

Optimal Power Efficient Photonic SWMR Buses

Eldhose Peter

Computer Science and Engineering
Indian Institute of Technology, New Delhi, India
Email: eldhose@cse.iitd.ac.in

Smruti R. Sarangi

Computer Science and Engineering
Indian Institute of Technology, New Delhi, India
Email: srsarangi@cse.iitd.ac.in

Abstract—In this paper we solve the problem of minimizing the power consumption in SWMR (single writer, multi reader) optical buses for both the ring and tree based configurations. To minimize the power consumption it is necessary to design the beam splitters in a SWMR bus optimally. A prior approach by Binzhang et al. [5] proposes to use an exponential time NLP based solution for solving a related problem, and there are other approaches for solving this problem sub-optimally. In comparison we propose an optimal linear time algorithm for both ring and tree based buses. Additionally, we take into account the fact that the power loss in a splitter is dependent on the split ratio. Subsequently, we propose a hardware version of our algorithm that takes 4 cycles for a 64-station ring, and 32 cycles for a 64-station balanced tree to compute an optimal configuration. With the help of exhaustive Synopsys RSoft simulations, we show that our SWMR bus is 87% more power efficient than a popularly used alternative approach proposed by Kurian et al. [9]. For large number of stations, an optimally designed tree based bus is a further 25% more power efficient than a ring.

I. INTRODUCTION

Optical networking is a disruptive technology, and is expected to feature in mainstream processors by the end of this decade. However, we need to solve several problems in the design and operation of optical networks before optical networks can be used in practical settings. This paper focuses on one such problem that is integral to most optical networks and has hitherto not received significant attention. The problem is to minimize the power consumption of SWMR optical buses that have multicast and broadcast capabilities. Such buses are extremely useful in implementing cache coherence protocols [9], [14], [22], optical barriers [2], and for implementing a power(backbone) waveguide [23] that distributes optical power to all the active transmitters.

The exact problem is as follows. Let us consider a single writer multiple reader (SWMR) bus that has a single transmitter, and multiple receivers. The transmitter typically uses multiple wavelengths to transmit a message (using DWDM – dense wavelength division multiplexing). The receivers receive a part of the signal using a beam splitter. This signal is sent to an array of photodetectors to detect the wavelengths present in the signal. In modern optical networks at least 16-64 optical stations [23], [15] (receivers or transmitters) are connected on a single SWMR bus. We show in this paper that due to restrictions imposed by optical physics, power losses in optical networks are multiplicative in nature, and are exponentially dependent on the number of splitters in series. To reduce the losses and minimize the optical power requirement, it is necessary to optimally design splitters and adjust their split ratios. The first problem that we solve in this paper is to optimally assign the split ratios in a SWMR bus, where the objective function is to minimize optical power.

Let us comment about the novelty of our approach. The observation that losses are multiplicative in nature, and it is necessary to correctly size splitters is not new (refer to [9]). The most closely related work by Binzhang et al. [5] uses an exponential time NLP (non-linear programming) solution for optimally finding the split ratios in a tree based network for a related problem. Kurian et al. [9] have proposed another sub-optimal algorithm to size splitters that is similar to the method use by the authors of [24]. In comparison, our approach is a linear time optimal dynamic programming algorithm. We further use variants of the base algorithm to solve the optimal splitter assignment problem for trees that can either be in a single optical layer, or in multiple layers. Additionally, we do not require the splitters to be of the same type. We can choose among a portfolio of splitters. Moreover, we take into account the complex relationship between the splitter loss and the split ratio. We derive this relationship between the splitter loss and split ratio by conducting exhaustive simulations using the industry standard tool, Synopsys RSoft. To the best of our knowledge, this is a novel contribution.

Next, we look at implementing our solution in hardware for architectures that have tunable splitters (similar to [24]). Recent proposals [24], [10] propose to dynamically reconfigure an optical network based on workload characteristics. In this case, it is necessary to compute the optimal configuration of splitters at a processor’s time scale (tens of cycles). This requires a very fast hardware implementation. We propose a hardware implementation that takes 4 cycles for a 64-station ring, and 32 cycles for a 64-station balanced tree.

Lastly in Section V we explore the design space of optical buses that are optimally designed using our algorithms. The effect of optimizing splitters is the most visible when we consider the problem of broadcasting a single bit to multiple stations at the same time. For this problem, we show experimentally that a SWMR bus (designed with our algorithm) is 87% more power efficient than a bus designed using the algorithm proposed in [9], [24]. An optimally designed tree is a further 25% more power efficient than an optimal bus.

II. BACKGROUND

A. Optical Communication

Figure 1(a) shows a typical optical communication system. We first need a light source, which can either be an off-chip laser, or an on-chip laser. Off chip lasers can produce more power; however, they require sophisticated methods for routing the optical power within the chip. On-chip lasers such as *vertical cavity surface emitting lasers* (VCSELS) produce less power, but can be integrated easily with the rest of the optical devices. We use waveguides to carry the optical signal

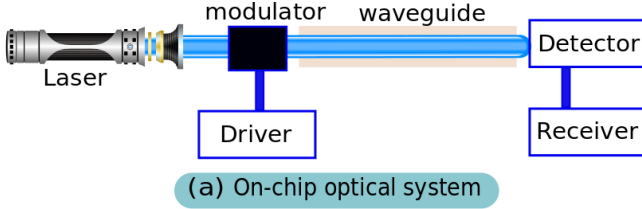
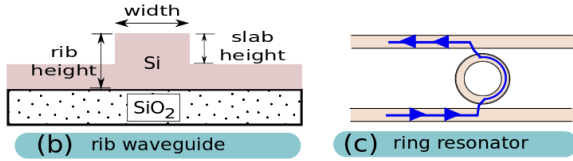


Fig. 1. An optical communication system

around the chip. A waveguide is a slab of silicon or a polymer that guides light along its path. A waveguide used in photonics (on-chip optical communication) has a typical ribbed structure (see Figure 1(b)). A waveguide is typically $0.5\text{-}1\mu\text{m}$ wide. We use Si waveguides with a refractive index of 3.46.

