LiTPiC – Light-trails and Photonic Integrated Circuits: Issues of Network Design and Performance

Ashwin Gumaste and Nasir Ghanı

1 Dept. of Computer Science and Engineering, Indian Institute of Technology, Bombay, India
2 Dept. of Electrical and Computer Engineering, University of New Mexico at Albuquerque, USA
Email: ashwing@ieee.org, and nghanı@ece.unm.edu

Abstract: A recent advance in high-speed networks is the concept of Photonic Integrated Circuits (PICs) on an Indium Phosphide substrate, allowing miniaturization and integration of multiple OE and EO modules in a chip. PICs have the potential to do away with the opto-electronic bandwidth mismatch as well as result in severe cost reduction, hence questioning the need for all-optical networking. In this paper we analyze the impact of this digital optical networking concept by combining PIC technology with an all-optical solution — light-trails, resulting in a new solution called LiTPiC (Light-Trail Photonic Integrated Circuit). LiTPiCs offer the best of both digital and all-optical worlds. By enabling selective (and on-demand) regeneration of the signal, we are able to enhance the reach of all-optical light-trails. In addition, we are able to partition light-trail buses into multiple geographically disjoint sub-buses called PIC-Trails, thus for the first time introducing wavelength reuse within a light-trail. The LiTPiC concept gives PICs a new direction — that of being a technology enabler for next generation services using ROADM (Reconfigurable Optical Add-Drop Multiplexer) architecture and a complementary solution to the light-trail technology.

I. INTRODUCTION

Lightpath communication has dominated much of optical networking in the past decade or so. A lightpath is a wavelength circuit that connects a source-node to a destination node facilitating full wavelength connectivity. Lightpath flow suits legacy (voice-oriented circuit) networks where connections occupy full wavelength granularity. However, emerging applications such as triple play, video-on-demand, IPTV, pseudo wires have requirements that are dynamic as well as flexible. Dynamic requirements of such services imply the ability to provision services in real-time — on-the-fly, while flexibility requires emerging services to facilitate sub-wavelength granularity.

Optical networking today is based on two schools of thought — the concept of opto-electronic (OEO) networks (also called digital optical networks) and the concept of all-optical networks (AONs). Digital optical networks involve OEO conversion and regeneration of data at every node in the network which is perceived to be expensive/bit-rate-protocol dependent. AONs fundamentally aim to keep the data in the optical domain making effective use of the bandwidth provided by the optical fiber. OEO solutions based on electrical grooming and switching like SONET/SDH and RPR (Resilient Packet Rings – IEEE 802.17) have been considered but are found to be expensive, bit-rate/protocol dependent and rigid (in granularity). Hence pure OEO based solutions fall short of being able to meet emerging demands of next generation services.

The OEO and AON approaches are sufficiently diverse in their implementations as well as governing principles. While the AON concept facilitates a node to keep the data in the optical domain enabling protocol/bit-rate independent networking, the OEO concept requires processing of the signal (at every node) and enables intelligence (of add-drop etc) often at the packet level granularity. The in-line intelligence added at nodes by the introduction of electronic (packet-switch – such as STS level cross-connect) fabrics, increases the cost of the node as well as creates dependencies for protocols and bit-rates. The advantage of cost in AON implementation is offset by performance (inability to provide sub-wavelength grooming, on-demand dropping). Hence for the growth and sustenance of future services on optical networks it is desired to have a networking mechanism that is low in cost as well as provides for efficiency in terms of sub-wavelength provisioning and grooming.

In this paper we propose a network solution that uses all-optical technology for reducing in-line switching and processing as well as uses OEO technology to increase the reach of communication and improve utilization. The all-optical technology is based on the light-trails concept [2, 6], while the OEO technology is based on recently proposed Photonic Integrated Circuits (PICs) [1]. The resulting networking solution is called LiTPiC – integration of light-trails (LiT) and Photonic Integrated Circuits (PIC). The resulting LiTPiC concept leads to long-reach networking with sub-wavelength granularity, programmable, self-defined, virtual topology growth and an efficient network.

The paper is organized as follows: Section II briefly discusses light-trail and PIC technology, Section III introduces the LiTPiC concept describing node architecture. Section IV discusses LiTPiC properties and network design. Section V showcases a protocol for communication as well as growth of LiTPiC topology. Simulation study for ring networks is shown in Section VI, while Section VII concludes this paper.

