J. Parallel Distrib. Comput. 72 (2012) 1226–1236

Contents lists available at SciVerse ScienceDirect

J. Parallel Distrib. Comput.

journal homepage: www.elsevier.com/locate/jpdc

Scheduling light-trails on WDM rings☆

Soumitra Pal *, Abhiram Ranade

Department of Computer Science and Engineering, Indian Institute of Technology Bombay, Powai, Mumbai 400076, India

ARTICLE INFO

Article history:
Received 21 June 2011
Received in revised form
24 February 2012
Accepted 29 May 2012
Available online 7 June 2012

Keywords:
Wavelength division multiplexing
WDM
Light-trail
Reconfigurable bus architecture
Scheduling
Routing
Wavelength assignment
Approximation
On-line
Algorithm
Simulation

ABSTRACT

We consider the problem of scheduling communication on optical WDM (wavelength division multiplexing) networks using the light-trails technology. We seek to design scheduling algorithms such that the given transmission requests can be scheduled using a minimum number of wavelengths (optical channels). We provide algorithms and close lower bounds for two versions of the problem on an n processor linear array/ring network. In the *stationary* version, the pattern of transmissions (given) is assumed to not change over time. For this, a simple lower bound is c, the congestion or the maximum total traffic required to pass through any link. We give an algorithm that schedules the transmissions using $O(c + \log n)$ wavelengths. We also show a pattern for which $\Omega(c + \log n/\log\log n)$ wavelengths are needed. In the *on-line* version, the transmissions arrive and depart dynamically, and must be scheduled without upsetting the previously scheduled transmissions. For this case we give an on-line algorithm which has competitive ratio $\Theta(\log n)$. We show that this is optimal in the sense that every on-line algorithm must have competitive ratio $\Omega(\log n)$. We also give an algorithm that appears to do well in simulations (for the classes of traffic we consider), but which has competitive ratio between $\Omega(\log^2 n/\log\log n)$ and $O(\log^2 n)$. We present detailed simulations of both our algorithms.

1. Introduction

Light-trails [6] 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 block the optical signal or let it pass through. By configuring some shutters ON (signal let through) and some OFF (signal blocked), the network can be partitioned into subnetworks, called *light-trails*. At any given time and using a given wavelength, there can be at most one communication in progress in a light-trail. Thus by increasing the number of light-trails, more simultaneous communications are possible, albeit going a shorter distance. How to do this is the central question in the paper. 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.

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 auxiliary network is typically electronic, and the shutter switching time is of the order of milliseconds as opposed to optical signals which have frequencies of several gigahertz. We further note that time division multiplexing may be used inside a single light-trail; in other words, a single light-trail can be used to serve several communication requests, provided the communicating processors lie within the light-trail.

We examine this problem in the *stationary* setting, in which interprocessor communication demands are known and do not change, as well as the *dynamic* setting in which communication requests arrive and depart after being served, in an on-line manner. For both problems, our objective is to minimize the number of wavelengths needed to accommodate the given traffic, using the best possible partitioning of the network into light-trails (for each wavelength), and the best possible assignment of requests to light-trails. Our results are applicable to the setting in which a fixed number of wavelengths is available as follows. 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

☆ A preliminary version of a part of the work was published in the Proceedings of the Seventeenth Annual International Conference on Advanced Computing and Communications (ADCOM) at the Indian Institute of Science (IISc), Bengaluru, 14–17 December 2009.

* Corresponding author.
E-mail addresses: mitra@cse.iitb.ac.in, soumitrakp@gmail.com (S. Pal), ranade@cse.iitb.ac.in (A. Ranade).

0743-7315/$ – see front matter © 2012 Elsevier Inc. All rights reserved.
doi:10.1016/j.jpdc.2012.05.010

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 analyze what fraction of requests are served.

The input to the stationary problem is a matrix B, in which $B(i, j)$ gives the bandwidth demanded between nodes i and j, expressed as a fraction of the bandwidth supported by a single wavelength. We give an algorithm which schedules this traffic using $O(c + \log n)$ wavelengths, where $c = \max_k \sum_{i,j|i\leq k<j} B(i, j)$ is the maximum total bandwidth demand, or the *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 c is small, but which yet require $\Omega(c + \log n/\log\log n)$ wavelengths. In some sense, this justifies the form of our result.

For the on-line problem, we use the notion of competitive analysis [5,1]. In this, an on-line algorithm which must respond without knowledge of the future is evaluated by comparing its performance to that of an *off-line adversary*, an algorithm which is given all the transmission requests at the beginning. Clearly, the off-line adversary must perform at least as well as the best on-line algorithm. We establish that our first algorithm is $\Theta(\log n)$-competitive, i.e., it requires $\Theta(\log n)$ times as many wavelengths as needed by the off-line adversary. We also prove that no on-line algorithm can do better by showing the lower bound on the competitive ratio of any algorithm for the problem to be $\Omega(\log n)$. A multiplicative $\Theta(\log n)$ factor might be considered to be too large to be relevant for practice; however, the experience with on-line algorithm design is that such algorithms often give good hints for designing practical algorithms. We also give a second algorithm for this problem: it is in fact a simplified version of the first. It actually performs better than the first algorithm in many situations; however, we can prove that its competitive ratio is worse, between $\Omega(\log^2 n/\log\log n)$ and $O(\log^2 n)$.

That brings us to our final contribution: we simulate two algorithms based on our on-line 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 the 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 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 2 by comparing our work with previous related work. Section 3 discusses our algorithm for the stationary problem. Section 4 gives an example instance of the stationary problem where the congestion lower bound is weak. In Section 5 we describe our two algorithms for the on-line problem. In Section 6 we show that every on-line algorithm must have competitive ratio $\Omega(\log n)$. In Section 7 we give results of simulation of our on-line algorithms.

2. Previous work

After the light-trail technology was introduced in [6], a variety of hardware models have emerged. For example, [11] has a mesh implementation of light-trails for general networks. The paper [14] implements a tree-shaped variant of light-trails, called as clustered light-trail, for general networks. The paper [28] describes a 'tunable light-trail' in which the hardware at the beginning works just like a simple light-path but can be tuned later to act as a light-trail. There is some preliminary work on multi-hop light-trails [16] in which transmissions are allowed to go through a sequence of overlapping light-trails. Survivability in the case of failures is considered in [3] by assigning each transmission request to two disjoint light-trails.

A variety of performance objectives have been proposed. Several objectives are mentioned in the seminal paper [12] – to minimize the 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 the total number of light-trails used [9,15,2,27]. Though the paper [15] suggests that minimizing the total number of light-trails also minimizes the total number of wavelengths, it may not always be true. For example, consider a transmission matrix in which $B(1, 2) = B(3, 4) = 0.5$ and $B(2, 3) = 1$. To minimize the total number of light-trails used, we create two light-trails on two different wavelengths. Both light-trails extend all the way from 1 to 4. 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 the total number of wavelengths, we put each of them in a separate light-trail, and the three light-trails are created on a single wavelength. We believe that minimizing the number of light-trails is motivated by the goal of minimizing the book-keeping and the scheduler overhead. However, we do not think this can be more important than reducing the number of wavelengths needed (or reducing the slowdown the system will face if the number of wavelengths is fixed). There are a few other models as well, e.g. [4] minimizes the 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 [9,15] give two different ILP formulations. However, solving these ILP formulations takes prohibitive time even with moderate problem size since the problem is NP-hard. As a result a number of heuristics have been proposed and evaluated experimentally [3,2,15,27,21,10].

