Light-frames: A Pragmatic Framework for Optical Packet Transport

Ashwin Gumaste, Imrich Chlamtac and Jason P. Jue
Center for Advance Telecommunication Services and Systems (CATSS)
The University of Texas, Dallas
Richardson, USA
Email: ashwing@ieee.org

Abstract: **In this paper we propose an architecture and an algorithm for the realization of a pragmatic framework for optical packet transport. The architecture enables the transport of IP packets over optical frames in a network. While doing so, it relaxes the need for address recognition as well as for high speed switching, the two key hindering factors that have prevented contemporary optical packet transport solutions from being deployed. We propose the light-frames architecture for sub-lambda level provisioning of optical paths. The idea is to create a logical topology that allows N^2 connectivity using optical paths that are by themselves sub-lambda provisionable yielding in packet transport. We show that by using substantially fewer resources, we can provide N^2 connectivity in arbitrary graphs to support optical packet transport.**

Keywords: Light-trail, packet switching, IP over WDM

I. MOTIVATION

The promulgation of IP communication in the past decade and the parallel exploitation of the large bandwidth offered by optical fiber have motivated proposals for the pragmatic implementation of optical packet communication by coalescing IP and WDM [6-7, 11-12]. This optical packet communication concept [11-12] has been considered in its more seemingly efficient form as native IP packets over WDM, a method to efficiently utilize the enormous bandwidth offered by the optical fiber while adhering to the limitations posed by the opto-electronic bottleneck (WDM). To furnish a solution that implements optical packet communication, there are two primary functions at the optical layer that need to be carried out – address recognition in the optical domain and high speed switching of packets based on optical address recognition. Both of these techniques are in the nascent stages of development, and plausible cost effective technology solutions are not available. This means that there is currently no solution that is technologically feasible and economically viable for implementing optical packet transport. We propose a solution that is built on the concept of light-trails [1-4], further enhancing light-trails to accommodate packet transport. *A light-trail is a generalization of a lightpath in which data can be inserted or removed at any node along the path*. Light-trails are a group of nodes capable of achieving dynamic provisioning in an optical path through an out of band control channel (overlaid protocol), leading to multiple source-destination pairs being able to establish time differentiated connections over the light-trail while eliminating the need for high speed switching. The overlaid protocol for management within light-trails causes coarse granularity. Thus in their fundamental manifestation, light-trails do not solve our requirement for optical packet transport but still provide for dynamic communication at each node. To support packet transport we propose the concept of *light-frame communication*. Light-frame communication is a first pragmatic implementation of packet level transport at the optical layer. The proposed framework uses an architecture that has evolved from the basic light-trail's concept, hence uses mature available components for implementation. The light-frames concept results in the formulation of a logical topology that addresses the N^2 connectivity issue, by intelligently providing transport routes to packet flows in a network thereby circumventing the need for fast switching as well as optical layer packet/header recognition schemes. In this paper we only showcase the nodal configuration and study micro-level details. The macro network analysis is a subject for further study.

II. LIGHT-FRAMES COMMUNICATION CONCEPT

Light-frames present a framework for conducting optical communication at the packet level. The light-frame (LF) framework presents an architecture, a protocol, and a configuration methodology that enables the successful transport of packets in the optical domain. The architecture of nodes that supports LFs is based on two principles: (a) *admittance and bifurcation* and (b) *drop and continue*. A node can admit data (LF) from a particular path or can bifurcate data all optically, to a particular path. Similarly, a node can drop data as well as continue the data flow (pass through) entirely in the optical domain. These two philosophies coalesced together produce a node connection methodology that realizes in a packet transportable network.