II. ENABLING TECHNOLOGIES: LIGHT-TRAILS AND PHOTONIC INTEGRATED CIRCUITS

II. A Brief overview of Light-trails: Light-trail proposed in [2, 6] is an alternate solution to provide optical grooming, dynamic provisioning, sub-wavelength granularity and optical multicasting. A light-trail is defined as a generalized lightpath such that multiple nodes can take part in communication along a path [6]. A light-trail is analogous to a multi-point-to-multi-
point (unidirectional) wavelength bus. Arbitration in the wavelength bus is done through an out-of-band (OOB) control channel. The segmentation of data plane and control-plane makes the light-trail system unique. The data plane is characterized by node architecture as shown in Fig. 1 and proposed in [2, 6] to facilitate bus operation. At every node on a per-wavelength basis, a light-trail undergoes two characteristics – drop-and-continue and passive-add. The former allows incoming optical signal from the light-trail to be dropped locally while still enabling continuation of the signal (to downstream nodes); while the latter enables local addition of optical signal into the light-trail whenever the bus is idle. To perform these two functions, the node uses optical couplers – passive elements that combine or split signal. An ON/OFF switch is used for creation and deletion of static light-trails and is inserted between the two couplers (see Fig. 1). An arc of nodes forms a light-trail when a wavelength is regulated between the two extreme nodes of the arc. The two extreme nodes are called the convener and end node. Setting up and tearing down of the light-trail involves configuration of the optical switch – typically blocking the wavelength between the extreme nodes and allowing bus operation at the intermediate nodes.

When nodes send data over a light-trail they do so by forming a connection. A connection is a flow of data over the light-trail and is setup/torn down without any optical switching (using bus features of drop and continue/passive add) and provisioned through the OOB control channel.

The control channel is out-of-band, defined by the ITU as the Optical Supervisory Channel (OSC) and is responsible for arbitration in the bus as well as the setting up/tearing down of the bus itself. A light-trail network enables efficient spatial (amongst multiple nodes) wavelength sharing and optical grooming of data (no in-line optical switching) as well as enables dynamic connection provisioning (that involves no optical switching).

The light-trail architecture has three associated innovative features: (1) it uses low-cost 1x2 and 2x1 passive couplers for supporting bus operation (2) it differentiates the data plane with the control plane (similar to OBS) and (3) since multiple nodes time-share the light-trail, the transponders that send data onto the light-trail are engineered for providing electronic queuing and burst-mode operation [2]. We implemented such a transponder and called it the burstponder [2]. The burstponder has an electronic queue associated with it that stores data when other nodes use the light-trail for transmission. The burstponder transmits its data when it receives a connection grant via the control channel. Transmission (and reception) of data from (and to) the burstponder is done by burst-mode transmitters and receivers. These components are commonly used in PON deployments [5]. Despite providing the features of optical multicasting, dynamic provisioning and sub-wavelength granularity, the node architecture of Fig. 1 has an inherent drawback – passive couplers have a significant optical-power loss associated with them. A second and more decisive drawback is the absence of wavelength reuse within a light-trail (property of a bus). This means that two light-trails can share the same wavelength only if they are graphically disjoint, but it is not possible for two connections (within the same light-trail) to be concurrently present, even if the connections are graphically disjoint. The unconstrained optical multicasting property that prevents wavelength reuse causes poor utilization of the physical layer.

II. B. Brief overview of Photonic Integrated Circuits (PICs):
Photonic-Integrated-Circuits (PICs) is a technology that has recently gathered significant attention [1] as an alternative to all-optical networks. PICs based on an Indium Phosphide substrate bring for the first time, the concept of system-on-a-chip to optical networking. Recent advances in PIC technology enable incorporation of a number of high-speed lasers and detectors, monolithically integrated on the same substrate (chip). This advance in engineering aspects of integrating a series of DFB (Distributed Feedback) lasers and PIN/APD detectors on the chip enable low-cost OEO processing on a signal. The realization of PICs enables full 3R (retiming, reshaping and re-amplifying) of a signal. It is claimed in [1] that apart from OEO functionality PICs can support other networking features like add-drop etc. While this may be possible in the future, the advantage gained by low-cost OEO technology would be offset by the nuances of processing high-speed electrical signals especially for dropping sub-wavelength flows from a high-speed (wavelength) bit-stream. It has been shown with RPR and SONET grooming technologies that the digital switch fabric responsible for extracting a sub-wavelength flow from a high-speed electronic signal is both complex and expensive [5]. The complexities and price have a further impact while incorporating add/drop of sub-wavelength electronic flows (to maintain high bandwidth utilization at a node. The advantage seen in PICs due to monolithic integration and reduced footprint would be negated if we were to assume a digital fabric consisting of pass-through, drop and add electronic queues each involving network processors and each operating at the full wavelength speed.

However, light-trails and PICs are complementary technologies, which when grouped together provide the necessary functionality as well as reduces costs. The functionality includes ability to groom sub-wavelength flows, dynamic provisioning, optical multicasting (all due to light-trail), long-reach, good per-span utilization (all due to PIC) as well as a low-cost system. The integration of light-trails and PICs results in a new networking hierarchy that we call as
LiTPiC or Light-Trail Photonic Integrated Circuits and is the subject of this paper.

