



ELSEVIER

Available online at www.sciencedirect.com

Optical Switching and Networking xx (xxxx) xxx–xxx

**Optical
Switching and
Networking**
www.elsevier.com/locate/osn

Light-mesh — A pragmatic optical access network architecture for IP-centric service oriented communication

Ashwin Gumaste^{a,*}, Deepak Diwakar^a, Anuj Agrawal^a, Akhil Lodha^a, Nasir Ghani^b

^a Department of CSE, IIT Bombay, India

^b Department of ECE, University of New Mexico, Albuquerque, NM, United States

Received 8 January 2008; accepted 15 January 2008

Abstract

Contemporary deployments of optical access networks are based on the principles of Passive Optical Networks (PONs). PONs deploy a star topology and dual wavelength for communication between the center and ends of a star. The star topology requires that each end-user be connected to the star splitter (usually a passive coupler). We argue in this paper that while adhering to the requirements of access networks, we are able to provide a better topological solution in terms of the cost-factor and the ability to upgrade to a greater bandwidth. This solution, called a *light-mesh*, is based on the concept of pragmatic optical packet transport or *light-frames* results in a unique node architecture, interconnection matrix, and communication protocols. We begin by investigating into the node architecture that is required for a mesh network in the access area. The proposed node architecture has unique benefits in terms of being able to support the intermittent communication in the access area — nodes are not always powered ON, despite which, it is important to maintain mesh connectivity. Hence we propose the use of largely passive components in node architecture design. Passive components in a mesh lead to collisions of packets in the access area, for which we propose a unique collision detection and recovery scheme based on a logical time-overlap method. Collisions make the end-to-end delay uncertain. Analysis of the associated delay is performed. We then propose algorithms to build such a *light-mesh* network. These algorithms are investigated in terms of network built-out costs and these costs are compared to a PON topology. Cost differences and a performance comparison with PON are presented as part of the numerical analysis.

© 2008 Published by Elsevier B.V.

Keywords: PON; Light-trail; Light-frame Ethernet in optical access

1. Introduction

The growth of broadband applications and the potential for broadband to promote e-commerce, e-governance, and other such lifestyle supplementing applications has promoted fiber-to-the-end-user as a plausible access networking technology. Fiber systems

in the last mile are gathering impetus as the gap between the cost of network deployment and broadband return-on-investment is reducing, coupled with an acute need for high-speed broadband access to support applications like video and data-centers. Presently, fiber in the last mile is characterized by the deployment of Passive Optical Networks (PONs) [2] that, due to the use of ‘passive’ technology, enable a star architecture — connecting multiple end-users (homes/offices/premises) to a single service provider-owned Central Office (CO). The star architecture facilitates broadcasting in

* Corresponding author.

E-mail address: ashwing@ieee.org (A. Gumaste).

one direction (i.e. downstream — from the CO to the end-users), and time-shared communication in the other direction (i.e. upstream — from multiple end-users to a single CO). For duplex communication between COs and end-users, PONs deploy a Dynamic Bandwidth Allocation (DBA) algorithm for upstream communication and use MAC-based addressing for downstream communication. In addition, PON uses wavelength diversity for the segregation of upstream and downstream traffic. The star topology and associated passive coupler have some significant advantages in terms of simplicity and ease of deployment as well as manageability, and a cost advantage over dedicated lines by making good use of the immense bandwidth offered by the fiber. A star topology facilitates *virtual* point-to-point communication between end-users and a central office. To enable this dual-wavelength communication in the star, a new network component was developed — the burst-mode transceiver [7]. The burst-mode transceiver allows fast operation (ON/OFF) of a laser and also enables efficient reception of ad-hoc bit streams requiring a minimal amount of time for the training sequence (for clock recovery). This allows efficient time-sharing of the upstream and downstream bandwidths. The principle detractor of the PON approach as a first-mile access solution is the huge cost involved in network layout which is due to fiber deployment. In fact, almost 90% of the cost involved in optical networks is related to fiber layout while only 10% of the capital costs are due to network equipment. The cost of first-mile fiber access networks is hence primarily driven by the cost of laying the fiber. The goal of this paper is to investigate alternate network architectures that reduce costs of deployment and, in particular, focus and showcase a new technology that can provide an interconnection mechanism for first-mile fiber access networks.

Our argument is that the star topology suited network requirements only because technology was nascent and other architectures were not possible. With recent technological advances and by amalgamation of low-cost optics along with smart control algorithms, it is possible to create new network architectures for fiber networks for last-mile based access networks. In fact, our primary contention is that if such a network architecture is possible (which we show later), then deploying an alternate topology — say a mesh, enables significant cost advantages without compromising on performance.

The rest of the paper is organized as follows: In Section 2 we discuss the conceptual positioning of

the light-mesh, in Section 3 we showcase delay and throughput evaluation of the light-mesh framework. Section 4 discusses topology abstraction algorithms that aid in topology design, while Section 5 presents simulation results and Section 6 concludes the paper.

2. Conceptual positioning — light-mesh

In this section, we describe the *light-mesh* concept, positioning it as an alternate fiber-access solution. The principal claim of light-mesh is a mesh network in the access that is largely passive with a small number of active elements used for control and support. The network configuration of the mesh leads to lower net utilization in terms of bandwidth. Light-mesh is based on the following principles and distinguishes itself as a novel optical access technology:

