Scheduling Light-trails on WDM Rings

Soumitra Pal and Abhiram Ranade
Department of Computer Science and Engg.,
Indian Institute of Technology Bombay,
Powai, Mumbai 400076, India.

***Abstract*—We consider the problem of scheduling communication on optical WDM (wavelength division multiplexing) networks using the light-trails technology. We give two online algorithms which we prove to have competitive ratios $O(\log n)$ and $O(\log^2 n)$ respectively. We also consider a simplification of the problem in which the communication pattern is fixed and known before hand, for which we give a solution using $O(c + \log n)$ wavelengths, where c is the congestion and a lower bound on the number of wavelengths needed. While congestion bounds are common in communication scheduling, and we use them in this work, it turns out that in some cases they are quite weak. We present a communication pattern for which the congestion bound is $O(\log n/ \log \log n)$ factor worse than the best lower bound. In some sense this pattern shows the distinguishing character of light-trail scheduling. Finally we present simulations of our online algorithms under various loads.**

I. Introduction

Light-trails [1] are considered to be an attractive solution to the problem of bandwidth provisioning in optical networks. The key idea in this is the use of optical shutters which are inserted into the optical fiber, and which can be configured to either let the optical signal through or block it from being transmitted into the next segment. By configuring some shutters on (signal let through) and some off (signal blocked), the network can be partitioned into subnetworks in which multiple communications can happen in parallel on the same light wavelength. In order to use the network efficiently, it is important to have algorithms for controlling the shutters.[1]

In this paper we consider the simplest scenario: two fiber optic rings, one clockwise and one anticlockwise, passing through a set of some n nodes, where typically $n < 20$ because of technological considerations. At each node of a ring there are optical shutters that can either be used to block or forward the signal on each possible wavelength. The optical shutters are controlled by an auxiliary network ("out-of-band channel"). It is to be noted that this network is typically electronic, and the shutter switching time is of the order of milliseconds as opposed to optical signals which have frequencies of Gigahertz.

For this setting we give three algorithms for controlling the shutters, or *bandwidth provisioning*. The first two consider dynamic traffic, i.e. communication requests arrive and depart in an online manner, i.e. they have to be serviced as soon as they arrive. The algorithm must respond very quickly in this case. The third algorithm considers stationary traffic. In this case, our algorithm can be allowed to take more time, because the computed configuration will be used for a long time since the traffic pattern does not change. For both problems, our objective is to minimize the number of wavelengths needed to accommodate the given traffic.[2]

The input to the stationary problem is a matrix B, in which $B(i, j)$ gives the bandwidth demanded between nodes i and j. We give an algorithm which schedules this traffic using $O(c+\log n)$ wavelengths, where $c = \max_k \sum_{i,j|i\le k<j} B(i, j)$ is the maximum congestion at any link. The congestion as defined above is a lower bound, and so our algorithm can be seen to use a number of wavelengths close to the optimal. The reader may wonder why the additive $\log n$ term arises in the result. We show that there are communication matrices B for which the congestion is much smaller than 1, but which yet require $\Omega(\log n/ \log \log n)$ wavelengths. In some sense, this justifies the form of our result.

For the online problem, we use the notion of competitive analysis [2], [3], [4]. Specifically we establish that our first algorithm is $O(\log n)$-competitive, i.e. it requires at most a $O(\log n)$ factor more wavelengths as compared to the best possible algorithm, including an unrealistic algorithm which is given all the communication requests in advance. A multiplicative $O(\log n)$ factor might be considered to be too large to be relevant for practice (and indeed it is an important open problem whether a lower factor can be proved); however, the experience with online algorithm design is that such algorithms often give good hints for designing practical algorithms. We establish that our second algorithm for the online problem is $O(\log^2 n)$, nevertheless we mention it because it is a simplified version of the first algorithm and it seems to perform better in our simulations.

That brings us to our final contribution: we simulate two algorithms based on our online algorithms for some traffic models. We compare them to a baseline algorithm which keeps the optical shutter switched off only in one node for each wavelength. Note that at least one node should switch off its optical shutter otherwise light signal will interfere with itself after traversing around the ring. We find that except for the case of very low traffic, our algorithms are better than the

[1]Notice that in the on mode, light goes through a shutter *without* being first converted to an electrical signal – this is one of the major advantages of the light-trail technology.

[2]If our analysis indicates that some λ wavelengths are needed while only λ_0 are available, then effectively the system will have to be slowed down by a factor λ/λ_0. This is of course only one formulation; there could be other formulations which allow requests to be dropped and analyse what fraction of requests are satisfied.

baseline. For very local traffic, our algorithms are in fact much superior.

The rest of the paper is organized as follows. We begin in Section II by comparing our work with previous related work. In Section III we give the details of our algorithm for the stationary problem. Section IV gives an example instance of the stationary problem where congestion lower bound is weak. We describe our two algorithms for the online problem in Section V. We give results of simulation of our online algorithms in Section VI.

II. Previous Work

Our problem as formulated is in fact similar to the problem of reconfigurable bus architectures [5], [6]. These have been proposed for standard electrical communication; like the optical shutter in light-trails, there is a switch which connects one segment of the bus to another, and which can be set on or off. Again, even in this model, the switches are slow as compared to the data rates on the buses. So from an abstract view points this model is very similar to ours.

While there is much work in the reconfigurable bus literature, it mostly concerns *regular* interconnection patterns, such as those arising in matrix multiplication, list ranking and so on [7], [8], [9], [10]. The only work we know of dealing with random communication patterns is in relation to the PARBUS architecture. Such patterns are handled using standard techniques such as Chernoff bounds [11]. We do not know of any work which discusses how to schedule arbitrary irregular communication patterns in this setting. This is probably understandable because reconfigurable bus architectures have mostly been motivated as special purpose computers, except for the PRAM simulation motivation of PARBUS where the communication becomes random. However, if the network is used for general purpose computing, it does make sense to have algorithms to provision bandwidth for arbitrary irregular patterns, as we do here.

