Network Design for IP-Centric light-trail networks

Srivatsan Balasubramanian, Ahmed E. Kamal, and Arun K. Somani
Department of Electrical and Computer Engineering
Iowa State University, Ames, IA 50011
E-mail: {vatsan, kamal, arun}@iastate.edu

***Abstract*— We explore network design principles for next-generation all-optical wide-area networks, employing light-trail technology. Light-trail [1] is a light-wave circuit that allows multiple nodes to share the optical bandwidth through the inclusion of simple but flexible hardware overlaid with a light-weight control protocol. We develop light-trails as a novel and amenable control and management solution to address IP-centric communication problems at the optical layer. We propose optical switch architectures that allow seamless integration of lightpath and light-trail networks, and assess their costs and capabilities. We formulate the static light-trail RWA problem as an Integer Linear Program. Since this programming problem is computationally intractable, we split it into two subproblems: (a) trail routing, for which we provide three heuristics, (b) wavelength assignment, for which we use the largest first heuristic available in literature. The objective of our design is to minimize the optical layer and electronic layer costs in terms of the number of wavelengths and communication equipments required. We illustrate our approach by comparing the performance of our trail design heuristics on some test networks.**

I. Introduction

The rapid growth of data traffic has fueled the Internet transport infrastructure to evolve towards a model of wire speed IP routers interconnected by intelligent optical networks. In such a dynamic environment, a scalable and rapidly deployable network architecture that supports high data rates and that accommodates multi-granularity traffic, will be the key technology for future networks.

Optical circuit switched networks operate on the notion of creating a dedicated lightpath from source to destination. However, this concept, in general, is not consistent with the packet switching philosophy of the Internet. Circuit switched networks do not offer buffering capability and hence the capacity from source to destination must match the peak rate, which may be significantly higher than the average rate, thereby leading to low effective utilization. Thus, there is a need for a technology which supports a granularity of transmission and switching finer than that of a full wavelength.

Optical packet switching achieves high statistical multiplexing gains and is amenable for traffic engineering. One of the key components of this network is the switching fabric that needs to be configured on a per-packet basis. Pure optical switches with high port count, low loss, negligible cross talk, and nanosecond switching times are not commercially viable yet. The requirement for rapid synchronization, scalable packet parsing mechanisms and the lack of fast and large random access memory units prevent implementation of sophisticated optical router architectures that are possible in electronics.

Optical burst switching is a hybrid approach between coarse-grained circuit switching and fine-grained packet switching. In burst switching, the ratio of cross-connect configuration time to burst duration may be very high with present optical switching technologies and hence may lead to low network utilization. In addition, the edge router for burst switched networks needs to implement burst assembly, disassembly and queue fairness algorithms. Thus, the control unit design at high data rates may become challenging.

The existing architectures, thus, are either not effective in handling bursty IP traffic patterns or they utilize immature optical components, unproven for field deployment. As a solution to providing high resource utilization, sub-wavelength support and network transparency, we discuss light-trail technology. The goal of light-trails is to eliminate active switching, and leverage resource multiplexing to address the growing demands placed on WDM networks.

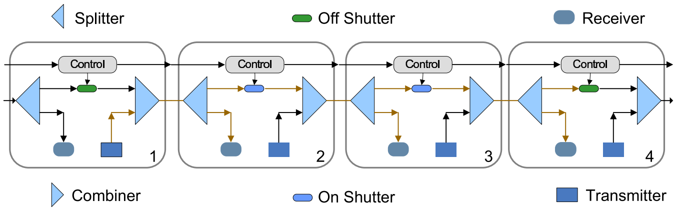


Fig. 1. Data transfer from node 1 to node 4 in a light-trail. Each node is equipped with LAUs. The optical shutter is in the off state on node 1 and node 4 and in the on state on node 2 and node 3. The arrows in lighter shade shows the packets being transmitted by node 1 to nodes downstream.

The work in [1] introduced light-trails as an optical solution for IP transport. The network utilization and blocking probabilities of light-trail networks is compared to that of OBS and lightpath routed networks in [5]. A tree-shaped variant of a light-trail called clustered light-trail is discussed and methods to create an efficient virtual topology for a given traffic is described in [4]. The trail routing problem is defined and an ILP formulation for routing is proposed in [6]. The study in [6], however, does not deal with the wavelength assignment problem and does not provide heuristics to solve the routing problem. None of the existing work that we are aware of investigate switch architectures for light-trail networks to demonstrate how light-trails can be made interoperable with lightpaths.

The rest of the paper is organized as follows. Section II of this paper introduces the light-trail architecture and formalizes the characteristics of communication over light-trails. Section

III proposes three switch architectures for seamless integration of light-trail and lightpath networks. Section IV defines the light-trail design problem and provides an ILP formulation. Section V deals with the trail-routing problem and proposes three heuristics for the same. The simulation results are discussed and analyzed in section VI following which the conclusions are presented in Section VII. Throughout this paper, we use the words light-trail and trail interchangeably.

II. Light-trail Architecture

A light-trail is similar to light path in that, it requires the establishment of a unidirectional optical circuit between the source and destination. The key difference is that some intermediate nodes can also receive and transmit data on the same channel in a time multiplexed manner.

Figure 1 shows a four node light-trail [1]. During operation, a node belonging to the trail obtains its turn to transmit based on a simple carrier-sensing medium access control protocol as discussed in [3], [5]. When a source transmits data towards a destination, the signal traverses all nodes downstream to it on the trail. At every node, the signal passes through a light-trail access unit (LAU) which consists of a splitter, a shutter and a combiner. The splitter and the combiner are attached to a receiver and a transmitter respectively. At the splitter, a part of the signal power is tapped by the receiver for local processing while the rest of the signal passes to the shutter. The shutter is a simple mirror-based optical attenuator that is configured to either block or let the wavelength pass through. If the node is the last or the first node on the trail, the shutter is configured in the off/blocking position, isolating this wavelength from the rest of the network. For all intermediate nodes on the trail, the shutter is in the on/pass-through position, letting the signal pass through. The signal, if not blocked by the shutter, travels through the combiner before exiting the node. The combiner allows the intermediate node to transmit data according to the light-trail access strategy.

The architecture utilizes an out of band control channel, which is dropped and processed at each node to actuate the shutters. The signaling channel carries information pertaining to the set up, tear down and dimensioning of light-trails, and is responsible for provisioning optical connections, ranging in duration from IP bursts to virtual circuits. The shutters are not reconfigured dynamically for every packet but is done on a longer time scale as traffic evolves. However, light-trails, using a flexible hardware platform and a simple over-laid protocol, effectively support dynamic traffic, by setting up new trails that vary slowly and tearing down unused trails in a distributed manner.