III. LiTPiC: LIGHT-TRAIL PHOTONIC INTEGRATED CIRCUITS

We will first define the abstract meaning of LiTPiC from a graphical and conceptual perspective, and then show the architectural integration of light-trails and PIC yielding LiTPiC. A LiTPiC is an arc of nodes in a network that satisfy the following properties. (1) A LiTPiC is analogous to a unidirectional optical bus with programmable wavelength reuse within the bus. (2) A LiTPiC can be divided into multiple sub-arcs each co-existing and exclusively satisfying bus-properties. (3) Each sub-arc within a LiTPiC is called a PIC-Trail. (4) The primary LiTPiC arc and its secondary sub-arcs (PIC-trails), all individually conform to light-trail properties of optical multicasting, dynamic provisioning and ability to groom sub-wavelength flows. While traditional bus architectures support only one source-destination pair at any given time, a LiTPiC can be partitioned into multiple sub-arcs (or sub-buses – PIC- Trails) allowing the existence of multiple concurrent connections, with the constraint that each concurrent connection is assigned to a separate PIC-Trail. Within a d-node PIC-Trail a maximum of $dC_2$ source-destination pairs are available for formation of statistically shared connections. A connection is formed when a source node sends data to a destination node through a PIC-Trail.

III. A. LiTPiC Setup:

A LiTPiC is setup as follows: an arc of nodes is transformed into a LiTPiC upon availability of a common data-wavelength (in the arc). Availability is checked by an initiating node through the control channel. To form the LiTPiC, the two extreme nodes in the arc, block the available wavelength while the remaining (intermediate) nodes enable access to the wavelength i.e. passive dropping (with continuation) and local addition, thus forming a bus.

III. B. LiTPiC Node Architecture:

In this subsection we focus on the LiTPiC node architecture. To facilitate PIC technology in a light-trail node, there are two modifications: (1) at the sub-system (architecture) level and (2) at the control channel/protocol level. Shown in Fig. 1 is the architecture of a light-trail node (for ring networks) and shown in Fig. 2 is the corresponding node architecture that supports LiTPiCs. Note that the architecture fully supports ROADM (Reconfigurable Optical Add-Drop Multiplexer) functionality. As mentioned before, every light-trail (on a pre-assigned wavelength) while passing through a node, has two signal-flow properties, that of drop-and-continue and passive-addition. These properties are due to two passive couplers (in 1x2 and 2x1 configuration). The two couplers are separated by an ON/OFF switch that enables the creation of the light-trail (regulating flow to a wavelength arc bounded by two extreme nodes, such that the switches at the extreme nodes are in the OFF position, while the switches at all the other intermediate nodes are in the ON position).

To incorporate PIC technology, the optical ON/OFF switch in Fig. 1 is replaced with a PIC transceiver switch as shown in Fig. 2. The PIC transceiver switch consists of a PIC detector circuit, PIC laser circuit and an external modulator, all housed in the same chip. The PIC transceiver switch first shown in [1] has two states of existence – ON and OFF. In the ON state, optical signal entering the switch is detected by the receiver. The data (in electronic domain) detected from the receiver, is coupled with the light emitted by a DFB laser (in the same chip) through an external modulator cavity. The DFB laser is fabricated (on the same InP substrate) just after the receiver. The laser emits an ITU grid wavelength and this light is coupled into an external modulator (within the same InP substrate) (refer Fig. 2–3). The data is hence regenerated and transmitted in optical format. In the OFF-state, data is detected and recovered by the receiver and available for local use.

From the perspective of LiTPiC, in the OFF-state, data in optical format enters the node and composite WDM signal is de-multiplexed into individual wavelengths. Each wavelength goes through the drop-coupler, which drops a copy of the signal (by optical power-splitting) and sends another copy to the PIC transceiver switch. The PIC transceiver switch being in the OFF-state now implies that this copy of the incoming optical signal is simply discarded. The OFF-state of the switch signifies that the node does not perform drop-and-continue operation but only performs drop operation – not allowing optical multicasting to nodes further downstream of itself (if any).

When the PIC transceiver switch is in the ON state, the bus function of a light-trail continues albeit there is OEO regeneration at the node. While, when the PIC transceiver switch is in the OFF-state, the light-trail is necessarily broken down into separate sub-arcs, each of which would continue to support bus-function.

Support for high-speed communication (10 Gb/s and beyond): High bit-rate signals, beyond 10 Gb/s require complex retiming and carrier extraction circuitry. Typically data recovery of such signals is complicated, and a dedicated clock-and-data recovery (CDR) circuit is required for recovery and regeneration of the signal to permit the PIC transceiver switch to function in the ON state. Current PIC technology does not permit the implementation of CDRs within the InP substrate, which, is presently done using ASICs [5]. Hence, PICs alone cannot be used as regenerators of data (OEO function) at high bit-rates. To enable high-speed data recovery and regeneration we make good use of existing CDR ASICs in conjunction with burst-mode receivers used for implementing the drop-and-continue function (as shown in Fig. 3). One copy of the signal is dropped at the burst-mode receiver (local drop) and another copy goes through the PIC. Using the CDR ASIC (which is a standard feature of the optical receiver), we are able to extract clock information from the incoming signal (follow the red-line in Fig. 3). This information is used to retime pulses (that were detected by the PIC receiver), by feeding the recovered carrier to the external modulator (see magenta-line in Fig. 3). In this fashion we are able to perform
OEO function for high bit-rate signals using PIC transceiver switch in LiTPiCs.