After the light-trail technology was was introduced in [1], much work has been published in the literature. For example, [12] has a mesh implementation of light-trails for general networks. The paper [13] implements a tree-shaped variant of light-trail, called as clustered light-trail, for general networks. The paper [14] describes 'tunable light-trail' in which the hardware at the beginning works just like a simple light-path but can be later tuned to act as light-trail. There is some preliminary work on multi-hop light-trails [15] in which transmissions are allowed to go through a sequence of overlapping light-trails. Survivability in case of failures is considered in [16] by assigning each transmission request to two disjoint light-trails.

Even with this basic hardware implementation, there are different works solving different design problems. Several objectives are mentioned in the seminal paper [17] – to minimize total number of light-trails used, to minimize queuing delay, to maximize network utilization etc. Most of the work in the literature seems to solve the problem by minimizing total number of light-trails used [18], [19], [20], [21]. Though the paper [19] suggests that minimizing total number of light-trails also minimizes total number of wavelengths, it may not be always true. For example, consider a transmission matrix in which $B(1,2) = B(3,4) = 0.5$ and $B(2,3) = 1$. To minimize total number of light-trails used, we create two light-trails on two different wavelengths. Transmission (2,3) is put in one light-trail and transmissions (1,2) and (3,4) are put in the other light-trail. On the other hand, to minimize total number of wavelengths, we put each of them in a separate light-trail on a single wavelength. We believe that minimizing the number of light-trails (while fixing the number of wavelengths) is an appropriate objective for the online case, where this is a measure of the work done by the scheduler. In our opinion, for the stationary problem, the number of wavelengths is a better measure. There are few other models as well, e.g. [22] minimizes total number of transmitters and receivers used in all light-trails.

The general approach followed in the literature to solve the stationary problem is to formulate the problem as an integer linear program (ILP) and then to solve the ILP using standard solvers. The papers [18], [19] give two different ILP formulations. However, solving these ILP formulations takes prohibitive time even with moderate problem size since the problem is NP-hard. To reduce the time to solve the ILP, the paper [20] removed some redundant constraints from the formulation and added some valid-inequalities to reduce the search space. However, the ILP formulation still remains difficult to solve.

Heuristics have also been used. The paper [20] solves the problem in a general network. It first enumerates all possible light-trails of length not exceeding a given limit. Then it creates a list of eligible light-trails for each transmission and a list of eligible transmissions for each light-trail. Transmissions are allocated in an order combining descending order of bandwidth requirement and ascending order of number of eligible light-trails. Among the eligible light-trails for a transmission, the one with higher number of eligible transmissions and higher number of already allocated transmissions is given preference. The paper [21] used another heuristic for the problem in a general network. For a ring network, [19] used three heuristics.

For the problem on a general network, [16] solves two subproblems. The first subproblem considers all possible light-trails on all the available wavelengths as bins and packs the transmissions into compatible bins with the objective of minimizing total number of light-trails used. The second subproblem assigns these light-trails to wavelengths. The first subproblem is solved using three heuristics and the second problem is solved by converting it to a graph coloring problem where each node corresponds to a light-trail and there is an edge between two nodes if the corresponding light-trails conflict with each other.

For the online problem, a number of models are possible. From the point of view of the light-trail scheduler, it is best if transmissions are not moved from one light-trail to another during execution, which is the model we use. It is also appropriate to allow transmissions to be moved, with some penalty. This is the model considered in [19], [23], where the goal is to minimize the penalty, measured as the number of light-trails constructed. The distributions of the transmissions that arrive is also another interesting issue. It is appropriate to assume that the distribution is fixed, as has been considered in many simulation studies including our own. For our theoretical results, however, we assume that the transmission sequence can be arbitrary. The work in [19] assumes that the traffic is an unknown but gradually changing distribution. They use a stochastic optimization based heuristic which is validated using simulations. The paper [20] considers a model in which transmissions arrive but do not depart. Multi-hop problems have also been considered, e.g. [24]. An innovative idea to assign transmissions to light-trails using *online auctions* has been considered in [25].

A. Remarks

As may be seen, there are a number of dimensions along which the work in the literature may be classified: the network configuration, the kind of problem attempted, and the solution approach. Network configurations starting from simple linear array/rings [9], [19], [23] to full structured/unstructured networks [8], [16], [18], [20], [21], [24] have been considered, both in the optical communication literature as well as the reconfigurable bus literature. The stationary problem as well as the dynamic problem has been considered, with additional minor variations in the models. Finally, three solution approaches can be identified. First is the approach in which scheduling is done using exact solutions of Integer Linear Programs [18], [19], [20]. This is useful for very small problems. For larger problems, using the second approach, a variety of heuristics have been used [16], [19], [20], [21]. The evaluation of the scheduling algorithms has been done primarily using simulations. The third approach could be theoretical. However, except for some work related to random communication patterns [11], we see no theoretical analysis of the performance of the scheduling algorithms.

In contrast, our main contribution is theoretical. We give algorithms with *provable* bounds on performance, both for the stationary and the online case. Our work uses the competitive analysis approach [2] for the online problem. We use techniques of approximation algorithms to solve the stationary problem. To our knowledge, this competitive analysis and approximation algorithm approach to solve the light-trail scheduling problem has not been used in the literature. We also give simulation results for the online algorithms.

III. The Stationary Problem

In this section, instead of considering two unidirectional rings, we consider a linear array of n nodes, numbered 0 to $n-1$. Communication is considered undirected. This simplifies the discussion; it should be immediately obvious that all results directly carry over to two directed rings mentioned in the introduction.