Node connection methodology – Threads and Strings: The LF framework consists of nodes that are interconnected in two ways: through threads or through strings. Let us first define these concepts from a connection perspective. A string is a collection of nodes that are interconnected in a linear way (like a trail of nodes). The first node of the string is called the *convener* while the last node is called the *end node*. The string is a unidirectional optical bus. Each node along the string can drop and continue data coming from an upstream node, and also has the ability to insert its own data to downstream nodes. Multiple strings can be connected to each other by threads which connect two nodes in different strings that are separated by a one hop (logical) path. A node in one string can thus send data all optically to a node in another string through a thread. Shown in Fig. 1 is the light-frame framework. Shown in thickened lines are strings, while those in dashes are threads.

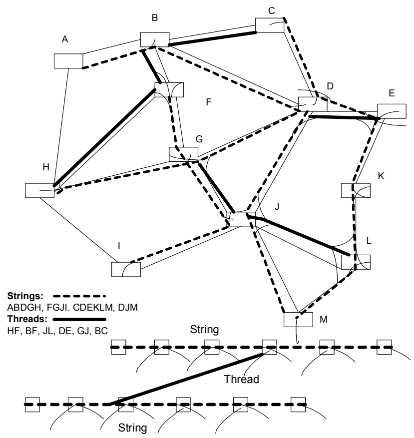

Fig. 1 Light-frame supportive network of strings and threads

The inherent idea in LF communication is to create a virtual topology whereby a source-destination pair falls into a path, which can be composed of multiple sub-paths (of strings and threads), the nodes of which use one of the two principles of drop and continue or admittance and bifurcation. The principle behind the virtual topology formulation is to have a number of opened *strings* that allow packets to be transported without the need for switching, and then to interconnect these strings and to ensure collision-free communication using the admittance and bifurcation approach by interconnects that are called *threads*. Hence a string of 'n' nodes can by itself support logical connections for ($^{n-1}C_2$) s-d pairs. If a node sends a packet into a string, then all of the downstream nodes will receive a copy of this packet at their local access interface. It is up to the nodes (based on a MAC level address match) to decide whether to keep the packet or discard it. The principle of this broadcast methodology is based on the drop and continue architecture (shown later). The second level of hierarchy in the creation of a network fabric entails in joining multiple strings together and this is done by *threads*. Threads are connected to nodes in a string and traffic is either *bifurcated* from a string to a thread, or *admitted* into a string from a thread based on the virtual topology requirement of the graph. The process of strings and threads leads to a WDM system where a node can select one of the possible strings it resides on and guarantee that the packet will travel (without intelligent switching or routing) to the destination through this preset maze of strings and threads.

III. NODE ARCHITECTURE TO SUPPORT LIGHT-FRAMES

The nodal architecture to support light-frame communication is shown in Fig. 2. This schematic is for a per wavelength basis and can be extended to a WDM system, not shown here to preserve clarity.