In some sense, the light-trail architecture may appear similar to the DQDB architecture specified in IEEE 802.6 standard. However, it is important to note the key differences between the two. DQDB is bi-directional, whereas the light trail has been chosen to be unidirectional to give the designer the flexibility to establish only those trails that will optimally meet the prevalent asymmetric traffic requirements. The second key difference is that DQDB is a physical topology whereas the collection of light-trails defines a virtual overlay, embedded on a mesh network to maximize the efficient use of the optical bandwidth based on traffic needs in the network.

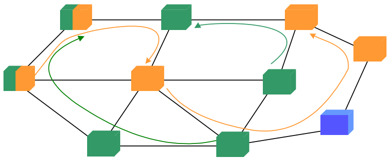


Fig. 2. A light-trail network. A node shaded the same color as the wavelength indicates that this node is active on the trail. A node with multiple shades is active on multiple trails and a node with different shade from the wavelength passing through it is inactive on the trail.

The characteristics of communication through light-trails can be explained more formally as follows. Consider a network topology as a directed graph G(V,E), with V as the vertex set and E as the edge set. Let a light-trail instance, which is just a simple path in a graph, be defined by the sequence LT = $\{v_1,v_2,v_3,v_4\}$ such that v_1, v_2, v_3, v_4 $\in$ V and (v_1,v_2), (v_2,v_3), (v_3,v_4) $\in$ E. Let R be the request matrix that denotes the value of the request between any node pair. A light-trail is a circuit that can carry multiple connection requests subject to the following constraints:

Containment Constraint: The light-trail can support any request (v_i,v_j) if v_i, v_j $\in$ LT and v_j is downstream of v_i in LT. That is, LT can possibly support the requests $\{(v_1,v_2)$, (v_1,v_3), (v_1,v_4), (v_2,v_3), (v_2,v_4), $(v_3,v_4)\}$

Capacity Constraint: The sum of the traffic supported by a light-trail is at most the capacity of a wavelength. If C = 5, R_{v_1,v_2} = 3, R_{v_1,v_3} = 3, R_{v_2,v_3} = 2 and the other requests are zero units, then LT can support one of the following: $\{(v_1,v_2)$, $(v_2,v_3)\}$, $\{(v_1,v_3)$, $(v_2,v_3)\}$, $\{(v_1,v_2)\}$, $\{(v_1,v_3)\}$, or $\{(v_2,v_3)\}$.

Request Assignment Constraint: A node pair's request cannot be split across multiple trails (unless it is more than the wavelength capacity) and hence a request is assigned to exactly one trail.

Trail Length Constraint: The data signal incurs a power loss on every node of the trail due to the LAU. Let trail size be defined as the number of hops in LT. Due to power budget constraints, the maximum number of nodes in a trail may be limited.

A node is said to be *active* on a trail if it has a request assigned to the trail while it is said to be *inactive* if it does not have a request assigned, but lies in the path of the trail. The collection of active nodes in a trail constitute the virtual topology of the trail. In the example above, $\{v_1,v_2,v_3,v_4\}$ refers to the physical topology of LT. If LT supports the request (v_1, v_4) alone, the virtual topology is $LT_v = \{v_1, v_4\}$ and the nodes v_2 and v_3 are *inactive* on LT. Figure 2 shows an example network that describes how light-trails can be used as a complete network solution.

III. Switch Architecture

The work in [5] discusses only node architecture for single-fiber-in, single-fiber-out networks as shown in Figure 1. In a typical mesh setting, multiple fibers pass through a node and hence the architecture of the optical crossconnect becomes important. An architecture for light-trail networks is introduced in

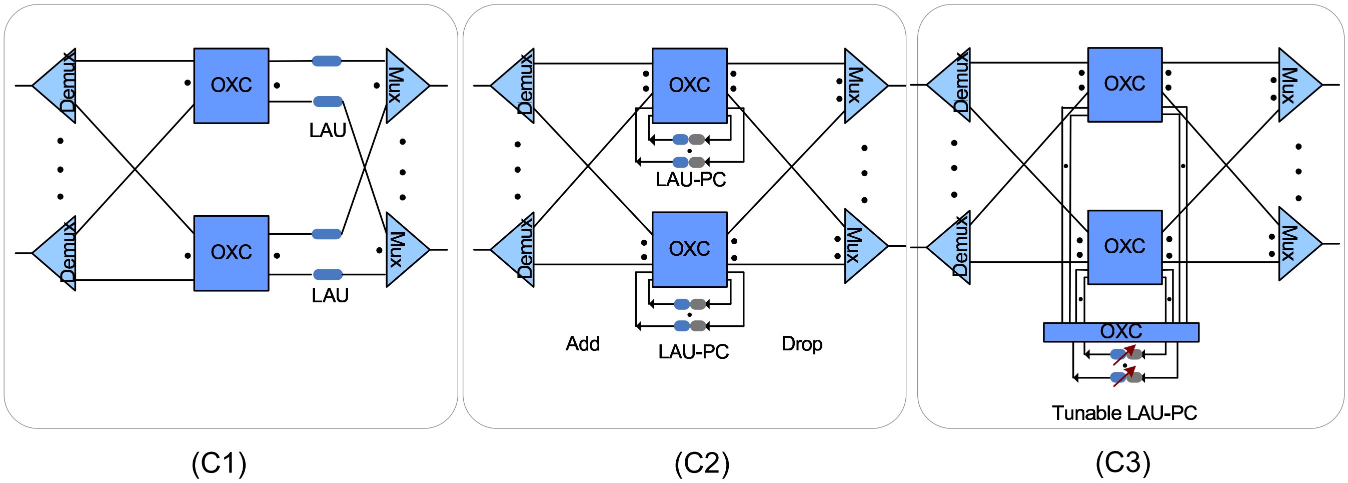


Fig. 3. The three suggested OXC configurations. LAU-PC refers to LAUs with local power compensation using SOAs. The optical layer costs increases from left to right, while the electronic layer cost increases from right to left.

the context of multicasting in [2]. However, it again considers only fiber-level switching and not wavelength-level switching.

We illustrate our motivation for different switch architectures using a simple example. Suppose that node A is active on trail t_1 and inactive on trail t_2. A design issue that needs to be addressed is whether t_2 needs to traverse the node A LAU or not. Having an optical switch that allows t_2 to bypass the node A LAU may seem to be the best thing to do to prevent t_2 from suffering unnecessary losses, but this leads to other transmission engineering issues. Namely, if t_2 bypasses node A LAU while t_1 traverses it, t_2 will have more signal strength than t_1 when both the signals exit node A. Signals of significantly different power levels may lead to amplification problems since one signal may saturate the EDFA because of its high power level while another may not get amplified much because of its low power level.