For the on-line 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 [15,20], where the goal is to minimize the penalty, measured as the number of light-trails constructed. The distributions of the transmissions that arrive are 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 [15] assumes that the traffic is an unknown but gradually changing distribution. It uses a stochastic optimization based heuristic which is validated using simulations. The paper [2] considers a model in which transmissions arrive but do not depart. Multi-hop problems have also been considered, e.g. [29]. An innovative idea to assign transmissions to light-trails using *on-line auctions* has been considered in [17]. The paper [13] gives a two-stage scheduling algorithm using heuristics based on the utility of each light-trail and estimates performance of the algorithm in terms of average delay and number of required light-trails by modeling a Markov chain.

Our problem as formulated is also similar to the problem of scheduling communications on reconfigurable bus architectures [8,26]. A reconfigurable bus architecture is modeled as a graph in which processors are vertices and edges are communication links; however, a processor can choose to electrically connect (or keep separate) the communication links incident to it. If links are connected together (like setting the shutter ON), the communication goes through (as well as being read by the processor). In this way the entire network can be made to behave like a few long or many short buses, as per the needs of the application running on the network.

At an abstract level, the reconfigurable bus system is similar to our light-trail model, as both models use controllable switches to dynamically reconfigure a bus into multiple subbuses. In both models, changing the state of the switch takes very long as compared to the data rates on the buses. However, typically, reconfigurable bus systems have only one bus, rather than allowing multiple wavelengths like the light-trail model. A second difference is in the context in which the two models have been studied. The light-trail model has been studied more by the optical network community, and the focus has been how to schedule relatively long duration communication requests (connection based) without having any graphical regularity. Reconfigurable bus systems have been studied more in the context of parallel computing, and the analyses have been more of entire algorithms running on them. These typically concern short messages and the communication patterns are often regular, as in matrix multiplication [18], problems on graphs [25], and sorting [23]. PRAM simulation on a reconfigurable bus [19], particularly in the case of randomized assignment of shared memory cells, generates random communication patterns. However, because these patterns are drawn from a uniform distribution, they end up being quite regular (and much of the analysis is to find regular patterns that are supersets of what is required). So even this work does not consider truly arbitrary/irregular patterns which are our prime interest, for the on-line as well as off-line (stationary) scenarios.[1]

2.1. Remarks

As may be seen, there is a fair amount of work in the literature on the stationary problem as well as the on-line problem. However, except for some work related to random communication patterns [19,24], we see no theoretical (polynomial time) analysis of the performance of the scheduling algorithms.

In contrast, we give polynomial time algorithms with *provable* bounds on performance, both for the stationary and the on-line case. Our work uses the competitive analysis approach [5,1] for the on-line 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 problems has not been used in the literature. We also give simulation results for the on-line algorithms.

3. The stationary problem

In this section, instead of considering two unidirectional rings, we consider a linear array of n nodes, numbered 0 to $n-1$. The link between the two consecutive nodes i and $i+1$ is numbered i. 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.

In WDM, the physical optic fiber carrying signals of k different wavelengths is logically thought of as k independent parallel fibers each carrying signals of a single wavelength. Each node can be thought of as having a separate shutter on each of the k fibers. Each shutter can be set ON, meaning it allows the optical signal to pass, or OFF, meaning it does not. The segment between two OFF shutters is a light-trail. A transmission request from node i to node j can be assigned to a light-trail if the following conditions are met:

1. $u \le i < j \le v$ where u, v are the OFF nodes of the light-trail.
2. The sum of the bandwidth requirements of all requests assigned to any single light-trail does not exceed the capacity of a wavelength.

The requests assigned to a light-trail are served by time division multiplexing, with service duration proportional to the bandwidth requirement. Thus at any time instant the light-trail is used by at most one request.

The input for the stationary problem is a matrix B with $B(i,j)$ denoting the bandwidth requirement for the transmission request from node i to node j, as a fraction of the bandwidth capacity of a single wavelength which we define to be 1 without loss of generality. The goal is to schedule these in a minimum number of wavelengths w. The output must give w as well as the light-trails used on each wavelength and the mapping of each transmission to a light-trail that serves it.

It will be convenient to represent/visualize schedules geometrically. We will use the x axis to represent our processor array, with processors at integer points and the y axis to represent the wavelengths numbered 0, 1, 2, and so on. The region bordered by $y=k$ and $y=k+1$ will be used to depict the transmissions assigned to the wavelength numbered k. The region will be partitioned with vertical lines at the nodes where the shutters are OFF. Each of the rectangular parts in the partition represents a light-trail created on the corresponding wavelength. A transmission from node i to node j having bandwidth requirement b will be denoted as $[i,j]$ and represented as a rectangle of height b located horizontally in the region between $x=i$ and $x=j$ and vertically within the region corresponding to the light-trail in which it is scheduled. We will also use the terms *length*, *extent*, and *height* of transmission $[i,j]$ to mean $j-i$, the interval $[i,j]$, and b respectively. Unless there is ambiguity in the context we will also use $[i,j]$ to denote a light-trail with end nodes i and j. Similarly we will also use the terms *length*, and *extent* of light-trail $[i,j]$ to mean $j-i$ and the interval $[i,j]$ respectively.

As an example consider a network with 3 nodes, 0, 1, 2 and a transmission matrix B in which $B(0,1)=B(1,2)=0.6$ and $B(0,2)=0.4$ are the only non-zero entries. In order to enable the transmission [0, 2], we must have a wavelength with a light-trail which goes all the way from 0 to 2. In this light-trail, we cannot put both the remaining communications, because then the bandwidth would become $0.6+0.6+0.4=1.6$, i.e., larger than 1. Say we put the transmission [0, 1] in this light-trail. Then the remaining communication, [1, 2], would require its own light-trail, and for that we will need another wavelength. This is shown in Fig. 1(a). Another way is as follows. We put transmission [0, 2] in a single light-trail extending from 0 to 2. Then on another wavelength, we create two light-trails, with the shutter at 1 in the OFF position. The transmissions [0, 1] and [1, 2] can now be placed in these respective light-trails. This solution is given in Fig. 1(b). Both the solutions require 2 wavelengths, and they are optimal because the required communication cannot be implemented using just 1 wavelength.

It is customary to consider two problem 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 requirement, and each of them can be assigned to a different light-trail. Note that when a transmission is split into multiple transmissions, the length and extent remain the same, only the height is divided. Our results hold for both variations.

[1] It is interesting to note that the communication patterns for PRAM simulation are uniformly random across the network because the PRAM address space is *hashed*, i.e., distributed randomly. Hashing has the effect of converting possibly local communication going a short distance to a random communication which most likely goes a long distance. Such a strategy is inherently wasteful in utilization of bandwidth. It seems much better to directly deal with the arbitrary communication pattern which arises in PRAM simulation in the first place.

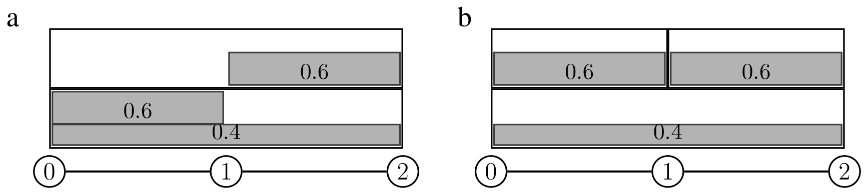


Fig. 1. Solutions to the stationary problem example.

Note that the Bin Packing problem which is NP-hard, is a special case of the stationary problem where each item corresponds to a transmission from node 0 to node $n-1$ and each bin corresponds to a light-trail (and to a wavelength too because each light-trail completely occupies a wavelength). Thus the non-splittable stationary problem is NP-hard. We do not know whether the splittable problem is also NP-hard.

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 $c(S) = \max_l c_l(S)$, the maximum congestion over all links, is a lower bound on the number of wavelengths needed. 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. Let R be the set of all transmissions of an instance of the stationary problem. Let m be the size of R. We will use c to denote the overall congestion $c(R)$.