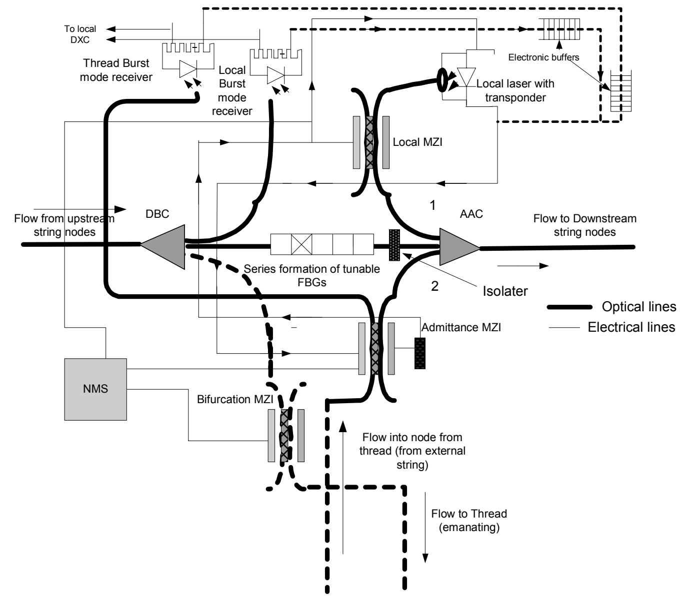

Fig. 2 Conceptual layout of node that supports light-frames

The node is essentially a 7 port device. The light-frames architecture assumes that the local transponders (for add and drop) are part of the nodal apparatus. The node in this case is assumed to have a flow from left to right and this optical signal flow through the node goes through the 2 couplers (dark triangles). The first coupler is called drop and bifurcate coupler (DBC) while the second coupler is called add and admittance coupler (AAC). The two couplers are separated by an ON/OFF switch whose implementation could be done through a fiber Bragg gratting (FBG) and isolator. Typical switching times can be millisecond or more. Signal flow through the node, happens when the switch is ON and in this case the optical signal is dropped (at the DBC) and it also continues (to the next node downstream of the incumbent node). If the switch is OFF then the optical signal is purely dropped and the second coupler (AAC) can add its own signal (LF) hence allowing spatial wavelength reuse. The two couplers are 1 x 3 and 3 x 1 in splitting ratio configurations. The DBC (1 x 3) has 1 input line and 3 output lines. The input line is connected to a node that is upstream (left) of this node on the same string. The three output lines of the DBC coupler are: a local drop output which is connected to a local burst mode receiver that either collects or discards packets by making a MAC address match of the detected LFs. The second port is a pass-through port, used to connect to the FBG which in turn is connected to the AAC. The last output of the DBC is the bifurcation output which essentially is connected to a bifurcation MZI (Mach Zehneder Interferometer) that leads to the emanation of a thread from this node to some other node in some other string. The AAC is a 3 x 1 coupler, whose primary function is to add traffic into the string. The added traffic may be from a local transponder (input port 1), or from a node in some other string via a thread and feeds into the node (through admittance MZI) – input port 2. Finally the third input port of the AAC is used for pass-through purpose of the string and is connected from the FBG. The AAC output port is connected to the line that feeds downstream nodes in the string. There are three kinds of MZI that complete the sub-system apparatus of the light-frame node and are called Local MZI (LMZI), Admittance MZI (AMZI) and Bifurcation MZI (BMZI). The individual functions of these shall be discussed in the next section on working of the node, but collectively the MZIs' regulate traffic (light-frames) into the node and serve for collision avoidance as well as for establishing/tearing down threads in the network.

IV. Functioning of the Node

Having described the subsystems, we shall now see how these sub-systems interact to support the LF framework. Let us assume that the node we are examining is an intermediate node (not convener, not end node) of a string say S and that one thread emanates from this node, to a node S_1 in some other string while one thread arrives into this node from a node S_2 part of another a string. There are five LF related functions that the node can carry out and these are: (a) local LF add, (b) local LF drop, (c) LF pass-through , (d) admit LF from thread (incoming), (e) bifurcate LF to thread (outgoing).

Before we define these functions let us now define the physical meaning of a light-frame: A light-frame is an optical version of an Ethernet frame that has essentially two information keys encoded in it: (1) a MAC layer destination and source address (2) clock recovery-carrier information useful for a burst mode receiver (through opto-electronic means as in [16-17]).

Function of local LF add: Assume the node has a light-frame (LF) at a local interface which it desires to send into the string. The node momentarily stores the frame (in electronic format) at an output queue in the local transponder for a period of time that corresponds to the listening of the underlying channel (string). By listening to upstream nodes in the string it confirms that the string is not occupied. The local transponder then sends the packet as an optical LF into the string (network). Just before sending the packet into the string, the node also 'closes' its admittance MZI and we term the position of the closed AMZI as a *restricted* position. This ensures that any LFs arriving from the thread do not collide with the local transmission but are instead sent by the AMZI to the thread burst mode receiver, which detects the LFs and sends the packets to the transponder for retransmission.