We could have two approaches to counter this problem. One solution is to let both t_1 and t_2 traverse the node A LAU so that they will have the same, but low power level. The second solution is to let t_2 bypass node A LAU through a switch, use a low gain amplifier like semiconductor optical amplifer (SOA) to compensate for the local losses t_1 suffers on node A LAU, so that when t_1 emerges out, it will have the same power level as t_2. In both the approaches, finally when the span losses and DWDM component losses have accumulated, amplification by a high gain EDFA becomes possible since both the trails have the same power level. However, the first solution may require higher gain amplification or more number of amplifer/regeneration stages than the second. Based on the two approaches here, we propose different cross-connect configurations (C1, C2, and C3) as shown in Figure 3 and analyze their capabilities and hardware requirements. Let F be the number of fibers and W be the number of wavelengths per node.

Configuration C1: In this configuration, the output of every wavelength plane switch goes to an LAU before getting multiplexed. It requires F Mux, F Demux, W $F \times F$ OXCs, and WF LAUs. Here, similar to the first approach mentioned earlier, all the signals entering a node, have to go through the LAUs on this node. This ensures that every signal emerging out has the same, but low power level.

Configuration C2: This configuration allows K LAU-PCs (LAUs with power compensation) for every wavelength with K being a maximum of F. It requires F Mux, F Demux, W $(F+K) \times (F+K)$ OXCs, WK LAUs, and WK SOAs.

Configuration C3: This configuration consists of two levels of cross-connects, tunable lasers and broadband receivers. This allows support for K LAU-PCs for the entire node, with K being a maximum of F. This configuration requires F Mux, F Demux, W $(F+K)\times(F+K)$ OXCs, $(WF+K)\times(WF+K)$ OXC, and K LAU with tunable transponders, and K SOAs.

In C1, all signals go through LAUs, and hence suffer high losses, requiring high gain EDFAs. However, only few of the signals that are tapped at the LAUs are required by the higher layer. In C2 and C3, some signals can bypass the LAU-PC units by directly being switched from the demultiplexer port to the appropriate multiplexer port. The signals that need local processing are routed to one of the LAU-PC ports to be tapped. These packets get recirculated into the fabric to be switched back out on the right multiplexer port. Note that, the LAU-PC units can be used to process both light-trails and lightpaths. This allows lightpaths to interoperate with the light-trails. Recall that a lightpath is just a special case of a light-trail where in the end points alone have access to the channel. If the cross-connects are configured so that the wavelength bypasses all of the intermediate LAUs, the circuit that results would be a lightpath. This gives the network designer the opportunity to seamlessly integrate lightpaths and light-trails as per the network requirements.

The switch sizes required by C3 is significantly bigger than that of C2 which in turn is bigger than that of C1. Large switches are harder to build since they need analog beam steering micromirrors whereas small switches can be realized using a variety of technologies. In C2 and C3, SOAs compensate for the local insertion losses caused by the splitter, shutter and combiner. SOAs are noisy, expensive and prone to cross-talk. However, providing a gain just before the LAU can help the local receiver achieve a better dynamic range.

The number of LAUs required by C2 and C3 are much fewer than the requirement of C1. For instance, if W=64, F=4, and if the node is active only on 8 trails, C1 still has to provision 256 LAUs while just 8 per wavelength are required in C_2 and 8 per node are required in C_3. In both C1 and C2, the transponders need to be deployed ahead of time so that they are available when needed. It is expensive to have a transponder deployed and not used while the associated signal is being ignored by the higher layer on the node. But, it can be argued that this cost is offset by the value addition in the capability to set up and take down circuits remotely and rapidly. For instance, in C3, if the number of circuits added/dropped at a node grows beyond K, remote configuration is not possible and the service provider has to manually provision equipments at the node. C1 is completely future-proof in this context while the same cannot be said about the other configurations. In C2, though the OXCs are reconfigurable, the transponders themselves are not. So, a decision needs to be made ahead of time regarding the number of transponders required for each wavelength. This may make the network planning constrained. These problems can be best avoided by making the transponders also reconfigurable as supported by C3. A detailed trade-off among these configurations will include issues related to capital expenses, operational expenses, expected traffic growth, ease of network deployment and management and is specific to a deployment scenario.

IV. LIGHT-TRAIL DESIGN

The transmission and switching granularity at the optical layer is at the wavelength or at the waveband level. This requires that a light path be provisioned for every request even if the request requirement is less than half the capacity, thereby leading to low wavelength utilizations. Traffic grooming can help address the low network utilization problem but the network loses transparency in the process [8]. The drawback with grooming is that the whole network is required to have a unique upper layer. The bit rates, frame formats, encodings and protocols should be interoperable for the entire network. In transparent networks, it is possible to assign different services to different light paths by running service-specific protocol on different wavelengths, but in a grooming network, all the services need to be supported by the higher layer with their varying traffic and quality requirements.

When grooming is performed, the packets need to be buffered on higher layers while it is being demultiplexed and remultiplexed, and since the packet processing delay is orders of magnitude higher than propagation delay, the latency incurred by groomed connections is typically high. Besides, the speed of the switching fabric may not scale well with the requirement for ever-increasing line speed and port count. Grooming requires that two layers (both electrical and optical) be optimized in a combined way so as to achieve significant gains, which is a much harder problem to handle than just optimizing the optical layer. This becomes even more apparent when dynamic routing is considered, where the more complex peer model is preferred for optimal routing. On the whole, electrical grooming is an expensive option and so there is some merit in investigating alternative methods for optimizing resource utilization.

Light-trails, in some sense, offer a form of grooming in the optical layer. The number of transmitters and receivers required to support a given traffic may be much fewer on light-trail networks than on all-optical light path networks. For instance, consider a trail $\{1,2,3\}$. If requests (1,2) and (1,3) are sub-wavelength in nature, and can be multiplexed on the same trail, the trail requires one transmitter on node 1 and one receiver each, on nodes 2 and 3. The transmitter on node 1 can send to both node 2 and node 3 in a time multiplexed fashion without the signals being terminated on any of the intermediate nodes. Moreover, multicasting is implemented at no additional cost since the transmission from node 1 to node 2 is also received at node 3 (any upstream activity is known at a downstream node on the trail). However, if the same traffic is carried by conventional networks, each request pair would be treated as a different light path, thereby requiring two transmitters and two receivers. Thus, light-trails, while not resorting to grooming in the electrical layer, is still able to provide improvement in network utilization and resource consumption by packing multiple connections onto the same trail. The more the number of requests multiplexed onto a trail, the more the gains one can expect.