Fig. 2. LiTPiC node architecture

Fig. 3. LiTPiC node architecture with provision for high-speed pass-through

III. C. Three Levels of Hierarchy within a LiTPiC

Management and functioning of LiTPiCs can be understood through 3-levels of hierarchy (1) the LiTPiC, (2) PIC-Trails and (3) Connections.

1. LiTPiC management and functioning: A LiTPiC is an arc of nodes, each of which has the architecture as shown in Fig. 4 resulting in a super-bus structure. A LiTPiC is semi-permanent and setting up as well as tearing down a LiTPiC necessarily involves PIC transceiver switch configuration at the extreme nodes of the arc – this sort of provisioning is called hard-provisioning. Setting up a LiTPiC also involves control channel based arbitration (to find a common wavelength etc.).

2. PIC-Trail management and functioning: A PIC-Trail is a sub-arc of the LiTPiC that individually functions as a light-trail. A PIC-Trail is formed by keeping the state of the PIC transceiver switch at the extreme nodes in the sub-arc in the OFF state while keeping all the switches at the intermediate nodes in the ON state, hence realizing bus-function. A PIC-Trail is essentially a bus-within-a-larger-bus (LiTPiC). Selecting any two nodes in the LiTPiC and changing the state of the PIC transceiver switch from ON to OFF, realizes a PIC-Trail. There is no wavelength reuse within a PIC-trail. Multiple PIC-Trails that are graphically disjoint (non-overlapping with one-another) can coexist within a LiTPiC (property of concurrency).

3. Connection assignment: A connection is a flow of data from a source node to one or more destination nodes (plurality specifies optical multicasting) over a PIC-Trail/LiTPiC. Connection provisioning is dynamic and arbitration is done purely through the control channel.

III. D. Example of LiTPiC, PIC-Trail and Connection

Consider Fig. 4(a-b) that shows a LiTPiC divided into PIC-Trails and connections provisioned over each PIC-Trail. We have in Fig. 4a LiTPiC A-F where each node has the architecture shown in Fig. 3. In Fig. 4a the LiTPiC is divided into two PIC-Trails A-D and D-F such that PIC-Trail A-D provisions connection BCD (note the optical multicast from B to C and D), while PIC-Trail DEF support connection DEF (with multicast to E and F). In Fig. 4b we have the same LiTPiC (A-F) now divided into three PIC-Trails AB, BE and EF supporting connections AB, CDE and EF. Note the state of the PIC transceiver switch at each node in the LiTPiC. Also note that though the LiTPiC supports three PIC-Trails not all the PIC-Trails have connections through all their spans e.g PIC-Trail BE has connection CDE with span BC is not utilized.

IV. LiTPiC NETWORK DESIGN

Since light-trails are an efficient way to spatially (across multiple node) groom sub-wavelength flows, this property naturally extends to LiTPiCs. Hence we desire to design LiTPiC networks so as to minimize the number of LiTPiCs while incorporating every traffic request and also ensuring that the delay bound of each request is met. The problem of connection allocation to LiTPiCs is analogous to bin-packing (connections are objects to be packed in a LiTPiC bin) and is shown to be NP-complete. Hence we develop a linear program which solves for a realistic metro ring network (node count <16, 30-40 wavelengths).

For a given ring network with N nodes and W wavelengths, we define T as a traffic matrix of size I X N(N-1), where T_iw denotes the m^i connection with source node m mod N and destination node \( \lceil m/W \rceil \); where m mod N denotes the quotient when m is divided by N and
\( \lceil m/N \rceil \) denotes the remainder after the division of \( m \) by \( N \).

Our objective is to assign every connection \( m \in T \) to a LiTPiC \( k \). \( v(m) \) implies the granularity of connection \( m \). \( v(m)=1 \) implies a full wavelength is required to facilitate connection \( m \), while \( v(m)<1 \) implies sub-wavelength granularity.

Let \( PLT \) be the set of PIC-trails possible. \( PLT \) is a vector of dimension \( I \times 2N(N-1)W \).

**Processing traffic for delay tolerance:** Since connection provisioning leads to stochastic interdependence, it is imperative to take into consideration the delay profile of each connection when optimizing the set of chosen LiTPiCs.

Let \( \delta^m(k-m') \) denote the delay that connection \( m \) experiences when assigned to LiTPiC \( k \) such that \( \{k-m'\} \) denotes the set of connections also assigned to LiTPiC \( k \).

Let \( \theta_m \) denote the maximum allowable delay for the \( m \)th connection.

