An Autonomic Virtual Topology Design and Two-Stage Scheduling Algorithm for Light-trail WDM Networks

Ashwin Gumaste, Tamal Das, Ashish Mathew and Arun Somani

***Abstract*[1]—Light-trails have been proposed as a solution for optical networking to provide support for emerging services such as *Video-on-Demand*, Pseudo-wires, data-center etc. To provision these services we require features such as dynamic bandwidth provisioning, optical multicasting, sub-wavelength grooming and a low-cost hardware platform – all of which are available through the light-trail concept. Architectural, performance, resilience and implementation studies of light-trails have led to consideration of this technology in metropolitan networks. In the area of architecture and performance, significant literature is available in terms of static network optimization. An area that has not yet been considered and which is of service provider importance (from an implementation perspective) is the stochastic behavior and dynamic growth of the light-trail virtual topology. In this paper, we propose a two-stage scheduling algorithm that efficiently allocates bandwidth to nodes within a light-trail and also grows the virtual topology of light-trails based on basic utility theory. The algorithm facilitates growth of the light-trail topology fathoming across all the necessary and sufficient parameters. The algorithm is formally stated, analyzed using Markov models and verified through simulations resulting in 45% betterment over existing LP or heuristic models. The outcome of the growth algorithm is an *autonomic optical network* that suffices service provider needs while lowering operational and capital costs. The paper presents the first work in the area of dual topology planning – at the level of connections as well as at the level of the network itself.**

***Index Terms*—light-trail, ROADM.**

I. Introduction

Next generation services such as video-on-demand and data-center/cloud computing are expected to dominate much of operator revenues [1]. Optical networks that provision such services need to demonstrate characteristics that support dynamic bandwidth provisioning, optical multicasting, sub-wavelength (preferably all-optical) grooming and low-cost node architecture [2]. We proposed light-trail (LT) in [3-6] as a solution that facilitates the aforementioned features. A LT is a generalized lightpath such that nodes along the path can take part in time-shared communication. A LT is analogous to a multipoint-to-multipoint unidirectional optical-wavelength bus. To support the wavelength bus, nodes have properties of allowing an incoming signal to be *dropped-and-continued*, as well as to *passively add* signal [7]. The resulting wavelength bus can cause potential conflicts amongst transmitting nodes. To avoid conflicts and for management (creation/deletion/modification) of LTs we deploy an *out-of-band* (OOB) control channel that is optically dropped and electronically processed at every node.

On one level of abstraction, nodes in a LT compete for the LT bandwidth and at another level, the network has to be configured to create/delete/modify (i.e. shrink/dilate) LTs. The dual-problem of bandwidth allocation within a LT and the creation/deletion/modification of LTs is complicated, and requires a pragmatic management plane. *We propose, analyze, simulate and verify an autonomic virtual topology design and scheduling algorithm that solves this dual problem in LT WDM networks*. The algorithm has two-stages – (1) Stage 1: whereby bandwidth allocation occurs within a LT, subject to delay sensitivity and bandwidth intensity of the requesting nodes/flows and (2) Stage 2: that governs the virtual topology of the LT system.

In stage 1, one of the LT nodes is selected as an *arbiter/controller,* and is responsible for bandwidth arbitration. The arbitration process involves nodes periodically advertising their utility through the OOB control channel to the arbiter. Bandwidth requests are made in advance using a time-slotted model described in Section II. The utility model further gives inputs and facilitates in designing the virtual topology, whereby new LTs are created/destroyed/modified, depending on the utility exhibited by the interacting flows/nodes. The algorithm presented in this paper is important from the perspective of providing an autonomic control to LT WDM networks – in solving the bandwidth and topology design issues together.

A partial class of design problems resulting from LTs shown in [8-12] only consider topology design, while we consider both intra-LT scheduling as well as topology design. *This work realizes the true potential of LTs as an optical grooming solution whose aim is to reduce the inventory while achieving good performance due to LT features.* The proposed topology design algorithm can in principle also be extended to next generation Carrier Ethernet, SDH/SONET and OTN networks [13].

This paper is organized as follows: Section II describes the System Design that we consider for our two-stage scheduling algorithm. Section III describes the virtual

Manuscript received September 7, 2010.

Ashwin Gumaste, Tamal Das and Ashish Mathew are with Gigabit Networking Laboratory, IIT Bombay, India. Arun Somani is with the ECE Dept., Iowa State University, Ames, IA 50010

topology growth algorithm. Section IV analyzes the scheduling algorithm from a stochastic perspective. Section V describes the behavior of system through simulations while Section VI concludes the paper.