3.1. Algorithm overview

Getting an algorithm which requires only $O(c \log n)$ wavelengths is easy. If $c_{n/2}$ denotes the congestion of the link between node $n/2$ and node $n/2+1$, then the transmissions crossing this link can be scheduled in $\lceil c \rceil \geq \lceil c_{n/2} \rceil$ wavelengths for the splittable case, and twice that many for the non-splittable case (using ideas from Bin Packing [7]). The remaining transmissions do not cross the middle link, and hence can be scheduled by separately solving two subproblems, one for the transmissions on each half of the array. The two subproblems can share the wavelengths. If $\lambda(n, c)$ denotes the number of wavelengths used for scheduling transmissions of congestion at most c in a linear array of n nodes, we have the recurrence $\lambda(n, c) = O(\lceil c \rceil) + \lambda(n/2, c)$. This solves to $O(c \log n)$. But we can do better.

Note that it is relatively easy to get a good schedule if all the transmissions have the same length (see Section 3.2). So we divide the transmissions into classes based on their lengths, then schedule each class separately and finally merge the schedules. The merging step is also somewhat sophisticated. This is the outline of our algorithm.

1. *Partition into classes.* Say a transmission belongs to class i if its length is between 2^{i-1} (exclusive) and 2^i (inclusive). Let R_i denote the set of transmissions of class i, for $i = 0$ to $\lceil \log_2(n-1) \rceil$. Let m_i denote the size of R_i.
2. *Schedule transmissions of each class separately.* It will be seen that each class can be scheduled efficiently, i.e., using $O(1 + c(R_i))$ wavelengths.
3. *Merge the schedules of different classes.* We do not simply collect together the schedules constructed for the different classes, but do need to mix them together, and repartition.

Scheduling classes R_0, R_1 is easy. Note that each transmission in R_0 has length 1. So they can be assigned to light-trails created by simply putting shutters OFF at every node on all the wavelengths that are to be used. Now for a fixed l consider the light-trails $[l, l+1]$ on all the wavelengths. Each of these light-trails can be thought of as a bin in which the transmissions $[l, l+1]$ are to be assigned. Clearly, $\lceil c_l(R_0) \rceil$ light-trails will suffice for the splittable case, and twice that many for the non-splittable case (using ideas from Bin Packing [7]). Since the light-trails for different l do not overlap, they can be on the same wavelength. So $\max_l O(\lceil c_l(R_0) \rceil) = O(\lceil c(R_0) \rceil)$ wavelengths will suffice. Transmissions in R_1 have length 2. So they can be assigned to light-trails created on two sets of wavelengths – one having shutters OFF at even nodes and the other having shutters OFF at odd nodes. Transmissions starting at an even (odd) node are assigned to a light-trail on a wavelength of the first (second) set. Using an argument similar for the transmissions in R_0, we can show that each of these sets requires $O(\lceil c(R_1) \rceil)$ wavelengths. So for the rest of this paper we only consider classes 2 and larger.

3.2. Schedule class $i \geq 2$

It seems reasonable that if the class R_i is further split into subclasses each of which has $O(1)$ congestion, then each subclass could be scheduled using $O(1)$ wavelengths. This intuition is incorrect for an arbitrary collection of transmissions with congestion $O(1)$, as will be seen in Section 4. However, the intuition is correct when the transmissions have nearly the same length, as they do when taken from any single R_i.

Lemma 1. *There exists an $O(nm_i c)$ time procedure to partition R_i into sets $S_1, S_2, \ldots, S_k$ where $k \leq \lceil c(R_i) \rceil$ such that* (i) *$c(S_j) < 4$ for all j, and* (ii) *if a transmission in S_j uses link l then $\lceil c_l(R_i) \rceil \geq j$.*

Proof. We start with $T_1 = R_i$, and in general given T_j we pick a subset of transmission S_j from T_j using a procedure described below and repeat with the remaining transmissions $T_{j+1} = T_j \backslash S_j$ until T_{j+1} becomes empty for some value k of j.

For each link l from left to right, we greedily pick transmissions crossing link l into S_j until we have removed at least unit congestion from $c_l(T_j)$ or reduced $c_l(T_j)$ to 0. Note that if the transmissions already picked while considering the links on the left of l also have congestion at least 1 at link l then we do not add any more transmission while considering link l. 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. It can be seen that the construction of a single S_j takes at most $O(n|T_j|) = O(nm_i)$ time in both the pick-up step and the move-back step. For all S_j it takes $O(nm_i c)$ time.

Now we show that condition (i) of the lemma is satisfied, i.e., $c(S_j) < 4$ for all j. At the end of the 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 $c_x(L_1) = c_y(L_1) \le c_y(S_j) < 2$. 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, $c(S_j) < 4$.

To show that condition (ii) is also satisfied, 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 1 through $j - 1$. Thus $c_l(R_i) > j - 1$. That implies $\lceil c_l(R_i) \rceil \ge j$. This also implies that $k \le \max_l \lceil c_l(R_i) \rceil = \lceil c(R_i) \rceil$. □

A transmission t is said to cross a node u if t starts at a node on the left of u and ends at a node on the right of u. Since every transmission t in S_j has length at least $2^{i-1} + 1$, t must cross some node whose number is a multiple of 2^{i-1}. The smallest such numbered node is called the *anchor* of t. The *trail-point* of a transmission t is the rightmost node numbered with a multiple of 2^{i-1} that is on the left of the anchor of t. If the transmission has trail-point at node $q2^{i-1}$ for some q, then we define q mod 4 as its *phase*.

Lemma 2. *The set S_j can be scheduled using $O(1)$ wavelengths in $O(n|S_j|)$ time.*

Proof. We partition S_j further into sets S_j^p containing transmissions of phase p. This takes time $O(|S_j|)$. 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 each transmission is at most 2^i, the two transmissions cannot intersect.

So for the set S_j^p, consider 4 wavelengths, each having shutters OFF at nodes numbered $(4q + p)2^{i-1}$. Let $x = (4q + p)2^{i-1}$ and $y = (4(q + 1) + p)2^{i-1} = x + 2^{i+1}$ be two nearest nodes having shutters OFF. Among the $O(n/2^i)$ light-trails thus created, for a fixed q, each of the 4 light-trails $[x, y]$ can be thought of as a bin in which the transmissions having extent totally within $[x, y]$ and total bandwidth requirement at most 1 are to be assigned. This is an instance of the Bin Packing problem. Clearly, for a fixed q, these 4 light-trails will suffice for the splittable case, because $c(S_j^p) < 4$. This takes time proportional to the number of requests considered. Since the light-trails for different q do not overlap, the instances of the Bin Packing problem can share wavelengths and hence these 4 wavelengths will suffice. For the non-splittable case, 8 wavelengths will suffice, using standard Bin Packing ideas, e.g., First-Fit [7]. Overall it takes at most $O(|S_j|n/2^i) = O(n|S_j|)$ time.

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. □

Lemma 3. *The entire set R_i can be scheduled in time $O(nm_ic)$ such that at each link l there are $O(C_l(R_i) + 1)$ light-trails.*

Proof. We first consider the light-trails as constructed in Lemma 2. For all S_j the construction takes time $O(nm_ic)$. In this construction, uniformly at all links there are at most $\lceil c(R_i) \rceil \le c(R_i) + 1$ sets of light-trails such that each set corresponds to $O(1)$ light-trails created to schedule the transmissions of an S_j. Note that $c(R_i) = \max_l c_l(R_i)$. So, in this construction the condition of the lemma is surely satisfied for the link where the congestion is maximum. For other links the condition of the lemma may not be satisfied because (1) there may be empty light-trails and (2) some light-trails may contain links that are not used by any of the transmissions associated with the light-trail. So we remove empty light-trails and in case (2) 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). This modification takes time proportional to the number of light-trails which is $O(m)$. We prove next that with this modification, the condition of the lemma is satisfied.