Let us define \( a_i \) a Boolean variable such that:

\[
 a_i = \begin{cases} 
 1, & \text{if PLT } k \text{ is chosen as a LiTPiC.} \\
 0, & \text{otherwise} 
\end{cases} 
\]

Let us define \( b_{km} \) a Boolean variable such that:

\[
 b_{km} = \begin{cases} 
 1, & \text{if PLT } k \text{ is chosen and connection } m \text{ is assigned to } k. \\
 0, & \text{otherwise} 
\end{cases} 
\]

We denote the objective function as:

\[
 \min \text{PLT}.a_i 
\]

Subject to the following constraints:

1. **Assignment constraint:**

\[
 \sum_{k \in \text{PLT}} T_{km}b_{km} = 1, \forall m 
\]

This implies that each connection \( m \) is assigned to exactly one LiTPiC \( k \). This also implies that no connection is assigned to more than one LiTPiC. Traffic between a source-destination pair that has granularity greater than a wavelength is separated into lightpaths (each of capacity \( C_i \)) and the remaining traffic is available for optimization into PIC-trails.

2. **Per-span capacity constraint:**

\[
 v(u) \leq C_j \quad \forall u : u \in \{ \text{PLT}_i \} 
\]

This constraint implies that the load assigned to every span \( u \) within the LiTPiC \( k \) is less than the capacity of the span \( C_j \). Note that while in light-trails the optimization is considered over the entire bus (i.e. the capacity constraints take the entire bus bandwidth into consideration) [7], in LiTPiCs the optimization takes a per-span value into consideration. This deviation is necessary to take into consideration the formation of PIC-Trails, as PIC-Trails can divide a LiTPiC into several buses. The maximum number of concurrent PIC-Trails formed in a LiTPiC of \( n(k) \) nodes is \( n(k)-1 \) which is the number of spans in the LiTPiC and hence for optimal choice of LiTPiCs we take into consideration the total load assigned to each span.

3. **Delay constraint:**

\[
 \delta^m(k-m') < \theta_m, \forall m 
\]

This constraint implies that the delay experienced by the \( m \)th source-destination pair when allocated to the \( k \)th LiTPiC (with \( k-m' \) other connections allocated to \( k \)) is less than the maximum allowable delay for that connection \( \theta_m \).

**V. LiTPiC CONTROL PROTOCOL**

For the purpose of management and connection provisioning every LiTPiC is allocated a controller (node) that arbitrates and manages communication within the LiTPiC. Protocols designed for communication and efficient sharing of bus networks can easily be extended to light-trails – note that the protocol for arbitration in a light-trail is housed within an out-of-band control channel unlike most existing polling based bus-protocols. The primary requirement of bus-protocols in the past has been to allocate bandwidth to a single node in the bus, so that conflict-free communication is facilitated. In LiTPiCs however, the bandwidth allocation problem is significantly different – more than one node can send data into the LiTPiC at the same time, as long as the multiple connections formed by each source node are **geographically diverse** (leading to different PIC-Trails). A protocol for bandwidth allocation (connection provisioning) within a LiTPiC, as well as for governing the growth of LiTPiCs in a network is now described.

Consider a LiTPiC \( k \), and let us assign a node \( \Omega_k \) as the controller of the LiTPiC. The controller has following properties:

1. All nodes forming new PIC-Trails, do so through \( \Omega_k \)
2. Any connection that desires to use the entire LiTPiC as a PIC-Trail has to do so through \( \Omega_k \)
3. Dimensioning and opportunistic growth functions (defined later) are carried through \( \Omega_k \)

The primary design principle followed by LiTPiC networks is to minimize the number of times a LiTPiC is formed, while facilitating all legitimate connection requests either directly or through formation of PIC-Trails. The problem of creating on-demand PIC-Trails in a LiTPiC is complex. The complexity arises from the fact that the choice of PIC-Trail affects the stochastic properties of other connections (within the LiTPiC). Typically, it is desired to form PIC-Trails such that every connection request conforms to its latency requirement and the utilization of each fiber segment in a LiTPiC is maximized.

**Network metrics:** we assume a 2-fiber N-node ring network with \( W \) wavelengths in each of the counter-propagating ringlets, with a wavelength in each ring allocated as the control-channel. The control channel is optically dropped, electronically processed and then reinserted at every node in the network enabling nodes to be synchronized through the control channel. Synchronization implies that the data channels (LiTPiCs) are also **loosely synchronized** – the control packets determine start and end times of connections,
and due to control channel synchronization we assume data channel is also loosely synchronized. Further, without loss of generality we assume that each LiTPiC is time-slotted (to make the analysis easier) and that time-slots have a value $T_s$ – between 2–5 millisecond for 1Gb/s or 10 Gb/s LiTPiC. The time required to set up a LiTPiC is 3 ms while that to set up a connection is 10 $\mu$s and to set up a PIC-Trail is 100 $\mu$s.