The next step is to modulate the optical signal to encode information. The presence of light at a time slot indicates 1, and absence indicates 0. Typically, a circular waveguide based structure called a *micro-ring resonator* is used to modulate light as shown in Figure 1(c). The micro-ring resonator can couple the light from a waveguide to another waveguide and remove almost all of the light to encode a logical 0. Alternatively, it is possible to take the resonator off resonance by applying a small electrical charge to it. This changes the refractive index of a portion of the circular waveguide, and the optical signal is not removed from the original waveguide (logical 1). It is possible to do this billions of times a second, and thus encode a sequence of 0s and 1s.

To detect a signal at a station, we need to first consider whether there are other stations downstream that might require the signal. If this is the case, then we need to use a beam splitter to split a fraction of the signal, and transfer it to another waveguide. A beam splitter can be created by forking the waveguide (Y-junction), or by using a directional coupler (two parallel waveguides). The signal then needs to be fed to a photodetector. An *optical station* typically corresponds to a set of cores (2-8). The individual cores are connected by electrical links to the optical stations.

III. ANALYTICAL MODEL

Let us consider a generic SWMR bus. One station writes to the bus, and all the other stations read from it. Each station that reads data is connected to a beam splitter that extracts a **fixed fraction** of the optical power. We shall take a look at two fundamental structures proposed in prior work – a logical ring [15], [23], and a tree [13], [5].

A. Hypothetical SWMR-S Bus

Let us consider a hypothetical SWMR bus (SWMR-S). Let us now consider any single transmitter. It is connected to the rest of the $n - 1$ stations.

Let the input signal power be y units in a channel, and the minimum amount of power that is required to detect the value

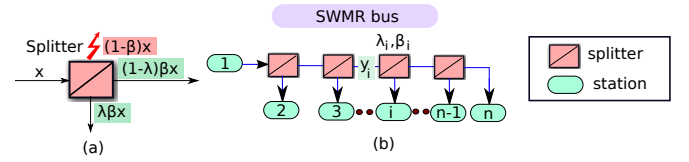


Fig. 2. A SWMR bus

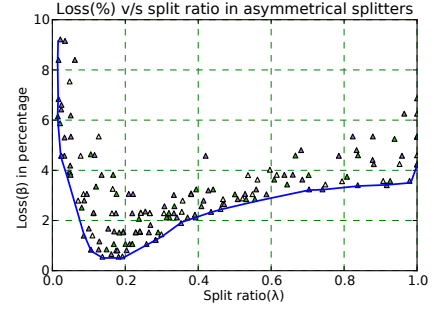


Fig. 3. Split ratio v/s power loss in percentage (Y junction)

of a signal be 1 unit. Let us model a splitter as a black box as shown in Figure 2(a). It takes x units of power as input, and it loses $(1 - \beta)x$ units of power. Out of the remaining power, it sends a fraction (λ) to a station, and the remaining part of the signal ($\beta(1 - \lambda)x$) down the bus. Let us now make a simplistic assumption that all the splitters have the same β and λ . Note that this is a simple and hypothetical simulation that has been made to simplify analytical modeling. We shall break this assumption when we consider the general SWMR bus.

Now, let us assume that the first station transmits a signal. The splitter of the second station splits the signal. The second station receives $\beta\lambda y$ units of power. The third station receives $\beta^2(1 - \lambda)\lambda y$ units of power, and ultimately the n^{th} station receives: $\beta^{n-1}(1 - \lambda)^{n-2}\lambda y$ units of power. This should be more than the detection threshold (equal to 1 unit). We thus have:

$$\beta^{n-1}(1 - \lambda)^{n-2}\lambda y \geq 1 \quad (1)$$

The aim is to minimize the input power (y). Let us first consider the ideal case where $\beta = 1$. After differentiating Equation 1, we find the optimal value of λ to be equal to $1/(n - 1)$. Let us now find the ratio between the ideal power and the actual power transmitted, and refer to it as the **power utilization efficiency** (PUE). Ideally, we need to send 1 unit of power to each station (for all the stations other than the transmitting station). Hence, the total ideal power is equal to $n - 1$. The actual power is y . Hence, the PUE is equal to $(n - 1)/y$.

$$\frac{n - 1}{y} = \left(1 - \frac{1}{n - 1}\right)^{n-2} = \left(1 + \frac{1}{n - 2}\right)^{2-n} \approx \frac{1}{e} \quad (2)$$

We thus observe that as the number of stations increases, we need to transmit roughly e (2.718) times the amount of ideal power. Sadly, $\beta \neq 0$, and for a Y junction, the relation between β and λ is shown in Figure 3. The y-axis shows the

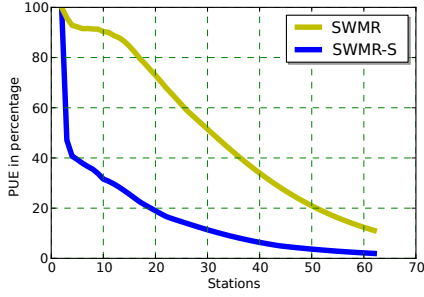


Fig. 4. Power consumption of SWMR-S and SWMR buses

power loss (in %), and the x-axis shows the split ratio(λ) for various configurations of splitters. The results were obtained by conducting exhaustive simulations using Synopsys RSoft, and the optical device models given in OptiKit [16]. Note that our splitters have a lower loss than the ones used in prior work such as Corona [23]. The envelope (Figure 3) represents the best configuration for a given split ratio from the point of view of power loss and we can use it to relate λ with the optimal value of β ($\beta = \mathcal{F}(\lambda)$). We do similar simulations (with 200+ candidate designs) for all kinds of splitters (Y, DC, MMI).

B. SWMR Bus with Different Split Ratios

1) *CASE: $\beta = 0$* : Now, let us consider a general SWMR bus (Figure 2(b)) [15], [21], where we can have splitters with different split ratios. The first splitter can split $1/(n-1)^{th}$ of the signal. The next splitter can split $1/(n-2)^{th}$ of the remaining signal and so on. Finally, the last splitter can split half the signal, and we will thus not have any loss. The PUE in this case will be 1. Kurian et al. [9] have used a similar bus. Zhou et al. [24] have also used this concept to design their bus. Let us refer to this bus as SWMR-P.