The input is a matrix B with $B(i,j)$ denoting the bandwidth requirement for the transmission from node i to node j, without loss of generality, as a fraction of the bandwidth of a single wavelength. The goal is to schedule these in minimum number of wavelengths w. The output must give w as well as a partitioning of each wavelength into a set of segments. The partitioning may be specified as an increasing sequence of numbers (what we refer to as *configuration*) between 0 and $n-1$; if u appears in the sequence it indicates that the shutter in node u is off, otherwise the shutter in node u is on. The segment between two off shutters is a light-trail. A transmission from i to j can be assigned to a light-trail L only if $u \leq i, j \leq v$ where u, v are the endpoints of the light-trail. Further the sum of the required bandwidths assigned to any single light-trail must not exceed 1.

It is customary to consider two variations: non-splittable, in which a transmission must be assigned to a single light-trail, and splittable, in which a transmission can be split into two or more transmissions by dividing up the bandwidth, and the resultant transmissions can be assigned to different light-trails. Our results hold for both variations.

We will use $c_l(S)$ to denote the congestion induced on a link l by a set S of transmissions. This is simply the total bandwidth requirement of those transmissions from S requiring to cross link l. Clearly $\max_l c_l(S)$, the maximum congestion over all links, is a lower bound on the number of wavelengths needed. We use $c(S)$ to denote $\max_l c_l(S)$. Finally if t is a transmission, then we abuse notation to write $c_l(t), c(t)$, instead of $c_l(\{t\}), c(\{t\})$, for the congestion contributed by t only, which is equal to the bandwidth requirement of t.

The key observation behind our algorithm for the stationary problem is: if all transmissions go the same distance in the network, then it is easy to get a nearly optimal schedule. Thus we partition the transmissions into classes based on the length of the transmission. We then stitch back the separate schedules.

Define the length of a transmission to be the distance between the origin and the destination. Transmissions with length between 2^{i-1} (non-inclusive) and 2^i (inclusive) are said to belong to the ith *class* where $0 \leq i \leq \lceil \log_2(n-1) \rceil$.

Let R denote the set of all transmissions, and R_i the set of transmissions in class i. Class 0 is served simply by putting shutters off at every node. Clearly, $\lceil c(R_0) \rceil$ wavelengths will suffice for the splittable case, and twice that many for the non-splittable (using ideas from bin-packing [26]). For R_1 also it is easily seen that $O(\lceil c(R_1) \rceil)$ wavelengths will suffice. So for the rest of this paper we only consider classes 2 and larger.

A. Schedule Transmissions of Class i

Our aim is to partition R_i further into sets $S_0, S_1, \ldots$ each with congestion at most some constant value so that overall it does not take many wavelengths. We start with $T_0 = R_i$, and in general given T_j we construct S_j and $T_{j+1} = T_j \setminus S_j$ as follows:

We add transmissions greedily into S_j starting from the leftmost link l moving right, i.e. for each l pick transmissions crossing it and move them into S_j until we have removed at least unit congestion from $c_l(T_j)$ or reduced $c_l(T_j)$ to 0. Then we move to the next link. So, at the end the following condition holds:

$$\forall l,\ c_l(S_j) \begin{cases} = c_l(T_j) & \text{if } c_l(T_j) \leq 1, \text{ and} \\ \geq 1 & \text{otherwise.} \end{cases} \tag{1}$$

However, to make sure that $c(S_j)$ is not large, we move back transmissions from S_j, in the reverse order as they were added, into T_j so long as condition (1) remains satisfied. Now we claim the following:

Lemma 1. *Construction of S_j, T_{j+1} from T_j takes polynomial time and $c(S_j) < 4$.*

Proof: For the first part, it can be seen that the construction takes at most $n|T_j|$ time in the pick-up step and also in the move-back step.

For the second part, at the end of move-back step, for any transmission $t \in S_j$ there must exist a link l such that $c_l(S_j) < 1 + c(t)$ otherwise t would have been removed. We call l as a *sweet spot* for t. Since $c(t) \leq 1$ we have $c_l(S_j) < 2$ for any sweet spot l.

Now consider any link x. Of the transmissions through x, let L_1 (L_2) denote transmissions having their sweet spot on the left (right) of x. Consider y, the rightmost of these sweet spots of some transmission $t \in L_1$. Note first that $c_y(S_j) < 2$. Also all transmissions in L_1 pass through both x, y. Thus $c_x(L_1) = c(L_1) = c_y(L_1) \leq c_y(S_j) < 2$. Similarly, $c_x(L_2) < 2$. Thus $c_x(S_j) = c_x(L_1) + c_x(L_2) < 4$. But since this applies to all links x, $c(S_j) < 4$. ■

Next we show that not too many S_j will be constructed.

Lemma 2. *Suppose S_j is created for class i. Then $j \leq c(R_i)$.*

Proof: Suppose S_j contains a transmission that uses some link l. The construction process above must have removed at least unit congestion from l in every previous step 0 through $j-1$. Hence $j \leq c_l(R_i) \leq c(R_i)$. ■

Every transmission in S_j has length at least $2^{i-1}+1$, and must cross some node whose number is a multiple of 2^{i-1}. The smallest numbered such node is called the *anchor* of the transmission. The *trail-point* of a transmission is the right most node numbered with a multiple of 2^{i-1} that is on the left of the anchor. If the transmission has trail-point at some node with number of the form $t2^{i-1}$, then we define $t \bmod 4$ as its *phase*.