**Control protocol:** Nodes send their bandwidth request (in real-time) through the control channel to $\Omega$. Bandwidth requests are sent as *bids* based on their quantitative and qualitative requests [8]. Quantitative requests imply the net utility that the node would have (for forming a connection/PIC-Trail), derived from the amount of data it possesses in its buffer (as compared to its buffer size). Qualitative requests specify the urgency in forming a connection, specifying how long the node can wait before packets belonging to some service would be timed out. Qualitative requests also take into consideration the idle period – the time since the node first received packets (after formation of its last connection in the LiTPiC). For voluminous bandwidth oriented requests the bid follows a Poisson distribution (denoting the arrival process) while the qualitative aspect is shown in [4] to follow a sigmoidal like function (distribution).

The bid value each node sends to the controller is computed as follows [4, 10]:

Let us denote $\text{Buff}_i(t)$ as the value of the buffer (as shown in Fig 2) at time $t$ at node $i$ for transmission in LiTPiC $k$. Also, let $\text{Buff}M$ denote the maximum size of the buffer at node $i$ and allocated for transmission in LiTPiC $k$.

Let $s = S_1, S_2, S_3, .., S_k$ denote the set of services, and let $D = d_1, d_2, d_3, .., d_s$ denote the latency tolerance limits of the corresponding services.

Let $x^h_i$ denote the time since the first packet of the $h$th service entered the buffer at node $i$ (desiring for transmission in LiTPiC $k$).

Then, $cr_i(t) = \min_{s_j, t_h} (d_s - x^h_i)$ denotes the time criticality value – noting the time before which the node must be serviced else packets belong to some service would be timed out.

Also let $ne_i(t) = \max_{s_j, t_h} (x^h_i)$ denote the *idleness* parameter. $ne_i(t)$ denotes the time since the node has not been serviced (with a successful bid) and has been receiving data (from the client side). The bid value is computed [4, 10]:

$$Bid_i(t) = \max \left[ \frac{1}{1 + cr_i(t)/ne_i(t)} \cdot \frac{\text{Buff}_i(t)}{\text{Buff}M} \right]$$

$$= \max \left[ \frac{1}{1 + \min_{s_j, t_h} (d_s - x^h_i)/ \max_{s_j, t_h} (x^h_i)} \cdot \frac{\text{Buff}_i(t)}{\text{Buff}M} \right]$$

The bid value computed above has a maxima taken over a sigmoidal like component (first term in the bracket) and a concave component (second term). This type of bid computation has been shown to be an efficient way to allocate connections within a light-trail as well as to enable growth of the light-trail topology [4]. However, for LiTPiC this mechanism is not sufficient – the arbitration (and hence bandwidth, connection allocation) must take into account the wavelength reuse aspect. That is, a priority is to be given to those connections that can exist concurrently. Hence, the bid value that a node would send should be scaled (up) if there is possibility of concurrency of the connection with connection(s) from other node(s) within the same LiTPiC. The scaling favors the provisioning of all the connections that have a higher possibility of concurrency. To incorporate concurrency based scaling we introduce a parameter $R_{sw}^h_i$ that denotes the set of all connections that can exist concurrently when node $i$ requests for a connection $m$ in LiTPiC $k$. Note that when node $i$ requests for a connection $m$, then the destination node (or farthest destination node in case of multicast) is $m \mod i$ which implies the remainder after the division of $m$ by $i$. (we assume nodes are labeled in numbers from $1..N$). The bid value obtained is then scaled up as:

$$fbid_{ne}(t) = \left[ \frac{\sum_{m \in R_{sw}^h_i \forall i} [Bid_m(t) - sw_p(t)]}{Bid_i(t)} \right]$$

where, $sw_p(t)$ denotes a switching penalty (required to create PIC-Trails). $sw_p(t)$ is computed as:

$$sw_p(t) = \frac{C_T}{\eta} \sum_{m \in R_{sw}^h_i \forall i}$$

where $\eta$ is the efficiency of the system (assumed to be 0.75 from simulation – Section VI), $C$ is the wavelength capacity and $T_s$ is the time-slot duration. Note that $sw_p(t)$ depends on the number of switching transitions to be made – proportional to the total amount of reuse possible while incorporating connection $m$.

Upon receiving bids in a time-slot, the controller $\Omega$ determines the highest bidder(s) and allocates the LiTPiC bandwidth to the corresponding node(s). The set of successful bidders is denoted as:

$$suc_i(t) = \arg \max_{s \in \Omega} [fbid_i(t)]$$

Further, $\Omega$ computes the highest (winning) bid as:

$$mbid_i(t) = \max_{s \in \Omega} [fbid_i(t)]$$

and announces this value to the rest of the network (even to those nodes not part of $k$ through the control channel). $mbid_i(t)$ is called the trophy price of the LiTPiC and denotes the pay-off value at which any node can expect to win bandwidth in LiTPiC $k$.

**Opportunistic growth and Dimensioning of LiTPiCs:**

The protocol mentioned above accomplishes bandwidth arbitration in a LiTPiC leading to setup/tear-down of PIC-Trails and connections. However, the unanswered question that remains is, what happens when a node does not receive bandwidth (to form a connection), while its packets are either being timed-out or its buffer is full to almost its capacity? To
prevent loss of data due to buffer overflow or service time-out
we propose three possible solutions (1) opportunistic bidding,
(2) dimensioning and (3) hard-provisioning.