Collision from transmission of a node that is upstream of the incumbent node and local transmission: Assume that, while the incumbent node is transmitting a light-frame, some other node that is upstream (in the string) as compared to the incumbent sends a LF. These two light-frames would collide resulting in both being lost. The LF that is being sent by the incumbent is also copied and electronically stored in a buffer at the local transponder until it is successfully sent (may involve multiple tries). However the LF from the upstream node that collides with the LF from the incumbent also needs to be retransmitted due to collision. The LF framework does not have a mechanism for upstream nodes to be informed of lost frames, and it is assumed that once a frame is injected into the network it will reach the destination successfully, possibly in an all optical way or possibly through multiple OEO hops. In fact the LF framework is designed such that any collision that occurs is the responsibility of the node where the collision happens, and the node has to take corrective action. The LF from the upstream node (that collided) goes through the DBC before colliding with the local LF at the AAC (point of collision). The collision hence occurs at the AAC. Despite this collision happening, the DBC is still able to drop an uncorrupted copy of the entire LF (from upstream node) on account of the drop and continue characteristic of the coupler. The other copy of the LF that collides with the locally sent LF does not hinder the dropping process at the DBC, as backward reflection of light due to collision is prevented by an isolator (shown in Fig. 2) which absorbs the reverse optical power.

Local LF drop (string): The nodal functionality for local drop is quite simple as compared to that for local add. A LF is completely *dropped* only at the end node of a string, otherwise a LF is dropped and it still continues (to the next downstream node) in a string (except at convener and end nodes). A LF that enters a node is dropped by the DBC. The local burst mode receiver detects and reframes the optical LF into electronic layer 2 frame. The local receiver decides whether to keep the frame or discard it depending on the address match as well as depending whether the frame needs to be retransmitted (due to collision) in which case the local burst mode receiver sends the frame to the transponder queue for retransmission into the network. The key technology for implementation is a burst mode receiver shown in [9].

LF Pass through function: The pass through function of the LF node can be summarized as, when the node behaves as a pure intermediate node of a string and does not take part in either of admittance or bifurcation features nor adds its own LFs.

Admittance from thread function: consider a case when a LF from a thread arrives at a node that admits this LF into its local string. Let us observe the signal flow for this case. The LF enters the node through the *Admittance MZI* (AMZI). The AMZI has two states of existence – the *open state* and the *restricted state*. In the open state the admittance MZI acts as a 1x2 splitter and splits the incoming optical power (of the LF) into two copies and sends one copy to the thread burst mode receiver, while the other copy is sent to the AAC and hence into the string. In the restricted state the AMZI couples the entire incoming LF signal power to the thread burst mode receiver. A LF that enters the node through the AMZI finds the MZI either in open state or in restricted state depending on the local transmitting state of the node. If the AMZI is in open state then that means the node is not feeding any local LFs into the string. In such a case, the incoming LF from the thread is admitted into the string, while the thread burst mode receiver stores a copy of the incoming LF (from the thread) momentarily for reasons of conflict management explained subsequently. The issue of conflict between the LF incoming from the thread and a local LF (Fig. 4) does not arise when the admittance MZI is in open state because whenever the local node wants to send in its LF at the AAC, it switches the AMZI into restricting state forcing the incoming LF to be routed to the thread burst mode receiver and hence collision does not happen. Of course the node will wait till all the traffic from thread is clear by 'listening' to the thread burst mode receiver. This means the incoming LF has no conflict issues with local transmission in open state but conflict can still happen between the LF that is coming from the thread through the AMZI and the LF coming from the node upstream of the incumbent in the same string. This conflict cannot be avoided but damage can be prevented as follows: In the open state of AMZI a copy of the LF from the thread is locally stored at the thread burst mode receiver, similarly a copy of the LF from the node upstream of this incumbent, is stored through the DBC at the local burst mode receiver. Now a logic circuit running at the NMS realizes that a collision occurred. The logic circuit then sends the buffered LFs (in electronic format) from the both the thread as well as from the upstream node at the output queue (shown in Fig.2) of the local transponder for transmission. The two LFs are then queued and sent into the string when the channel is free.