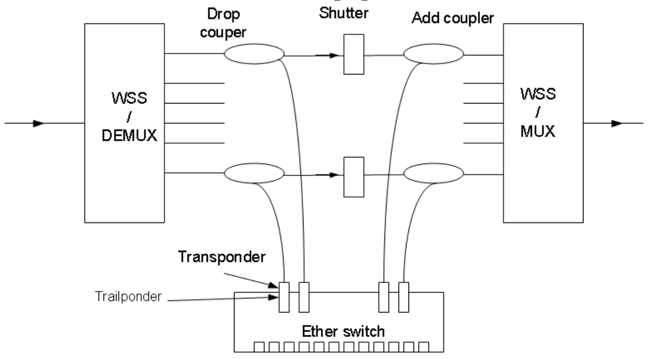


Fig. 1. Light-trial node architecture.

II. System Design

This Section describes the LT system design and node architecture. For simplicity, we assume a 2-fiber WDM ring network, though our work can easily be extended to mesh networks [12, 14]. The node architecture of LTs is shown in Fig. 1, and a comparison of LTs to lightpaths is shown in Fig. 2. Details regarding node functioning are presented in [6, 15]. Specific engineering assumptions on LTs node architecture for our algorithm are now presented.

The LT node architecture (Fig. 1) has evolved from a basic ROADM – Reconfigurable Optical Add-Drop Multiplexer. The architecture in Fig. 1 is for a unidirectional fiber system and two such elements make up a duplex node on a ring network. Composite WDM signal from each fiber passes through a de-multiplexer (a Wavelength Selectable Switch – WSS) that separates individual wavelengths. Each wavelength passes through a LT Optical Retrieval Section (LORS) that facilitates the node to locally access (drop/add) data from/to the LT. The LORS consists of two passive optical couplers in 1×2 and 2×1 configuration, separated by an ON/OFF optical switch. The first coupler *drops-and-continues* the incoming signal, while the second coupler enables the node to (passively) add signal into the LT. One of the channels called the optical supervisory channel (OSC) is used for control. This out-of-band (OOB) control channel (dropped and processed at each node) along with the bus property is necessary for nodes to *time-share* the LT bus bandwidth. Time sharing implies that at any given time, only one node can transmit data (*i.e.* enable a *connection*) across a LT. Conventional transponders are replaced with burst-mode *trailponders* [6, 15] to support fast-connection setup/tear down.

To time-share efficiently, we propose the use of a scheduling algorithm whose functioning is now defined. The LT channel bandwidth is assumed to be divided into time-slots. By processing the OOB control channel at every node, we are able to assume synchronization between the nodes at the control layer. This synchronization extends to the data channels implying a stage set for time-slotted communication. The size of the time-slots is of particular importance and represents an interesting trade-off: small time-slots of the duration 0.125~0.625 ms are desired to achieve performance that is comparable to SONET/SDH, while for changing the virtual topology (LT-reconfiguration) larger durations of 2-3ms are desired. To solve this trade-off, we consider pragmatic networks [16-17] and it is observed that the virtual topology seldom changes – implying that it is desirable to have small time-slots (of duration 0.125~0.625 ms). The choice of small time-slots helps in replicating SONET/SDH like latency-sensitive performance. In addition, to support topology changes, we assume an integral multiple of time-slots giving sufficient time without compromising on service quality.

A. Network Model

An N-node time-slotted system with various service classes from the universal set of services $S = \{S_1, S_2, \ldots, S_h, S_{|S|}\}$ is assumed. Each service arrival is characterized by a Poisson-arrival process and packet sizes are exponentially distributed. T_S denotes the duration of a time-slot. Each node N_i in the network comprises of $|S|$ input buffers, represented by B_{hi}, $\forall h = 1,\ldots, |S|$. We are interested in the system dynamics at node N_i at time t_o, $\forall i$ (which we eventually generalize to a steady state). Let λ_{hi} be the arrival rate of service requests of type S_h and $B_{hi}(t_o)$ be the size of the buffer consisting of requests of type S_h at node N_i. Let the cumulative buffer size at N_i be denoted as $B_i(t_o) = \sum_{h=1}^{|S|} B_{hi}(t_o)$. Assuming independence, $\lambda_i = \sum_{h=1}^{|S|} \lambda_{hi}$ denotes the effective arrival rate at N_i.

Let $\{\Delta_h$: $h = 1,\ldots,|S|\}$ denote the maximum tolerable delay of each service class, whereas $\{\delta_{ji}$: j=1,2,...$\}$ denotes the time interval within which the jth service request must be serviced after its arrival at N_i. Let H_{ji} denote the absolute arrival time of the jth request at N_i. Further, x_{ji} denotes the time elapsed since the arrival of the jth service request at N_i.

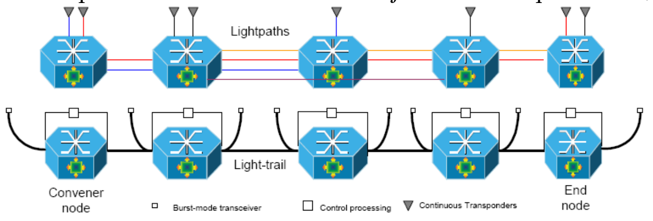


Fig. 2. Comparison of lightpaths and LTs

Then, $\sigma_i(t_o) = \max\{x_{ji}, \forall h : S_h \in B_i(t_o)\}$ is the time elapsed since the first packet (of any service) enters the buffer. Where, $S_h \in B_i(t_o)$ implies the existence of the S_h^{th} service at the buffer at N_i.