Lemma 3. *The set S_j can be scheduled using $O(1)$ wavelengths.*

Proof: We partition S_j further into sets S_j^p containing transmissions of phase p. Note that the transmissions in any S_j^p either overlap at their anchors, or do not overlap at all. This is because if two transmissions in S_j^p have different anchors, then these two anchors are at least 2^{i+1} distance apart. Since the length of transmission is at most 2^i, the two transmissions can not intersect.

So for the set S_j^p, consider 4 wavelengths, each having shutters off at nodes numbered $(4q+p)2^{i-1}$. Clearly, for the splittable case, the transmissions will be accommodated in these wavelengths, since $c(S_j^p) < 4$. For the non-splittable case, 8 wavelengths will suffice, using standard bin packing ideas [26].

Thus all of S_j can be accommodated in at most 16 wavelengths for the splittable case, and at most 32 wavelengths for the non-splittable case. ■

Theorem 4. *The entire set R_i can be scheduled such that at each link x there are $O(C_x(R_i)+1)$ light-trails.*

Proof: We first consider the light-trails as constructed in Lemma 3. In that construction, it is possible that some light-trails contain links that are not used by any of the transmissions associated with the light-trail. In such cases we shrink the light-trails by removing the unused links (which can only be near either end of the light-trail because all transmissions assigned to a light-trail overlap at their anchor).

Let j be largest such that x has a transmission from S_j. Then we know that $c_x(R_i) \geq j$. For each $k = 0, 1, \ldots, j$ we have $O(1)$ light-rails at x as described above. Thus we have a total of $O(j+1) = O(c_x(R_i)+1)$ light-trails at x. ■

B. Merge Light-trails of All Classes Together

If we simply collect together the wavelengths as allocated above, we would get a bound $O(c \log n)$. Note however, that if two transmissions, one in class i and the other in class j, are spatially disjoint, then they could possibly share the same wavelength. Given below is a systematic way of doing this, which gets us the sharper bound.

We know that at each link l there are a total of $O(c_l(R_i)+1)$ light-trails. Thus the total number of light-trails at l are $O(c_l(R) + \log n)$, summing over all classes.

Think of each light-trail as an interval, giving us a collection of intervals such that any link l has at most $O(c_l(R)+\log n) = O(c+\log n)$ intervals. Now this collection of intervals can be colored using $O(c+\log n)$ colors. So we put all the intervals of the same color in the same wavelength.

IV. On the Congestion Lower Bound

We now consider an instance of the stationary problem. For convenience, we assume $m = n-1 = 2^k$ for some k, and all logarithms with base 2. All the transmissions have same bandwidth requirement $\alpha = 1/(\log m + 1)$.

First, we have a transmission going from 0 to 2^k. Then a transmission from 0 to 2^{k-1} and a transmission from 2^{k-1} to 2^k. Then 4 spanning one-fourth the distance, and so on. Thus we have transmissions of $\log m + 1$ classes, each class having transmissions of same length. In class $i \in \{0, 1, \ldots, \log m\}$ there are 2^i transmissions $B(s_{ij}, d_{ij}) = \alpha$ where $s_{ij} = jm/2^i, d_{ij} = (j+1)m/2^i$ and $j = 0, 1, \ldots, 2^i - 1$. All other entries of B are 0. This is illustrated in Fig. 1 for $n = 17$.

Clearly the congestion of this pattern is uniformly 1. Consider an optimal solution. There has to be a light-trail in which the first transmission from node 0 to m is scheduled. Thus we must have a wavelength with no off shutters except at node 0 and node m. In this wavelength, it is easily seen

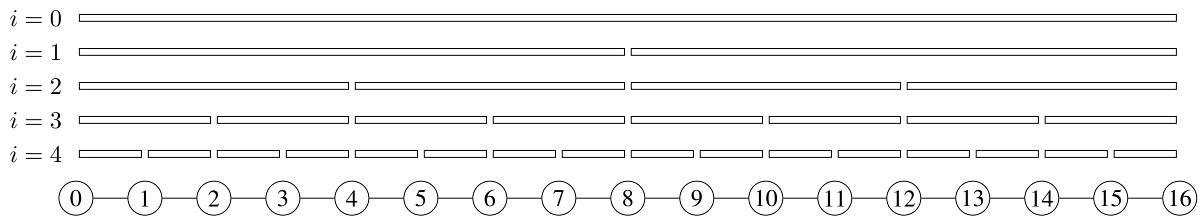


Fig. 1. An example instance where congestion bound is weak

that the longest transmissions should be scheduled. So we start assigning transmissions of first few classes in this light-trail. Suppose, all the transmissions for first l classes are assigned. Then we have total $1 + 2 + 4 + \cdots + 2^l = 2^{l+1} - 1$ transmissions assigned to this light-trail. Total bandwidth requirement of these transmissions should be less than 1. This gives us $(2^{l+1} - 1)(1/(\log m + 1)) \leq 1$ implying $l \leq \log(\log m + 2) - 1 \approx \log\log m$.

For the subsequent classes of transmissions, we allocate a new wavelength and create light-trails by putting shutters off at nodes numbered multiples of $m/2^{l+1}$. It can be seen that again transmissions of next about $\log\log m$ classes can be put in these light-trails. We repeat this process until all transmissions are assigned.

In each wavelength we assign transmissions of $\log\log m$ classes. There are total $(1 + \log m)$ classes. Thus the total number of wavelengths needed is $\lceil(1 + \log m)/\log\log m\rceil = O(\log n/\log\log n)$ rather than the congestion bound of 1. For the example in Fig. 1, using this procedure, we have $\log\log m = 2$. Thus we require $\lceil(1 + \log m)/\log\log m\rceil = 3$ wavelengths. The first wavelength is used for the transmissions of classes {0,1}, the second wavelength for classes {2,3} and the third for class 4.