Bifurcate to thread: The last function that characterizes a LF node is that of thread bifurcation. Here one of the outputs of the DBC feeds to the thread through the bifurcation MZI or BMZI. If a thread exists from the node to a node in some other string then the BMZI is "open" else it is closed- blocks the signal. In the open condition, the BMZI feeds the LF to the emanating thread.

V. LIGHT-FRAME NODE AS A 7-PORT DEVICE AND CONFLICT RESOLUTION

The nodal architecture that supports LF can be viewed as a 7 port device, at the optical layer, on a per wavelength basis. Shown in Fig. 3 is a conceptual schematic of the 7 port device.

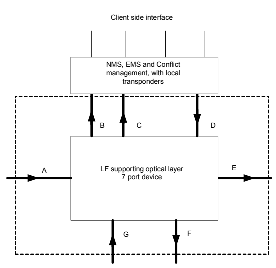

Fig. 3. The 7 port conceptual realization

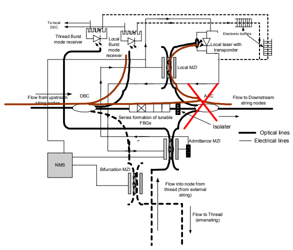

Fig 4 a

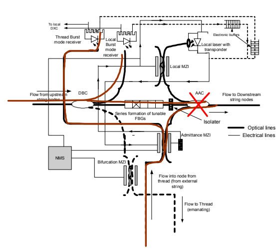

Fig. 4.b

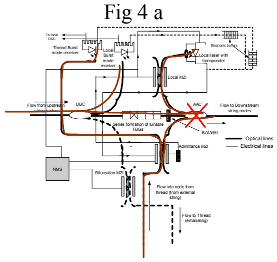

Fig.4.c

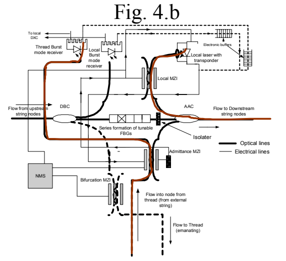

Fig. 4.d

Fig. 4: LF Conflict cases

Shown are 7 ports A through G, of a LF supporting node at the optical layer, also shown is the network management system and element management system blocks (NMS, EMS), the electronic control yields in client interfaces. Ports B, C, D are local ports indicating local add or drop, while ports A and E are input and outputs to the string. Ports G and F are thread input and thread output ports. In principle, we can scale the node to have multiple threads emanating or submerging, as well as have multiple strings supported by the same node.

In Table 1. we have shown the conflict management and remedial action (shown in Fig. 4) that is taken by a LF node. In the table we see that there are 4 cases when conflicts can happen between LFs in a node. The first case is when a LF arrives from an upstream node and the incumbent simultaneously sends its own LF. In this case, both LFs are recovered. The one from upstream node is recovered by the local burst mode receiver, while the incumbent LF is stored in the transponder itself until it is successfully sent. In the second case, a LF arrives from a thread and collides with a LF from an upstream node. Here the LF from the upstream node is stored in the thread burst mode receiver buffer, while the one from the thread goes into the local burst mode receiver. In the third case, when a LF comes from a thread and another LF is injected locally, with the injection procedure ensuring that the AMZI is in the restricted state; hence collision is not allowed to happen at all. The LF from the thread is routed to the thread burst mode receiver and hence collision is completely avoided. In the final case, 3 LFs collide – one from an upstream node, one from the thread, and one that is locally injected. Here the AMZI is in the restricted state; hence, the LF from the thread is routed to the thread burst mode receiver. The LF from the upstream node is routed to the local burst mode receiver.

We show in the table how the conflicting LFs are recovered and the state of the corresponding MZI that are responsible for this recovery. Note that when a LF is recovered, it is queued up and retransmitted into the string. This process also allows LFs to be detected and re-directed as their MAC addresses are matched in the opto-electronic recovery process. LFs, since they are broadcast in the string (due to the drop and continue architecture), may often end up floating in the network, unnecessarily past the destination and on paths that do not reach the destination. A node's NMS can analyze if the LF is just a floating frame in the network – meaning that another copy of the LF has already reached its destination. If that is the case, then the NMS removes this redundant LF from the network during the OEO regeneration process. However, to do so, the NMS must be aware of the virtual topology that exists in the network. We briefly study the method of building the topology in the next section.