Let (Δ_h–x_{ji}) represent the remaining time for a particular service S_h, then $\psi_i(t_o) = \min\{\Delta_h - x_{ji}, \forall h : S_h \in B_i(t_o)\}$ represents the absolute allowable delay limit.

We define criticality as $\sigma_i(t_o)/(\sigma_i(t_o) + \psi_i(t_o))$. While criticality is important in computing a node's utility, a second factor of interest is a node's buffer occupancy. Since nodes share the LT bandwidth the two components of occupancy and criticality are imperative in the computation of utility. We further desire to normalize these two components. Buffer occupancy is stated as a ratio of the occupation of packets (bits) in a buffer to its maximum size. Criticality is normalized over an activity period – how long since the buffer is active in receiving packets since its last transmission. Hence, our utility metric intuitively takes into consideration both the criteria of buffer occupancy and criticality; and after normalizing these two criteria; maximizes them to obtain the utility value. The utility measure that a node sends to an arbiter for bandwidth assignment in the LT is defined as:

$$bid_i(t_o) = \max\left\{\frac{B_i(t_o)}{B_{\max}}, \frac{\sigma_i(t_o)}{\sigma_i(t_o)+\psi_i(t_o)}\right\} \quad (1)$$

This utility value is computed over all the requests provisioned at a node. The requests are assumed to have their own independence, such that requests (flows) are free (with appropriate utility constraints) to associate with LTs. *Functioning*: At the beginning of every time-slot for every LT to which a node has an active trailponder, the node computes its utility for the next slot and sends it to the arbiter of the LT through the control channel. The arbiter is typically the end-node of a LT. In LT k, the highest utility is then $\max\{bid_i(t_o), \forall i : N_i \in k\}$, and the winning node is $\arg\max\{bid_i(t_o), \forall i : N_i \in k\}$. The winning node would transmit (establish a connection) in the next $(t_o+1)^{\text{th}}$ time-slot. It may happen that a flow at a node may not be serviced within its permissible delay limit, or a buffer may overflow. Such a flow or such a buffer implies an event called *recourse*, which requires the algorithm to re-evaluate the flow and provision it through some other LT or create a new LT.

B. Transmission Scheduling Discipline

As a node is allotted a time-slot for transmission, its buffer is emptied based on the following scheme. If the utility value of the winning node was dominated by *buffer occupancy*, then its buffer is emptied in the decreasing order of its occupancies with respect to the different services; whereas if the winning utility value was dominated by the *service criticality* at the node, the node's buffer is emptied according to the Earliest-Deadline-First (EDF) discipline [18] implying the service with most critical packets goes out first.

C. Role of the Network Management System (NMS)

We assume a centralized NMS that validates all topology change decisions. The NMS uses the topology growth algorithm explained in Section III and directs requests to appropriate LTs or creates/modifies LTs. *All intra-LT decisions are taken by the LT arbiter node, while all inter-LT decisions involving dimensioning, or creation of new LTs are validated by the NMS* (*in real time*). For intra-LT decisions, the arbiter uses the control channel to get information from all the nodes in the LT. Likewise, the control channel is also used for communication between the arbiter of each LT and the NMS. The NMS can be viewed as a two-layer hierarchy: (1) all the nodes communicating to the arbiter via the control channel (for intra-LT decisions) and (2) all the arbiters communicating to the NMS (using the control channel) for inter-LT decisions.

III. The Light-trail Virtual Topology Growth Algorithm

In this Section, we describe the proposed LT topology growth algorithm. The heuristic topology growth algorithm works by encompassing an exhaustive set of growth options. The algorithm uses the notion of utility described in Section II. The algorithm assumes two types of utility: (1) that of a flow or a node towards a time-slot and (2) that of a LT towards a network. At an intrinsic level, we first attempt to accommodate new requests through existing LTs. If existing LTs are not available, then the algorithm makes an effort to map the requests to LTs by dimensioning the LTs. Only under the event that suitable LTs cannot be found, or cannot be dimensioned, does the algorithm create a new LT for accommodating a request. Moreover, the LT is given a finite amount of time to attract other flows that have graphical overlap with the newly created LT. The LT continues to exist if it is able to find flows that together are within a defined *utility* threshold. This threshold is called the *operating range (OR)* of an LT and will be defined later. A second parallel thread that the algorithm pursues is that of reconfiguration. Reconfiguration is important to distribute requests amongst existing LTs or to create new LTs (while destroying existing LTs) that lead to network-wide benefits and better utilization. The algorithm facilitates reconfiguration at two levels – a flow/request level and at a LT or virtual topology level.

