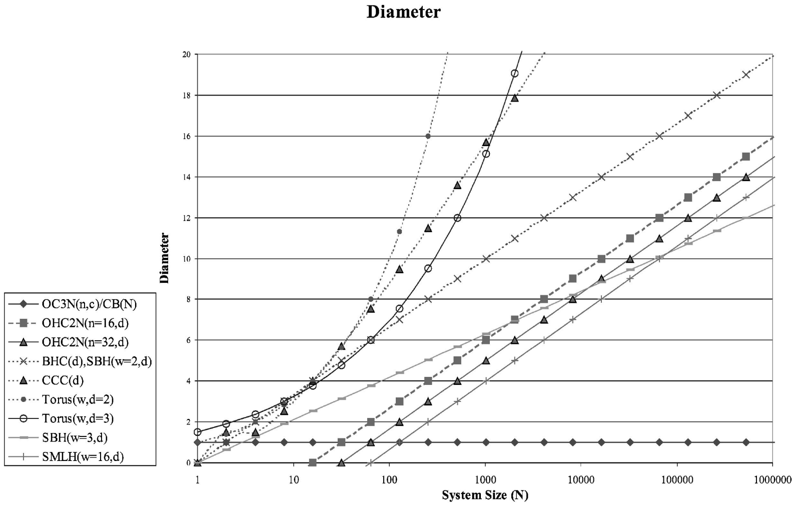

Fig. 5. Comparison of network diameter with respect to system size for various networks.

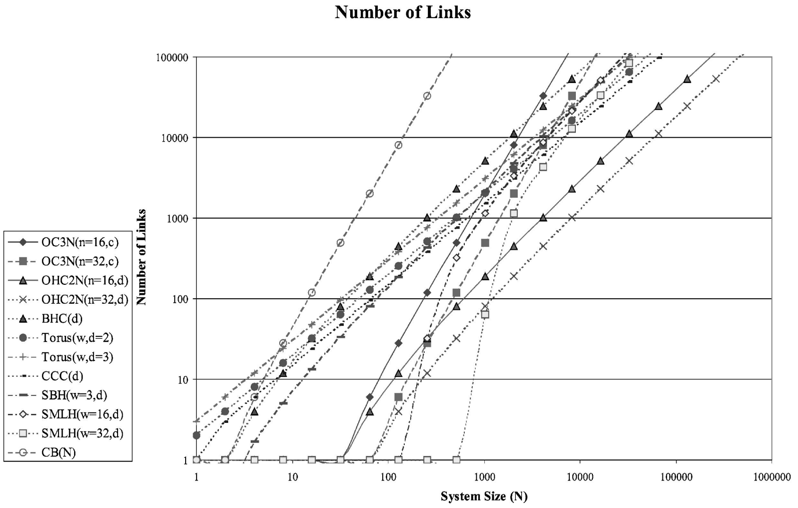

Fig. 6. Comparison of the total number of network interconnection links with respect to system size for various networks.

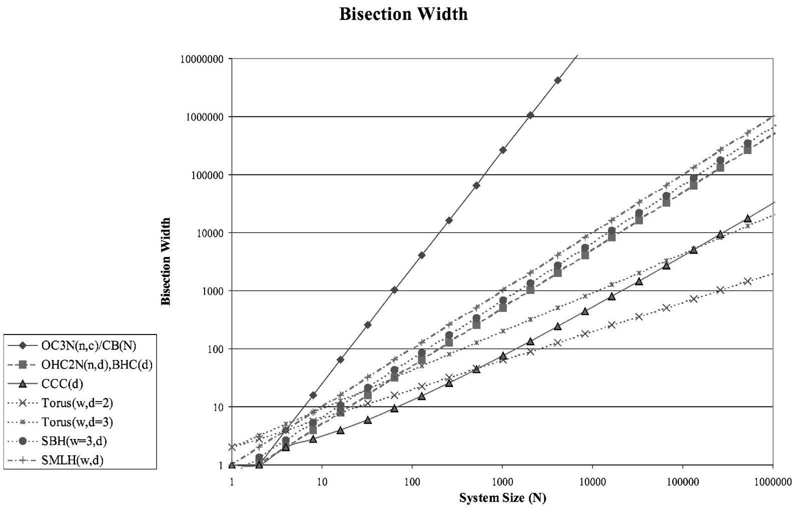

Fig. 7. Comparison of the bisection width with respect to system size for various networks.