The cost incurred in the process of multiplexing is in the additional hardware (which is not much) and delays caused due to the medium access protocol (which is not high) while the extent of multiplexing gain is decided by the nature of the carried traffic and the packing algorithm employed. If a specific request is large and cannot be multiplexed with other requests, then the corresponding light-trail supports only one request. The cross-connect architecture C2 and C3 that we have proposed ensures that the light-trail, in this case, behaves no different from the traditional light-path as discussed in the previous section.

An important next step would be to quantify the multiplexing gains achieved using light-trails. The previous work that is most relevant to this study is [6]. The work in [6] provides ILP formulation for the trail routing problem alone and does not consider it in the context of crossconnect architectures. Besides, it does not take wavelength assignment into account and does not minimize the wavelength usage or the communication equipment resources as we do here. Also, it does not provide heuristics for network design. The current work suggests three heuristic methods and studies the effect of trail sizes on multiplexing gains. First, we state the ILP formulation. The light-trail routing and wavelength assignment problem can be defined as follows: Given a network graph G(V,E), a request matrix R(V×V), identify the trails required to carry all requests so as to minimize resource utilization. The objective is to minimize the optical layer and the electrical layer costs. The electrical layer costs include the number of transmitters and receivers and the optical layer costs include the number of different wavelengths required in the network. By minimizing the number of communication equipments required to carry a given static traffic, more

resources are left available in the network to handle the incremental and dynamic traffic efficiently. We define our notation first.
N - number of nodes in the network (data)
C - capacity of a wavelength (data)
W - number of wavelengths on each link of capacity C (data)
S - maximum allowable trail size in the network (data)
LT - set of all possible light-trails in the network of size S or less (data)
LT_t - an instance of a light-trail $LT_t \in LT$ (data)
LT_t^r - set of requests that can be supported by LT_t based only on the containment constraint (data)
$LT_t^{i,j}$ - assumes 1 if trail LT_t traverses link (i, j), 0 otherwise (data)
$s, d = 1..N$ number assigned to each node (index)
$R_{s,d}$ - traffic request between node s and node d, assumed to be sub-wavelength(data)
t, t_1, t_2 =1..$||LT||$ - number assigned to each light-trail(index)
$w = 1..W$ - number assigned to each wavelength (index)
$X_t^{s,d}$ - assumes 1 if (s, d) is assigned to t, 0 otherwise (variable)
T_t - assumes 1 if trail t supports at least one node pair, 0 otherwise (variable)
L_t^w - assumes 1 if wavelength w is assigned to trail t, 0 otherwise (variable)
$W_t^{r,s}$ - assumes 1 if node s on trail t needs a receiver, 0 otherwise (variable)
$W_t^{x,s}$ - assumes 1 if node s on trail t needs a transmitter, 0 otherwise (variable)
U_t - number of transmitters and receivers required for the trail t (variable)

Objective :

$$Min \quad \sum_t U_t \tag{1}$$

Request Assignment Constraint :

$$\sum_t X_t^{s,d} - 1 = 0 \qquad \forall s, d \in V \tag{2}$$

Capacity Constraint :

$$\sum_{(s,d) \in LT_t^r} R_{s,d} X_t^{s,d} - C \le 0 \ \ \forall t \tag{3}$$

Trail Assignment Constraint :

$$T_t - X_t^{s,d} \ge 0 \ \ \forall (s, d) \in LT_t^r, \ \forall t \tag{4}$$

Wavelength Continuity Constraint :

$$\sum_w L_t^w - T_t = 0 \ \ \forall t \tag{5}$$

Wavelength Assignment Constraint :

$$\sum_t L_t^w - 1 \le 0 \ \ \forall w, \{t : LT_t^{i,j} = 1\}, \forall (i, j) \in E \tag{6}$$

Receiver Usage Constraint :

$$W_t^{r,s} - X_t^{d,s} \ge 0 \ \ \forall (d, s) \ \in LT_t^r, \ \forall t \tag{7}$$

Transmitter Usage Constraint :

$$W_t^{x,s} - X_t^{s,d} \ge 0 \ \ \forall (s, d) \ \in LT_t^r, \ \forall t \tag{8}$$

Resources Usage Constraint :

$$U_t - \sum_{s \ \in LT_t} W_t^{r,s} - \sum_{s \ \in LT_t} W_t^{x,s} = 0 \ \ \forall t \tag{9}$$

Variable Range Constaint :

$$X_t^{s,d}, T_t, L_t^w, W_t^{r,s}, W_t^{x,s} \in (0, 1), U_t \in I \tag{10}$$

Wavelength continuity constraint ensures that every trail is given exactly one wavelength throughout its route and the wavelength assignment constraint prevents two trails traversing the same link from being assigned the same wavelength. Equation (7) accounts for the number of transmitters required on a trail t and equation (8) accounts for the number of receivers required on a trail t. Equation (9) keeps track of the total number of transmitters and receivers counted separately. The objective function minimizes the total number of equipments required to support the entire traffic. We first fix the number of wavelengths and check if ILP yields a solution. If a solution is not found, then larger number of wavelengths are attempted. Otherwise, fewer number of wavelengths are attempted. This procedure is iterated until the minimum number of wavelengths is found. Let T be the number of trails required to carry a traffic matrix R. Then, it can be seen that,

$$\lceil \frac{\sum_{s,d} R_{s,d}}{C} \rceil \le \ T \ \le \sum_{s,d} \lceil \frac{R_{s,d}}{C} \rceil$$

For $R_{s,d} << C$, the bounds are loose, but as $R_{s,d}$ approaches C, the bounds become tighter. The bounds become exact when $R_{s,d} = C \ \ \forall s, d \in V$, in which case, the ILP reduces to the static lightpath RWA problem.

V. The trail routing and wavelength assignment

The static lightpath establishment problem is a well studied problem. In [10], an approach based on relaxation of ILP, followed by a randomized rounding was chosen. In [12], a set of alternate paths that include routes longer than the shortest paths is considered with an objective to minimize the number of fibers in a multi-fiber network. In [11], a shortest-hop path is randomly chosen for a source-destination pair. The route for the pair of nodes is switched to an alternate path if doing so leads to reduction in the load of the most congested link in the original shortest path route. We call this last heuristic LP-LB (LP heuristic with load balancing) and we compare the results from our heuristics with the method suggested in [11].