The growth algorithm uses three premises: birth-death, validity and rationality; which are now described.

1. Birth-Death property: The decision to create, destroy or dimension (grow/shrink) a LT is determined by the *utility* that the LT provides to the network. The utility of a LT is the cumulative utilities provided by all the flows, divided by the number of hops (spans) in the LT. For example in a 4-node LT N_1–N_4 (of capacity 1Gbps), if the flows f_{12} (=150Mbps), f_{13} (=200Mbps), f_{23} (=75Mbps) and f_{24} (=125Mbps) are provisioned, then we define the absolute-utility of the LT as $\overline{U}(k)$ and is for this particular case given as (1/3)*(550Mbps/1Gbps)=0.18. Generalizing,

$$\overline{U}(k) = \frac{1}{(n(k)-1)\times C}\sum_{f_{ij}\to k} f_{ij} \quad (2)$$

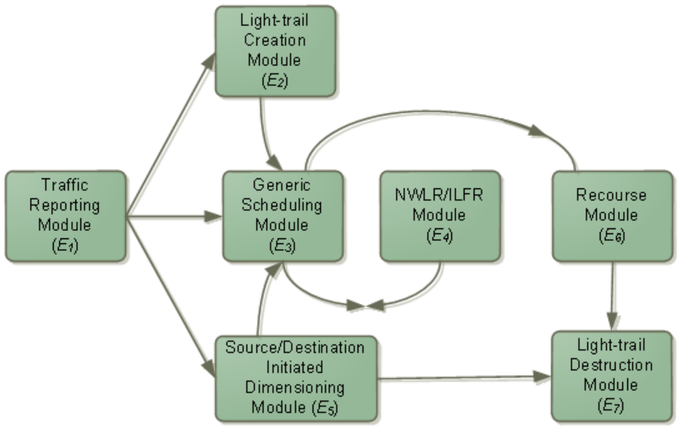
The concept of absolute utility will lead to better per-span utilization. In the computation of LT utility, we neglect the loss of LT bandwidth due to presence of guard-bands [15]. LT utility leads to the birth-death property: we set an upper and lower bound on the utility associated with a LT called UB_{Thresh} and LB_{Thresh}. Setting up UB_{Thresh} and LB_{Thresh} presents a trade-off: greater the range of UB_{Thresh}–LB_{Thresh}, higher the probability of packets being dropped (due to lack of a cushion in the buffer at the trailponder), and greater the LT utilization; conversely, lesser the range of UB_{Thresh}–LB_{Thresh}, lesser the LT utilization, but lower the probability of packets being lost. The range [LB_{Thresh} ,UB_{Thresh}] is called the Operating Range (*OR*).

2. Validation property: The validation property is based on the notion of utility – a LT is "valid" while its utility is within the *OR*. Once the net utility of a LT falls below

LB_{Thresh}, we destroy the LT. Conversely, when the utility increases above UB_{Thresh}, the algorithm assigns some of the flows from this LT to a new LT, reducing the utility (of the original LT) to within the *OR*. Validity also involves conformance to physical-layer parameters like Optical-Signal-to-Noise Ratio (OSNR) and wavelength continuity. When a LT is created, for duration of *TTL* we allow the LT to survive without checking for validity. During *TTL*, the LT is expected to attract requests thereby supporting the optical grooming property of LTs. If however, it cannot attract enough requests to meet the *OR* requirements, then the LT is destroyed and requests are to be reassigned.

3. *Rationality property*: The growth algorithm takes a *particular* rational decision so as to maximize net utility towards the network. The justification of associating a flow with *a* particular LT and creating *a* particular LT is dependent on the property of rationality. Rationality determines "*which*" new LT must be created, while birth-death property implies "*when*" the LT is to be created.

The virtual topology growth algorithm, uses the above 3 properties to minimize the network-wide number of LTs or conversely facilitates maximal per-LT utilization.



Light-trail Creation Module (E_2)
Traffic Reporting Module (E_1)
Generic Scheduling Module (E_3)
NWLR/ILFR Module (E_4)
Recourse Module (E_6)
Source/Destination Initiated Dimensioning Module (E_5)
Light-trail Destruction Module (E_7)

Fig. 3. State Diagram of the topology growth algorithm.

The utility based topology growth algorithm is now presented. Shown in Fig. 3 is a 7-state diagram that describes the algorithm. Specifically we can view Fig. 3 from the left to the right – thus describing how an arriving request at module E_1 is treated, or describing the virtual topology reconfiguration (modules E_3, E_4, E_6 and E_7). The algorithm works as follows: A request arrives and is assigned to an existing LT, or to a LT that is dimensioned to accommodate this request, or if both are not possible, then a new LT is created. We facilitate interaction between LTs by allowing flows (requests) to move from one LT to another or a group of requests currently assigned to various LTs to form a new LT. These procedures of flow interaction called – *Inter LT Flow Reconfiguration (ILFR) and Network Wide LT Reconfiguration (NWLR)* – occur in concurrence to creation, deletion and dimensioning modules.