TABLE I. SHOWS THE CONFLICT MANAGEMENT SCHEME: NOTE THAT D_Q NOTES THE OUTPUT BUFFER AT THE LOCAL TRANSPONDER.

LF at port A	LF at port F	LF at port D	LF at A (recov-ered)	LF at G (recov-ered)	LF at D (recov-ered)	AMZI	BMZI	LMZI
X	X		B	C		O	O	O

X		X	B		D_Q	R	O	O
	X	X		C	SENT	R	O	O
X	X	X	B	C	D_Q	R	O	O
X=LF at port, B,C-ports in Fig. 3, R=Restricted, O=Open								

Shown in Fig. 4 are the 4 possible conflict cases in a LF node. We can identify these cases in clockwise rotation: case in Fig. 4.a, signifies when a local LF collides with a LF from some upstream node, the Fig. 4.b shows the collision of a LF from a thread and a LF from some upstream node. The case in Fig. 4.d shows the event when a LF arrives from a thread and a local LF is being transmitted (note here that there is no collision happening). Finally the case in Fig. 4.c shows a collision when all 3 inputs have LFs i.e. a local LF, a LF from an upstream node and a LF from a thread. The remedial actions are shown in the figure (4) as well as table.

Virtual topology dissemination- string and thread set up and tear down procedure: For effective LF communication, the two critical network related tasks that need to be carried out are setting up of threads as well as strings subject to source destination demands in the network. The setting up of strings and threads is the solution to the virtual topology design that we will now discuss.

The principle idea behind formation of strings and threads is to form an optimized virtual topology – a matrix of interconnected strings and threads such that each node is able to reach every other node in the network through the interconnection matrix. We first build the virtual topology of strings and threads and then create these static strings and threads by the following procedure: For any given node, we have a logical connection to every other node in the network through multiple or single strings and threads. To avoid cycles we assume fixed wavelength converters (opto-electronic transponders) at *certain* threads as we traverse from one string to another (not compulsory). The issue of assigning fixed converters is a separate problem and not discussed here. These OEO devices rejuvenate signal strength by 3R regeneration. The graph can then be viewed as multiple logical trees rooted at every node such that each tree reaches every other node in the network. Our interest is to reduce the number of strings and threads that make up *all* these trees. We introduce the term – residual capacity of a string for a given node (not necessarily in the string). This term indicates the set of nodes in a string that the node can reach by using interconnected threads and strings (of course in the most optimized way). Thus residual pairs for a given logical tree are the set of source-destination pairs that are reachable in the interconnection graph.

For formulating an LP for the virtual topology, we identify S and T as two one dimensional vectors for all the possible strings and threads in the network. The objective function then is to minimize the number of strings and threads we choose from S and T subject to given constraints.

From this we derive the optimization equation:

$$\min\left[\sum_{i=1}^{S} X_i a_i + \sum_{i=1}^{T} Y_i b_i\right] \qquad (1)$$

Subject to the constraints:

$$R\left[\sum_{i=1}^{S} X_i a_i + \sum_{i=1}^{T} Y_i b_i\right] = [Tr], \qquad (2)$$

where $a_i = 1,$ if string X_i is selected X_i is the cost of choosing string i. $a_i = 0,$ otherwise.

$b_i = 1,$ if string Y_i is selected Y_i is the cost of choosing thread i. $b_i = 0,$ otherwise

Note: $R[A + B]$ indicates the total residual pairs in the set of 'A' strings and 'B' threads.