However, in a realistic scenario, we need to factor in splitter losses ($\beta \neq 0$), and we shall thus see in Section V that the SWMR-P bus is not optimal. Also, we will have a fixed portfolio of splitters and we have to choose a splitter with the most suitable split ratio. Let us thus try to optimally design a bus with n splitters with different split ratios (referred to as the *splitter assignment problem*) such as the input optical power is minimized.

2) *Linear Time Algorithm for the Splitter Assignment Problem*: Let us consider a sequence of n splitters, and focus on the i^{th} splitter. Let the input power at the i^{th} splitter be y_i , and let λ_i denote the split ratio of the i^{th} splitter. If the station is active, it needs to split the signal into two parts. One part needs to be greater than 1 (let 1 unit be the minimum power of signal to be detected), and the other part needs to be greater than the minimum power required to send a signal to the remaining $i-1$ stations (y_{i-1}). We further note that $\beta_i = \mathcal{F}(\lambda_i)$. Let us slightly redefine β to also include the loss of the segment of a waveguide associated with the splitter. Now, if the station is inactive, then y_i needs to be equal to y_{i-1}/β . For both the cases, we have:

$$\mathcal{F}(\lambda_i)\lambda_i y_i \geq A_i \quad (3)$$

$$\mathcal{F}(\lambda_i)(1-\lambda_i)y_i \geq y_{i-1} \quad (4)$$

Here, A_i is the activity of the i^{th} station. It is 1 if the station is active, and 0 if it is inactive. The aim is to minimize y_i by finding the optimal values of $\lambda_1 \dots \lambda_i$. We observe that the optimal value of y_i for the i^{th} splitter is related to y_{i-1} , which is the optimal power for a system with $(i-1)$ splitters. This problem has two important properties – (1) overlapping subproblems and (2) optimal substructure. We can thus use dynamic programming to solve this problem. We find the optimal designs for splitters $1 \dots i$. We use this information to find the optimal design of the $(i+1)^{th}$ splitter.

Algorithm 1: ASSIGNSPLITTERS

Input: Number of nodes: N , Activity of each station $A[]$

Output: $(\Lambda[], \phi)$: $\Lambda[i] = \lambda_i$, $\phi = \text{Min. optical power for } N \text{ stations}$

```

1 if  $N = 1$  then
2   return  $(1, A[1])$ ;
3 else
4    $(\Lambda, y_{N-1}) \leftarrow \text{AssignSplitters}(N-1)$ ;
5    $(\beta_N, \lambda_N) \leftarrow \text{findBestSplitter}(A[N], y_{N-1})$ 
   /* Find the best splitter with
   split ratio 1:  $y_{N-1}$  */
6    $y_N \leftarrow y_{N-1}/((1-\lambda_N)\beta_N)$ ;
7   return  $(\Lambda \oplus \lambda_N, y_N)$ ;

```

The dynamic programming algorithm for *splitter assignment* is shown in Algorithm 1. We start with a single station. This clearly does not require a splitter. Then, we consider the general case, where we have computed the optimal solution for $N-1$ nodes. The minimum input power for $N-1$ nodes is y_{N-1} . The next step is to find the best split ratio, where the minimum power that needs to be supplied in one branch is y_{N-1} , and in the other branch is $A[i]$. We find the optimal split ratio (λ_N) and associated splitter loss (β_N) such that the input power to N stations is minimized (using the method *findBestSplitter*). A fast method to implement this method is to have a table that stores the optimal values of (β_N, λ_N) for different input powers (y_{N-1}) and activity levels (0 or 1). This table can be created offline and subsequently accessed in $O(1)$ time. In this manner, we find the optimal power for N stations. We finally return the minimum input power for N stations, and the split ratios of each splitter. Note that in the algorithm, \oplus is the concatenation operator. The overall time complexity of the algorithm is $O(N)$ (with the fast single step lookup table).

A quick comparison of both the approaches, SWMR-S and SWMR (optimized splitters with different split ratios), is shown in Figure 4. In the SWMR-S bus, we use the same splitter for all the nodes. We choose the splitter with the best (β, λ) combination to minimize the total input power. We observe from the figure that the PUE for the SWMR bus is significantly better than the SWMR-S bus. Thus, we should always design buses with different split ratios.

C. Tree Based Broadcast Schemes

Let us look at generic tree based structures that have their root at the center of the die and branches span out in different directions. The stations are the leaves of the tree.

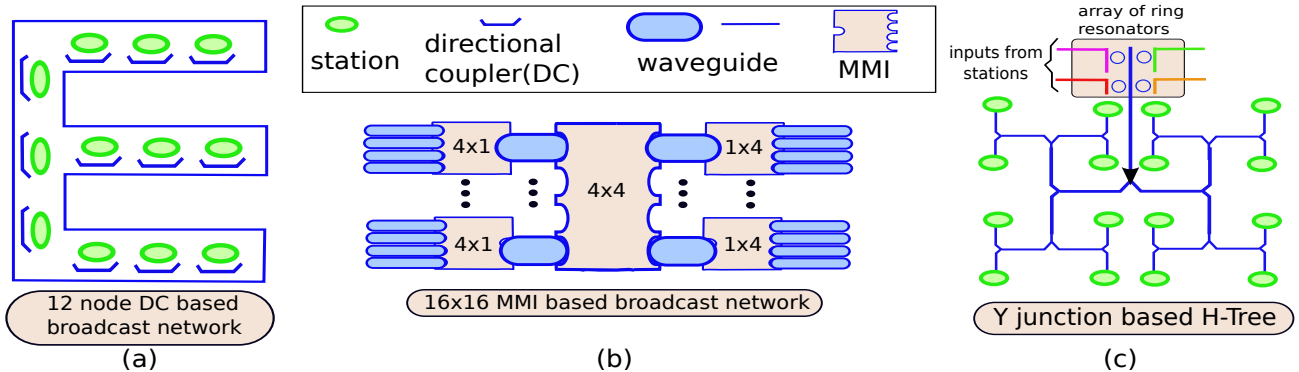


Fig. 7. Broadcast structures

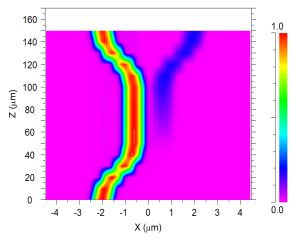


Fig. 5. Directional coupler (RSoft simulation)