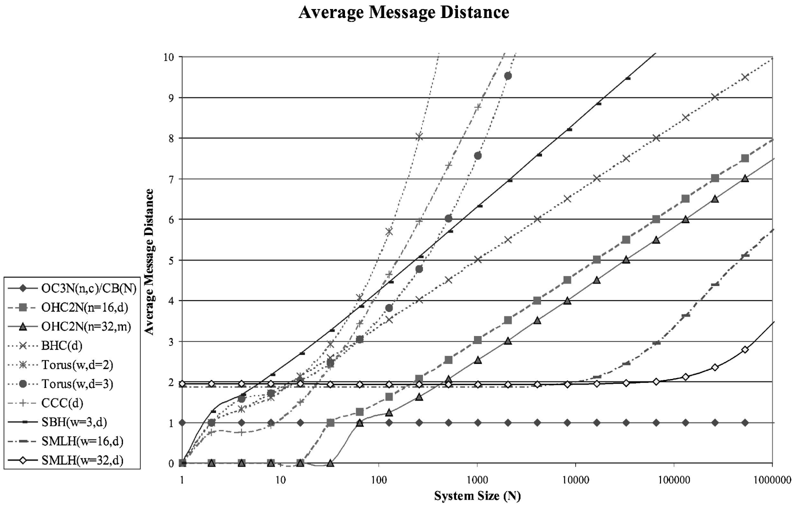

Fig. 8. Comparison of the average message distance within the network with respect to system size for various networks.

transceivers and the processor electronics provides an all-optical path directly from processor to processor, taking full advantage of the bandwidth and latency advantages of optics in the network.

The optical signal from each processor is directly coupled into polymer waveguides that route the signal around the PC board to a waveguide based optical combiner network. Polymer waveguides are used for this design because they provide a potentially low cost, all-optical signal path that can be constructed using relatively standard manufacturing techniques. It has been shown that polymer waveguides can be constructed with relatively small losses and greater than $30dB$ crosstalk isolation with waveguide dimensions on the order of $50\mu m \times 50\mu m$ and with a $60\mu m$ pitch [31], implying that a relatively large-scale crossbar and optical combiner network could be constructed within an area of just a few square centimeters.

The combined optical signal from the optical combiner is routed to a free-space optical demultiplexer/crossbar. Within the optical demultiplexer, passive free-space optics is utilized to direct the beam to the appropriate destination waveguide. As can be seen in the inset in Fig. 9, the beam emitted from the input optical waveguide shines on a concave, reflective diffraction grating that diffracts the beam through a diffraction angle that is dependent on the wavelength of the beam, and focuses the beam on the appropriate destination waveguide. The diffraction angle varies with the wavelength of the beam, so the wavelength of the beam will define which destination waveguide, and hence, which processor receives the transmitted signal. Each processor is assigned a particular wavelength that it will receive based on the location of its waveguide in the output waveguide array. For example, for processor **1** to transmit to processor **3**, processor **1** would simply transmit on the wavelength assigned to processor **3** (e.g., λ_3). If each processor is transmitting on a different wavelength, each signal will be routed simultaneously to the appropriate destination processor. Ensuring that no two processors are transmitting on the same wavelength is a function of the