1. In PON (as shown in Fig. 1), every node is connected to the Splitter, which results in high cost due to laying of fiber. It is possible that the arm that connects the node to the splitter could connect many such arms in a sub-tree topology. However, this would result in an increased complexity of control algorithms for bandwidth arbitration in the tree.
2. A mesh that consists of nodes connected to each other with minimal connection to the central office as shown in Fig. 2 would result in significant fiber savings, as evident from the figure. Mesh nodes are naturally able to provide bypass to traffic in addition to being able to add/drop traffic.
3. The condition for a mesh implementation in an access network to result in lower fiber requirement than a star based PON was earlier investigated by us and reported in [1].
4. A node in an access network mesh has the following networking properties:
 - (a) *Sub-wavelength/packet-mode behavior*: In the access area, a single/two wavelength solution (like PON) is preferred due to cost considerations. Further, multiple nodes share the same wavelength bandwidth efficiently, implying that each node will receive sub-wavelength granularity. To enable efficient statistical multiplexing we desire packet-centric communication. Hence, a mesh solution in the access has to support packet-mode communication, leading to: (i) sub-wavelength granularity, and (ii) good statistical multiplexing.
 - (b) *Largely passive standalone equipment*: End-user equipment in the access area suffers from what can be described as a *consumer syndrome* i.e., electrical power is unpredictable. Since in a

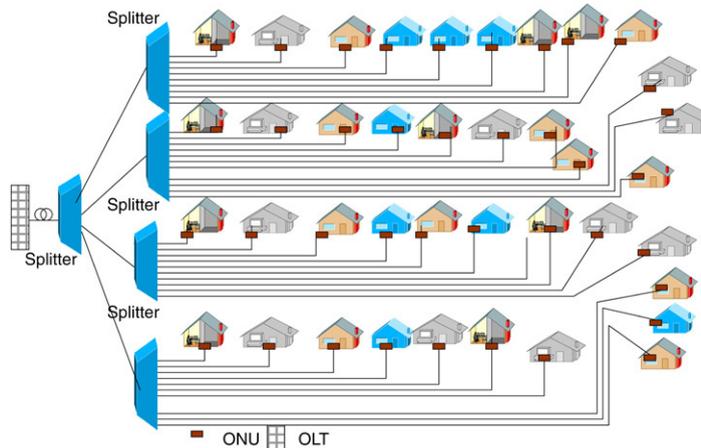


Fig. 1. Sample Passive Optical Network (PON) Topology.

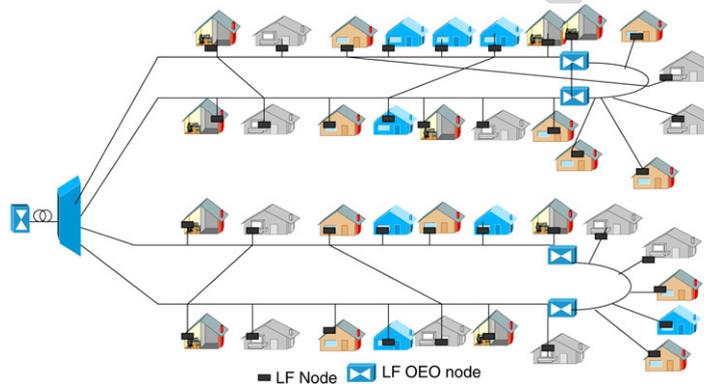


Fig. 2. Sample light-mesh network topology.

mesh traffic can hop across multiple nodes (using bypass properties) while traveling from a source to a destination, it is imperative that the node should be able to forward traffic even while it is switched OFF. Simple bypass using optical drop-and-continue techniques for optical circuits will not work as it does not support packet-mode communication, which, as we have seen earlier, is an essential technology for the mesh in the access area.

- (c) *Simple routing in a mesh with a good throughput:* Since node power behavior is uncertain, the routing capability of a node is uncertain. This implies the need for a simplistic routing mechanism with minimal dependence on the behavior of each node. The best behavior in such a situation is to have a virtual N^2 connectivity, whereby a node that is passive and gets switched OFF does not affect the overall routing mechanism. However, issues like flooding and learning preferred routes (to avoid

duplication) are critical and need to be solved in a mesh network.

- (d) *Fault-tolerance:* A mesh network comprising the end-user nodes can also provide a protection path in case of a fiber/node snap. This is due to the ability to provide alternate routes resulting from the N^2 connectivity.
- (e) *Simplified protocol and ability to support services:* An N^2 connected graph which is largely passive implies a shared medium. Another constraint on the graph is the need for efficient statistical multiplexing. Passive N^2 connectivity combined with good statistical multiplexing results in a paradox. This paradox implies requirement of an efficient protocol that can support new services. The protocol needs to be simple, for example it has to say something that uses carrier-sense type access, without unnecessary overheads, while maintaining low delay at higher loads.

2.1. Light-mesh principles of design

The light-mesh network is a mesh network that consists of two types of nodes: *OOO* and *OEO*. The former allows local add/drop as well as supports bypass — with the key being that the bypass involves no electronic regeneration. The latter allows local add/drop of signal and supports *filtered bypass* i.e. the signal flowing through an *OEO* node is converted into the electronic domain, processed, and then reinserted into the optical domain. The central idea behind the choice of *OOO* and *OEO* nodes is that *OOO* nodes are much lower in cost than the *OEO* counterpart (due to less processing) and can function for bypass traffic in the mesh even when they are powered OFF. *OEO* nodes, in contrast, work as gatekeepers — allowing for the specialized layer-2-over-optical neighbor discovery algorithm that is presented in [1].

The light-mesh network is designed based on the principles below:

1. *OOO* nodes are connected to each other. A contiguous set of *OOO* nodes is called a *string* (see Fig. 7 for an example).
2. Two strings are separated by a common *OEO* node. Hence, a string consists of *OOO* nodes with *OEO* nodes at both the ends of the string.
3. Signal flow in the string is assumed to be from one *OEO* node to the other *OEO* node, i.e. unidirectional. The first *OEO* node in the string is called the *start-node* while the last *OEO* node is called the *end-node*.
4. Within each string we have only *OOO* nodes, and each *OOO* node has two properties: (i) the ability to bypass signal without switching and, (ii) the ability to drop and add signal (also without switching). These properties imply that the string is an optical bus.
5. The entire light-mesh is assumed to have N^2 connectivity i.e. a route is available from every node to every other node. In summary, the light-mesh is a virtually connected optical network [3].
6. Two strings that share a common *OEO* node are called back-to-back strings. One of the two strings is an ingress string while the other is an egress string. This implies that one string feeds the signal into the common *OEO* node while the other string receives the signal from the *OEO* node.
7. A second type of connection is possible exclusively between *OEO* nodes and is called a *thread*. A thread is an all-optical point-to-point connection between two *OEO* nodes.
8. The layout of strings and threads is critical in maintaining N^2 connectivity, i.e. the position of nodes with respect to strings and creation of threads. This problem is addressed in detail in [1] which is a similar design problem. The variation in [1] is that threads can be between *any* two nodes, while in a light-mesh, threads are exclusively between two *OEO* nodes.
9. We now propose a node architecture for mesh networks that is largely passive for the access area. We will see that this architecture results in significant fiber savings. A node in the light-mesh network must not only support passive communications but also be able to be converted to support active communication (with the help of the in-line switch). The node architecture is based on the light-frame node architecture [1] and associated framework that we modify to suit the access area, and is described next. The architecture supports a 2×2 network element with additional access ports for client traffic. We first understand the light-mesh interconnection mechanism and then understand the node architecture in a top-down approach. The mesh consists of several single-wavelength strings (which could be enhanced to support multiple wavelengths, if required), and a few point-to-point thread interconnections between *OEO* nodes on different strings. In order to scale to multiple wavelengths, we must ensure that while designing the node architecture the individual components are wavelength agnostic; for example, the use of passive optical couplers. The strings are regulated in size by a *start-node* and an *end-node*, each of which is an active device, while intermediate nodes are entirely passive (implying no requirement of electrical power for communication in the mesh to carry through these nodes). Note, that for the nodes themselves to transmit/receive data into the mesh, they do need electrical power (for driving their lasers and receivers). The passive nodes are called *OOO* nodes as there is no conversion of passing light into the electric domain, while the active nodes are called *OEO* nodes as they convert the signal that passes through into the electronic domain (and reconvert back to the optical domain). Each node supports a string and can possibly support threads as well. To support a string (bus), each node is required to be able to drop-and-continue as well as have passive add functionality. Together, these two functions enable bus functionality. In addition, the node has to be able to support thread functionality as well. A conceptual schematic of

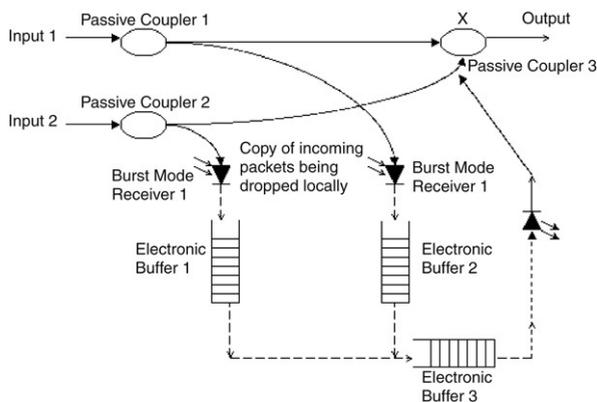


Fig. 3. Conceptual architecture of light-mesh node.

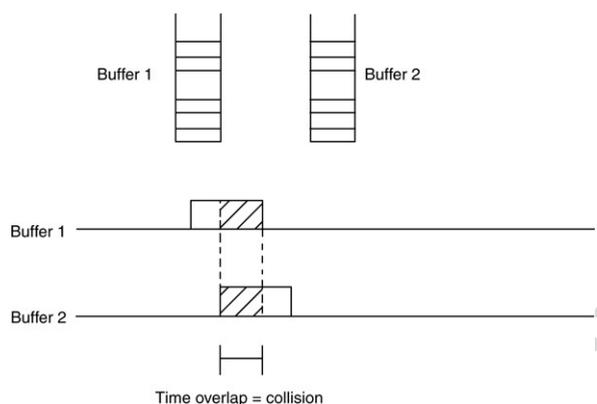
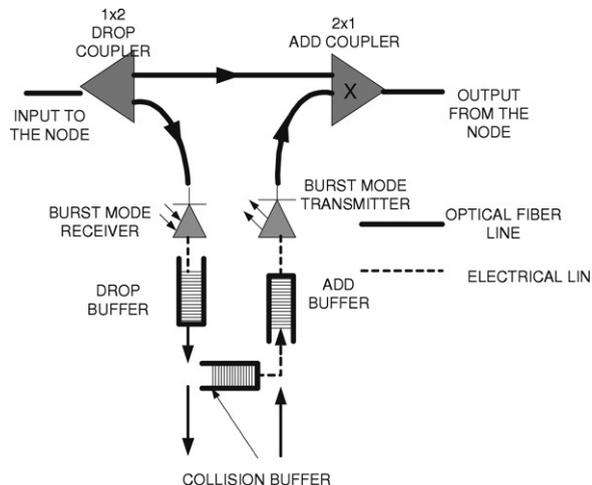
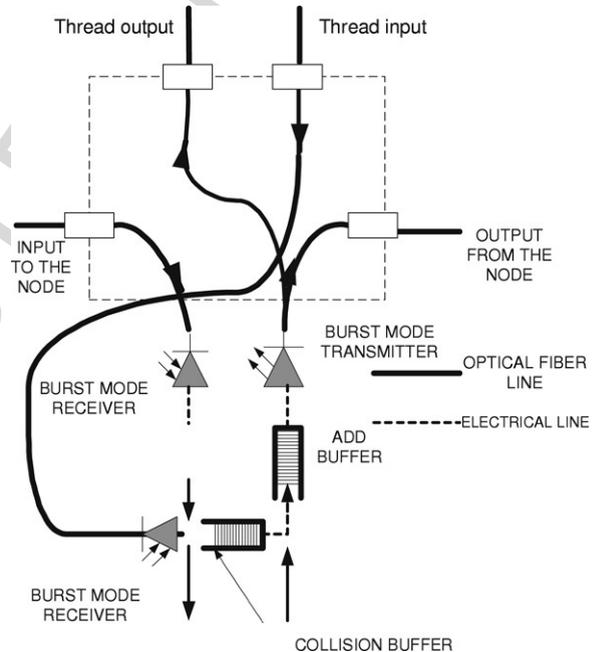


Fig. 4. Occurrence of collision at a light-mesh node.

such a node architecture is shown in Fig. 3 with working and collision described in Fig. 4. Detailed node architecture for the *OOO* variant is described in the next sub-section and shown in Fig. 5, while the architecture for *OEO* is shown in Fig. 6.