Let j be the largest such that a transmission from S_j uses link l. After the modification the light-trails that carries transmissions from $S_{j'}$ for $j' > j$ do not use link l. So now there are j sets of light-trails using link l such that each set has $O(1)$ light-trails. However we know from Lemma 1 that $j \le \lceil c_l(R_i) \rceil \le c_l(R_i) + 1$. Thus there are a total of $O(j) = O(c_l(R_i) + 1)$ light-trails at link l. □

3.3. Merge schedules of all classes

If we simply collect together the wavelengths as allocated above, we would get a bound $O(c \log n)$. Note, however, that if two light-trails, one for transmissions in class i and the other for transmissions 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 a sharper bound.

Theorem 4. *The entire set R can be scheduled using $O(c + \log n)$ wavelengths in time $O(nmc + m \log m)$.*

Proof. We know that after the modification in Lemma 3, at each link l there are a total of $O(c_l(R_i) + 1)$ light-trails for each class i. Thus summing over all classes, the total number of light-trails at l are $O(c_l(R) + \log n)$, and total time taken is $O(nmc)$.

Think of each light-trail as an interval, giving us a collection of, say k, intervals such that any link l has at most $O(c_l(R) + \log n) = O(c + \log n)$ intervals. Now this collection of k intervals can be colored using $O(c + \log n)$ colors [22] in time $O(k \log k)$. Now for each color w, we use a separate wavelength and configure the light-trails corresponding to the intervals of color w by setting the shutters OFF at the nodes corresponding to the endpoints of the intervals. Hence $O(c + \log n)$ wavelengths suffice. Overall it takes time $O(nmc + m \log m)$ as k can be at most m. □

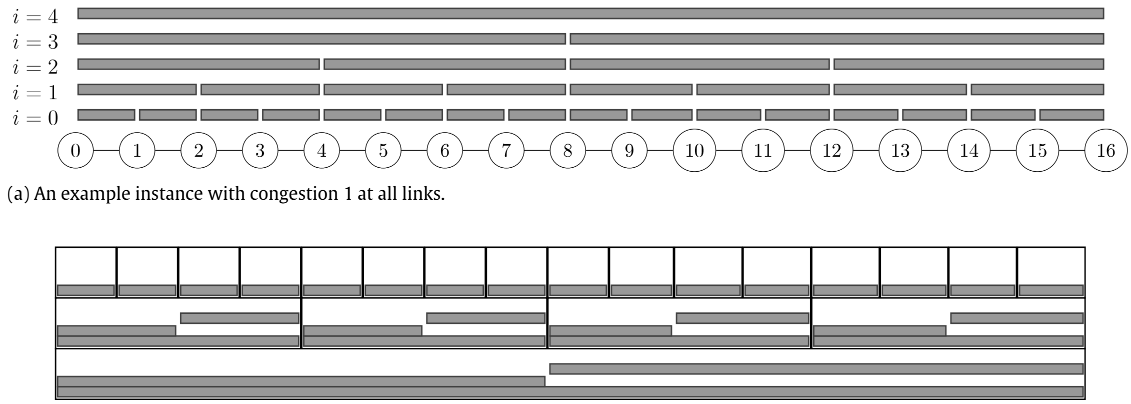
4. On the congestion lower bound

We show an instance of the stationary problem for which the congestion lower bound is weak. For convenience, we assume there are $n + 1$ nodes numbered $0, \ldots, n$ where $n = 2^k$ for some k and all logarithms are with base 2. All the transmissions have the same bandwidth requirement $b = 1/(\log n + 1)$.

First, we have a transmission going from 0 to n. Then a transmission from 0 to $n/2$ and a transmission from $n/2$ to n. Then four transmissions spanning one-fourth the distance, and so on. In class $i \in \{0, 1, \ldots, \log n\}$ there are $n/2^i$ transmissions $B(s_{ij}, d_{ij}) = b$ where $s_{ij} = j2^i$, $d_{ij} = (j + 1)2^i$ for all $j = 0, 1, \ldots, (n/2^i) - 1$. All other entries of B are 0. This is illustrated in Fig. 2(a) for $n = 16$. Clearly the congestion of this pattern is uniformly 1.

Consider an optimal solution for the splittable case. There has to be a wavelength with a light-trail in which the transmission $[0, n]$ is scheduled. This light-trail might have additional transmissions besides $[0, n]$. Clearly, we can assume without loss of generality that the light-trail contains the longest transmissions. Suppose the transmissions from the longest l classes are completely contained in this light-trail. Thus we have a total of at most $1 + 2 + 4 + \cdots + 2^l = 2^{l+1} - 1$ transmissions assigned to this light-trail. Total bandwidth requirement of these transmissions should be at most 1. This gives us $(2^{l+1} - 1)(1/(\log n + 1)) \le 1$ implying $l \le \log(\log n + 2) - 1 = O(\log \log n)$.

The remaining transmissions do not cross nodes with numbers $qn/2^{l-1}$, for all integers q. Thus, we have 2^{l-1} separate parallel problems, each having $n/2^{l-1} = \Omega(n/\log n)$ nodes. Thus the total number of wavelengths $W(n)$ needed must satisfy the



(a) An example instance with congestion 1 at all links.

(b) An optimal solution for the above example using 3 wavelengths.

Fig. 2. An example instance where congestion bound is weak.

recurrence $W(n) = 1 + W(n/\log n)$. This solves to $W(n) = \Omega(\log n/\log\log n)$.

Suppose we add $c - 1$ transmissions of extent $[0, n]$ and bandwidth requirement 1 to this pattern of transmissions of congestion 1 uniformly at all links. We can similarly show that an optimal solution will require $c - 1 + \Omega(\log n/\log\log n)$ wavelengths. Thus we will get an instance of congestion c uniformly at all links but which requires $\Omega(c + \log n/\log\log n)$ wavelengths, for any c.

5. The on-line problem

In the on-line 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 belongs 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 of transmissions S. This is simply the total bandwidth requirement of those transmissions from S requiring to cross link l at time t. The congestion lower bound $c(S)$ is $\max_l \max_t c_{lt}(S)$, the maximum congestion over all links over all time instants.

For the on-line problem, we present two algorithms – (1) SeparateClass having competitive ratio $\Theta(\log n)$ and (2) AllClass, a simplification of SeparateClass. We show that this simplified algorithm, AllClass has a competitive ratio in between $\Omega(\log^2 n/\log\log n)$ and $O(\log^2 n)$.

In both the on-line algorithms, when a transmission request arrives, we first determine its class i and trail-point x (defined in Section 3.2). The transmission is allocated to some light-trail $[x, x+2^{i+1}]$. However, the algorithms differ in the way a light-trail is configured on some candidate wavelength.

5.1. *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 class i and phase p. Whenever a transmission t 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 w is found, and a light-trail L on w starting at the trail-point of t has space, then t is assigned to the light-trail L. If no such wavelength is found, then a new wavelength w' is allocated, it is labeled and configured for class i and phase p as above and t is assigned to the light-trail on w' that starts at the trail-point of t.

When a transmission finishes, it is removed from its associated light-trail. When all transmissions in a wavelength finish, then its labels are removed, and it can subsequently be used for other classes or phases.

Time complexity: for a class i and phase p there are $n/2^i$ possible light-trails. For each of these light-trails, we can maintain a list of wavelengths on which the light-trail is present. So on an arrival, searching for a candidate light-trail takes $O(w)$ time where w is the number of wavelengths used. On departure also, it takes $O(w)$ time.

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

Proof. Suppose at some instant of time, wavelengths $w_1, w_2, \ldots, w_x$, ordered according to the time of allocation, had non-empty light-trails $L_1, L_2, \ldots, L_x$, respectively, of the same class and phase across link l'. Let u be the anchor (defined in Section 3.2) of the transmissions assigned on these light-trails and l be the link between node u and node $u + 1$.