V. The Online Problem

In the online case, the transmissions arrive dynamically. An arrival event has parameters (s_i, d_i, r_i) respectively giving the origin, destination, and bandwidth requirement of an arriving transmission request. The algorithm must assign such a transmission to a light-trail L such that s_i, d_i belong to the light-trail, and at any time the total bandwidth requirement of transmissions assigned to any light-trail is at most 1. A departure event marks the completion of a previously scheduled transmission. The corresponding bandwidth is released and becomes available for future transmissions. The algorithm must make assignments without knowing about subsequent events.

Unlike the stationary problem, congestion at any link may change over time. Let $c_{lt}(S)$ denote the congestion induced on a link l at time t by a set S of transmissions. This is simply the total bandwidth requirement of those transmissions from S requiring to cross link l at time t. The congestion bound $c(S)$ is $\max_l \max_t c_{lt}(S)$, the maximum congestion over all links over all time instants.

For the online problem, we present two algorithms, SeparateClass and AllClass having competitive ratios $O(\log n)$ and $O(\log^2 n)$ respectively. They are inspired by the analysis of the algorithm for the snapshot problem, as may be seen.

In both the online algorithms, when a transmission request arrives, we first determine its class i and trail-point x (defined in Section III-A). The transmission is allocated to some light-trail with end nodes x and $x + 2^{i+1}$. However, the algorithms differ in the way a light-trail is chosen from some candidate light-trails.

A. *Algorithm* SeparateClass

In this algorithm, every allocated wavelength is assigned a class label i and a phase label p, and has shutters off at nodes $(4q + p)2^{i-1}$ for all q, i.e. is configured to serve only transmissions of that class and phase. Whenever a transmission of class i and phase p is to be served, it is only served by a wavelength with the same labels. If such a wavelength is found, and light-trail starting at its trail-point has space, then the transmission is assigned to that light-trail. If no such wavelength is found, then a new wavelength is allocated and labeled and configured as above.

When a transmission finishes, it is removed from its associated light-trail. The wavelength can be relabeled only when there are no transmissions in any of its light-trail.

Lemma 5. *Suppose, at some point of time, among the wavelengths allocated by the algorithm, x wavelengths had non-empty light-trails of the same class and phase across a link l. Then l must have congestion $\Omega(x)$ at some instant.*

Proof: Number these wavelengths in the order that they got allocated. Suppose the xth one was allocated due to a transmission t. This could only happen because t could not fit in the first $x - 1$ wavelengths.

For the splittable case this can only happen if the previous $x - 1$ wavelengths contain congestion at least $x - 1 - c(t)$ at the anchor of t, when t arrived. But this is $\Omega(x)$ giving us the result.

For the non-splittable case, suppose that $c(t) \leq 0.5$. Then each of the first $x - 1$ light-trails must have congestion of least 0.5 when t arrived, giving congestion $\Omega(x)$. So suppose $c(t) > 0.5$. Let k be the largest such that wavelength k contains a transmission t' with $c(t') \leq 0.5$. If no such k exists, then clearly the congestion is $\Omega(x)$. If k exists, then all the wavelengths higher than k have congestion at least 0.5 when t arrived. And the wavelengths lower than k had congestion at least 0.5 when t' arrived. So at one of the two time instants the congestion must have been $\Omega(x)$. ∎

Theorem 6. SEPARATECLASS *is* $O(\log n)$ *competitive.*

Proof: Suppose that SEPARATECLASS uses w wavelengths. We will show that the best possible algorithm (including off-line algorithms) must use at least $\Omega(w/\log n)$ wavelengths.

Consider the time at which the wth wavelength was allocated. At this time $w-1$ wavelengths are already in use, and of these $w' = (w-1)/4\log n$ must have the same class and phase. Among these w' wavelengths consider the one which was allocated last to accommodate some light-trail L serving some newly arrived transmission. At that time, each of the previously allocated $w'-1$ wavelengths was nonempty in the extent of L. By Lemma 5, $c(B) = \Omega(((w-1)/4\log n)-1) = \Omega(w/\log n)$. This is a lower bound on any algorithm, even off-line. ∎

B. *Algorithm* ALLCLASS

This is a simplification of the previous algorithm in that allocated wavelengths are not labeled. When a transmission arrives, if a light-trail of its class and phase capable of including it is found, then the transmission is assigned to it. If no such light-trail is found, then an attempt is made to create such a light-trail from the unused portions of any of the existing wavelengths. If such a light-trail can be created, then it is created and the transmission is placed in it. Otherwise a new wavelength is allocated, the required light-trail is created, and the rest of the wavelength is considered unused.

When a transmission finishes, it is removed from its associated light-trail. If this makes the light-trail empty then we consider it as unused. Then we check if there are adjacent unused light-trails for the same wavelength. If so, we merge them by turning on the off shutter between them.

Theorem 7. ALLCLASS *is* $O(\log^2 n)$ *competitive.*

Proof: Suppose ALLCLASS uses w wavelengths. We will show that an optimal algorithm will use at least $\Omega(w/\log^2 n)$. Clearly, we may assume $w = \Omega(\log^2 n)$.

We first prove that there must exist a point of time in the execution of ALLCLASS when there are $w/4\log n$ light-trails crossing the same link.

Number the wavelengths in the order of allocation. Consider the transmission t for which the wth wavelength was allocated for the first time. Let L be the light-trail used for t. Clearly, the wth wavelength had to be allocated because the other wavelengths contained light-trails overlapping with L. Of these if at least $w/4\log n$ light-trails crossed either end of L, then we are done. If this fails, there must be at least $w' = w - 1 - w/2\log n$ wavelengths that have light-trails which are strictly contained inside the extent of L. Let L' be the light-trail allocated on the w'th of these wavelengths. Note that L' is strictly smaller than L. Thus we can repeat the above argument by using L' and w' in place of L and w respectively, only $\log n$ times, and if we fail each time, we will end up with a light-trail L'' such that there are at least w'' wavelengths with light-trails conflicting with L'', where $w'' = w - \log n - \log n(w/2\log n) = w/2 - \log n \geq w/4\log n$ for $w = \Omega(\log^2 n)$. But L'' is a single link and so we are done.