A. Traffic Reporting module (E_1)

This module resides in the NMS and gets traffic requests from nodes in the network. An existing LT is selected to facilitate this request if the mapping of the request to the LT conforms to the validation property. Validation of mapping a connection request to a LT includes: (1) Satisfying graph sufficiency – implying whether the source and destination nodes are members of the same LT. (2) The direction of the flow in the LT is from the source to the destination node. (3) Satisfying the condition that with the added flow, the LT utility is within the *OR*. (4) If there are multiple LTs, then the choice of a particular LT for facilitating the flow is determined, based on the rationality property. The case where a request f_{ij} is assigned to a LT k is defined by: $f_{ij}: N_i, N_j \in k; \overline{U}(k) \in OR; k \in K(t); \left\| N_i \oplus N_j \right\|_k = 1$ where, $K(t)$ denotes the set of LTs at time t and $\left\| N_i \oplus N_j \right\|_k = 1$ implies that node N_i is upstream of N_j in LT k.

B. Dimensioning Module (E_5)

If no LT *exists* that can accommodate f_{ij}, the algorithm *searches* for an existing LT that can be *dimensioned* or modified to accommodate f_{ij}. Dimensioning implies growing a LT and happens in one of the following two ways: (1) Source Initiated Dimensioning or (2) Destination Initiated Dimensioning; depending on whether the source or destination of the request is in the LT (that is to be dimensioned). If an arriving request is such that the source is within an existing LT and the destination is not in the same LT; the LT is dimensioned to include the destination and this is called *Source Initiated Dimensioning* (SID). If however, the destination is part of an existing LT, and the LT is dimensioned to include the source node then this is called as *Destination Initiated Dimensioning* (DID).

In case of SID, a request f_{ij} at N_i in LT k desires to send data to N_j which is not an element of LT k, but is further downstream of LT k. The node then sends a request to the LT controller to dimension LT k to include N_i. The request is approved by the NMS based on satisfaction of the validation and the rationality properties. In this case, validation implies that N_j is downstream of k and segment $S_{end(k)N_j}$ is available on the same wavelength as k (wavelength continuity), where $end(k)$ is the end node of LT k. Rationality implies that the added node N_j enhances its absolute utility.

If, $f_{ij}: N_i \in k, N_j \notin k$ implying a request desires to be provisioned from N_i to N_j and N_j is not in LT k, then the condition for dimensioning is:

$$if\ LB_{Thresh} < \overline{U}(k') < UB_{Thresh}\ and\ w_k : \overline{\overline{path_{end(k)\sim N_j}}} = 1$$

$$then\ create\ k'{:}end(k') = N_j,\ conv(k') = conv(k)\ and\ \lambda_{k'} = \lambda_k \quad (3)$$

where, $w_k : \overline{\overline{path_{N_i \sim conv(k)}}} = 1$ represents availability of a wavelength λ_k on the path from N_i to $conv(k)$. The *if* condition tests (1) whether the newly created (dimensioned) LT k' conforms to the validity property – the total flow is within *OR* and (2) whether the wavelength on which LT k was set up is available from the end-node of k to N_j (which will be the new end-node of the dimensioned LT k'.

In case of DID, a node N_i desiring to establish a flow to N_j finds an existing LT k, such that $N_j \in k$ and the path from N_i to $conv(k)$ is available on the same wavelength as that of LT k, thereby dimensioning k to k' to include N_i . To do so, N_i sends a request (dimensioning) packet to N_j, through the control channel. The controller upon conforming to properties of rationality and validity (through the NMS) approves the dimensioning of k to k' to include N_i. The new LT k' has N_i as its convener. DID is thus defined as:

$$\text{if } f_{ij}: N_i \notin k, N_i \in k, \forall k \in K(t) \text{ and } w_k: \overline{\overline{path_{N_i \sim conv(k)}}} = 1$$
$$\text{then if } \overline{U}(k') < OR$$

then dimension k to produce k':$conv(k')=N_i$, $end(k')=end(k)$, $\lambda_{k'}=\lambda_k$. (4)

C. Light-trail Creation Module (E_2)

For an arriving request (f_{ij}) if no existing LT or a dimensioned LT is available, then we create a new LT. This module facilitates creation of a *new* LT whose convener and end nodes are the source and the destination nodes of the traffic request. The newly created LT is subject to validity and rationality properties with some exception. Validation here implies the availability of a free wavelength between the request's source–destination pair. The created LT need not conform to the *OR* requirement for a duration of *TTL*. After *TTL* seconds, the LT should have its utility within *OR*. The condition for creation of a new LT is given by:

$\exists f_{ij}$: f_{ij} cannot be added to $k \in K(t)$,
& f_{ij} cannot be added to k by SID,
& f_{ij} cannot be added to k by DID.