The light-trail RWA problem belongs to the NPC class because of the wavelength assignment problem and hence to make it tractable, it is decoupled into two subproblems (1) trail routing and (2) wavelength assignment, and each problem is solved independently. Trail routing can be formally defined as follows. Given a network G(V,E), a request matrix R(V×V), a cost function that assigns a cost to each trail (typically, to be the sum of the cost of the edges that are traversed by the trail), identify the minimum cost packing of requests to trails. In the current work, we assume that the cost of all the links are identical and hence, shortest paths are defined in terms of number of hops. Heuristics for the trail routing problem have not be studied yet and we propose three heuristics for it. Our heuristics work as follows:

- The traffic matrix preprocessing step is performed to address the trail length constraint (refer Section II).
- The requests are routed on the trails using the LT-LB, LT-RT, and LT-SP heuristics each using the increasing, decreasing and 0-1 knapsack packing methods outlined below.
- A postprocessing step is performed to improve the obtained solution.
- Finally, each trail is assigned a wavelength.

A. Preprocessing

Let S be the maximum allowable trail size due to systems engineering constraints, let D be the network diameter and $d_{i,j}$ be the shortest distance between the node pair (i, j). If $S < D$, it may not be possible to carry the requests for node pairs (i, j) whose $d_{i,j} > S$ in a single hop. In this case, we use a minor variant of the traffic rearrangement pre-processing step suggested in [6]. For every node pair (i, j) with nonzero traffic $R_{i,j}$, and with $d_{i,j} >$ S, an intermediate node k is found such that $d_{i,k} \leq$ S, $d_{k,j} \leq S$ and the traffic matrix is modified such that, $R_{i,k}$ += $R_{i,j}$, $R_{k,j}$ += $R_{i,j}$, and $R_{i,j}$ = 0. k is first searched on shortest path from i to j and if such a node is not found, an arbitrary node conforming to the above conditions is identified. This allows the traffic to follow multiple hops to reach their destination. At the end of the pre-processing step, all node pairs with non-zero traffic between them can be reached with the trails allowed by the system. If $S > D$, multihop is not allowed and the signals remain purely in the optical domain.

B. Trail Routing

`LT-LB heuristic`

The primary focus of this heuristic is load balancing (hence LB) on links while packing trails. It finds the shortest path length between any node pair with non-zero request and sorts them in the non-increasing order of their path lengths. It then chooses the node pair which are farthest apart, and finds all possible shortest routes between them. It selects the route with the least load similar to [12], and tries to pack as many other requests as possible on the same route adhering to the containment and the capacity constraints. Only the requests that exceed the value of a wavelength capacity, can be split into multiple streams. All the other requests adhere to the request assignment constraint. The procedure is repeated by sequentially stepping through the ordered list of requests until each of them is packed onto a trail. The details of the algorithm are presented in Figure 4.

`LT-RT heuristic`

The primary objective of this heuristic is to minimize the receiver and transmitter requirements (hence RT). Our approach here is based on our observation of the results yielded by the ILP formulation for various topology and traffic. The set of all possible trails of size up to S are given as input to the ILP for various values of S. We find that almost all the time, the optimal solution involves primarily trails of size exactly S. So, for this heuristic, we give all possible trails of length exactly S as the input. The heuristic works as follows.

Trails of size S are listed out and each trail is packed to the best possible extent using the increasing, decreasing or knapsack packing methods conforming to the containment and capacity constraints. As described in Section IV, whenever a node is assigned to send data to multiple nodes or receive data from multiple nodes in a time multiplexed way, there is savings in communication equipment. The savings for each trail is computed and the trail that leads to the maximum savings is chosen. In the request matrix, the requests that have already been assigned to this chosen trail are zeroed out, and the remaining trails are again packed with the new request matrix. The procedure repeats until all requests are packed. The details of the algorithm are presented in Figure 5.

`LT-SP heuristic`

The LT-SP heuristic is similar to LT-LB heuristic except that it does not take network load into account. That is, it sorts the requests in the non-increasing order of shortest path lengths (hence SP), scans the requests sequentially and it tries to multiplex as many requests as possible along every route. It considers multiple possible shortest routes between a node pair and chooses the routes which packs the best. The rest of the details are identical to the description of the LT-LB heuristic.

C. Postprocessing

For all the three heuristics, an additional post-processing step is implemented. The trails that are in the final solution set are scanned sequentially. It is possible that there are some intermediate nodes in each trail that are inactive on that trail. They are merely there because they happen to be in the path of another request being served by the trail. These nodes are marked and the OXCs on these nodes are configured so as to let the packets sent on this trail bypass this node. Based on the requests that are carried by each trail, the required number of transmitters and receivers are counted individually. The total number of equipments, defined to be the sum of transmitters and receivers, is logged for various values of S.

D. Wavelength Assignment

After all requests are routed on the graph, each trail needs to be assigned a wavelength according to the wavelength assignment and continuity constraints. We first construct an auxiliary graph, G', such that each light-trail in the system is represented by a node in G'. An undirected edge between two nodes is introduced in G' if the trails corresponding to the two nodes pass through a common physical link. Now, the nodes in the auxiliary graph are colored using the largest-first algorithm discussed in [9].

VI. Simulation Results

The proposed heuristics are implemented and the design algorithms are run on various networks of different diameters

STEP 1: Run Flloyd Warshall's algorithm and identify the shortest path lengths between any node pair in the network.

STEP 2: Let L be the list of all non-zero request pairs arranged in the non-increasing order of their shortest path lengths. Let T be the list of trails required to carry the traffic. Initialize $T = \emptyset$

STEP 3: IF L is empty, STOP. ELSE, PROCEED.

STEP 4: Consider the first item (s,d) in L and its request value $R_{s,d}$. Find all possible shortest routes between (s,d) and assign it to list SP.

STEP 5: FOR each route $LT_i \in SP$, define AR_i to be the final assigned requests to this route, and VAR_i to be the value of the total traffic packed onto LT_i. Initialize AR_i = $\{(s,d)\}$.

STEP 6: IF $R_{s,d} < C$, assign k = $R_{s,d}$. GOTO STEP 7. ELSE repeat steps 7, 11 and 12 with k = C, $\lfloor \frac{R_{s,d}}{C} \rfloor$ number of times and finally once more with k = $R_{s,d}$ - $C\lfloor \frac{R_{s,d}}{C} \rfloor$

STEP 7: Initialize VAR_i = k. Repeat steps 8,9 and 10 for each i.

STEP 8: Find $Load_i$ which is the maximum of the loads (number of trails) found on all the links of the route LT_i.

STEP 9: IF $VAR_i < C$, for LT_i, generate LT_i^{req}, which corresponds to all the requests, consistent with the containment constraint. Remove (s,d) from LT_i^{req} since it has already been assigned to this route while AR_i was initialized in STEP 5.