Of these $w/4\log n$ light-trails, at least $w/16\log^2 n$ must have the same class and phase. But Lemma 5 applies, and hence there is a link having congestion at least $w/16\log^2 n$. But this is a lower bound on the number of wavelengths required by any algorithm, including an offline algorithm. ∎

VI. SIMULATIONS

We simulate our two online algorithms and a baseline algorithm on a pair of oppositely directed rings, with nodes numbered 0 through $n-1$ clockwise.

We use slightly simplified versions of the algorithms described in Section V (but easily seen to have the same bounds): basically we only use phases 0 and 2. Any transmissions that would go into class i phase 1 (or phase 3) light-trail are contained in some class $i+1$ light-trail (of phase 0 or 2 only), and are put there. We define a class i and phase 0 light-trail to be one that is created by putting off shutters at nodes $jn/2^i$ for different j, suitably rounding when n is not a power of 2. A light-trail with class i and phase 2 is created by putting off shutters at nodes $(jn/2^i + n/2^{i+1})$, again rounding suitably. For ALLCLASS, there is a similar simplification. Basically, we use light-trails having end nodes at $jn/2^i$ and $(j+1)n/2^i$ or at $jn/2^i + n/2^{i+1}$ and $(j+1)n/2^i + n/2^{i+1}$. As before, in SEPARATECLASS, we require any wavelength to contain light-trails of only one class and phase; whereas in ALLCLASS, a wavelength may contain light-trails of different classes and phases.

For the baseline algorithm in each ring we use a single off shutter at node 0. Transmissions from lower numbered nodes to higher numbered nodes use the clockwise ring, and the others, the counterclockwise ring.

A. *The simulation experiment*

A single simulation experiment consists of running the algorithms on a certain load, characterized by parameters λ, D, r_{min} and α for 100 time steps. In our results, each datapoint reported is the average of 150 simulation experiments with the same load parameters.

In each time step, all nodes j that are not busy transmitting, generate a transmission (j, d_j, r_j) active for t_j time units. After that the node is busy for t_j steps. After that it generates another transmission as before. The transmission duration t_j is drawn from a Poisson distribution with parameter λ. The destination d_j of a transmission is picked using the distribution D discussed later. The bandwidth is drawn from a modified Pareto distribution with scale parameter $= 100 \times r_{min}$ and shape parameter $= \alpha$. The modification is that if the generated bandwidth requirement exceeds the wavelength capacity 1, it is capped at 1.

We experimented with $\alpha = \{1.5, 2, 3\}$ and $\lambda = \{0.01, 0.1\}$ but report results for only $\alpha = 1.5$ and $\lambda = 0.01$; results for other values are similar. We tried four values $0.01, 0.1, 0.25$

and 0.5 for r_{min}. Here we report the results for $r_{min} = 0.01, 0.5$. We considered four different distributions D for selecting the destination node of a transmission. 1) *Uniform*: we select a destination uniformly randomly from the $n-1$ nodes other than the source node. 2) *UniformClass*: we first choose a class uniformly from the $\lceil \log n/2 \rceil + 1$ possible classes. It should be noted that there can be a destination at a distance at most $n/2$ in any direction since we schedule along the direction requiring shortest path. 3) *Bimodal*: first we randomly choose one of two possible modes. In mode 1, a destination from the two immediate neighbors is selected and in mode 2, a destination from the nodes other than the two immediate nodes is chosen uniformly. For applications where transmissions are generated by structured algorithms, local traffic, i.e. unit or short distances (e.g. $\sqrt{n}$ for mesh like communications) would dominate. Here, for simplicity, we create a bimodal traffic which is mixture of completely local and completely global. 4) *ShortPreferred*: we select destinations at shorter distance with higher probability. In fact, we first choose a class i in the range $0, \ldots, \lceil \log n/2 \rceil$ with probability $\frac{1}{2^{i+1}}$ and then select a destination uniformly from the possible destinations in that class. We report the results only for the distributions *Uniform* and *Bimodal*.

B. Results

Fig. 2 shows the results for the 4 load scenarios. For each scenario, we report the number of wavelengths required by the 3 algorithms and the measured congestion as defined in Section V. Each data-point is the average of 150 simulations (each of 100 time steps) for the same parameters on rings having $n = 5, 6, \ldots, 20$ nodes. We say that the two scenarios corresponding to $r_{min} = 0.01$ have *low load* and the remaining two scenarios ($r_{min} = 0.5$) have *high load*.

For low load, the baseline algorithm outperforms our algorithms. At this level of traffic, it does not make sense to reserve different light-trails for different classes. However, as load increases our algorithms outperform the baseline algorithm.

For the same load, it is also seen that our algorithms become more effective as we change from the completely global *Uniform* distribution to the more local *Bimodal* distribution. This trend was also seen with the other distributions we experimented with.

Though we could not show analytically that ALLCLASS is better than SEPARATECLASS always, our simulation shows that ALLCLASS performs better. It may be noted that our algorithm for the stationary case mixes up the light-trails of different classes, and so suggests that the ALLCLASS might work better.

VII. Conclusions and Future Work

It can be shown that the non-splittable stationary problem is NP-hard, using a simple reduction from bin-packing. We do not know if the splittable problem is also NP-hard. We gave an algorithm for both variations of the stationary problem which takes $O(c + \log n)$ wavelengths. It will also be useful to improve the lower bound arguments; as Section IV shows, congestion is not always a good lower bound. This may lead to a constant factor approximation algorithm for the problem.