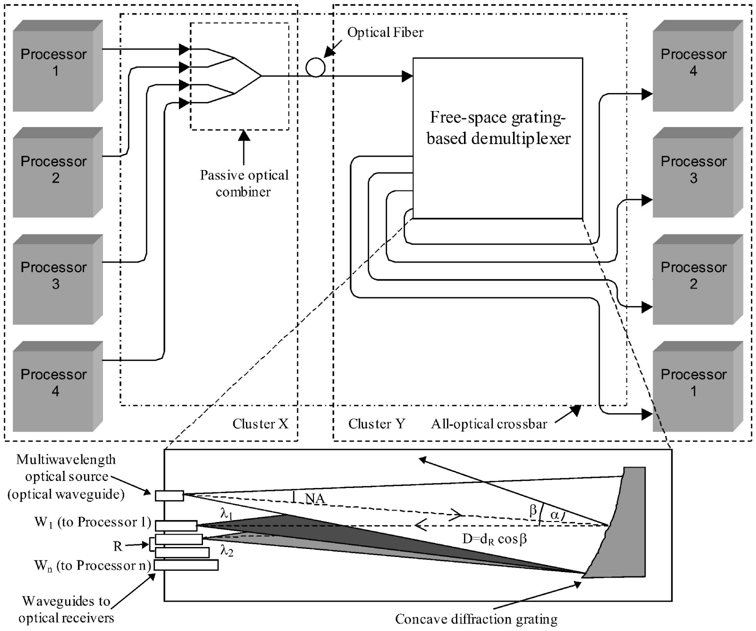

Fig. 9. A proposed compact optical crossbar consisting of polymer waveguides directly coupled to processor mounted VCSELs, a polymer waveguide based optical combiner, and a compact free-space optical crossbar/demultiplexer. The proposed optical crossbar can be connected to remote processors using a single optical fiber or connected locally by eliminating the optical fiber.

media access control (MAC) protocol (detailed in a later section).

After routing through the free-space optical demultiplexer, the separate optical signals are routed to the appropriate destination processor via additional integrated optical waveguides. As can be seen in Fig. 9, the combined optical signal between the optical combiner network and the demultiplexer can be coupled into a single optical fiber to route to a remote PC board to implement an intercluster optical crossbar, or a short length of polymer waveguide could replace the optical fiber to implement a local (intracluster) optical crossbar.

A power budget and signal-to-noise ratio (SNR) analysis have been conducted for the intracluster and intercluster optical crossbars [32], [33], [34]. The result of the power budget analysis is shown in Table 3. Assuming a necessary receiver power of $-30dBm$, a VCSEL power of $2dBm$ [35] and a required bit-error rate (BER) of 10^{-15}, it was determined that with current research level technology, 16 processors could be supported by such a network with 32 processors very nearly possible.

Details of the optical implementation of the SOCN crossbar interconnect and a thorough analysis of the optical implementation can be found in references [32], [33], [34], and [36].

7 Media Access in the SOCN

An SOCN network contains a local intracluster WDM subnetwork and multiple intercluster WDM subnetworks at each processing node. Each of these intracluster and intercluster subnetworks has its own medium that is shared by all processors connected to the subnetwork. Each of these subnetworks are optically isolated, so the media access can be handled independently for each subnetwork. One advantage of an SOCN network is that each subnetwork connects processors in the same cluster to processors in a single remote cluster. The optical media are shared only among processors in the same cluster. This implies that media access control interaction is only required between processors on the same cluster. Processors on different clusters can transmit to the same remote processor at the same time, but they will be transmitting on different media. This could cause conflicts and contention at the receiving processor, but these conflicts are an issue of flow control, which is not in the scope of this paper.

7.1 SOCN MAC Overview

In a SOCN network, the processors cannot directly sense the state of all communication channels that they have access to, so there must be some other method for processors to coordinate access to the shared media. One method of accomplishing this is to have a secondary broadcast control/reservation channel. This is particularly advantageous in a SOCN class network because the coordination need only happen among processors local to the same cluster. This implies that the control channel can be local to the cluster, saving the cost of running more intercluster cabling, and ensuring that it can be constructed with the least latency possible. For control-channel based networks, the latency of the control channel is particularly critical because a channel must be reserved on the control channel before a message is transmitted on the data network, so the latency of the control channel adds directly to the data transfer latency when determining the overall network latency.

Since there are multiple physical channels at each cluster (the local intracluster network and the various intercluster network connections), it is conceivable that each physical data channel could require a dedicated control channel.