STEP 10: IF $VAR_i < C$, pack other requests in this route LT_i consistent with the capacity constraint. This can be done in one of the three possible ways.
Decreasing Packing: Arrange all items $\in$ to LT_i^{req} in non increasing order of their request value. Select all the items from the left until taking any extra item would defy the capacity constraint. Update AR_i and VAR_i if any additional request has been packed.
Increasing Packing: Arrange all items $\in$ to LT_i^{req} in non decreasing order of their request value. Select all the items from the left until taking any extra item would defy the capacity constraint. Update AR_i and VAR_i if any additional request has been packed.
0-1 Knapsack Packing: Assuming the capacities and request values are integral, a dynamic programming formulation of the 0-1 knapsack packing is used to squeeze as many requests as possible into LT_i. Update AR_i and VAR_i if any additional request has been packed.

STEP 11: Among LT_i, select the route which corresponds to minimum $Load_i$. In case of a tie, select the route which corresponds to the maximum packing, VAR_i. If there is still a tie, break arbitrarily.

STEP 12: Update the traffic matrix R for all the requests carried by the chosen trail. FOR every request, that has been completely satisfied, remove the corresponding item from L. Update T. Update network load.

STEP 13: Go to STEP 3.

Fig. 4. LT-LB heuristic

STEP 1: Let L be the list of all routes of length S in the network. Let T be the list of trails required to carry the traffic. Initialize $T = \emptyset$

STEP 2: FOR every s,$d \in V$ with $R_{s,d} \geq C$, repeat STEP 3 $\lfloor \frac{R_{s,d}}{C} \rfloor$ number of times.

STEP 3: List all shortest routes between s and d. Choose the route with the minimum congestion on the most loaded link. Update $R_{s,d}$ = $R_{s,d}$ - C. Update T.

STEP 4: IF $R_{s,d} = 0 \quad \forall (s,d) \in V$ GOTO STEP 11. Else PROCEED.

STEP 5: FOR each route $LT_i \in L$, define AR_i to be the final assigned requests to this route, and VAR_i to be the total traffic carried by LT_i. Initialize AR_i = $\{\emptyset\}.VAR_i$ = 0. Repeat STEPS 6-8 for each i.

STEP 6: Find $Load_i$ which is the maximum of the loads found on all the links of the route LT_i.

STEP 7: For route LT_i , generate LT_i^{req} consistent with the containment constraint. Pack as many requests as possible on LT_i using the following methods:
Decreasing Packing
Increasing Packing
0-1 Knapsack Packing
The methods are outlined for the LT-LB heuristic in Figure 4. Update AR_i that holds the final requests that is assigned to this route and VAR_i which shows the total traffic that is packed into the route.

STEP 8: FOR each LT_i, calculate $Savings_i = x_i - y_i$ where x_i refers to the number of transmitters and receivers (counted individually) used on this trail and y_i refers to the number of transmitters and receivers that my have been used if each of the packed requests were routed separately and without being groomed.

STEP 9: Among LT_i, select the route which corresponds to maximum $Savings_i$. In case of a tie, select the route which corresponds to minimum $Load_i$. If there is still a tie, break arbitrarily.

STEP 10: Update traffic matrix R for all the requests carried by the chosen trail in STEP 9. For every request, that has been completely satisfied, remove the corresponding item from L. Update T. Update network load. Go to STEP 4.

STEP 11: Scan all the trails $\in$ T sequentially. Recall that the list L has all routes of length S. It is possible that some trails may just serve a request of a few nodes located somewhere in the middle of the trail. The endnodes may not be active on this trail. So, prune the trail from the left until the first node that is involved with transmission.

STEP 12. Repeat the pruning process from the right until the first node that is involved with reception. For instance, say N = 4, and we have a trail T_i = $\{1,2,3,4\}$ that serves the request (2,3) alone, then, after pruning using STEPs 11 and 12, T_i becomes $\{2,3\}$.

Fig. 5. LT-RT heuristic

shown in Figure 6. Every link is assumed to have two fibers to facilitate bi-directional communication. This allows the same wavelength to be used in the forward and reverse direction. Wavelength conversion is not present in the network. Every link is provisioned as many wavelengths as is required. We assume the traffic to be primarily sub-wavelength. The capacity of a wavelength is arbitrarily assigned to be 48 units. All the simulations described below were done with three different packing methods. It was observed that increasing-packing does a little better than knapsack-packing which in turn does consistently much better than decreasing-packing. For lack of space, we do not show all our graphs here, but only the results obtained with the increasing-packing method.

First, we run the ILP formulation described in section III on the NSFNET using the CPLEX Linear Optimizer 8.1.0. We give the set of all possible paths of hop length 3 or less (S=3) as input to the ILP. For NSFNET, there are 354 such paths. Since D=3 for NSFNET, there is no need for traffic rearrangement. In NSFNET, there are 182 possible source-destination pairs. With probability p, we make a node pair active, and generate requests that are uniformly distributed between 5 and 15 units. We call this value of p the load of the network since it decides the number of active pairs in the network. We vary the value of p from 0.1 to 1, and solve the ILP to find the optimal RWA. Heuristics LT-SP, LT-LB, LT-RT and LP-LB are run on the same network under identical traffic conditions and the generated results are plotted in Figure 7.

We see that for small values of load, LT-SP, LT-LB and LT-RT yield results very similar to that of the ILP in assessing the equipment and wavelength requirements. For peak load, when every node pair in the network is active, the maximum error margin for LT-RT in estimating equipment usage is about 6% while for LT-LB and LT-SP, it is about 11%. LT-RT does better than LT-SP and LT-LB in conserving equipments, because LT-RT is designed for that. The number of wavelengths required by LT-LB is at most one in excess of the optimal number of wavelengths. The wavelength requirements of the LT-LB approach is lesser than that of LT-RT and LT-SP, because LT-LB reduces congestion by spreading the load evenly. LP-LB cannot groom requests and hence it has to provide a full light path for every fractional request. So, the number of equipments required will be exactly twice the number of active node pairs. Due to lack of multiplexing, the number of circuits required is more and hence the number of wavelengths required is also much higher for the LP-LB approach. The non grooming lightpath approach consumes at least 1.5 times more number of wavelengths and equipments than the light-trail approach.

We study the impact of trail size on the network design problem when load p of the network equals 1. We provide 1000 instances of randomly generated traffic matrix, where the requests are uniformly distributed in the range (0,24) units. We vary the value of the trail size S from 3 to 8 and observe the performance of LT-LB and LT-RT heuristics on NJLATA, NSFNET, COST239 and ARPANET. If $S < D$, then the preprocessing step outlined in Section V.A is performed. The average wavelength and equipment requirements are plotted in Figures 8 and 9.

In general, if trail sizes are too small, multiplexing capability is limited due to the inability to reach nodes which are beyond the trail size but have the requests that could have been possibly included on this trail. However, if trail sizes are too large, then requests may be packed inefficiently and hence the connections may end up having large number of wavelength links. Recall that in LT-RT, trail sizes of exactly S are provided and hence it can be seen that the wavelength consumption of this heuristic grows with increasing S due to increasing wavelength links. LT-LB heuristic always tries to balance the load over multiple shortest paths, and hence the wavelength requirement is almost a constant.