In the online case we gave two algorithms which we prove to have competitive ratios $O(\log n)$ and $O(\log^2 n)$ respectively. In practice we found that the second algorithm was better, and showing this analytically is an important open problem.

Our online model is very conservative in the sense that once a transmission is allocated on a light-trail, the light-trail cannot be modified. However, other models allow light-trails to shrink/grow dynamically [17]. It will be useful to incorporate this (with some suitable penalty, perhaps) into our model.

It will also be interesting to devise special algorithms that work well given the distribution of arrivals.

Acknowledgment

We would like to thank Ashwin Gumaste for encouragement, insightful discussions and patient clearing of doubts related to light-trails.

References

[1] I. Chlamtac and A. Gumaste, "Light-trails: A solution to IP centric communication in the optical domain," *Lecture notes in computer science*, pp. 634–644, 2003.

[2] A. Borodin and R. El-Yaniv, *Online computation and competitive analysis*. Cambridge University Press, New York, NY, USA, 1998.

[3] D. Sleator and R. Tarjan, "Amortized efficiency of list update and paging rules," *Communications of the ACM*, vol. 28, no. 2, pp. 202–208, 1985.

[4] A. Karlin, M. Manasse, L. Rudolph, and D. Sleator, "Competitive snoopy caching," *Algorithmica*, vol. 3, no. 1, pp. 79–119, 1988.

[5] R. Wankar and R. Akerkar, "Reconfigurable architectures and algorithms: A research survey," *IJCSA*, vol. 6, no. 1, pp. 108–123, 2009.

[6] K. Bondalapati and V. Prasanna, "Reconfigurable meshes: Theory and practice," in *Fourth Workshop on Reconfigurable Architectures, IPPS*, 1997.

[7] K. Li, Y. Pan, and S. Zheng, "Parallel matrix computations using a reconfigurable pipelined optical bus," *Journal of Parallel and Distributed Computing*, vol. 59, no. 1, pp. 13–30, 1999.

[8] C. Subbaraman, J. Trahan, and R. Vaidyanathan, "List ranking and graph algorithms on the reconfigurable multiple bus machine," in *Parallel Processing, 1993. ICPP 1993. International Conference on*, vol. 3, 1993.

[9] Y. Pan, M. Hamdi, and K. Li, "Efficient and scalable quicksort on a linear array with a reconfigurable pipelined bus system," *Future Generation Computer Systems*, vol. 13, no. 6, pp. 501–513, 1998.

[10] Y. Wang, "An efficient O (1) time 3D all nearest neighbor algorithm from image processing perspective," *Journal of Parallel and Distributed Computing*, vol. 67, no. 10, pp. 1082–1091, 2007.

[11] S. Rajasekaran and S. Sahni, "Sorting, selection, and routing on the array with reconfigurable optical buses," *IEEE Transactions on Parallel and Distributed Systems*, vol. 8, no. 11, pp. 1123–1132, 1997.

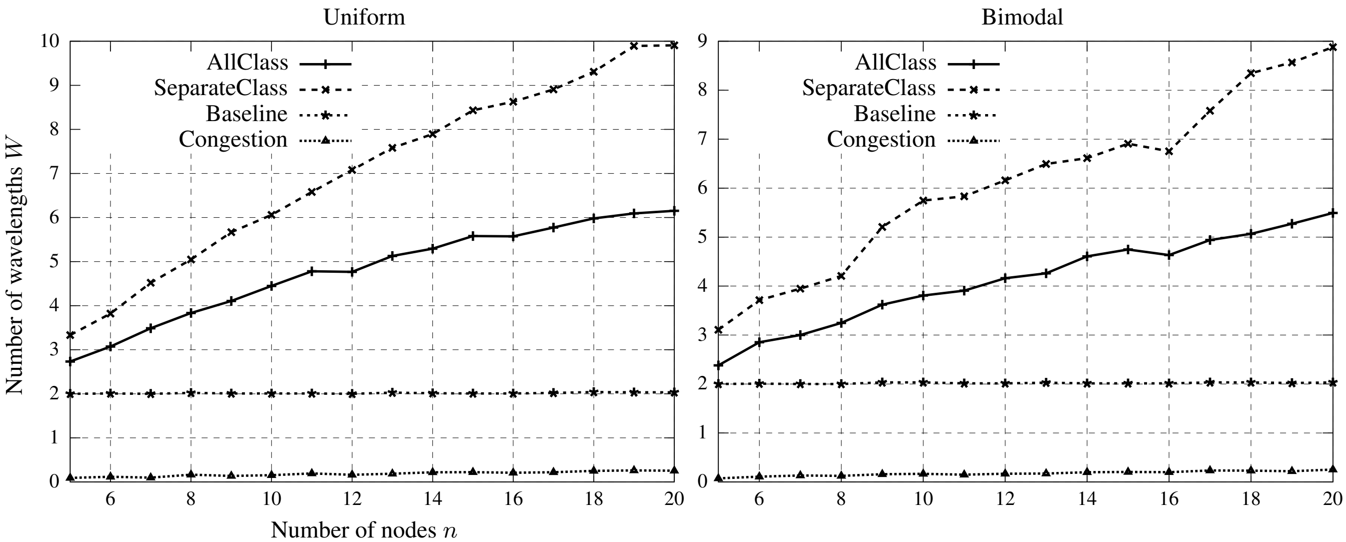
[12] A. Gumaste and I. Chlamtac, "Mesh implementation of light-trails: a solution to IP centric communication," *Proceedings of the 12th International Conference on Computer Communications and Networks, ICCCN'03*, pp. 178–183, 2003.