TABLE 3
Losses (in dB) for Each Component of the Optical Crossbar

Loss mechanism	*Loss (dB)*
VCSEL-waveguide coupling (L_{vc})	$-1dB$
Waveguide (L_w)	$-6dB$
Y-coupler (L_Y)	$-3dB \times log_2(n)$
Waveguide-to-Fiber Coupling (L_{wf})	$-0.5dB$
Fiber-to-Waveguide Coupling (L_{fw})	$-0.5dB$
Demultiplexing (L_d)	$-9dB$
Receiver coupling(L_{rc})	$-0.5dB$
total	$-17.5dB - 3dB \times log_2(n)$

Fortunately, each physical data channel on a given cluster is shared by the same set of processors, so it is possible to control access to all data channels on a cluster using a single control channel at each cluster. Each WDM channel on each physical channel is treated as a shared channel, and MAC arbitration is controlled globally over the same control channel.

7.2 A Carrier Sense Multiple Access with Collision Detection (CSMA/CD) MAC Protocol

If we assume that a control channel is required, one possible implementation of a MAC protocol would be to allow processors to broadcast channel allocation requests on the control channel prior to transmitting on the data channel. In this case, some protocol would need to be devised to resolve conflicts on the control channel. One candidate might be the Carrier Sense Multiple Access/Collision Detection (CSMA/CD) protocol.

Running CSMA/CD over the control channel to request access to the shared data channels is similar to standard CSMA/CD protocols, such as that used in Ethernet networks, except that Ethernet is a broadcast network, where each node can see everything that is transmit, so the CSMA/CD used within ethernet is run over the data network and a separate control channel is not required. There are some advantages to using CSMA/CD as a media access control protocol. The primary advantage is that the minimum latency for accessing the control channel is zero. The primary disadvantage to using such a protocol for a SOCN based system is that it requires that state information be maintained at each node in the network. Each processing node must monitor the control channel and track which channels have been requested. When a channel is requested, each processor must remember the request so that it will know if the channel is busy when it wishes to transmit. There is also a question about when a data channel becomes available after being requested. A node could be required to relinquish the data channel when it is finished with it by transmitting a data channel available message on the control channel, but this would double the utilization of the control channel, increasing the chances of conflicts and increasing latency. The requirement that a large amount of state information be maintained at each node also increases that chances that a node could get out-of-sync, creating conflicts and errors in the data network.

7.2.1 A THORN-Based Media Access Control Protocol

Another very promising control channel based media access control protocol was proposed for the HORN network [24]. This protocol, referred to as the Token Hierarchical Optical Ring Network protocol (THORN) is a token based protocol based on the Decoupled Multichannel Optical Network (DMON) protocol [37]. In the THORN protocol, tokens are passed on the control channel in a virtual token ring. As can be seen in Fig. 10, THORN tokens contain a bit field containing the active/inactive state of each of the data channels. There is also a bit field in the token that is used to request access to a channel that is currently busy. In addition, there is an optional payload field that can be used to transmit small, high priority data packets directly over the control channel. All state information is maintained in the token, so local state information is not be required at the processing nodes in the network, although processors may store the previous token state in the eventuality that a token might be lost by a processor going down or other network error. In this eventuality, the previous token state could be used to regenerate the token. This still requires that processors maintain a small amount of state information, but this state information would be constantly refreshed and would seldom be used, so the chances of the state becoming out-of-sync is minimal.

As can be seen in Fig. 11, there is a single control channel for any number of data channels, and tokens are continuously passed on the control channel that hold the entire state of the data channels. If a processing node wishes to transmit on a particular data channel, it must wait for the token to be received over the control channel. It then checks

Busy Bits (one bit / channel)	Request Bits (one bit / channel)	Payload (optional)

Fig. 10. The layout of a THORN-based token request packet. Each token packet contains one bit per channel for busy status and one bit per channel for the channel requests. The token packet also contains an optional payload for small, low latency messages.