Now suppose wavelength w_x was allocated due to a transmission t. This could only happen because t could not fit in the wavelengths w_j for all $j \le x - 1$.

For the splittable case this can only happen if light-trails L_1 through L_{x-1} together contain transmissions of congestion at least $x - 1 - c(t) = \Omega(x)$ crossing the anchor u of t, when t arrived. Thus at that time l had congestion $\Omega(x)$, giving us the result.

For the non-splittable case, suppose that $c(t) \le 0.5$. Then the transmissions in each of the light-trails L_j, $1 \le j \le x - 1$, must have congestion of at least 0.5 at l when t arrived, giving congestion $\Omega(x)$. So suppose $c(t) > 0.5$. Let k be the largest such that light-trail L_k contains a transmission t' with $c(t') \le 0.5$ when t arrived. If no such k exists, then clearly the congestion at l when t arrived is $\Omega(x)$. If k exists, then all the light-trails $L_j, j > k$ have transmissions of congestion at least 0.5 at l when t arrived. And the light-trails $L_j, j \le k$ had transmissions of congestion at least 0.5 at l when t' arrived. So at one of the two time instants the congestion at l must have been $\Omega(x)$. □

Theorem 6. SeparateClass *is $\Theta(\log n)$ 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. That will prove that SeparateClass is $O(\log n)$ competitive.

Consider the time at which the wth wavelength was allocated by SeparateClass. At this time $w-1$ wavelengths are already in use, and of these at least $w' = (w-1)/(4\log n)$ 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, there is a link that had congestion $\Omega(w'-1) = \Omega(((w-1)/(4\log n))-1) = \Omega(w/\log n)$ at some time instant. This is a lower bound on any algorithm, even off-line. Thus the competitive ratio of SeparateClass is $O(\log n)$.

We show the lower bound $\Omega(\log n)$ using the following example. Let $n = 2^k + 1$. At each time $t = 0, 1, \ldots, k$, a transmission $[0, 2^t]$ arrives. All transmissions have bandwidth requirement $1/(k+1)$. At time $k+1$ all transmissions depart together. SeparateClass takes k wavelengths because each transmission is of a different class. The optimal off-line algorithm assigns all of them to a single light-trail spanning the entire network and hence takes only one wavelength. □

5.2. Algorithm AllClass

This is a simplification of SeparateClass in that the allocated wavelengths are not labeled. When a transmission t of class i and trail-point x arrives, we search the wavelengths in the order they were allocated for a light-trail L of extent $[x, x+2^{i+1}]$ such that L has enough space to serve t. If such a light-trail L is found, then t is assigned to L. If no such light-trail is found, then an attempt is made to create a light-trail $[x, x+2^{i+1}]$ from the unused portions of one of the existing wavelengths in a first-fit manner in the order they were allocated. If such a light-trail L can be created, then L is created and t is assigned to L. Otherwise a new wavelength w is allocated, the required light-trail L of extent $[x, x+2^{i+1}]$ is created on w, and t is assigned to L. The portion of the wavelength w outside the extent of L is marked unused.

When a transmission finishes, it is removed from its associated light-trail. If this makes the light-trail empty then we mark its extent on the corresponding wavelength as unused.

Time complexity: using a binary search tree based data structure for the light-trails and the transmissions, the algorithm can be implemented in $O(\log m + w)$ time on each arrival and in time $O(\log m)$ on each departure where m is the number of active requests and w is the number of wavelengths used.

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

Proof. Suppose AllClass uses w wavelengths. Since the optimal must use at least one, we only need consider the case $w = \Omega(\log^2 n)$.

The key idea is to argue that at some time during the execution of AllClass there will be least $w/(4\log n)$ non-empty light-trails (not necessarily of the same class and phase) crossing the same link. If this holds, then of these light-trails, at least $w/(16\log^2 n)$ must have the same class and phase. But it can be shown that Lemma 5 is also true for AllClass, and hence there is a link having congestion $\Omega(w/(16\log^2 n))$ at some time instant. But this is a lower bound on the number of wavelengths required by any algorithm, including an off-line algorithm. Thus the competitive ratio of AllClass is at most $O(\log^2 n)$.

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 at that time the $w-1$ previously allocated wavelengths contained light-trails overlapping with L. Let S' denote this set of light-trails, each from a different wavelength, but overlapping with L.

If S' contains at least $w/(4\log n)$ light-trails which cross the leftmost link in L or the rightmost link, we are done. So assume the contrary. Thus there must be at least $w' = w-1-2w/(4\log n) = w-1-w/(2\log n)$ in S' whose extent is completely contained in the extent of L. Among these light-trails, let L' be the largest numbered. 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 to find at least $w/(4\log n)$ light-trails crossing a link, we will end up with a light-trail L'' such that there are at least w'' wavelengths having 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. □

5.3. Lower bound for AllClass

We give a sequence of transmissions for which AllClass takes $\Omega(\log^2 n/\log\log n)$ wavelengths but an optimal off-line algorithm, Opt, requires only one wavelength.

Theorem 8. AllClass *is* $\Omega(\log^2 n/\log\log n)$ *competitive.*

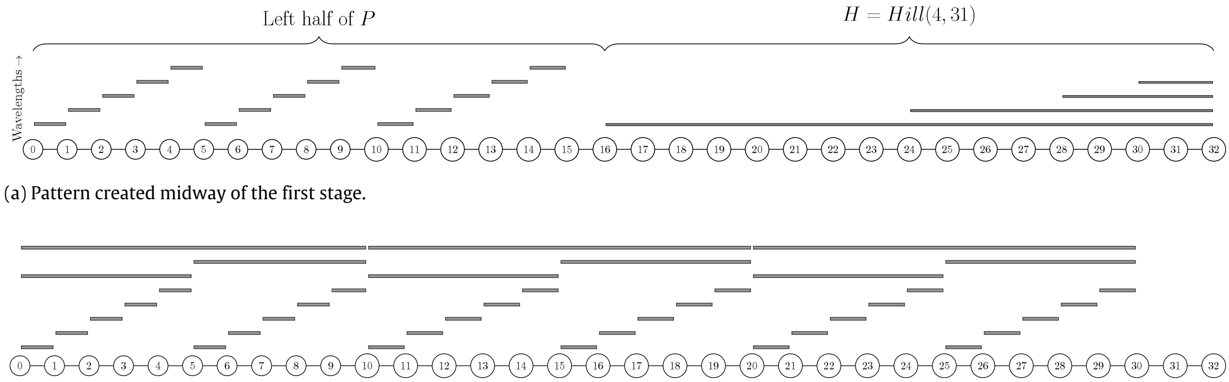
Our transmission sequence consists of several (the exact count will be shown later) subsequences, which we call stages. In all stages, all transmissions have a height (i.e., bandwidth requirement) of $1/n^2$. Our transmission sequence is such that, at any point of time, there are less than n^2 active transmissions. Opt will put all transmissions in a single light-trail using the full length of a wavelength. On the other hand, it will be seen that AllClass will allocate $\Omega(\log^2 n/\log\log n)$ wavelengths in total for all stages. We describe the first stage only; the other stages are scaled versions of the first stage. The goal of the first stage is to force AllClass to allocate wavelengths with light-trail patterns given in the following lemma.

Lemma 9. *Let the network have* $q+1$ *nodes numbered* $0, \ldots, q$. *Then there is a transmission sequence for which* AllClass *allocates* $k = \lfloor \log q \rfloor$ *wavelengths numbered* $0, \ldots, k-1$, *with the following staircase pattern: each wavelength i has* $\lfloor n/k \rfloor$ *unit-length light-trails* $[jk+i, jk+i+1]$, *each containing a single transmission, for all* $j = 0, \ldots, \lfloor n/k \rfloor - 1$.