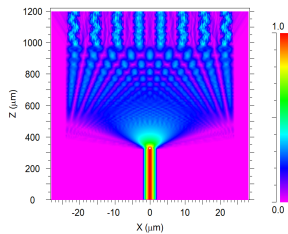


Fig. 6. MMI (RSoft simulation)

We can either use a single MWSR bus to route requests to the central node in a tree, or use separate wave guides to transfer signals to the central node. We define a MWSR (multiple writer single reader) bus that has multiple transmitters and a single reader (see [23], [15]). Since multiple writers use the same optical channel, we typically need to perform arbitration to choose the writer. In either case, we need to have an array of ring resonators to transfer the signal to a waveguide as shown in Figure 7(c). Subsequently, the signal can be sent along the edges of the tree to all the stations. We can use any of the splitters in our portfolio of devices namely the Y junction, directional coupler, or $1 \times N$ MMI ($N \in \{2,4,8\}$). A directional coupler (DC) is a structure that has two waveguides placed side by side. The evanescent part of the signal couples with the other waveguide, and thus the structure works as a splitter (see Figure 5 and Figure 7(a)). A MMI is a structure that splits an optical signal into M identical parts with a reduced intensity as shown in Figures 6 and 7(b). For lack of space, we are not describing the details of these devices in this paper. Readers can refer to [16], [17] for a deeper discussion.

We consider balanced trees in this paper. For such balanced trees, we use the H-Tree topology as shown in Figure 7(c). The distance to each leaf is the same from the center. We believe that using unbalanced trees might be beneficial especially if we have a bundle of waveguides. However, computing the topology of an optimal unbalanced tree is a difficult routing problem because we need to take the maximum optical path length, the number of crossings, and the variable splitter loss into account. This is a difficult problem and is a subject of future work. However, once the topology is decided, the splitter assignment algorithm will also work for unbalanced trees.

D. The Splitter Assignment Problem for Trees

We can extend Algorithm 1 to optimally design a tree based SWMR bus. Here, each transmitter is the root of a tree [5], [13], the splitters are internal nodes, and the rest of the stations might be turned off. We have a choice regarding the type of splitters. They might be binary splitters ($1 \text{ input} \rightarrow 2 \text{ outputs}$), or be generic $1 \rightarrow M$ splitters. Furthermore, we might have a combination of homogeneous splitters (same power at all ports), and heterogeneous splitters. We claim that all of these variants can be solved in polynomial time using our dynamic programming approach.

Let us consider one specific example in this paper. Assume that we wish to design a binary tree that allows us to broadcast to n stations from the central node, and the objective is to maximize the PUE. Here, again we can use a dynamic programming formulation. The optimal solution for two active stations uses a single 50-50 beam splitter. If one station is inactive, then the split ratio is 0-100. For all other combinations of activities and subtree sizes the split ratio will vary from 0-100 to 100-0. Now, let us assume that we have an optimal solution for the left and right subtrees. Then to obtain the optimal solution for the entire tree, we need to appropriately set the split ratio of the splitter associated with the root of the tree such that the total input power to the tree is minimized.

The algorithm for this procedure is shown in Algorithm 2. It takes as its input the root of the tree. In each iteration, we find the optimal input power values and the split ratios for the left and right subtrees respectively. The minimum power requirement for the left and right subtrees is denoted by ϕ_l and ϕ_r respectively. We assume a function *BestSplitter* that finds the optimal split ratio (λ) and minimum input power (ϕ) given the power values in the left and right branches (ϕ_l and ϕ_r). We assume that this function runs in $O(1)$ time. In this function, we first compute a ratio of ϕ_l and ϕ_r , and then access a fast lookup table based on the value of the ratio. This table stores the optimal configuration of the splitter (at the root of the subtree) and the normalized minimum input power.

We need to handle the special case of a node being a leaf. In this case, a node is a station and not a splitter. Thus, it does not have a split ratio. However, its power requirement depends upon its activity (0 or 1). Additionally, we need to factor in the losses in the waveguide that connects the station to its nearest

splitter. We assume a function, $MinPower$, which returns the minimum amount of power required given the id of a station, and its activity.

Regarding the time complexity, we observe that Algorithm 2 does a simple tree traversal, and it thus takes $O(N)$ steps. Each step takes $O(1)$ time.

Algorithm 2: ASSIGNSPLITTERST

Input: Root: R
Output: $(\Lambda[], \phi) : \Lambda[i] = \lambda_i, \phi = \text{Min. optical power for } N \text{ stations}$

```

1 if  $R.isLeaf()$  then
2   return (null, MinPower(R, R.activity()))
3 else
4    $(\Lambda_l, \phi_l) \leftarrow \text{AssignSplittersT}(R.left())$ 
5    $(\Lambda_r, \phi_r) \leftarrow \text{AssignSplittersT}(R.right())$ 
6    $(\lambda, \phi) \leftarrow \text{BestSplitter}(\phi_l, \phi_r)$ 
7    $\Lambda \leftarrow \Lambda_l \oplus \lambda \oplus \Lambda_r$ 
8   return  $(\Lambda, \phi)$ 

```

A tree is a good option, if we have a single waveguide (or a few waveguides). However, we have a bundle of K waveguides then at each splitter we will have $1 \dots K - 1$ waveguide crossings. Given that the power loss at a crossing is 0.05 dB [3], [4], [14], the total loss can be significant for a large tree based network. If we have two optical layers, then we can avoid crossings altogether. Note that multi layer photonics is at the moment a futuristic technology. However, researchers have begun prototyping chips with 2 optical layers [1], and through-silicon optical vias [19], [11]. Let us also comment on the practicality of splitters in SWMR buses. Let us first look at splitters that are not tunable. Given nanometer level feature sizes, we can very accurately fabricate micron-scale splitters, and thus ensure that they have optimal split ratios. Thus our algorithm is very suitable for design time optimizations. For designs that use tunable splitters, our algorithm can be used to properly tune them dynamically. Research in this area is picking up, and the architecture/EDA community has begun to take cognizance of highly tunable (tuning range of 20dB) splitters as described in [24]. These are fast MMI based splitters that are very versatile and rely on the electro-optic effect to generate a change in refractive index (maximum $\Delta n = 0.01$). Their utility has already been proven in off-chip networks [18].