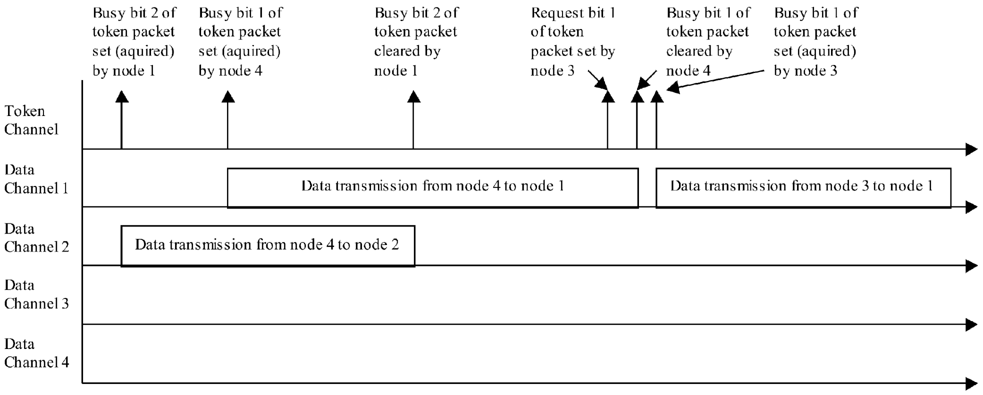
Busy bit 2 of token packet set (aquired) by node 1
Busy bit 1 of token packet set (aquired) by node 4
Busy bit 2 of token packet cleared by node 1
Request bit 1 of token packet set by node 3
Busy bit 1 of token packet cleared by node 4
Busy bit 1 of token packet set (aquired) by node 3
Token Channel
Data Channel 1
Data transmission from node 4 to node 1
Data transmission from node 3 to node 1
Data Channel 2
Data transmission from node 4 to node 2
Data Channel 3
Data Channel 4

Fig. 11. A timing diagram for control tokens and data transfers in a SOCN architecture using a form of the THORN protocol. A node may transmit on a data channel as soon as it acquires the appropriate token bit. Setting the request bit forces the relinquishing of the data channel.

the the busy bit of the requested data channel to see if it is set. If the busy bit is not set, then the data channel is not currently active, and the processing node can immediately begin transmitting on the data channel. It must also broadcast the token, setting the busy bit in the data channel that it is transmitting on. If the busy bit is already set, it implies that some other transmitter is currently using the requested data channel. If this is the case, then the processing node must set the request bit of the desired data channel, which indicates to the processing node that is currently transmitting on the desired channel that another transmitter is requesting the channel.

A disadvantage of a token-ring based media access control protocol is that the average latency for requesting channels will likely be higher that with a CSMA/CD protocol. If we assume a single control channel per cluster, with a cluster containing n processors and m physical data channels (one intracluster subnetwork and $m-1$ intercluster subnetworks), the control token would contain $n \times m$ bits busy bits and $n \times m$ request bits. For example, if a system contains $n = 16$ processors per cluster and $m = 8$ WDM subnetwork links, the control token would require 128 busy bits and 128 request bits. If we assume a control channel bandwidth of $2Gbps$, and if we ignore the possibility of a token payload, we can achieve a maximum token rotation time (TRT) of $128ns$. This is assuming that a node starts retransmitting the token as soon at it starts receiving the token, eliminating any token holding latency. This would imply a minimum latency for requesting a channel of close to zero (assuming the token is just about to arrive at the requesting processing node) and up to a maximum of $128ns$, which would give an average control channel imposed latency of approximately $64ns$. If a lower latency is required, a CSMA/CD protocol could be implemented, or multiple control channels could be constructed that would reduce the latency proportionally.

7.3 Control Channel Optical Implementation

Irrespective of the media access control protocol, a dedicated control channel is required that is broadcast to each processor sharing transmit access to each data channel. Since each physical data channel is shared among only processors within the same cluster, the control channel can be implemented local to the cluster. This will simplify the design and implementation of the control channel because it will not require routing extra optical fibers between clusters, and will not impose the optical loss penalties associated with routing the optical signals off the local cluster.

An implementation of a broadcast optical control channel is depicted in Fig. 12. The optical signal from a dedicated VCSEL on each processor is routed through a polymer waveguide based star coupler that combines all the signals from all the processors in the cluster and broadcasts the combined signals back to each processor, creating essentially an optical bus. The primary limitation of a broadcast based optical network is the optical splitting losses encountered in the star coupler. Using a similar system as a basis for a power budget estimation [38] yields an estimated optical loss in the control network of approximately $-8dB - 3dB \times log_2(n)$ (Table 4), which would support approximately 128 processor per cluster on the control channel if we assume a minimum required receiver power of $-30dBm$ and a VCSEL power of $2dBm$.