10. Communication in the network: the principle for communication is that when a packet is transmitted by a node into the network, it is guaranteed to reach the destination. This happens on a hop-by-hop basis. It may suffer multiple collisions in the process (hops) but once sent into the network, the source node does not have the responsibility of ensuring end-delivery. If a packet suffers a collision, it is the responsibility of the node at which the collision occurred to detect it and recover the collided packet. Collisions occur due to the largely passive nature of the network.
11. Collision management: since collisions can happen in the light-mesh network, our objective is to recover the data lost in collision. To do so, the following set of principles are adapted:

Fig. 5. *OOO* node architecture.Fig. 6. *OEO* node architecture.

- Collision happens only at nodes and in particular, at a certain specific position within a node. In Fig. 5, the passive coupler marked as 'X' is the point at which the collision occurs.
- All incoming packets that could possibly collide are optically split and a copy is stored at the node in an electronic buffer prior to the other copy colliding.
- No delay lines or optical-memory are assumed for storing the copy. The original packet is simply split and one copy is sent to a burst-mode-receiver while the other copy goes through. This

is the copy that can possibly collide with another packet. The copy that is sent to the burst-mode receiver is electronically saved in a buffer.

- d. Similarly, a copy of a locally injected packet is also stored (in another electronic buffer).
- e. Collision is detected when two or more electronic buffers indicate to a central logic about receiving packets in an overlapping time-interval. This is shown in detail in Fig. 4.
- f. The stored copies of the packets are then queued up for re-transmission.

2.2. Detailed node architecture

The node architecture for *OOO* nodes is now described in detail. The *OEO* node architecture is then presented as an amendment to the *OOO* architecture. The node architecture is shown in Fig. 5. It consists of two inputs and two outputs, with one input connected to the network and the other input used for the local addition of data. One of the two output ports is connected to the network, while the other output port is used for local drop. Inside the node we have two passive couplers connected back to back, with the first coupler in 1×2 configuration and the second coupler in a 2×1 configuration. The first coupler (also called the drop coupler) allows an incoming signal from the string to be split into two copies and drops one copy to a burst-mode receiver (BMRX) and all-optically forwards the other copy to the second coupler. This second coupler (add coupler), allows for passive addition of packets into the network. To insert packets into the network, we use a burst-mode transmitter (BMTX) that is connected to the add-coupler. The BMTX is connected on the client side to an electronic buffer, while it is connected on the network side to the add coupler. Electronic buffers are used for collision recovery as mentioned, and their role is now highlighted.

There are three buffers in an *OOO* node, the drop buffer, the add buffer, and a collision buffer. The drop buffer collects packets that are split from the drop coupler. The add buffer stores packets that are to be inserted into the network. These packets could be either locally generated or recovered after collision. The collision buffer stores exclusively packets that have collided — i.e. when a copy of the packet collides, the non-collided copy is stored in the collision buffer.

OEO architecture: The *OEO* variant (shown in Fig. 6) differs from the *OOO* version as follows: It has 3 input ports and 3 output ports. The first input port is the string input port that is connected to another node upstream in the string. The second input port is the

thread input port that allows packets to come to the node from another node (in another string, through a thread). Finally, the third input port is used for local addition. The first output port is a string output and this starts a new string. This means that an *OEO* node is a start-node for one string and an end-node for another string. The second output port is connected to the thread, allowing locally generated packets as well as packets from the string to be transmitted to the thread. The third output port is used for local drop of packets from the ingress string. There are no couplers present in the *OEO* version, indicating that there is no drop-and-continue or passive add functionality. All the incoming and outgoing ports are connected to one of the three burst-mode transceivers for communication. There are two BMRXs and one BMTX used for communication. The BMTX sends packets (in the optical domain) to both the string and the thread (splitting packets). The two BMRXs are connected to the string input port and the thread input port respectively. Note that in the *OEO* variant there is no possibility of collision. Also note that an *OOO* node can be converted into an *OEO* node by simply disconnecting the connection between the add-coupler and drop-coupler in the *OOO* node.

2.3. Working of the node

To send data into the network, the node generates the frame (presumably from a client side device) and then stores this frame electronically in the add-buffer. It simultaneously senses the drop-buffer for any activity on the string line, and if there is no upstream node transmitting data, it sends the frame stored in the electronic buffer. A collision can happen while a node is transmitting a packet into the network, another packet sent by an upstream node arrives in time overlapping with the present transmission (as shown in Fig. 4). However, both packets are recovered by using the principle of splitting-prior-to-collision, storing, and then forwarding the two copies later in time.

Based on the working of the node, we compute the average delay that a packet would incur while traveling from one node to another in the light-mesh network and the associated throughput in Section 3.