IV. IMPLEMENTATION IN HARDWARE

There have been some recent proposals [24], [10], [15] that propose to dynamically allocate bandwidth to groups of nodes based on their declared or predicted usage. For example, Zhou et al. [24] use a tree based power waveguide to distribute optical power to the nodes. They use a variant of the SWMR-P (see Section III-B1) algorithm. Based on the results of their predictor they vary the split ratios of the splitters in the tree. This strategy reduces the optical power requirement. We envisage similar proposals in the future that will propose to dynamically change the split ratios based on characteristics of the workload.

The only requirement of such algorithms is that they need to be fast and compute the optimal set of split ratios quickly.

This needs to be at the processor's time scale and should typically take tens of cycles (not hundreds of cycles). Let us thus propose implementations of Algorithm 1 and 2 in hardware.

A. SWMR Ring

A naive solution is to simulate the dynamic programming algorithm in hardware. However, this will take roughly N iterations, and each iteration involves a costly lookup operation for finding the best splitter.

Let us instead adopt an alternative approach. Let us consider K consecutive calls to the function $AssignSplitters$, and treat it as one large function. Let the indices of the splitters be $i \dots (i + K - 1)$. The inputs to this large function are: $y_{i-1}, A[i] \dots A[i + k - 1]$, and $\Lambda[1 \dots (i - 1)]$. The split ratios in the array Λ do not change the outcome of the function; hence, let us ignore them for the time being. The inputs are essentially the input power to the first $i - 1$ nodes (y_{i-1}) and the activity of K stations ($A[i] \dots A[i + k - 1]$). Let us use this insight to design a lookup table.

Let us represent each power value by 8 bits (quantization error limited to 1%). Let us choose K as 4 (experimentally determined to be a good choice). We can thus represent the combined input as: $y_{i-1} \oplus A[i] \oplus A[i + 1] \oplus A[i + 2] \oplus A[i + 3]$ using 12 bits (here \oplus is the concatenation operator). Let us have a lookup table that is indexed by these 12 bits. Each entry in this table needs to contain four split ratios, and the input power for $i + 3$ stations. If we represent each split ratio using 6 bits (output power accurate within 1%), and the power value using 8 bits, we require a total of 32 bits (4 bytes). Thus, the total size of the table is 16 KB. Since we need to store one such table in the entire system, the area overhead is very small.

We simulated such a table using the memory modeling tool, Cacti 5.3 [20]. It is a read-only structure with 1 read port (associativity of 1, 1 bank). We designed the table in 18 nm technology (scaled the results from 32 nm using the results in [7]). Cacti predicted an access time of 85 ps (after scaling). Assuming a clock frequency of 2.5 GHz (clock period = 400 ps), we can easily fit four such accesses in one cycle with a 10% timing margin for process corners. Hence, it is possible to generate the optimal configuration for $4 \times 4 = 16$ stations in a single cycle. Accessing this lookup table four times in a cycle is not very difficult. We can use either of two standard approaches. We can either use a frequency divider to locally produce a faster clock (at 4X the frequency), or we can use three separate locally generated clock signals that are obtained by shifting the master clock signal by $\pi/2, \pi$ and $3\pi/2$ radians respectively. Using this approach we will take $\lceil N/16 \rceil$ cycles for generating the optimal split ratios for N stations.

B. Tree

In the case of a tree, the problem is slightly more complicated. Still a combination of a lookup table based approach and the dynamic programming based approach (as shown in Algorithm 2) can be used.

We use a lookup table for trees with up to 8 nodes. Since each node can either be active or inactive, we have $2^8 = 256$ combinations. We thus have a 256 entry table, where each row

stores the split ratios of the 7 splitters (in the 8 node tree) and the input power (8 bits). The total number of bits per entry that we require is: $7 \times 6 + 8 = 50$ bits. The total size of the table is thus 1.8 KB. The time per access is 113 ps at 18nm, which is well within 1 cycle (generated by Cacti 5.3, and scaled). If we have 8 stations in a tree, then we can get the optimal configuration within a single cycle.

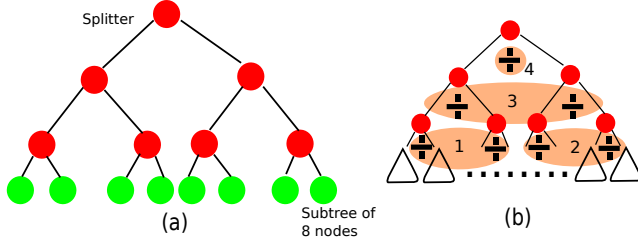


Fig. 8. Reconfiguration of splitters in trees

Let us now discuss the steps for a system with more than 8 stations by considering an example 64-station architecture. First, we can compute the optimal configurations for groups of 8 stations (leaves of the tree) by using the lookup table based approach. Then we need to combine the solutions to produce a solution for the 64 station system. If we assume that the lookup table for 8 stations has 2 read ports, then it will take us 4 cycles to complete this phase (Phase I).