The above condition states that neither any existing LT nor an existing LT on dimensioning can accommodate f_{ij}. We then create a new LT k:

$$k: conv(k) = N_i, end(k) = N_j, w_k: \overline{\overline{path_{N_i \sim conv(k)}}} = 1$$

Once a LT is setup, nodes provision requests using module E_3. When a request is assigned to a LT, two modules are in parallel invoked: (1) intra-LT scheduling module (E_3), (2) Opportunistic growth (flow-reconfiguration and LT-reconfiguration) module (E_4).

Module E_3 performs intra-LT scheduling within existing light-trails and monitors recourse – buffer time-out or delay time-out. A node/flow referred to the recourse module implies a need for a change between the present association of the node and the LT.

The opportunistic module E_4 concurrently functions with the intra-LT scheduling module. The earlier modules of creation/deletion and dimensioning (LT-modification) are good aids in topology growth, especially if traffic increases or decreases but these cannot handle situations when the network needs to be reconfigured in its existing form. E_4 consists of two sub-modules – *ILFR* and *NWLR*. The ILFR module enables flows to change their associations with LTs; while, the NWLR module facilitates creation of new LTs from the reorganization of an existing set of flows. In case of both ILFR and NWLR, the movement of flows and creation of new LTs happens considering birth-death, validity and rationality, leading to utility enhancement at all times.

D. ILFR Module (E_4)

A request f_{ij} currently provisioned through LT k forms a new association with LT k' if: (1) its association with k' results in greater benefit (to f_{ij}) than its association with k, and (2) there is network-wide utility enhancement (rationality) as a result of this association. ILFR is invoked, if the flow can be provisioned in LT k' subject to conditions of validity. Rationality for successful use of ILFR implies the condition that the *perceived success* as a result of association with another LT is greater than the success achieved in the present LT. Perceived success implies the probability of establishing a successful connection in the new LT (k'). To compute perceived success, we define for a request f_{ij} a value $pbid_{ijk}(t)$ as the *partial* utility that node N_i would advertise to the arbiter of LT k assuming f_{ij} was the *only* flow provisioned at N_i. We further define $apbid_{k'}(t)$ as the average partial utility of LT k' at time t. The average partial utility is computed by considering each flow provisioned in LT k' as a separate entity (hence partial utility) over which the average is computed. The two conditions for ILFR to be invoked are:

(i) $pbid_{ijk}(t) > apbid_{k'}(t)$

(ii) $\overline{U}(k' \cup f_{ij}) + \overline{U}(k - f_{ij}) > \overline{U}(k') + \overline{U}(k)$

Explanation: Condition (i) states that the partial utility $pbid_{ijk}(t)$ should be less than the average partial utility in k' *viz.*, $apbid_{k'}(t)$. This means that the flow that wants to end its association with k is actually contributing more than the average utility that k obtains from each of its current (flow) associations. Condition (ii) states that network-wide absolute utility increases due to ILFR when f_{ij} moves from k to k'. Condition (ii) is satisfied if the probability of success of N_i in k is less than that in k' after f_{ij} associates with k'.

Validation for ILFR implies that the ingress LT k and the egress LT k', both graphically support f_{ij}, *i.e.* $N_i, N_j \in k, k'$ and the direction of communication in k' is from N_i to N_j. The conditions of validity are defined as:

1. $\text{if } N_i, N_j \in k' \text{ and } \|N_i \oplus N_j\|_k = 1$
2. $\text{if } LB_{Thresh} < \overline{U}(k' \cup f_{ij}) < UB_{Thresh}$.

E. NWLR Module (E_4)

We define NWLR as the grouping and movement of requests in existing LTs to form a new LT, leading to increase in utility towards the network.

Method for NWLR: NWLR is described through the following functional rules.

1. At a given time, there is only one instance of NWLR, *i.e.* only one LT (due to NWLR) is created in the network.
2. LTs can either be in *metastable* or *unstable* states. LT k is metastable if its absolute utility is greater than $LB_{Thresh} + |OR|/2$, implying k is a candidate for NWLR. LT k is otherwise unstable if the time elapsed since it has been created (defined as ($t_{live}(k)$)) is less than *TTL* or utility is $< LB_{Thresh} + |OR|/2$.
3. Amongst all the LTs in the metastable state, we find a node in any LT k called the *originator* that initiates NWLR. A node is chosen as the *originator* if it satisfies the following condition:

$$orig(t) = N_i : i = \arg\max_{\substack{\forall j: N_j \in k, k \in K(t), \\ k \text{ is metastable}}} \left\{ \frac{B_{jk}(t)/CT_S}{P_{succ}^{jk}(t)} \right\} \qquad (5)$$

The above condition denotes that N_i is an element of k such that its probability of success as compared to its time-slot utilization is the least amongst that of all the nodes in LT *k*.