Opportunistic bidding: As can be seen from the LiTPiC node
architecture, a node has access to multiple LiTPiCks and can
decide to transmit data into a particular LiTPiCk by purely
adjusting its electronic grooming fabric (standard feature in
MSPPs [5]). Hence, a node i part of LiTPiC k continues to
monitor the bidding process in LiTPiC k', if k' is of interest
to node i. LiTPiC k' is of interest to node i if (a) k' contains all
destination nodes that node i is interested in transmitting data
to, and (b) the total value of traffic in k' can accommodate
additional traffic from i, if node i so chooses to become a
member of k'. The deciding point that determines when node i
should leave LiTPiC k for k' (though it still continues to be a
passive [7] member of k – without transmitting or bidding for
bandwidth in k) is:

\[ \text{Fbid}_{i,k}(t) > \text{Mbid}_{i,k}(t-1) \]  
\[ Y \geq c_{\beta}(t) + t_{\text{HP}} + t_{\text{SP}} \text{ or Buff}_{i,k}(t)/C > Y \]

where, Y is a constant of the system (set at \(1/\text{t}(k)\)), t_{\text{HP}} is the
time required to provision a LiTPiC and t_{\text{SP}} is connection
setup time. Equations (11-12) imply the condition for node i to
move from LiTPiC k to k', denoting the instance when the bid
sent by node i in k is greater than the maximum bid (trophy
price) of k' in the previous time-slot and that there is a need
for i (either due to approaching time-out or buffer overflow) to
move from k to k' (12).

Dimensioning: If a node \( \alpha \in k \) in LiTPiC k desired to transmit
data to a node \( \beta \in k \) (not part of LiTPiC k) and if k can
accommodate this additional capacity request (by node \( \alpha \)),
then k can be dimensioned (in size) to include node \( \beta \).

Dimensioning a LiTPiC means adding/deleting nodes. The
procedure to dimension a LiTPiC is shown below.

**Dimension**

If \( \alpha \beta \in k \) and \( f_{\alpha\beta} > 0 \), \( \alpha \in k \) or \( \beta \in k \) (but not both) and
\( f_{\alpha\beta} + v(k) < C \)

\( P1: \) if \( \alpha \in k \) then \( \overline{\alpha \beta} \) is incident on k, and \( \overline{\beta} \) on \( \lambda_k \) is
available then go to P3.

\( P2: \) if \( \beta \in k \) \( \overline{\alpha \beta} \) is incident on k and \( \overline{\beta} \) on \( \lambda_k \) is available
then go to P3.

\( P3: \) dimension \( k \leftarrow k U \overline{\alpha \beta} \)

The if statement implies whether segment \( \overline{\alpha \beta} \) is not
part of k (i.e. either \( \alpha \) or \( \beta \) are not part of k) and the flow due
to the traffic \( \overline{\alpha \beta} \) denoted by \( f_{\alpha\beta} \) can be accommodated in k.

P1 and P2 state that there are two ways to dimension the
LiTPiC depending on which of \( \alpha \) and \( \beta \) are elements of k. P3
dimensions the LiTPiC to include the segment \( \overline{\alpha \beta} \). In the
above pseudo code \( \lambda_k \) denotes the wavelength allocated to
LiTPiC k.

**Hard-provisioning:** if a node is on the verge of either buffer
overflow or packet-time out, then the controller of the LiTPiC
signals to the node to form a new LiTPiC (or join another
existing LiTPiC through opportunistic bidding). The condition
of the controller signaling to a node to move out of a present
LiTPiC is stated as follows.

If \( Y \geq c_{\beta}(t) + t_{\text{HP}} + t_{\text{SP}} \text{ or Buff}_{i,k}(t)/C > C(1-Y\text{Buff})M \)
then the controller \( \Omega_k \) signals to node i to form a new LiTPiC
\( k' \).

### VI. SIMULATION MODEL AND RESULTS

The constrained optimization shown in Section IV and the
heuristic proposed in Section V were numerically
evaluated for ring WDM networks. We assumed a 2-fiber ring
with 40 wavelengths in each fiber, the line-rates were assumed
to be either 1Gb/s (GigE). Node architecture was assumed to
be as shown in Fig. 3. For the heuristic case, traffic was
generated as follows: Traffic was assumed to be of four types –
iovoice, video, data and triple-play. Voice traffic was modeled
to be Poisson, data traffic to be Pareto distributed, video traffic
was assumed to be a mixture of Poisson and Pareto traffic
(with a ratio 40:60) and triple-play traffic assumed a mix of
voice, video and data arriving at the same port. Load was
computed as the total amount of bits injected into the network
in unit time divided by the maximum traffic (in bits) the
network could carry (obtained by multiplying the wavelength
capacity with number of nodes and dividing by optimal
wavelength reuse factor \( \approx 2.43 \) for rings [11]). To compare
performance of the network (for both optimization and
heuristic), we considered light-trail based networks as well as
lightpath networks (GigE) with similar topologies and bit-
rates. Shown in Table I are the numerical results of the
constrained optimization process (solved using LINDO) as
well as the heuristic algorithm presented in Section IV. We
assumed each LiTPiC to be time-slotted, with slot size of 3 ms
and time to provision \( (1) \) connection is 10 \( \mu s \), \( (2) \) PIC-Trail
100 \( \mu s \) and \( (3) \) LiTPiC/light-trail 2.5 ms. The table compares
the number of light-trails and LiTPiCs created to suffice
network traffic. As can be seen the number of LiTPiCs created
is almost half the number of light-trails required at low loads
(indicating excellent wavelength reuse within a light-trail), but
as load increases the discrepancy tends to decrease such that
the number of light-trails and LiTPiC is the same for heavily
loaded networks. Similarly, the performance heuristic
algorithm proposed in Section IV increases with load for both