2.4. Related work: Variation of light-mesh from light-frame

While the light-mesh node architecture is similar to the light-frame architecture, there are some significant differences between the two that lead to a lower cost implementation as well as a lower delay profile for our approach. The differences are listed below:

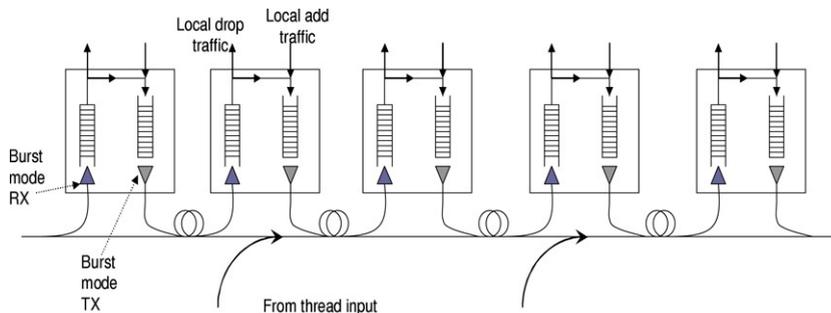
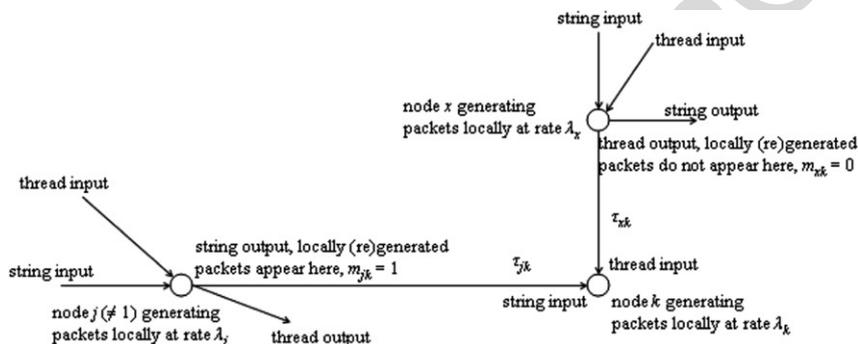


Fig. 7. Example of a string.

Fig. 8. Computation of traffic incoming to a given node k .

1. In the light-mesh, threads are supported only at *OEO* nodes while in the light-frame concept any node can potentially support a thread.
2. Due to the above feature of threads, the light-mesh node architecture has a far smaller probability of collision, as the only case of collision is between a packet arriving from an upstream node and a locally generated packet, whereas in the light-frame framework, there are five cases of collisions and these are described in [1].
3. In the light-mesh node, there are no moving parts like Mach Zehnder Interferometers (MZIs). In the light-frame framework, each node is equipped with three MZIs used for regulating flow into the node.
4. The light-mesh is finally designed as a specific solution for access, as an alternate technology to PON [5,6].

3. Network parameter computation

In this section, we discuss the analytical model that governs the light-mesh framework. In particular, we are interested in computing the end-to-end delay that a packet experiences when sent into the light-mesh. This delay is not deterministic but depends on the number of collisions and the time lost in retransmission. This stochastic behavior, as a result of collisions, also has an

effect on the net throughput of the system. To compute the delay and the throughput, we need to compute the blocking probability at a node as a function of the load.

3.1. Computation of blocking probability at each node of an optical string

Consider a light-mesh of n nodes where any given node i generates packets as per the Poisson distribution with parameter λ_i . Let the propagation delay between any two consecutive nodes be T . Since, we use a buffer to save a copy of packets incoming to the node (for collision recovery) while the packets are being transmitted, it affects the effective transmission rate, which also follows the Poisson distribution but with parameter λ'_i .

We also define τ_{jk} as the traffic injected from node j to node k in the link (string or thread) connecting j to k (as shown in Fig. 8). Mathematically, τ_{jk} is stated as

$$\tau_{jk} = \begin{cases} m_{jk}\lambda'_j + \sum_{\forall i} a_{ij}\tau_{ij}, & \text{if } j \text{ is not the first node,} \\ \lambda_j, & \text{otherwise} \end{cases} \quad (1)$$

where,

$$a_{ij} = \begin{cases} 1, & \text{if there exists a directed edge from } i \text{ to } j \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

$$m_{jk} = \begin{cases} 1, & \text{if there exists a string connection} \\ & \text{from } j \text{ to } k \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

$$\lambda'_j = \begin{cases} \lambda_j [1 + P_B(i)] + P_B(i) \sum_{\forall i} \tau_{ij}, & \text{if } j \text{ is an OOO node} \\ \lambda_j + \sum_{\forall i} s_{ijk} [1 - P_B(i)] \lambda'_i, & \text{if } j \text{ is an OEO node} \end{cases} \quad (4)$$

and

$$s_{ijk} = \begin{cases} 1 & \text{if the shortest pat from } i \text{ to } k \text{ is through } j \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

Let A_{ij} be an event which represents collision of packets at node j in a light-mesh due to a packet received from node i (where $i \in I$ and $I = \{i : a_{ij} = 1\}$). A packet received from node i at time t collides if node j transmits a packet between the time interval of $t - (j - i)T$ to $t - (j - i)T + L/C$, where L is the length of the packet, C is the capacity of the network, and $i < j$. Further, let X_i be a random variable which represents the generation of packets at node i . We assume that the generation of packets follows the Poisson distribution with parameter λ_i . We also assume L to be a random variable for the length of packets that follows the exponential distribution with parameter μ . Thus, the event A_{ij} is formulated as:

$$A_{ij} \equiv t - (j - i)T < X_i < t - (j - i)T + L/C \quad (6)$$

where $i < j$, C is the channel capacity and t is time.

Thus,

$$P(A_{ij}) = P(t - (j - i)T < X_i < t - (j - i)T + L/C)$$

$$\text{or, } P(A_{ij}) = \int_0^\infty P(t - (j - i)T < X_i < t - (j - i)T + L/C) P(L = l) dl$$

$$\text{or, } P(A_{ij}) = \int_0^\infty P(t - (j - i)T < X_i < t - (j - i)T + L/C) \mu e^{-\mu l} dl$$

$$\text{or, } P(A_{ij}) = \int_0^\infty [1 - P(t - (j - i)T \text{ and } t - (j - i)T + L/C)] \mu e^{-\mu l} dl. \quad (7)$$

Since we take only those nodes into consideration which have a directed edge (string or thread) to node j , we use the traffic parameter τ_{ij} to compute $P(A_{ij})$. Thus,

$$P(A_{ij}) = \int_0^\infty (1 - e^{-\tau_{ij}l/C}) \mu e^{-\mu l} dl. \quad (8)$$