Again, the optical implementation of the SOCN MAC network has been throughly analyzed, but due to page limitation the analysis could not be included in this article.

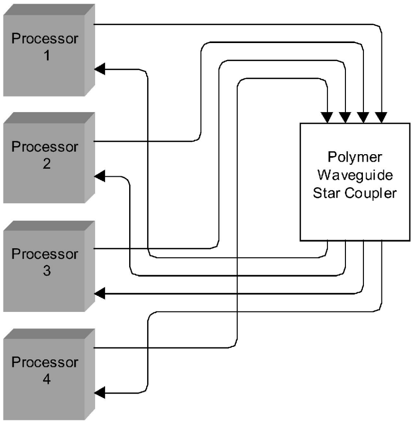

Fig. 12. An optical implementation of a dedicated optical control bus using an integrated polymer waveguide-based optical star coupler.

TABLE 4
Losses (in dB) for Each Component of the Optical Crossbar

Loss mechanism	*Loss (dB)*
VCSEL-waveguide coupling (L_{vc})	$-1dB$
Waveguide (L_w)	$-6dB$
Star coupler excess (L_{sc})	$-0.625dB$
Star coupler splitting (L_s)	$-3db \times log_2(n)$
Receiver coupling(L_{rc})	$-0.5dB$
total	$-8.125dB - 3dB \times log_2(n)$

8 Conclusions

This paper presents the design of a proposed optical network that utilizes dense wavelength division multiplexing for both intracluster and intercluster communication links. This novel architecture fully utilizes the benefits of wavelength division multiplexing to produce a highly scalable, high bandwidth network with a low overall latency that could be very cost effective to produce. A design for the intracluster links, utilizing a simple grating multiplexer/demultiplexer to implement a local free space crossbar switch was presented. A very cost effective implementation of the intercluster fiber optic links was also presented that utilizes wavelength division multiplexing to greatly reduce the number of fibers required for interconnecting the clusters, with wavelength reuse being utilized over multiple fibers to provide a very high degree of scalability. The fiber-based intercluster interconnects presented could be configured to produce a fully connected crossbar network consisting of tens to hundreds of processors. They could also be configured to produce a hybrid network of interconnected crossbars that could be scalable to thousands of processors. Such a network architecture could provide the high bandwidth, low latency communications required to produce large distributed shared memory parallel processing systems.

References

[1] J. Duato, S. Yalmanchili, and L. Ni, *Interconnection Networks and Engineering Approach,* 1997.

[2] L.M. Ni, "Issues in Designing Truly Scalable Interconnection Networks," *Proc. 1996 ICPP Workshop Challenges for Parallel Processing,* pp. 74-83, Aug. 1996.

[3] D.J. Kuck, *High Performance Computing: Challenges for Future Systems.* New York: Oxford University Press, 1996.

[4] S. Frank, J. Rothnie, and H. Burkhardt, III, "The KSR1: Bridging the Gap between Shared Memory and MPPs," *Proc. Compcon,* pp. 285– 294, Mar. 1993.

[5] D. Lenoski, J. Laudon, K. Gharachorloo, W. Weber, A. Gupta, J. Hennessy, M. Horowitz, and M. Lam, *The Stanford Dash,* vol. 25, no. 3, pp. 63–79, Mar. 1992.

[6] D. Gajski, D.J. Kuck, D. Lawrie, and A. Sameh, "CEDAR—A Large Scale Multiprocessor," *Proc. Int'l Conf. Parallel Processing,* pp. 524-529, Aug. 1983.

[7] D. Cheriton, H. Goosen, and P. Boyle, "ParaDiGM: A Highly Scalable Shared-Memory Multicomputer Architecture," *IEEE Computer,* vol. 24, no. 2, pp. 33-46, Feb. 1991.

[8] C. Chen, D.P. Agrawal, and J.R. Burke, "Design and Analysis of a Class of Highly Scalable Hierarchical Networks: PdBCube," *J. Parallel and Distributed Computing,* vol. 22, pp. 555–564, 1994.

[9] V. Cantoni, M. Ferreti, and L. Lombardi, "A Comparison of Homogeneous Hierarchical Interconnection Structures," *Proc. IEEE,* vol. 79, no. 4, pp. 416–427, Apr. 1991.