Optimization Results: The numerical evaluation of the framework for supporting light-frame communication is shown in Table II. We calculate and verify through simulation the performance of the network for multiple loads. The network is a 14 node random graph. Though the optimization problem results in the virtual topology design, we however want to see the direct benefit of virtual topology – the network utilization advantage. We show that in column 4 against load, and also show the similar observation made through pure simulation model (where the topology is not optimized). The results show significant benefit due to optimization. The other important parameter we see is the percentage of redundant traffic that continues to exist in the network due to the drop and continue architecture. We see that due to the redundant traffic the network becomes saturated with light-frames at around 80 % of the total load (busy time) – shown in column 1 and 2.

VI. Procedure to Set Up Strings and Threads and Evaluation

From the virtual topology design problem, we can set up strings and threads in the network, based on the following procedure: initially assume that the network has no strings or threads, and bi-directional sporadic communication is permissible within the data plane itself. A prospective convener node sends a frame to each prospective member of a string, by spoofing the respective MAC addresses. Each member then independently acknowledges acceptance by the same 'listening' procedure explained in the functioning section IV (analogous to CSMA/CA). Acceptance of all nodes to a convener's request then yields in a string where by the convener and end nodes block the optical path between them (by changing the FBG/optical switch states). Subsequently, threads are formed by a node sending a control frame (with the thread end node's destination address) and initiating a request for a thread. Upon acceptance by the end node of the thread, the thread is formed. Note here that the string and thread inputs at a DBC are inter-convertible through minor AMZI adjustments.

TABLE II. NETWORK PERFORMANCE AND EVALUATION

Load in the network (14 nodes)	%of redundant frames (non-optimal topology	%of redundant frames (Optimal topology)	Network Utilization (Simulation)	Network Utilization (Optimization)
0.05	3	1.1	0.09	0.056
0.1	3.8	2.4	0.18	0.112
0.15	6	3.8	0.25	0.168
0.2	7	5.3	0.33	0.224
0.25	9	7.1	0.38	0.28
0.3	10.4	7.9	0.45	0.336
0.35	12	8.9	0.51	0.392
0.4	13.2	9.4	0.58	0.448
0.45	14.1	10.2	0.66	0.504
0.5	14.8	10.8	0.72	0.56
0.55	15.6	11.5	0.78	0.616
0.6	17.1	12.2	0.84	0.672
0.65	19	13.6	0.92	0.728
0.7	21.2	14.2	0.95	0.784
0.75	23.2	14.9	0.98	0.84
0.8	25.1	15.7	NA	0.896
0.85	27.8	16.3	NA	0.952
0.9	29.9	17.1	NA	1.008
0.95	31.2	17.5	NA	1.064
1	33.5	17.9	NA	1.12

In course of communication, LFs are multi-hopped to their destination whenever collision occurs or if there is an intermediate OEO device in between. It is thus important to find the behavior of the LF framework in terms of delays incurred while transporting LFs from source to destination. In Fig. 5 is shown the maximum delay for LF's delivery in a 14 node network. The simulation model, constructs an optimal topology (based on the LP solution) and we model LF arrival as Poisson. LFs are queued up and retransmitted when a collision occurs. We observe in the figure that as the route has to encounter more number of strings to the destination increase, the end to end delay increases almost exponentially. Likewise, as the load increases, there is more of redundant traffic and hence the end to end delay increases. In the network, extreme delay is seen at loads in excess of 0.8 signifying high queueing or establishment of dedicated paths as opposed to the shared path system of LFs.