On solving and re-arranging (8), we get,

$$P(A_{ij}) = \frac{\tau_{ij}}{\tau_{ij} + \mu C}. \quad (9)$$

Blocking probability at node j is a union of all such events A_{ij} under the set $I = \{i : a_{ij} = 1\}$. In a light-mesh, any given node j can have $a_{ij} = 1$ for a maximum of two values of i . Let these two values of i be i_1 and i_2 . Thus, the blocking probability at node j can be mathematically stated as:

$$P_B(j) = P(A_{i_1j}) + P(A_{i_2j}) - P(A_{i_1j} \cap A_{i_2j}). \quad (10)$$

Since all X_i are independent random variables and all A_{ij} are independent for all I , we can say

$$P(A_{i_1j} \cap A_{i_2j}) = P(A_{i_1j})P(A_{i_2j}). \quad (11)$$

Combining (2) and (3) we get,

$$P_B(j) = P(A_{i_1j}) + P(A_{i_2j}) - P(A_{i_1j})P(A_{i_2j}). \quad (12)$$

Thus, on combining (5) and (6), we get the complete expression of blocking probability at node j ,

$$P_B(j) = \frac{\tau_{i_1j}}{\tau_{i_1j} + \mu C} + \frac{\tau_{i_2j}}{\tau_{i_2j} + \mu C} - \frac{\tau_{i_1j}}{\tau_{i_1j} + \mu C} \frac{\tau_{i_2j}}{\tau_{i_2j} + \mu C} \quad (13)$$

where $P_B(1) = 0$ because no collision is possible at node 1.

3.2. Computation of delay

In a light-mesh, we assume the propagation delay and processing delay to be negligible and assume that the average end-to-end delay will depend upon the queuing delay suffered by the packets at every node. We model each node as an M/M/1 queue to compute the queuing delays at each node. The service rate of any such queue j in a light-mesh is λ'_j and the arrival rate is, $\lambda_j + P_B(j) \sum_{\forall i} \tau_{ij}$.

Thus, the queuing delay at each node can be computed as

$$\frac{1}{\lambda'_j - \lambda_j + P_B(j) \sum_{\forall i} \tau_{ij}}.$$

3.3. Computation of throughput

We define the throughput of a network as the amount of useful information received by the receiver over the

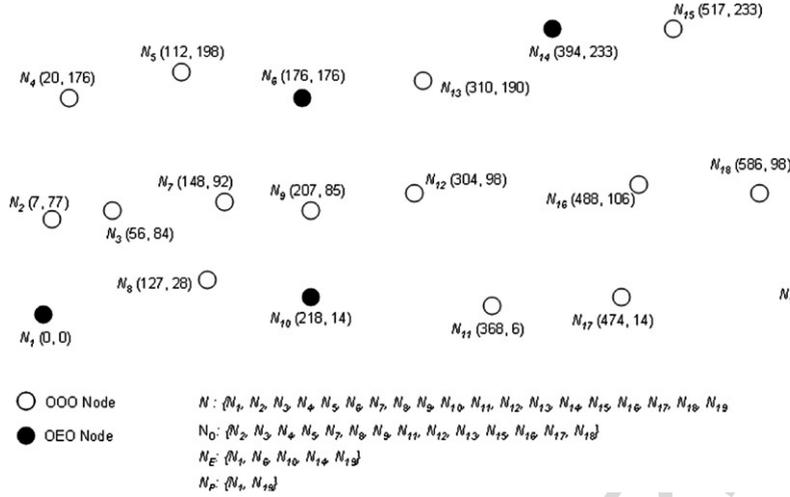


Fig. 9. Sample node layout — design problem in light-mesh networks.

network per unit of time. In our case, the throughput is equal to the number of bits transmitted by the OEO node connected to the splitter. Since an OEO node removes all corrupt packets from the system, we can define the throughput for the light-mesh as the traffic injected by the last OEO node (l). Thus the throughput for a light-mesh is defined as

$$S = \lambda_l + \sum_{\forall i} s_{ilk} [1 - P_B(i)] \lambda'_i \quad (14)$$

where, s_{ijk} is defined in (5).

4. Topology abstraction algorithm

While designing a light-mesh network, of particular interest to us are optimal routes, i.e. strings and threads that result in maximum fiber savings while maintaining N^2 connectivity and an acceptable upper bound on delay. We hence propose a topology abstraction algorithm that results in the creation of strings and threads and an assignment of nodes to them in tractable time. The optimal solution for the topology abstraction algorithm is shown to be NP-complete by reducing this to a multiple Steiner tree as in [1].

The input to our topology abstraction algorithm is a set of nodes denoted by $N = \{N_1 \dots N_v\}$. Let N_E (a subset of N) be the set of OEO nodes and N_O (again, a subset of N) be the set of OOO nodes such that $N = (N_E \cup N_O)$. Let N_P (a subset of N_E) be the set of OEO nodes connected to the passive splitter. A sample topology-design problem is shown in Fig. 9.

The output of the algorithm is a graph $G(V, E)$ which satisfies the following constraints denoted by $C_1(G)$:

1. The end-to-end delay between any two nodes N_i and N_j (such that $N_i, N_j \in N$) does not exceed Δ_{\max}
2. The optical signal at any point in G does not drop below a threshold P_{th} .

The algorithm is divided into two parts. The first part (shown in Algorithm 1) creates strings and works as follows: begin by selecting an OEO node connected to the passive splitter. This node becomes the start-node of the first string. The power threshold and delay constraints determine the OEO members and the end-node of this string. The OEO node selected as the end-node for the previous string becomes the start-node of the next string, and the process continues until all the nodes have become part of a string.

Algorithm 1 (String Creation from a Set of Nodes N).