[13] A. Gumaste, G. Kuper, and I. Chlamtac, "Optimizing light-trail assignment to WDM networks for dynamic IP centric traffic," in *The 13th IEEE Workshop on Local and Metropolitan Area Networks, LANMAN'04*, 2004, pp. 113–118.

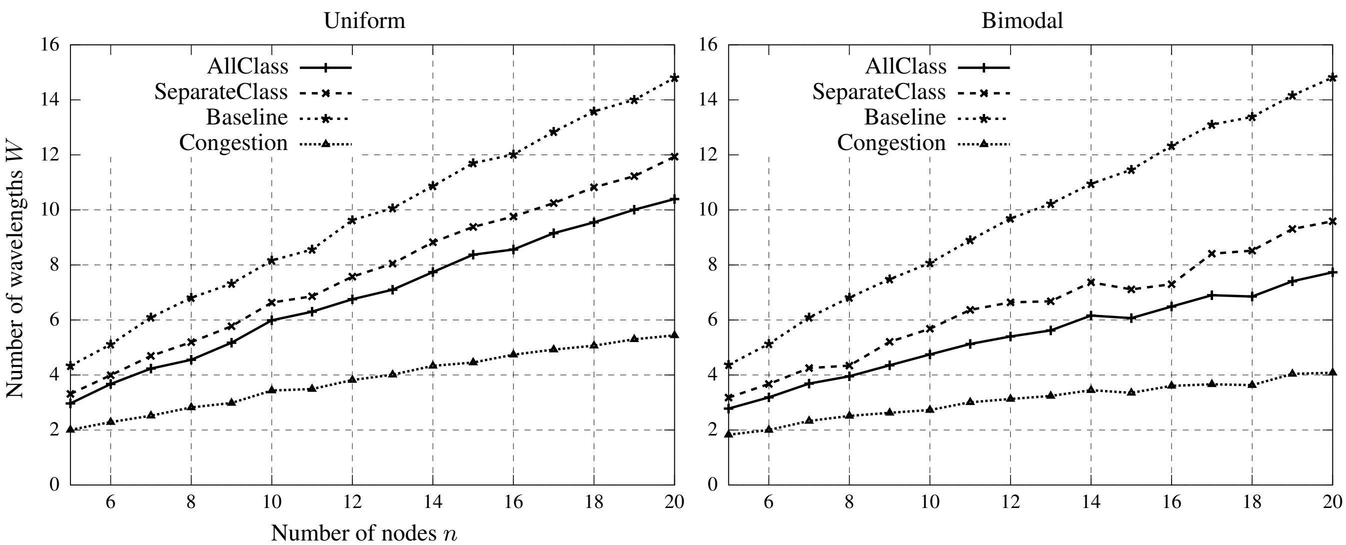
[14] Y. Ye, H. Woesner, R. Grasso, T. Chen, and I. Chlamtac, "Traffic grooming in light trail networks," in *IEEE Global Telecommunications Conference, GLOBECOM'05*, 2005.

[15] A. Gumaste, J. Wang, A. Karandikar, and N. Ghani, "MultiHop Light-Trails (MLT) - A Solution to Extended Metro Networks," *Personal Communication*.

[16] S. Balasubramanian, W. He, and A. Somani, "Light-Trail Networks: Design and Survivability," *The 30th IEEE Conference on Local Computer Networks*, pp. 174–181, 2005.



(a) Low Load



(b) High Load

Fig. 2. Simulation results

[17] A. Gumaste and I. Chlamtac, "Light-trails: an optical solution for IP transport [Invited]," *Journal of Optical Networking*, vol. 3, no. 5, pp. 261–281, 2004.

[18] J. Fang, W. He, and A. Somani, "Optimal light trail design in WDM optical networks," in *IEEE International Conference on Communications*, vol. 3, June 2004, pp. 1699–1703.

[19] A. Gumaste and P. Palacharla, "Heuristic and optimal techniques for light-trail assignment in optical ring WDM networks," *Computer Communications*, vol. 30, no. 5, pp. 990–998, 2007.

[20] A. Ayad, K. Elsayed, and S. Ahmed, "Enhanced Optimal and Heuristic Solutions of the Routing Problem in Light Trail Networks," *Workshop on High Performance Switching and Routing, HPSR'07*, pp. 1–6, 2007.

[21] B. Wu and K. Yeung, "OPN03-5: Light-Trail Assignment in WDM Optical Networks," in *IEEE Global Telecommunications Conference, GLOBECOM'06*, 2006, pp. 1–5.

[22] S. Balasubramanian, A. Kamal, and A. Somani, "Network design in IP-centric light-trail networks," in *2nd International Conference on Broadband Networks, IEEE Broadnets'05*, 2005, pp. 41–50.

[23] A. Lodha, A. Gumaste, P. Bafna, and N. Ghani, "Stochastic Optimization of Light-trail WDM Ring Networks using Benders Decomposition," in *Workshop on High Performance Switching and Routing, HPSR'07*, 2007, pp. 1–7.

[24] W. Zhang, G. Xue, J. Tang, and K. Thulasiraman, "Dynamic light trail routing and protection issues in WDM optical networks," in *IEEE Global Telecommunications Conference, GLOBECOM'05*, 2005, pp. 1963–1967.

[25] A. Gumaste and S. Zheng, "Dual auction (and recourse) opportunistic protocol for light-trail network design," in *IFIP International Conference on Wireless and Optical Communications Networks*, 2006, p. 6.

[26] E. G. Coffman, Jr., M. R. Garey, and D. S. Johnson, "Approximation algorithms for bin packing: a survey," *Approximation algorithms for NP-hard problems*, pp. 46–93, 1997.