[10] Cray Research Inc., "Cray T3D," technical summary, Oct. 1993.

[11] C. Chen, D.P. Agrawal, and J.R. Burke, "dBCube: A New Class of Hierarchical Multiprocessor Interconnection Networks with Area Efficient Layout," *IEEE Trans. Parallel and Distributed Systems,* vol. 4, pp. 1,332–1,344, Jan. 1993.

[12] J.M. Kumar and L.M. Patnaik, "Extended Hypercube: A Hierarchical Interconnection Network of Hypercubes," *IEEE Trans. Parallel and Distributed Systems,* vol. 3, pp. 45–57, Jan. 1992.

[13] D. Nussbaum and A. Agrawal, "Scalability of Parallel Machines," *Comm. ACM,* vol. 34, pp. 57–61, Mar. 1991.

[14] S. Dandamudi and D. Eager, "Hierarchical Interconnection Networks for Multicomputer Systems," *IEEE Trans. Computers,* vol. 39, pp. 786–797, June 1990.

[15] P. Dowd, K. Bogineni, K.A. Aly, and J.A. Perreult, "Hierarchical Scalable Photonic Architectures for High Performance Processor Interconnection," *IEEE Trans. Computers,* vol. 42, pp. 1,105–1,120, Sept. 1993.

[16] S.L. Johnsson, "Issues in High Performance Computer Networks," *Newsletters of the Computer Architecture Technical Committee,* pp. 14–19, Summer/Fall 1994.

[17] J.L. Hennessy and D.A. Patterson, *Computer Architecture: A Quantitative Approach.* Palo Alto, Calif.: Morgan Kaufmann, 1990.

[18] S.L. Johnsson, "Data Motion and High Performance Computing," *Proc. Int'l Workshop Massively Parallel Processing Using Optical Interconnections,* pp. 1–19, Apr. 1994.

[19] J. Neff, "Optical Interconnects Based on Two-Dimensional VCSEL Arrays," *IEEE Proc. First Int'l Workshop Massively Parallel Processing Using Optical Interconnections,* pp. 202–212, Apr. 26-27, 1994.

[20] A.A. Sawchuk, C.S. Raghavandra, B.K. Jenkins, and A. Varma, "Optical Crossbar Networks," *IEEE Computer,* vol. 20, no. 6, pp. 50– 62, June 1987.

[21] F.T.S. Yu, *Optical Information Processing.* New York: Wiley, 1983.

[22] A.D. McAulay, *Optical Computer Architectures: The Application of Optical Concepts to Next Generation Computers.* John Wiley, 1991.

[23] H.S. Hinton, *An Introduction to Photonic Switching Fabrics.* New York: Plenum Press, 1993.

[24] A. Louri and R. Gupta, "Hierarchical Optical Ring Interconnection (Horn): A Scalable Interconnection Network for Multiprocessors and Multicomputers," *Applied Optics,* vol. 36, no. 2, pp. 430–442, Jan. 1997.

[25] A. Louri, B. Weech, and C. Neocleous, "A Spanning Multichannel Linked Hypercube: A Gradually Scalable Optical Interconnection Network for Massively Parallel Computing," *IEEE Trans. Parallel and Distributed Systems,* vol. 9, no. 5, pp. 497–512, May 1998.

[26] F.A.P. Tooley, "Optically Interconnected Electronics: Challenges and Choices," *Proc. Int'l Workshop Massively Parallel Processing Using Optical Interconnections,* pp. 138–145, Oct. 1996.

[27] L.N. Bhuyan and D.P. Agrawal, "Generalized Hypercube and Hyperbus Structures Constructing Massively Parallel Computers," *IEEE Trans. Computers,* vol. 33, pp. 323–333, 1984.

[28] F.P. Preparata and J. Vuillemin, "The Cube-Connected Cycles: A Versatile Network for Parallel Computation," *Comm. ACM,* pp. 300–309, May 1981.

[29] G.H. Barnes, R.M. Brown, M. Kato, D.J. Kuck, D.L. Slotnick, and R.A. Stokes, "The Illiac IV Computer," *IEEE Trans. Computers,* vol. 17, no. 8, pp. 746–757, Aug. 1968.