- 1: **let** y be an empty set of nodes
- 2: **select** an OEO node ($starNode$) from set N_P
- 3: **add** it to Y
- 4: **while** ($Y \neq V$)
- 5: **find** OOO members (M) from set N_O for the string which satisfy the delay and power constraints
- 6: **add** M to Y
- 7: **find** an available OEO node ($endNode$) from N_E to be the end-node which satisfies the delay and power constraints
- 8: **add** E to Y
- 9: **create** a string S with $starNode$, M and $endNode$
- 10: **add** S to the $stringList$
- 11: **let** $starNode = endNode$
- 12: **end while**
- 13: **populate** G as per the set $stringList$

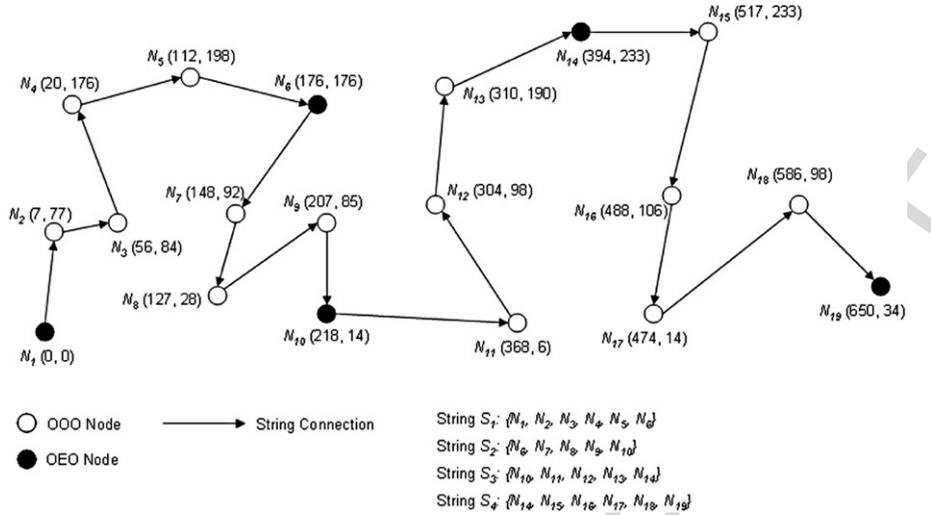


Fig. 10. Light-mesh with only strings.

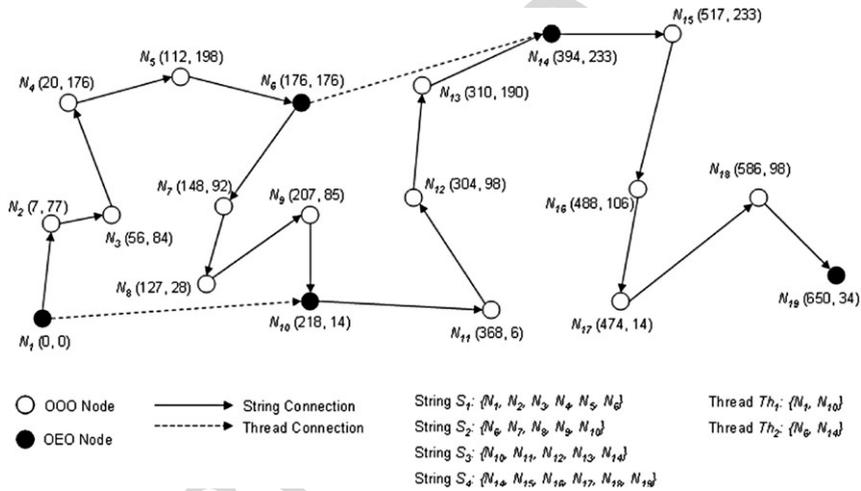


Fig. 11. Network topology design of light-mesh (strings and thread).

Thus, we now have a graph containing a set of all strings (as shown in Fig. 10) denoted by *stringList*, where each node is part of a string. Now the second part of the algorithm creates threads between two OEO nodes N_i and N_j if they satisfy the following constraints denoted by $C_2(N_i, N_j)$:

1. N_i and N_j are OEO nodes
2. Thread output of N_i is free
3. Thread input of N_j is free
4. Connecting N_i and N_j does not create a cycle
5. N_i and N_j are not part of the same string.

Thus, to obtain the final graph (as shown in Fig. 11), we use Algorithm 2 to create threads.

Algorithm 2 (Thread Creation from a Set of String *stringList*).

- 1: **while** (G is not N^2 connected OR end-to-end delay $> \Delta_{\max}$)
- 2: **find** nodes N_i and N_j that satisfy the constraints $C_2(N_i, N_j)$
- 3: **create** a thread between N_i and N_j in G
- 4: **end while**

5. Simulation results

We present our simulation results in two parts: (1) fiber savings as a result of using a light-mesh over PON; and (2) delay simulation results for the light-mesh.

5.1. Fiber savings simulation

We assume every house is 20 m apart (i.e. each plot is about 400 sq meters), every row has 10 houses and

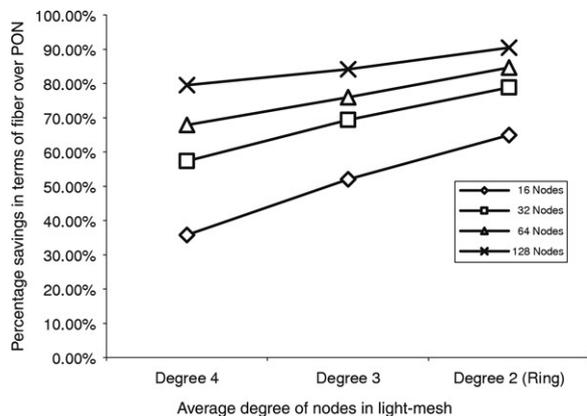


Fig. 12. Percentage fiber savings over PON by using light-mesh.

rows are connected in a Manhattan Street Network type topology (as shown in Fig. 1). In the PON model, a splitter is assumed to connect either a row or a series of rows and in the light-mesh model, strings and threads are mapped on to the rows (as shown in Fig. 2) to facilitate connectivity to the end-users.