Proof. For simplicity we assume $q = 2^k$, i.e., k is exactly equal to $\log q$. The general case can be similarly proved.

We first describe how to create a unit-length light-trail $[x, x+1]$ on any wavelength h. We will repeatedly use this procedure to create our pattern. Define $Hill(h, x)$ to be an ordered sequence of h transmissions as follows. For each $i = 0, 1, \ldots, h-1$, $Hill(h, x)$ contains a transmission that uses the link $[x, x+1]$ and has class $k-1-i$, and some suitable phase. The key point is that all the transmissions in a hill overlap but have different classes, and hence AllClass must assign them in distinct light-trails on different wavelengths. Thus starting from scratch, the arrival of the transmissions in a hill will cause h wavelengths to be allocated. For example, we show $Hill(4, 31)$ on the right half of Fig. 3(a). Further, if a new transmission $[x, x+1]$ arrives, it will cause one more wavelength to be allocated. From now on, by creating (deleting) a hill we mean the arrival (departure) of transmissions in a hill.

Now we describe how to generate the staircase using several hills. The idea is to build the staircase one wavelength at a time from top to bottom, i.e., first create all light-trails of the staircase on wavelength $k-1$, then all light-trails on wavelength $k-2$ and so on. Each unit-length light-trail $[x, x+1]$ on an wavelength is created by temporarily creating an appropriate hill underneath it,



(a) Pattern created midway of the first stage.

(b) Final pattern.

Fig. 3. An instance for which AllClass is $\Omega(\log^2 n/\log\log n)$ competitive.

then creating the transmission $[x, x+1]$ and finally deleting the temporary hill. The left half set of staircases is created first, and then the right half set.

Before creating the left half, we first create hill $H = Hill(k-1, q-1)$. This hill will survive until the left half is completely created. Its sole purpose is to ensure that the numbering of its $k-1$ wavelengths does not change as the left half is created. The left half is created top to bottom as given in Algorithm 1. Consider the first execution of the insertion marked as belonging to the staircase. Because of the hill H', this transmission will clearly be assigned to a light-trail on wavelength i. Note further that when transmissions in H' depart, the wavelengths $0, \ldots, i-1$ do not become empty because of the presence of hill H in the right half. Thus the subsequent iterations also force the transmissions to be assigned in wavelength i, and so on.

Algorithm 1 Create left half of the staircase

```
for i = k − 1 downto 0 do
  for j = 0 to q/2k do
    Create a hill H′ = Hill(i, jk + i)
    Insert an arrival event for transmission [jk + i, jk + i + 1]
    {belongs to staircase}
    Remove hill H′
  end for
end for
```

At the end of the above, we will have created a pattern as shown in Fig. 3(a). Since each light-trail contains only one transmission, we just show the transmissions instead of the light-trails.

Next we remove H, and execute the same code to create the right half of the staircase on the k wavelengths already allocated. Note that the light-trails created in the left half now serve the purpose that H did earlier. At this point we will have the complete staircase. □

The first stage is created by using Lemma 9 with $q = n$. In the second stage, we can treat every $k = \lfloor \log n \rfloor$ nodes as a single node, and think of the network as having $n' = \lfloor n/k \rfloor$ nodes. We create a staircase of height $\lfloor \log n' \rfloor$ but with light-trails of length k using Lemma 9 with $q = n'$. Since these light-trails are longer than the light-trails in the previous stage, we can stack up the new pattern on top of the previous pattern. We can keep doing this until n' becomes less than 2. Thus the number of stages is $\Omega(\log_{\log n} n) = \Omega(\log n/\log\log n)$.

Let $T(n)$ denote the total height of the patterns thus created for $n+1$ nodes, then $T(n)$ is computed using the following recurrence:

$$T(n) = \lfloor \log n \rfloor + T(\lfloor n/\lfloor \log n \rfloor \rfloor)$$

$$\text{or simply } T(n) = \log n + T(n/\log n) \tag{2}$$

with the base condition $T(n) = 0$ for $n \le 1$. It can be seen that the recurrence has solution $T(n) = \Omega(\log^2 n/\log\log n)$.

Thus AllClass will use $\Omega(\log^2 n/\log\log n)$ wavelengths for the patterns created. Fig. 3(b) shows all the transmissions active at the end of all stages, for the example considered in Fig. 3(a).

5.4. Remarks

It is interesting to note that AllClass is more flexible than SeparateClass, and it is this flexibility that is exploited in the lower bound argument to show a worse ratio for AllClass than SeparateClass.

Indeed, the more flexibility we give, the worse it seems the ratio will become. In AllClass, if we have a transmission of length $L = 2^k$ we assign it to a light-trail of length $2L$. This seems wasteful. But this is done to accommodate transmissions that do not start at a multiple of $L/4$, using only 4 phases. Suppose we decide to be more flexible, and allow light-trails to start anywhere (so long as their length is 2^k for some k) using 2^k phases. Although this strategy will handle the above transmission better, in general it is worse in that its competitive ratio can be shown to be $\Omega(\log^2 n)$. We omit the details.

6. Problem lower bound – $\Omega(\log n)$

Theorem 10. *Every on-line algorithm has competitive ratio $\Omega(\log n)$.*

Let Alg be any algorithm for the on-line problem and Opt be an optimal off-line algorithm. By observing the behavior of Alg we can create a sequence of transmissions for which Alg takes $\Omega(\log n)$ times as many wavelengths as Opt. This will prove the theorem.

For convenience we assume that the network has $n+1$ nodes numbered $0, 1, \ldots, n$ and $n = 2^k$ for some k. Our transmission sequence will have $k = \log n$ stages. For stage $i = 0, 1, \ldots, k-1$, consider the network broken up into $n' = n/2^i$ intervals of length 2^i. Let this set of intervals be $Q_i = \{[q2^i, (q+1)2^i]\}_{q=0}^{n'-1}$. At the beginning of the ith stage, for each interval $I \in Q_i$, k^2 transmissions having extent I arrive. We will denote this set of $n'k^2$ transmissions by A_i. All transmissions have a height (i.e., bandwidth requirement) of $1/k$. At the end of the ith stage, all but a subset S_i of A_i depart. The set S_i is determined by observing the behavior of Alg.

Lemma 11. *Among the $n'k^2$ transmissions arriving at the beginning of stage i, we can find a set S_i of $n'k$ transmissions such that* (1) *Exactly k transmissions from S_i are for a single interval $I \in Q_i$,* (2) Alg *assigns each transmission in S_i to a distinct light-trail.*

Proof. We have k^2 transmissions for each interval $I \in Q_i$. Partition these k^2 transmissions arbitrarily into k groups of k

transmissions each. So overall we have $n'k$ groups each containing k transmissions. Now form a bipartite graph (U, V, E) as follows.

1. U has $n'k$ vertices, each vertex corresponding to a group of k transmissions as formed above. Note that there are k groups for each interval, and hence we can consider a distinct group of k vertices of U to be associated with each $I \in Q_i$.
2. V has a vertex corresponding to each light-trail used by Alg for serving the transmissions of this stage.
3. E has following edges. Suppose a transmission t from the group associated with a vertex $u \in U$ is placed by Alg in the light-trail L associated with a vertex $v \in V$. Then for each such t there will be an edge (u, v) in E. Note that this may produce parallel edges if several transmissions in the group of u are placed in L.

The degree of each vertex in U is exactly k, one edge for each transmission in the associated group. Consider any vertex $v \in V$. Since its associated light-trail can accommodate at most k transmissions of height $1/k$, its degree must be at most k.