Let us now move to the next phase (Phase II). In Figure 8(a) each of the leaves is a subtree of 8 nodes, and the internal nodes are splitters (total: 7). To find the split ratios of these splitters we need to run the optimal offline algorithm. In each step of the algorithm we need to find the ratio of the powers of the left and right branches, and use this ratio to index a lookup table that yields the optimal configuration of the splitter. The time will be dominated by the slow division operation, where we need to divide two 8 bit numbers (powers of the left and right branches). We use a fast method to divide two small 8 bit numbers based on a reciprocal cache [12]. If the two power values are $(\phi_l$ and ϕ_r) respectively (w.l.o.g $\phi_l > \phi_r$), then we first compute $2^8/\phi_r$ (integer division) by accessing a 256 entry lookup table. This takes less than 1 cycle (@2.5 GHz). Then we multiply ϕ_l with $2^8/\phi_r$, and right shift the result by 8 places. We assume 1 cycle for the reciprocal cache lookup operation, 2 cycles for 8-bit multiplication (a 64-bit multiplier on Intel Sandybridge takes 3 cycles), and 1 cycle for the shift (total: 4 cycles). We then need one more cycle to get the optimal configuration for the splitter from a lookup table. Since the ratio can only take 256 values, we need a lookup table containing 256 entries (very fast: <80 ps access time).

To compute the input power we need to multiply the power obtained from the lookup table by ϕ_r , which will take 2 more cycles. We need to do this because the lookup table stores normalized values, which assume $\phi_r = 1$. Now, if we have two such small circuits with a 8 bit multiplier, a reciprocal cache, and a lookup table, then it will take us 4 rounds, where each round is $(4+1+2=7)$ cycles (see Figure 8(b)). Thus, the total time requirement for computing the split ratios of the 7 splitters is 28 cycles. If we add the time for Phase I, then the total time requirement is 32 cycles for 64 stations. In general, for $N > 8$ stations where N is a multiple of 8, it will take us $\lceil N/16 \rceil + 7 \times (\lceil (N/8 - 1)/2 \rceil)$ cycles.

Optical Parameters	
Wavelength (λ)	1.55 μm
Width of waveguide (W_g)	3 μm
Slab height	1 μm
Rib height	3 μm
Refractive Index of $\text{SiO}_2(n_r)$	1.46
Refractive Index of $\text{Si}(n_c)$	3.45
Insertion Coupling Loss	5 dB
Input Driver Power	43 μW (at 18 nm)
Output Driver Power	94 μW (at 18 nm)
Ring Heating	26 μW per ring
Ring Modulation	500 μW
Splitter Loss	0.1 dB - 0.3 dB
Waveguide Loss	1 dB/cm
Filter Drop	1 dB
Non Linearity	1 dB
Photodetector minimum power	40 μW
Output Coupling Loss	0.6 dB
Photodetector quantum efficiency	0.8 A/W
Laser wall plug efficiency	30%

TABLE I. OPTICAL SIMULATION PARAMETERS [6], [13]

V. EVALUATION

A. Setup

In this section, we evaluate the power efficiency of our SWMR bus for both rings and trees. Note that our approach relies on multiple stations reading a channel concurrently; hence, we need to evaluate our results with broadcast/multicast traffic. Broadcast/multicast based optical buses are used in optical networks [8], [23] for implementing barriers and cache coherence protocols. Additionally, the power waveguide that distributes power to the optical stations is conceptually a SWMR based broadcast bus. We compare different designs on the basis of the energy required to broadcast a single bit by one station. For the SWMR bus (ring or tree), we consider a single station that sends a signal to $N - 1$ stations, and each station gets a portion of the light using a beam splitter. It might be using multiple bits (in a DWDM configuration) for sending a message; however, we are only interested in the transmission of any single bit for the purposes of evaluation.

We compare our approach with the approach adopted in [9], [24], which we have referred to as the SWMR-P bus in Section III-B1. Its split ratios are of the form (1:1, 1:2, 1:3, 1:4, ...). We evaluate a default SWMR-P bus that has a fixed loss of 0.2 dB per splitter (as assumed in [24]). Additionally, we evaluate SWMR-P buses with the splitters that we have designed with Synopsys RSoft. We define two new designs, SWMR-P-DC and SWMR-P-Y, that use directional couplers and Y junctions respectively. In both SWMR-P-DC and SWMR-P-Y the split ratio of each splitter is the same as the splitters in the SWMR-P bus. However, the power loss per splitter is not fixed at 0.2dB. Instead it is equal to the minimum power loss of a splitter in our portfolio of splitters with that specific split ratio. SWMR-DC and SWMR-Y are optimal configurations designed using our algorithms.

We wrote a C program to simulate all the topologies. To get the relationship between the split ratio(λ) and the minimum power loss (β), we used the OptiKit component library, and Synopsys RSoft 8.0.2 to simulate a variety of splitters with 200+ candidate designs, and computed the relationship between the split ratio and the minimum power loss. Table I contains some of the additional parameters that we use. We report the energy that is consumed by the external laser by taking into account the wall plug efficiency, and the different different coupling losses.