4. Based on *orig*(*t*) and its corresponding LT *k*, we compute a set called *prey*(*t*) that denotes the set of LTs with which *k* has at least one overlapping segment or at least one adjacent segment. Segment S_{ij} is common to two LTs if N_i is adjacent to N_j on these two LTs.
   (A) find $prey(t)$:$k' \in prey(t)$, $|S_{ij}|=1$
      a) S_{ij}: $S_{ij} \in k$, $S_{ij} \in k'$ or
      b) $N_i=end(k)$, $N_i=conv(k')$ and $N_j \in k'$ or
      $N_i=end(k')$, $N_i=conv(k)$ and $N_j \in k$ or
      $N_j=end(k)$, $N_j=conv(k')$ and $N_j \in k$ or
      $N_j=end(k')$, $N_j=conv(k)$ and $N_j \in k'$
   (B) $\overline{U}(k') > LB_{Thresh} + |OR|/2$

   Above in (A) we compute the set of LTs that are possible prey (for NWLR). The conditions that a LT *k'* would be a prey LT so that an originator (in *k*) can defect is denoted by A(a) or A(b). If there exists a LT *k'*, such that *k'* and *k* have at least one common/adjacent segment, then the flows assigned to *k'* are candidates for a new LT. Condition (B) states that the LT *k'* has an absolute utility that is at least $LB_{Thresh} + |OR|/2$.

5. From *prey*(*t*), we populate $prey_{flow}(t)=\{f_{ij}:f_{ij} \in k, k \in prey(t)\}$. A subset of $prey_{flow}(t)$ is selected to create LT *k''*, such that *k''* contributes towards maximizing the utility towards the network over all such combination of flows from $prey_{flow}(t)$ and:
   $$\overline{U}(k'') + \sum_{k' \in prey(t), f_{ij} \in k', f_{ij} \in prey_{flow}(t)} \overline{U}(k' - f_{ij}) > \sum_{k' \in prey(t)} \overline{U}(k')$$

The detailed working of NWLR is described in Section IV.

F. *Recourse module* (*E*₆)

This module works at the node level and at the LT level. At the node level, recourse implies a request being timed-out or the buffer (at the node) is nearing to overflow. At the LT level, recourse implies the condition when the absolute utility (of the LT) either falls below LB_{Thresh} or exceeds UB_{Thresh} (assuming that the LT is metastable, *i.e.* it has lived for $t_{live}(k)>TTL$). The recourse conditions are:

(1) $f_{ij} \in k$ and $\overline{U}(k) < LB_{Thresh}$ and $t_{live} > TTL$

(2) $f_{ij} \in k$ and $\overline{U}(k) > UB_{Thresh}$

The LT level recourse conditions are:

(3) $t_{HP} < \psi_{ik}(t) < t_{HP}+T_S$

(4) $\left(1 - \frac{B_{\max} \times (t_{HP}+T_S)}{C}\right) < \frac{B_{ik}(t)}{B_{\max}} < \left(1 - \frac{B_{\max} \times t_{HP}}{C}\right)$

The first condition states that the net utility of a LT is less than the LB_{Thresh}, and that the LT has existed for some time greater than *TTL*. In the second case, the utility provided by LT *k* exceeds the UB_{Thresh}.

In the third case, a request f_{ij} reaches criticality based on limits of t_{HP} and $t_{HP}+T_S$, implying that this request must be reassigned to some other LT, else it will be timed out. Where t_{HP} is defined as the time required to provision a new LT. In the fourth case, a request f_{ij} reaches criticality, such that the buffer supporting the request has timed-out.

G. *Destruction Module* (*E*₇)

The recourse module is supported by the LT destruction module that destroys LTs when:

1. A LT *k* is dimensioned into another LT *k'*, thereby destroying *k*, or,
2. When *k* reaches recourse condition, specifically, such that the total flow is lesser than the LB_{Thresh}.

IV. Stochastic Analysis Of The Scheduling Algorithm

The scheduling algorithm is stochastically analyzed resulting in computation of performance metrics (such as probability of success) that are instructive in calculation of average delay and number of LTs. As a node wins a transmission slot, we define the interval between two consecutive wins as a *scheduling cycle* for that node. We assume that if a node has won a slot then at the end of this slot its buffer is empty, *except* for the requests that arrived within this slot. Hence, every scheduling cycle starts with a zero buffer and grows thereafter. This behavior is similar across every cycle, although separated in time.

We analyze the algorithm by first computing the probability of success for a node within a LT. This process is complex given the multiple dependencies of the associated R.Vs. Hence, this is divided as: (1) the computation at the startup of the system and (2) a generic state. For the generic state, a Markov model is developed whose steady state probabilities are of interest to us, leading to the computation of the success probabilities at a node. Subsequently, we consider the stochastic behavior of the growth algorithm, using the probability of success at a node as an enabler. The analysis is verified through simulation.