We assume that the distance between the passive splitter and the nodes is appreciably larger than the distance between any two nodes. If this were not the case, then the fiber could be terminated at the passive splitter and copper/coaxial cables could be drawn between the splitter and the end-user nodes, thus reducing the cost without affecting the performance.

The simulator calculates the length of fiber needed to deploy a PON-based network and the length of fiber needed to deploy a light-mesh (with an average of 2, 3, or 4 incoming or outgoing links at every node) based on the coordinates of the passive splitter and the end-user nodes. Percentage savings in terms of the fiber length by deploying a light-mesh over PON for 16-node, 32-node, 64-node, and 128-node light-meshes are shown in Fig. 12.

5.2. Delay simulation results

We observed the delay for 16-node, 32-node, 64-node, and 128-node LF frameworks as the load varied. λ is defined as the rate at which packets were generated at each node, and was varied from 100 packets per second to 1500 packets per second in steps of 100. The packets size was assumed to be 1500 bytes (denoted by L) and the transmission rate to be 1 Gbps. The overall load on the network (assuming N nodes) was computed as

$$\text{Overall Load} = \lambda \times N \times L. \quad (15)$$

Fig. 13 presents the results of our simulation. We observe that as the number of nodes increases, the

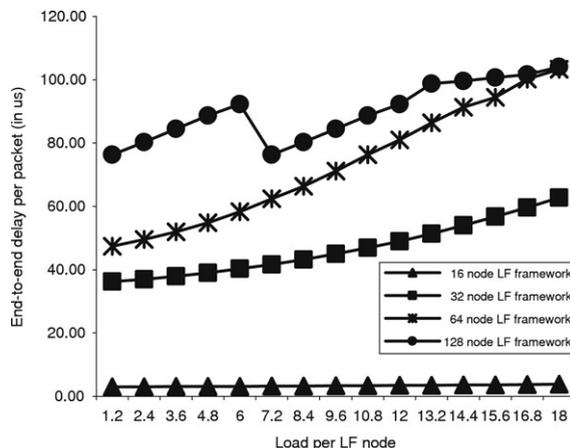


Fig. 13. Average end-to-end delay simulation results.

number of collisions also increases and hence the delay increases. This is shown in Fig. 13, where the average end-to-end delay increases more rapidly (as λ increases) for the 32-node and 64-node cases than for the 16-node case.

For the case of the 128-node LF framework, we see that as the load increases to 6 Mbps per node the average delay increases, after which it decreases till 7.2 Mbps. As we further increase the load the average delay increases again. This happens due to the fact that at any given load, the probability of successful transmission is higher for a node that is closer to the splitter.

This means that at a given load packets from nodes far from the splitter observe higher delays due to more collisions. As the load is increased, the number of collisions increases further. The average delay is computed for the packets that reach the splitter and at high loads the contribution of packets from nodes closer to the splitter is greater. Since such packets observe lower delay this decreases the average delay. As the load increases above 7.2 Mbps per node, the collisions for the packets reaching the splitter from nearer nodes increases and hence the average delay increases again.

6. Conclusion

In this paper, we present the light-mesh framework — a new architecture solution for first-mile access networks. In particular, we propose a unique node architecture and connection methodology. Our contention is that our light-mesh framework is lower in cost than the contemporary PON solution. We investigate the analysis and design of this framework. A rigorous stochastic evaluation is presented followed by design guidelines that lead to topological abstraction. Based on these

1 guidelines and evaluations we compare our proposed
 2 light-mesh to PON solution and show significant cost
 3 improvement while maintaining good networking pa-
 4 **Q1** rameters like delay and throughput for our approach.

5 **Uncited references**

6 [4].

7 **Acknowledgements**

8 The authors wish to thank Prof. Biswanath Mukher-
 9 jee (UC-Davis) for some of his insights into this work.
 10 The authors also wish to thank Prof. Krithi Ramam-
 11 ritham and Prof. Bernard Menezes (IIT Bombay) for
 12 their support. This work was supported by Cotrion Net-
 13 works Pvt. Ltd.

14 Ashwin Gumaste is currently on sabbatical with
 15 MIT, Cambridge, MA, United States.

16 **References**

[1] A. Gumaste, S. Zheng, Light-frames: A pragmatic solution to

- optical packet transport — extending the ethernet from LAN to
 optical networks, *IEEE Journal of Lightwave Technology* (2006).
 [2] G. Kramer, B. Mukherjee, G. Pesavento, Ethernet PON (ePON):
 Design and analysis of an optical access network, *Photonic
 Network Communications* 3 (3) (2001) 307–319.
 [3] Kang Xi, Shin'ichi Arakawa, Masayuki Murata, Virtual
 fully connected WDM network: Architecture, scheduling
 and performance evaluation, broadnets, in: *First International
 Conference on Broadband Networks, BROADNETS'04*, 2004,
 pp. 222–232.
 [4] A. Gumaste, I. Chlamtac, J. Jue, Light-frames: Pragmatic
 implementation for optical packet transport, in: *39th IEEE
 International Conference on Communication, ICC, Paris, France*,
 2004.
 [5] A. Agrawal, A. Gumaste, M. Chamania, N. Ghani, Topology
 abstraction algorithms for light-mesh — an alternate model
 for PON, in: *IEEE/OSA Optical Fiber Communication (OFC)
 Conference, San Diego, February 2008*.
 [6] G. Lambertsen, T. Yamada, PON mobile access network
 without Mobile IP FA switchover, in: *The 13th IEEE
 International Symposium on Personal, Indoor and Mobile Radio
 Communications*, vol. 5, Sept. 2002, pp. 2137–2141.
 [7] E. Sackinger, Y. Ota, T.J. Gabara, W.C. Fischer, A 15-mW,
 155-Mb/s CMOS burst-mode laser driver with automatic power
 control and end-of-life detection, *IEEE Journal of Solid-State
 Circuits* 35 (2) (2000) 269–275.