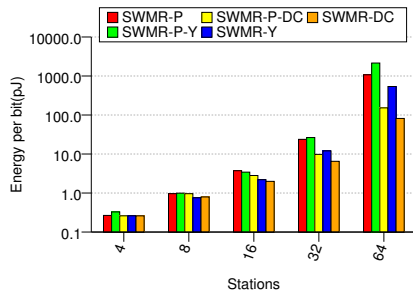


Fig. 9. Energy per bit

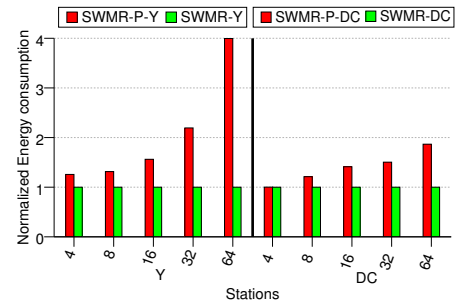


Fig. 10. Comparison of structures with Y and DC splitters

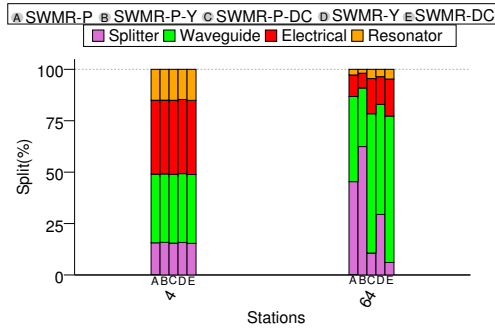


Fig. 11. Energy breakup

B. Rings

Figure 9 compares all the five designs based on the energy per bit. This graph is in the log-scale, and is meant to primarily highlight broad trends and indicate designs that are infeasible. Let us list the basic trends. The SWMR-P designs are almost as efficient as optimal designs (within 0.25pJ) up till 8 stations. For 16 stations, SWMR-P requires 3.73 pJ, and SWMR-P-Y requires 3.43 pJ, whereas SWMR-Y and SWMR-DC require 2.19 and 1.99 pJ respectively. For 32 stations, the energy requirement of SWMR-P and SWMR-P-Y is 2 times more than SWMR-Y. We can also see the advantage of having directional couplers for 32 stations. The energy requirement of SWMR-DC is roughly half of that of SWMR-Y. There are clear scalability issues for 64 stations. SWMR-P, SWMR-P-Y, and SWMR-Y are not scalable. Their energy requirement is in the range of 500-2000 pJ, which is roughly 8-26 times more than the DC based designs. SWMR-DC is the best design, and its power requirement for 64 stations is 81 pJ.

Let us now not consider the SWMR-P configuration and focus on the rest of the designs. Moreover, let us compare designs with the same type of splitters in Figure 10. This figure has a linear scale and is meant to show the reduction in energy consumption for the optimal designs. For designs using the Y junction, the reduction in energy is 25% for 4 stations, 120% for 32 stations, and 300% for 64 stations. Similarly, for DC, the reduction in energy is 0.1% for 4 stations, 50% for 32 stations, and 87% for 64 stations. To get a deeper insight into the results, let us look at Figure 11 that shows a relative breakup of the energy consumption. We observe that the power loss due to the splitters is the most dominant factor, which is what we have been trying to minimize by optimally sizing the splitters. The other sources of losses are waveguide losses, resonator losses (thermal tuning, ring losses), and electrical losses (power consumed by the transmitter and receiver circuits).

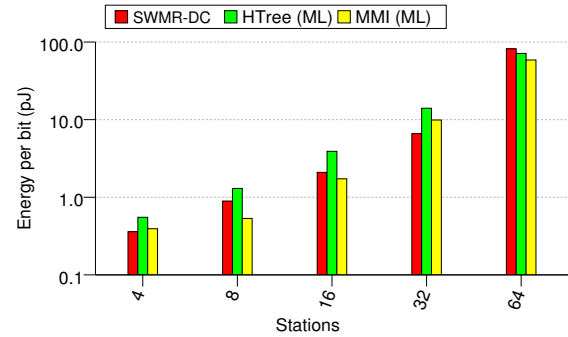


Fig. 12. Comparison between rings and trees

C. Trees

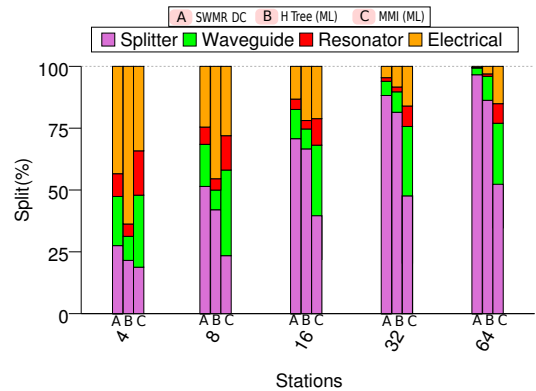


Fig. 13. Energy breakup (ring and tree)

A tree is slightly more difficult to design because it is hard to support multiple parallel waveguides. In bus based networks such as Firefly [15] and Corona [23] the entire system has $O(N)$ waveguides such that all pairs of nodes can communicate concurrently. Each waveguide supports DWDM for transmitting a multi-bit message. However, in the case of trees if we have multiple parallel waveguides, then we shall have an average of $K/2$ crossings per internal node, where K is the number of parallel waveguides. Given that the loss per crossing is 0.05 dB, the total power loss for a 64 station tree can be significant. In a system with a single writer, we can always use a tree in a single optical layer; however, if we plan to have many parallel waveguides, then we require two optical layers to eliminate crossings. Let us thus evaluate our design with 2 optical layers, where we use a dedicated waveguide to send a message to the root node.

We evaluate three configurations in Figure 12 – Ring

(SWMR-DC), HTree(ML) (balanced tree, multi layer) and MMI(ML) (multi layer 1×4 and 4×4 ,MMI). For the MMI based splitter we manually design a radix-4 tree based network (conceptually similar to the network shown in Figure 7(b)). In the HTree we use Y-junction based splitters primarily because the split ratio is typically close to 50-50, and for such split ratios Y-junctions are more efficient than DCs.

For, 4-32 stations, the ring(SWMR-DC) is very competitive with the HTree and MMI based networks. It is the second best configuration for 8 and 16 stations, and it is the best configuration for 4 and 32 stations. This is because of two reasons. First, multi-layer structures have losses associated with vias. We model a via as a waveguide with two bends, and two resonators for coupling the signal between the waveguides. Secondly, Y-junctions (used in the HTree), and MMI based splitters typically have a higher power loss than the directional coupler. However, the SWMR-DC based bus fails to scale to 64 stations. For 64 stations the MMI based bus is the most efficient structure. We show the breakup of the energy consumed in Figure 13. We see a gradual increase in the proportion of the energy lost by splitters. The resonator, and waveguide losses are never more than 10%. The electrical losses in the transmitter and receiver keep decreasing from 50%(4 stations) to 12%(64 stations).

VI. RELATED WORK

Kurian et al. [9] propose to use splitters with different split ratios. We have referred to their algorithm as SWMR-P in this paper, and we have compared our approach against their's and obtained an energy reduction of up till 75%. The authors of Probe [24] have used an algorithm similar to SWMR-P and have assumed a constant loss per splitter. We have also evaluated this configuration in Section V. Both of these strategies start having scalability issues beyond 16 stations.