From Figure 8(a) and 9(a), equipment requirements for NJLATA, COST239 and NSFNET do not vary much as a function of trail size. Figure 8(a) shows a near constant wavelength requirement for these networks. If these networks are designed to support long trail sizes, it may lead to severe power budget constraints. If the trail sizes supported are too short, multihop communication may be required. Electronic grooming has lots of disadvantages as stated in section IV and grooming switches are very expensive. Keeping in view that all-optical communications has strong advantages due to its transparency, a cost-effective operation point may be to engineer transmission systems that support trail sizes up to the diameter of the network ($S = D$). This avoids electrical grooming and does not suffer from heavy power budget constraints.

For ARPANET, there is a prominent increase in equipment requirements as S increases from 3 to 4. This can be explained as follows. The total number of possible active node pairs at peak load in ARPANET is 360 and the diameter of the network is 6. At S = 3, the total number of active node pairs is only 262, since the requests which need to traverse more than three hops have been electronically groomed. Since the total number of active pairs has been markedly reduced, the equipment requirements are also reduced. At S = 4, the number of active node pairs is a little higher, about 323, and the multiplexing capability in the optical layer also reduces a bit because electronic grooming leaves behind request values that cannot be efficiently packed. Hence, the equipment requirements increase. As expected, the number of wavelengths used also increases when S increases from 3 to 4. This trend is not seen in other networks since the diameters of other networks are small and S values we have considered for simulations are $\geq D - 1$ for these networks.

It can be inferred from the results for ARPANET that if $S < D$, wavelength, transmitter and receiver requirements are low, but the electronic grooming requirements are high. If $S \geq D$, equipment and wavelength requirements are high, but there is no grooming requirement. To determine an optimal network operation point, an approach similar to that specified in [13] is adopted. Let C_λ refer to the capital and operational cost of maintaining a wavelength in the network. The capital cost includes the cost of provisioning a transmitter and receiver for every link, and the cost of a wavelength plane switch on every crossconnect. The operational cost incorporates the monitoring, maintenance and management overhead to support a wavelength. Let C_g denote the cost to employ grooming capability in the network per node. Define the ratio R,

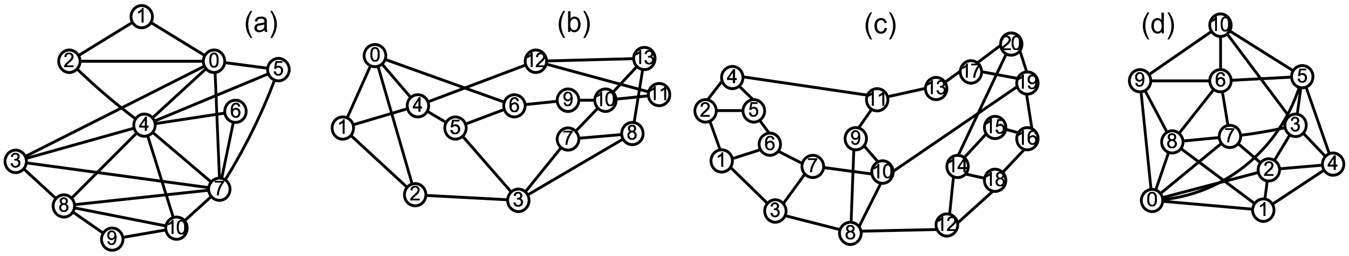
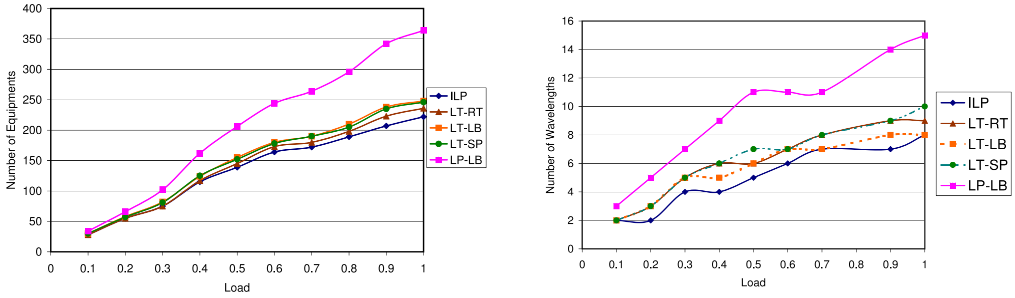
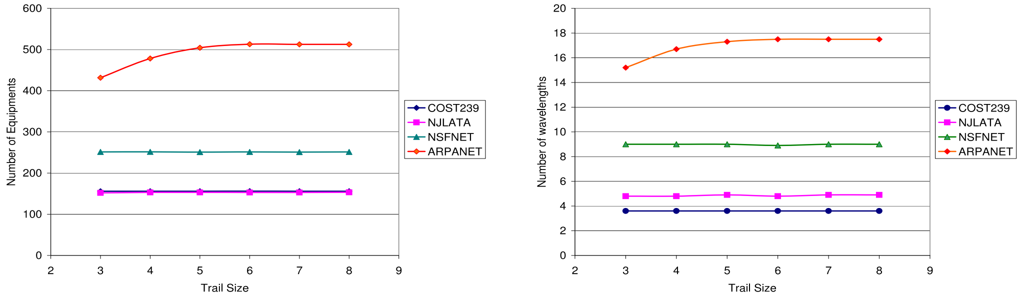


Fig. 6. Test networks used for simulations (a) NJLATA, D = 4 (b) NSFNET, D = 3 (c) ARPANET, D = 6 (d) COST239, D = 3



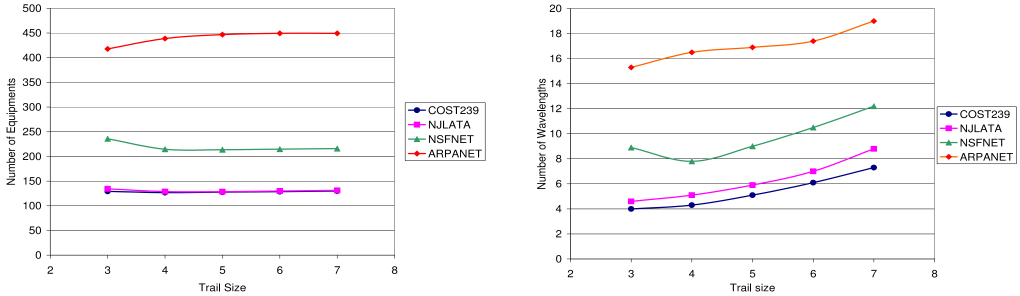
(a) (b)

Fig. 7. Comparison of optimal solution and heuristic results for wavelength and equipment requirements as a function of load on the NSFNET topology.