Now consider any subset S of vertices from U and its neighborhood T in V. Because vertices in U have degree exactly k there must be exactly $|S|k$ edges leaving S. These must be a subset of the edges entering T. But vertices in V have degree at most k. So there can be at most $|T|k$ edges entering T. Thus we have $|S|k \leq |T|k$, i.e., $|T| \geq |S|$, i.e., S has at least as many neighbors as its own cardinality. But this is true for any S. Thus by the generalization of Hall's theorem, there must be a matching M that includes an edge from every vertex of U to a distinct vertex in V.

Consider the set S_i of transmissions associated with each edge of M. Since there is exactly one edge in M for each node in U, S_i has one transmission per group of transmissions for each interval. Hence S_i has exactly k transmissions for each interval. Since M has exactly one edge per vertex in V, we know that each transmission in S_i is assigned to a distinct light-trail by Alg. □

We have now completely described the transmission sequence. At the end all transmissions have departed except those in some S_i. We will use D_i to denote the transmissions which depart in stage i. Clearly $A_i = S_i \cup D_i$.

Lemma 12. Opt *uses overall* $2k-1$ *wavelengths while processing the transmission sequence for all stages.*

Proof. Consider stage i. The set A_i has k^2 transmissions for each interval $I \in Q_i$. To serve these transmissions A_i, Opt uses k wavelengths configured as follows. Each wavelength is configured into light-trails as per Q_i, i.e., each interval $I \in Q_i$ forms one light-trail. Now the key point is that Opt places all transmissions in S_i into light-trails on a single wavelength. This can be done because the set S_i indeed has k transmissions for each $I \in Q_i$. The remaining transmissions D_i can be accommodated into $k-1$ additional wavelengths. Note now that at the end of the stage, the transmissions D_i depart. Hence although the stage used k wavelengths transiently, at the end $k-1$ of these are released.

Thus, at the end of stage i, there will be $i+1$ wavelengths in use, one for transmissions in each S_j, $j = 0, \ldots, i$. When A_{i+1} arrives, Opt will allocate k new wavelengths. So while processing A_{i+1} there will be $i+1+k$ wavelengths in use. These will drop down to $i+2$ at the end of stage $i+1$. Thus, over all the stages the maximum number of wavelengths used will be at most $\max_{i=0,\ldots,k-1}(i+k)$, i.e., $2k-1$. □

Lemma 13. Alg *uses at least* $k^2/2$ *wavelengths while processing the transmission sequence.*

Proof. Consider the light-trails used by Alg which are active at the end of the stage $k-1$. Each of these light-trails may contain several transmissions but only one transmission from each S_i. Since transmissions from each S_i have different lengths, each light-trail must hold transmissions of different lengths. Thus, each light-trail can have at most one transmission of length 1, one of length 2, and so on. The sum of the lengths of the transmissions assigned to a single light-trail of length l is thus at most $1 + 2 + 4 + \cdots + l = 2l - 1 \leq 2l$. But this applies to all light-trails in any wavelength, and hence the total length of the transmissions assigned to a single wavelength is at most $2n$. However, the transmissions that survive at the end consist of nk transmissions of length 1, $nk/2$ transmissions of length 2, and so on to $2k$ transmissions of length $n/2$. Thus the total length is nk^2. Thus Alg needs at least $(nk^2)/(2n) = k^2/2$ wavelengths at the end. □

But Opt requires at most $2k - 1 < 2k$ wavelengths. Hence the competitive ratio is at least $(k^2/2)/2k = k/4 = \Omega(\log n)$. This completes the proof of Theorem 10.

7. Simulations

We simulate our two on-line 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 5 (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. The class and phase of a transmission is determined by the light-trail of maximum class (note that now larger classes have shorter light-trails) and minimum phase that can completely accommodate it. 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.

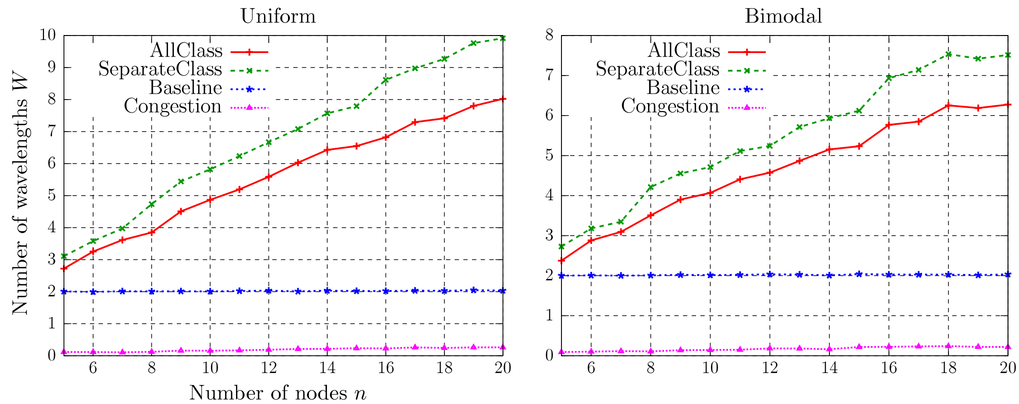
For the baseline algorithm in each ring we use a single OFF shutter at node 0. Transmissions from lower numbered nodes to higher numbered nodes use the clockwise ring, and the others, the counterclockwise ring.

7.1. The simulation experiment

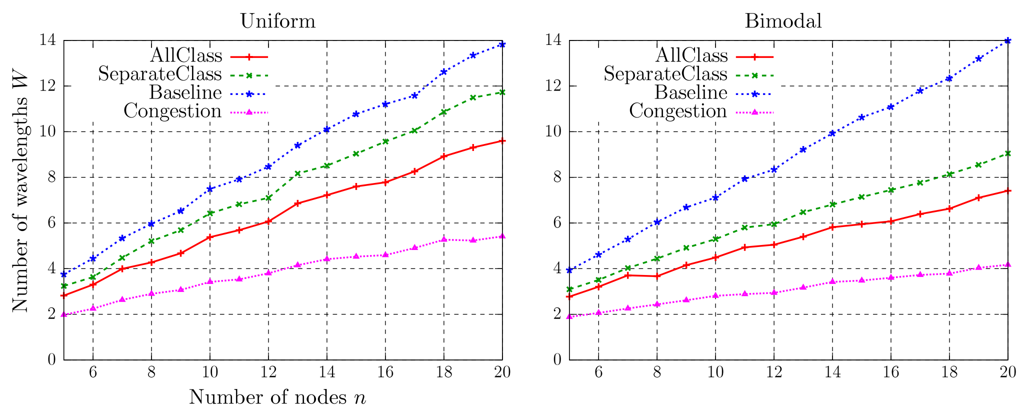
A single simulation experiment consists of running the algorithms on a certain load, characterized by parameters λ, D, r_{min} and α for 100 time steps. In our results, each data-point reported is the average of 150 simulation experiments with the same load parameters.

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}. We considered four different distributions D for selecting the destination node of a transmission.



(a) Low load.



(b) High load.

Fig. 4. Simulation results.

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 and then choose a destination uniformly from the nodes possible for that class. 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 the 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 a 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* and for $r_{min} = 0.01, 0.5$, i.e., a total of 4 load scenarios. Results for other scenarios follow a similar pattern.

7.2. Results

Fig. 4 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 5. 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.

It is also to be noted that AllClass performs better than SeparateClass in our simulations. This is perhaps surprising because in Section 5.3 we showed that SeparateClass has a better competitive ratio. Indeed, in that section we presented an input instance on which AllClass performs substantially worse than SeparateClass. But there is no contradiction here. The simulation results merely indicate that instances like the one we presented do not appear in our workload. For our workload, perhaps the extra flexibility of AllClass is very useful. So we feel that in practice the algorithm AllClass is an important candidate.

8. 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 4 shows, congestion is not always a good lower bound. This may lead to a constant factor approximation algorithm for the problem.