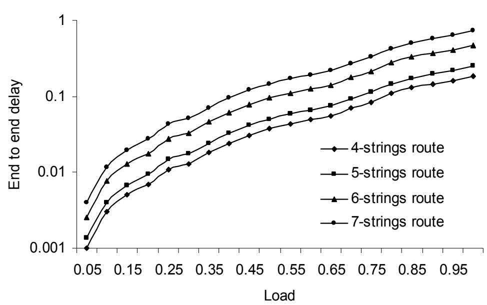

Fig. 5 Evaluation of LF framework

VII. SUMMARY

In this paper we have proposed a pragmatic framework for plausible implementation of optical frame transport. The framework based on light-frames utilizes two contemporarily known schemes of drop and continue and admittance and bifurcation. This architecture enhances these schemes in a way that leads to optical packet (frame) transport without the need for high speed switching or optical address processing the two contemporary problems faced in the implementation of optical packet transport. Despite the seemingly high degree of redundancy in frame transport the architecture performs well. This redundancy is still tolerable as the framework performs exceedingly well at load and medium loads. For heavily loaded links, the framework is backward compatible to bursts or for lightpaths that can be more gradually (statically) set up. We verify the framework through a tractable model as well as through simulation.

The viability of this scheme can be seen through the technologies used. All the component and sub-system level technologies are mature and shown in [15]. MZI switching speeds are assumed to be proportional to one LF transmission speed (listening time) and hence is in several tens of nanosecond. These can be deployed using acousto-optic MZI over Lithium Niobate Substrate as shown in [14-15]. Burst mode receivers can typically detect clocks in several nanoseconds (9 ns typical as in [14]). This indicates all sub-systems to be built from mature technology and hence demonstrate the conceptual viability of the concept of Light-frames.

REFERENCE

[1] A. Gumaste and I. Chlamtac, 'Light-trails: A Solution to IP centric communication at the Optical Domain,' Proc of IEEE High Performance Switching and Routing HPSR 2003, Torino Italy

[2] A. Gumaste and I. Chlamtac, "Mesh Implementations of Light-trails: A Solution to IP Centric Communication in the Optical Domain," 12th IEEE Proc of Intl Conf on Comp Commun. Networks ICCCN 2003 Dallas TX

[3] I. Chlamtac and A. Gumaste, "Light-trails A Platform for QoS at the Optical layer," Invited Talk IEEE 2nd QoSIP Workshop Milano Italy 2003

[4] A. Gumaste and I. Chlamtac, "GMPLS Adaptations to Providing End to End QoS: The Light-trails Approach," NFOEC 2003

[5] A. Gumaste and I. Chlamtac, " Optimizing Light-trail Assignment for IP Centric Communication," 13th IEEE LANMAN, San Francisco, April 2004

[6] L. Tamil, T. McDermott, F. Masetti, G. A. Castañon, A. Ge, L. Tancevski, "Optical IP Routers: Design and Performance Issues under Self-Similar Traffic", *Journal of High Speed Networks*, Vol. 8, pp. 59-67, 1999.

[7] I. Chlamtac et. Al, 'CORD: Contention Resolution using Delay Lines' *IEEE Journal of Select Areas in Commun* June 1996.

[8] Y. Ota, *et al*., "High-Speed, Burst-Mode, Packet-Capable Optical Receiver and Instantaneous Clock recovery for Optical bus Operation", *Journal of Lightwave Technology*, Vol.12, No.2, pp.

[9] Hernandez et al "First Field Trial of Optical Label-Based Switching and Packet Drop on a 477 km NTON/Sprint Link," OFC 2003 Anahiem CA

[10] L. Rau and D. Blumenthal, 'Optical packet communication using All Optical Label Swapping' Proc of OFC 2002 Anaheim CA.

[11] C. Qiao et al, 'Labelled Optical Burst Switching LOBS: IP over WDM Networks,' IEEE Communication Magazine. Sept 2001

[12] F. Calleghatti et al,' Packet Optical Networks for High-Speed TCP-IP Backbones. Solutions for Optical Transparent Networking. An Optical Transparent Packet Network. IP over OTP," IEEE Commun Mag March 1999

[13] I. Chlamtac, A. Farago et al, 'Lightpath (WDM) Routing for Large Networks' IEEE Trans. on Commun June 1996 pp 909

[14] A. Gumaste and T. Antony, "DWDM Network Designs and Engineering Solutions" Pearson Education Publisher Dec 2002

[15] D. Dutta et al, " WDM Technologies: Active Optical Components," Academic Press Publishers 2002