<table>
<thead>
<tr>
<th>Load</th>
<th>Number of LiTPiCs from Optimization</th>
<th>Number of LiTPiCs (Fixed)</th>
<th>Number of Light-trails from Optimization</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>20</td>
<td>35</td>
<td>79</td>
</tr>
<tr>
<td>0.2</td>
<td>36</td>
<td>59</td>
<td>86</td>
</tr>
<tr>
<td>0.3</td>
<td>59</td>
<td>82</td>
<td>87</td>
</tr>
<tr>
<td>0.4</td>
<td>77</td>
<td>101</td>
<td>108</td>
</tr>
<tr>
<td>0.5</td>
<td>95</td>
<td>114</td>
<td>126</td>
</tr>
<tr>
<td>0.6</td>
<td>112</td>
<td>130</td>
<td>136</td>
</tr>
<tr>
<td>0.7</td>
<td>148</td>
<td>152</td>
<td>150</td>
</tr>
<tr>
<td>0.8</td>
<td>185</td>
<td>180</td>
<td>172</td>
</tr>
<tr>
<td>0.9</td>
<td>222</td>
<td>227</td>
<td>217</td>
</tr>
</tbody>
</table>
LiTPiC and light-trail. The explanation for betterment of performance as load increases is that, light-trails and LiTPiCs are generalizations of lightpaths and as the load increases the number of connections per light-trail/LiTPiC decreases (implying an increase in connection granularity). This means that at full load, a light-trail/LiTPiC will have just one connection provisioned for all the time (no statistical sharing), implying a light-trail/LiTPiC has provisioned a lightpath.

Fig. 5. Delay in LiTPiC and comparison with Light-trails

Shown in Fig. 5 is the delay profile for LiTPiC and that for light-trail for traffic provisioning. We observe that for low-loads the delay incurred in LiTPiC is greater than that for light-trails – the reason being the lost time in creating PIC-Trails (100 microseconds). However, as load increases LiTPiCs have a better delay profile (due to more certainty of connection provisioning because of wavelength reuse within the LiTPiC) as compared to light-trails.

Fig. 6. Per span efficiency of LiTPiC, Light-trails and Lightpaths

Shown in Fig. 6 is the efficiency of light-trails, lightpaths and LiTPiCs. Efficiency is measured as the total amount of time the channel carries useful information in unit time. We observe that due to tight grooming of connections (excellent statistical multiplexing + reuse) in a LiTPiC, the efficiency (on a per-span basis) of LiTPiC is the better than light-trails and lightpaths. Light-trails naturally are better than lightpaths [2, 6].

Shown in Fig. 7 is packet loss (ratio of packets dropped in the buffer at a burstponder to buffer size) as a function of load. Here we assume the buffer size in a burstponder is 12 MB. We observe that LiTPiC has lower packet loss (due to higher possibility of forming connections) than light-trails. Finally, shown in Fig. 8 is the efficiency of LiTPiCs with and without concurrency, where concurrency is defined as the number of PIC-Trails that can co-exist in a LiTPiC averaged over time. We observe that there is an improvement by almost a factor of 2 in efficiency in LiTPiCs at low to moderate loads. The effect of concurrency fades as load increases. Theoretically, and intuitively we can expect that at low loads, concurrency would have an advantage factor of 2.43, but this does not happen in the simulation due to penalty in connection and PIC-Trail set up time.

VII. CONCLUSION

In this paper we proposed the concept of LiTPiC – by merging light-trails and PIC technology. LiTPiC concept is showcased through novel node architecture, management as well as network design. The node architecture functions as a ROADM – the first such implementation using PIC technology. LiTPiCs are super-buses that can be further divided into sub-buses called PIC-Trails. Connections (for communication) are provisioned over PIC-Trails. A protocol that provisions connections, PIC-Trails and LiTPiC is also presented. The protocol maximizes concurrency of PIC-Trails within a LiTPiC leading to good per-span utilization. A simulation model and comparison with lightpaths as well as light-trails is presented. The LiTPiC concept leads to enhanced utilization and enables dynamism.