In the on-line case we proved that the lower bound on the competitive ratio of any algorithm is $\Omega(\log n)$ and gave a matching algorithm which we proved to have competitive ratio $\Theta(\log n)$. We also gave a second algorithm which seems to work better in practice but can be as bad as $\Omega(\log^2 n/\log\log n)$ factor worse than an optimal off-line algorithm on some pathological examples as we have shown. We also proved an upper bound of $O(\log^2 n)$ for the algorithm but it will be an interesting problem to close the gap between the two bounds.

Our on-line model is very conservative: once a transmission is allocated on a light-trail, it cannot be moved to another light-trail, nor can the light-trail grow or shrink. However, there are models [12] which allow light-trails to shrink/grow dynamically, and those in which it is possible to transfer active transmissions from one light-trail to another [15]. It will be useful to incorporate these (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.

Acknowledgments

We would like to thank Ashwin Gumaste for encouragement, insightful discussions and patient clearing of doubts related to light-trails. We would also like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper.

References

[1] S. Albers, Online algorithms: a survey, Mathematical Programming 97 (1) (2003) 3–26.
[2] A. Ayad, K. Elsayed, S. Ahmed, Enhanced optimal and heuristic solutions of the routing problem in Light-trail networks, in: Workshop on High Performance Switching and Routing (HPSR), 2007 pp. 1–6.
[3] S. Balasubramanian, W. He, A. Somani, Light-Trail networks: design and survivability, Thirtieth IEEE Conference on Local Computer Networks, 2005, pp. 174–181.
[4] S. Balasubramanian, A. Kamal, A. Somani, Network design in IP-centric Light-trail networks, in: Second International Conference on Broadband Networks (Broadnets), 2005, pp. 41–50.
[5] A. Borodin, R. El-Yaniv, Online computation and competitive analysis, Cambridge University Press, New York, NY, USA, 1998.
[6] I. Chlamtac, A. Gumaste, Light-trails: A solution to IP centric communication in the optical domain, Lecture notes in computer science (2003) 634–644.
[7] E.G. Coffman Jr., M.R. Garey, D.S. Johnson, Approximation algorithms for bin packing: a survey, in: Approximation algorithms for NP-hard problems, 1997, pp. 46–93.
[8] H. ElGindy, H. Schroder, A. Spray, A. Somani, H. Schmeck, RMB – A reconfigurable multiple bus network, in: Second International Symposium on High-Performance Computer Architecture, IEEE, 1996, pp. 108–117.
[9] J. Fang, W. He, A. Somani, Optimal Light-trail design in WDM optical networks, in: IEEE International Conference on Communications, Vol. 3, 2004, pp. 1699–1703.
[10] P. Gokhale, R. Kumar, T. Das, A. Gumaste, Cloud computing over Metropolitan Area WDM networks: the Light-trails approach, in: IEEE Global Telecommunications Conference (GLOBECOM), IEEE, 2010, pp. 1–6.
[11] A. Gumaste, I. Chlamtac, Mesh implementation of Light-trails: a solution to IP centric communication, in: Twelfth International Conference on Computer Communications and Networks, ICCCN, 2003, pp. 178–183.
[12] A. Gumaste, I. Chlamtac, Light-trails: an optical solution for IP transport [Invited], Journal of Optical Networking 3 (5) (2004) 261–281.
[13] A. Gumaste, T. Das, A. Mathew, A. Somani, An autonomic virtual topology design and two-stage scheduling algorithm for light-trail WDM networks, Journal of Optical Communications and Networking 3 (4) (2011) 372–389.
[14] A. Gumaste, G. Kuper, I. Chlamtac, Optimizing light-trail assignment to WDM networks for dynamic IP centric traffic, in: Thirteenth Workshop on Local and Metropolitan Area Networks (LANMAN), IEEE, 2004, pp. 113–118.
[15] A. Gumaste, P. Palacharla, Heuristic and optimal techniques for light-trail assignment in optical ring WDM networks, Computer Communications 30 (5) (2007) 990–998.
[16] A. Gumaste, J. Wang, A. Karandikar, N. Ghani, Multihop light-trails (mlt)-a solution to extended metro networks, in: International Conference on Communications (ICC), IEEE, 2009, pp. 1–6.
[17] A. Gumaste, 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.
[18] K. Li, Y. Pan, S. Zheng, Parallel matrix computations using a reconfigurable pipelined optical bus, Journal of Parallel and Distributed Computing 59 (1) (1999) 13–30.
[19] K. Li, Y. Pan, S. Zheng, Efficient deterministic and probabilistic simulations of PRAMs on linear arrays with reconfigurable pipelined bus systems, The Journal of Supercomputing 15 (2) (2000) 163–181.
[20] A. Lodha, A. Gumaste, P. Bafna, N. Ghani, Stochastic optimization of Light-trail WDM ring networks using Bender's decomposition, in: Workshop on High Performance Switching and Routing (HPSR), 2007, pp. 1–7.
[21] X. Luo, B. Wang, Integrated scheduling of grid applications in WDM optical Light-trail networks, Journal of Lightwave Technology 27 (12) (2009) 1785–1795.
[22] S. Olariu, An optimal greedy heuristic to color interval graphs, Information Processing Letters 37 (1) (1991) 21–25.
[23] Y. Pan, M. Hamdi, K. Li, Efficient and scalable quicksort on a linear array with a reconfigurable pipelined bus system, Future Generation Computer Systems 13 (6) (1998) 501–513.
[24] S. Rajasekaran, S. Sahni, Sorting, selection, and routing on the array with reconfigurable optical buses, IEEE Transactions on Parallel and Distributed Systems 8 (11) (1997) 1123–1132.
[25] Y. Wang, An efficient O(1) time 3D all nearest neighbor algorithm from image processing perspective, Journal of Parallel and Distributed Computing 67 (10) (2007) 1082–1091.
[26] R. Wankar, R. Akerkar, Reconfigurable architectures and algorithms: a research survey, IJCSA 6 (1) (2009) 108–123.
[27] B. Wu, K. Yeung, OPN03-5: Light-trail assignment in WDM optical networks, in: IEEE Global Telecommunications Conference (GLOBECOM), 2006, pp. 1–5.
[28] Y. Ye, H. Woesner, R. Grasso, T. Chen, I. Chlamtac, Traffic grooming in Light-trail networks, in: IEEE Global Telecommunications Conference (GLOBECOM), 2005.
[29] W. Zhang, G. Xue, J. Tang, K. Thulasiraman, Dynamic Light-trail routing and protection issues in WDM optical networks, in: IEEE Global Telecommunications Conference (GLOBECOM), 2005, pp. 1963–1967.

Soumitra Pal received a B.E. degree in Computer Science and Technology from Bengal Engineering and Science University, Howrah, India in 2000 and an M.Tech. degree in Computer Science and Engineering from IIT Bombay in 2007. Since 2007 he is a doctoral student in Computer Science and Engineering at IIT Bombay. He worked with Texas Instruments, India as a Software Design Engineer during 2000–5. His research interests are in algorithm design, optimization, and parallel and distributed systems.

Abhiram Ranade received a B.Tech degree in Electrical Engineering from IIT Bombay in 1981 and a doctorate in Computer Science from Yale University, USA in 1989. He was an Assistant Professor of Electrical Engineering and Computer Science at the University of California, Berkeley during 1988–95. Since 1995 he is on the faculty of IIT Bombay, currently as a Professor of Computer Science and Engineering. He has also held a visiting Professor position at the Institute for Mathematics and its Applications, University of Minnesota, USA, and a visiting researcher position at Thinking Machines Corporation, USA. He has served as an associate editor of IEEE Transactions on Parallel and Distributed Systems, and as a member of the program committees of leading conferences in computer science. His research interests are in algorithm design, optimization, parallel computation and computer based education.