[30] L.D. Wittie, "Communication Structures for Large Networks of Microcomputers," *IEEE Trans. Computers,* vol. 30, pp. 264–273, Apr. 1981.

[31] Y.S. Liu et al. "Plastic VCSEL Array Packaging and High Density Polymer Waveguides for Board and Backplane Optical Interconnects," *Proc. 1998 IEEE Electronic Components and Technology Conf,* pp. 999–1,005, 1998.

[32] B. Webb, "SOCN: A Highly Scalable Optical Interconnect Network for Parallel Computing Systems," masters thesis, Univ. of Arizona, Tucson, June 1999.

[33] B. Webb and A. Louri, "An All-Optical Crossbar Switch Using Wavelength Division Multiplexing and Vertical-Cavity Surface-Emitting Lasers," *Applied Optics,* vol. 38, no. 29, Oct. 1999.

[34] B. Webb and A. Louri, "A Free Space Optical Crossbar Switch Using Wavelength Division Multiplexing and Vertical-Cavity Surface-Emitting Lasers," *Proc. Fifth Int'l Conf. Massively Parallel Processing Using Optical Interconnects (MPPOI),* pp. 50–57, June 1998.

[35] M.Y. Li, W. Yuen, and C.J. Chang-Hasnain, "Top-Emitting Micromechanical VCSEL with a 31. 6 nm Tuning Range," *IEEE Photonics Technology Letters,* vol. 10, no. 1, pp. 18-20, Jan. 1998.

[36] B. Webb and A. Louri, "A Scalable All-Optical Crossbar Network Using Wavelength Division Multiplexing and Tunable Vertical-Cavity Surface Emitting Lasers," *Proc. Symp. High Performance Interconnects,* pp. 169–180, Aug. 1999.

[37] T.M. Pinkston and C. Kuznia, "Smart-Pixel-Based Network Interface Chip," *Applied Optics,* vol. 36, no. 20, pp. 4,871–4,881, July 1997.

[38] K.W. Beeson, M.J. McFarland, W.A. Pender, J. Shan, C. Wu, and J.T. Yardley, "Laser-Written Polymeric Optical Waveguides for Integrated Optical Device Applications," *Proc. SPIE,* vol. 1,794, pp. 397–404, 1992.

Brian Webb received his BS degree in computer engineering from the University of Arizona in 1989 and his MS degree in electrical engineering from the University of Arizona in 1999. He has also been employed by Science Applications International Corporation (SAIC) since 1986 and is currently a senior staff member at the SAIC Tucson office. Mr. Webb has published several papers and journal articles in the fields of optical interconnects, parallel processing, and image processing. He has participated in research projects involved with parallel and distributed processing, parallel optical interconnects, image processing, image understanding, and image pattern recognition. He is a current member of the IEEE.

Ahmed Louri received his PhD degree in computer engineering in 1988, the MS degree in computer engineering in 1984, both from the University of Southern California, Los Angeles. He is currently a professor of electrical and computer engineering at the University of Arizona and director of the Optical Networking and Parallel Processing Laboratory. His research interests include parallel processing, optical computing systems, and scalable optical interconnection networks. He has published numerous journal and conference articles on the above topics. In 1991, he received the "Best Article of 1991 Award" from IEEE Micro. In 1988, he was the recipient of the U.S. National Science Foundation Research Initial Award. In 1994, he was the recipient of the Advanced Telecommunications Organization of Japan Fellowship, Ministry of Post and Telecommunications, Japan. In 1995, he was the recipient of the Centre Nationale de Recherche Scientifique (CNRS), Fellowship, France. In 1996, he was the recipient of the Japanese Society for the Promotion of Science Fellowship.

Prior to joining the University of Arizona, he worked as a researcher with the Computer Research Institute at the University of Southern California, where he conducted extensive research in parallel processing, multiprocessor system design, and optical computing. He has served as a member of the Technical Program Committee of several conferences, including OSA Topical Meetings on Optics in Computing, OSA/IEEE Conference on Massively Parallel Processors using Optical Interconnects. For more information, please see his web address: www.ece.arizona.edu/department/ocppl. He is a senior member of the IEEE and a member of OSA.