Binzhang et al. [5] take the loss of the splitters into account similar to our SWMR bus. This is the only work to the best of our knowledge that tries to optimally set the split ratios of different splitters to realize a tree based SWMR bus. However, they use a different objective function. They try to minimize the variance of the coupled power to the different nodes. Additionally, they use a non-linear programming formulation. Instead, we devised a linear time algorithm that guarantees that all the active stations get an input power of 1 unit (0 variance), and simultaneously the total input power is minimized. Our objective function is thus stronger, and our solution is faster.

VII. CONCLUSION

In this paper, we proposed optimal linear time dynamic programming based algorithms for the splitter assignment problem for rings and trees. Along with optimal offline implementations, we proposed very fast (1-4 cycles) hardware implementations for rings using a set of lookup tables. The hardware algorithm for trees takes 1 cycle for 8 stations and 32 cycles for 64 stations.

REFERENCES

- [1] A. Biberman, K. Preston, G. Hendry, N. Sherwood-Droz, J. Chan, J. S. Levy, M. Lipson, and K. Bergman, "Photonic network-on-chip architectures using multilayer deposited silicon materials for high-performance chip multiprocessors," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 7, no. 2, p. 7, 2011.
- [2] N. Binkert, A. Davis, M. Lipasti, R. Schreiber, and D. Vantrease, "Nanophotonic Barriers," in *Workshop on Photonic Interconnects & Computer Architecture (in conjunction with MICRO 41)*, 2009.
- [3] C. Chen and A. Joshi, "Runtime management of laser power in silicon-photonics multibus noc architecture," *IEEE Journal of Selected Topics in Quantum Electronics*, 2013.
- [4] C. Chen, T. Zhang, P. Contu, J. Klamkin, A. K. Coskun, and A. Joshi, "Sharing and placement of on-chip laser sources in silicon-photonics nocs,"
- [5] B. Fu, Y. Han, H. Li, and X. Li, "Accelerating lightpath setup via broadcasting in binary-tree waveguide in optical nocs," in *DATE*, 2010.
- [6] M. Haurylau, G. Chen, H. Chen, J. Zhang, N. A. Nelson, D. H. Albonese, E. G. Friedman, and P. M. Fauchet, "On-chip Optical Interconnect Roadmap: Challenges and Critical Directions," *Selected Topics in Quantum Electronics, IEEE Journal of*, 2006.
- [7] W. Huang, K. Rajamani, M. Stan, and K. Skadron, "Scaling with design constraints: Predicting the future of big chips," *Micro, IEEE*, vol. 31, no. 4, pp. 16–29, 2011.
- [8] N. Kirman, M. Kirman, R. K. Dokania, J. F. Martinez, A. B. Apsel, M. A. Watkins, and D. H. Albonese, "Leveraging Optical Technology in Future Bus-based Chip Multiprocessors," in *MICRO*, 2006.
- [9] G. Kurian, J. E. Miller, J. Psota, J. Eastepe, J. Liu, J. Michel, L. C. Kimerling, and A. Agarwal, "ATAC: A 1000-Core Cache-Coherent Processor with On-chip Optical Network," in *PACT*, 2010.
- [10] R. Morris, E. Jolley, and A. K. Kodi, "Extending the performance and energy-efficiency of shared memory multicores with nanophotonic technology," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 1, pp. 83–92, 2014.
- [11] A. Noriki, K.-W. Lee, J. Bea, T. Fukushima, T. Tanaka, and M. Koyanagi, "Through silicon photonic via (tspv) with si core for low loss and high-speed data transmission in opto-electronic 3-d lsi," in *3D Systems Integration Conference (3DIC), 2010 IEEE International*, 2010, pp. 1–4.
- [12] S. F. Oberman and M. J. Flynn, "Reducing division latency with reciprocal caches," *Reliable Computing*, vol. 2, no. 2, pp. 147–153, 1996.
- [13] I. O'Connor, "Optical Solutions for System-Level Interconnect," in *SLIP*, 2004.
- [14] Y. Pan, J. Kim, and G. Memik, "Flexishare: Channel sharing for an energy-efficient nanophotonic crossbar," in *HPCA*, 2010, pp. 1–12.
- [15] Y. Pan, P. Kumar, J. Kim, G. Memik, Y. Zhang, and A. Choudhary, "Firefly: Illuminating Future Network-on-Chip with Nanophotonics," ser. ISCA, 2009.
- [16] E. Peter and S. R. Sarangi, "Optikit: An open source kit for simulation of on-chip optical components," in *VLSI Design Conference (Poster)*.
- [17] G. T. Reed, *Silicon Photonics: The State of the Art*. John Wiley & Sons, 2008.
- [18] G. Rouskas, "Optical layer multicast: rationale, building blocks, and challenges," *Network, IEEE*, vol. 17, no. 1, pp. 60–65, Jan 2003.
- [19] A. Sharkawy, S. Shi, and D. W. Prather, "Implementations of optical vias in high-density photonic crystal optical networks," *Journal of Micro/Nanolithography, MEMS, and MOEMS*, 2003.
- [20] D. T. S. Thoziyoor, D. Tarjan, and S. Thoziyoor, "Cacti 5.3," Tech. Rep., 2006.
- [21] D. Vantrease, N. Binkert, R. Schreiber, and M. Lipasti, "Light speed arbitration and flow control for nanophotonic interconnects," in *MICRO*, 2009.
- [22] D. Vantrease, M. H. Lipasti, and N. Binkert, "Atomic coherence: Leveraging nanophotonics to build race-free cache coherence protocols," in *HPCA*, 2011.
- [23] D. Vantrease, R. Schreiber, M. Monchiero, M. McLaren, N. P. Jouppi, M. Fiorentino, A. Davis, N. Binkert, R. G. Beausoleil, and J. H. Ahn, "Corona: System Implications of Emerging Nanophotonic Technology," in *ISCA*, 2008.
- [24] L. Zhou and A. K. Kodi, "Probe: Prediction-based optical bandwidth scaling for energy-efficient nocs," in *NOCS*, 2013.