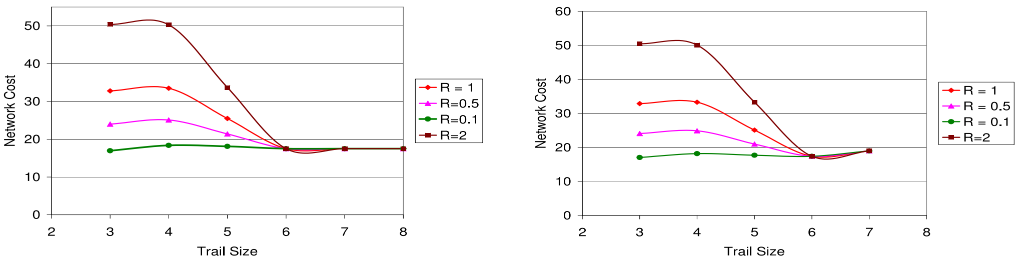
(a) (b)

Fig. 8. The equipment and wavelength requirements estimated by the LT-LB heuristic for peak loads on various networks as a function of trail size



(a) (b)

Fig. 9. The equipment and wavelength requirements estimated by the LT-RT heuristic for peak loads on various networks as a function of trail size



(a) (b)

Fig. 10. The network cost as a function of trail size for various values of R for the ARPANET operating at peak load (a) LT-LB heuristic (b) LT-RT heuristic. S = 6 and S =3 are the optimal operation points for $R \geq 1$ and $R \leq 0.1$ respectively.

$$R = \frac{C_g}{C_\lambda}$$

The cost of the network C_n can be be computed as

$$C_n = n_g \times C_g + n_\lambda \times C_\lambda = (n_g \times R + n_\lambda) \times C_\lambda$$

where n_g refers to the number of nodes equipped with grooming capability and n_λ refers to the number of wavelengths supported in the network. Normalizing the cost by C_λ,

$$C_n = n_g \times R + n_\lambda \tag{11}$$

We use equation (11), and results from Figure 8 (b) and 9 (b) to evaluate the cost of designing ARPANET as a function of the trail size for various values of R. We observe in Figure 10 that when the cost of a wavelength channel is cheaper compared to the cost of a grooming node ($R \geq 1$), the network cost decreases with increasing trail size for $S < D$. In figure 10(a), the network cost saturates at $S = D$, while in figure 10(b), the network cost increases beyond $S = D$. We conclude that trail sizes of up to D need to be supported by the network for cost-effective operation. However, when grooming becomes more economical than wavelength maintenance ($R \leq 0.1$), supporting only smaller sized trails and employing grooming may be a better option. Note that grooming algorithms based on nodal-degree or amount of by-pass traffic, as discussed in [13], may yield different operation points for different values of R. We leave this study for future work.

VII. Conclusions

We studied the light-trail network design problem and formulated ILPs to minimize equipments and wavelengths required to support a given traffic. We designed three different heuristics for solving the trail routing problem and showed that our heuristics yield results very close to the optimal solution. We observed that the light-trail based approach is more economical than lightpath based approach in the stated scenarios. We studied the effect of trail size on resource requirements. We concluded that if we set the allowable trail size to at least the diameter of the network, we derive the benefits of all-optical communications and still achieve reduced network costs. We introduced three crossconnect architectures that enable seamless integration of light-trail and lightpath networks and discussed their capabilities and limitations.

The light-trail design problem resembles a packing problem, and in general, appears to be hard, but its complexity class membership needs to be established formally which would be our next step. Solving the dynamic routing problem would be harder but practically more useful than the static problem and hence requires further work. The true merits of the new architecture needs to be evaluated elaborately by comparing light-trail networks and lightpath networks with full grooming and sparse grooming capabilities. A good direction to proceed would be to identify the specific conditions and scenarios involving topologies, traffic, equipment requirements and operational costs, under which the light-trail architecture would complement and supplement the lightpath architecture.

References

[1] A. Gumaste, I. Chlamtac, *Light-trails: A novel conceptual framework for conducting optical communications* , Proceedings of HPSR 2003 Torino, Italy June 2003.

[2] A. Gumaste and I. Chlamtac, *Mesh implementation of light-trails: a solution to IP-centric communication*, 12th International Conference on Computer Communication and Networks, 2003.

[3] S. Balasubramanian, A. E. Kamal, A. K. Somani,*Medium Access Control Protocols For light-trail and Light Bus Networks*, Proceeding of 8th IFIP Working Conference on Optical Network Design and Modeling, Feb 2-4, 2004.

[4] A. Gumaste, G. Kuper, I. Chlamtac, *Optimizing light-trail assignment to WDM networks for dynamic IP centric traffic*, 13th workshop on Local and Metropolitan Area Networks, LANMAN, 2004.

[5] A. Gumaste, and I. Chlamtac, *Light-trails: an optical solution for IP transport*, Journal of optical networking, Vol. 3, No.4, April 2004.

[6] J. Fang, W. He and A. K. Somani, *Optimal light-trail Design in WDM Optical Networks*, the proceedings of ICC 2004, vol.3, pp.1699-1703, Paris, 2004.

[7] R. Ramaswami, K.N. Sivarajan, *Optical Networks: A practical perspective*, second edition, Morgan Kaufmann publishers,2002.

[8] T. Cinkler, *Traffic and lambda grooming*, IEEE Network, pp. 16-21 March/April 2003.

[9] H. Zang, J.P.Jue, B. Mukherjee, *A review of routing and wavelength assignment approaches fro wavelength routed optical WDM networks*, Optical Networks magazine, pp. 46-60, January 2000.

[10] D. Banerjee and D. Mukherjee, *A practical approach for routing and wavelength assignment in large wavelength-routed optical networks*, IEEE Journal on Selected Areas in Communications, vol. 14, No. 5, pp. 903-908, 1996

[11] S. Baroni and P. Bayvel, *Wavelength Requirements in arbitrarily connected wavelength routed optical networks*, IEEE/OSA Journal of Lightwave Technology, Vol. 15, No. 2, pp. 242-251, 1997.

[12] M. Alanyali and E. Ayanoglu,*Provisioning algorithms for WDM optical networks*, IEEE/ACM Transactions on Networking, Vol 7, No 5, pp. 767-778, 1999.

[13] K.Zhu, H. Zang, B. Mukherjee, *Design of WDM mesh networks with sparse grooming capability*, in Proc. IEEE Globecom 2002, Vol. 3, pp. 2696-2700, November, 2002.