A. *Analysis of the first scheduling cycle*

Referring to the system model in Section II, we define, $V_i(t_o)=\{j \mid H_{ji}+\delta_{ji} \geq t_o\}$ comprising of all valid requests at node N_i at time t_o. Hence,

$$\sigma_i(t_o)=\max_{j \in V_i(t_o)}\{x_{ji}\} \text{ and } \psi_i(t_o)=\min_{j \in V_i(t_o)}\{\delta_{ji}-x_{ji}\} \qquad (6)$$

where, $x_{ji}=t_o-H_{ji}$. Hence, the utility of N_i for time-slot t_o reduces to,

$$bid_i(t_o)=\max\left\{\frac{B_i(t_o)/|S|}{B_{\max}}, \frac{\max_{j \in V_i(t_o)}\{x_{ji}\}}{\max_{j \in V_i(t_o)}\{x_{ji}\}+\min_{j \in V_i(t_o)}\{\delta_{ji}-x_{ji}\}}\right\}$$
$$=\max\left\{\frac{B_i(t_o)/|S|}{B_{\max}}, \frac{t_o-\min_{j \in V_i(t_o)}\{H_{ji}\}}{\min_{j \in V_i(t_o)}\{H_{ji}+\Delta_i\}-\min_{j \in V_i(t_o)}\{H_{ji}\}}\right\}$$

where, B_{max} is the buffer size. Amongst all the valid requests at N_i at time t_o, we define the earliest arrival time across all the services as $A_i(t)=\min_{j \in V_i(t)}\{H_{ji}\}$. In addition, the earliest instant that a packet is timed-out is $D_i(t)=\min_{j \in V_i(t)}\{H_{ji}+\delta_{ji}\}$. We require the joint distribution function of $\sigma_i(t_o)$, $\psi_i(t_o)$ and $B_i(t)$ to compute success probability.

Using (6) and rearranging, we get:

$$P\left(\sigma_i(t_o)<\alpha, \psi_i(t_o)<\beta, B_i(t_o)<\gamma \mid V_i(t_o) \neq \phi\right)$$

$$P\left(\begin{array}{c} \min_{j \in V_i(t_o)}\{H_{ji}\} > t_o-\alpha, \\ \min_{j \in V_i(t_o)}\{H_{ji}+\delta_{ji}\} < t_o+\beta, B_i(t_o)<\gamma \end{array} \middle| V_i(t_o) \neq \phi\right)$$

$$= P\left(A_i(t_o) > t_o - \alpha, B_i(t_o) < \gamma \middle| V_i(t_o) \neq \phi\right)$$
$$-P\left(A_i(t_o) > t_o - \alpha, D_i(t_o) > t_o + \beta, B_i(t_o) < \gamma \middle| V_i(t_o) \neq \phi\right) \quad (7)$$

First consider the second term of (7). From the definition of $A_i(t)$ and $D_i(t)$, we state that,

$$P\left(A_i(t_o) > t_o - \alpha, D_i(t_o) > t_o + \beta, B_i(t_o) < \gamma \middle| V_i(t_o) \neq \phi\right)$$
$$= P\left(\begin{matrix} H_{ji} > \max\left\{t_o - \alpha, t_o + \beta - \delta_{ji}, t_o - \delta_{ji}\right\}, \\ B_i(t_o) < \gamma \end{matrix} \middle| V_i(t_o) \neq \phi\right) \quad (8)$$

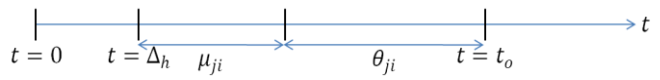


Fig. 4. Illustrating definition of μ_{ji} and θ_{ji}.

The arrival time (H_{ji}) of the jth service request at N_i can vary in the interval $[t_o – \Delta_h, t_o]$. We subdivide this interval into two non-overlapping sub-intervals, based on (8), corresponding to the jth service request arrival at N_i that belongs to S_h as follows, and which is illustrated in Fig. 4.

$$\mu_{ji} = \left[\max\left\{0, t_o - \Delta_h\right\}, \min\left\{t_o, \max\left\{\begin{matrix} t_o + \beta - \delta_{ji}, \\ t_o - \alpha, t_o - \delta_{ji} \end{matrix}\right\}\right\}\right]$$
$$\theta_{ji} = \left[\min\left\{t_o, \max\left\{t_o + \beta - \delta_{ji}, t_o - \alpha, t_o - \delta_{ji}\right\}\right\}, t_o\right]$$

Note that for all values of α and β, μ_{ji} and θ_{ji} are always valid intervals. Let us also consider the following events at N_i.

$G_i(t_o)$ = No request $\in V_i(t_o)$ arrived at N_i in μ_{ji}.
$L_i(t_o)$ = No request $\in V_i(t_o)$ arrived at N_i in θ_{ji}.

The jth service request at N_i will not time-out till t_o, if it arrives in either one of these intervals: μ_{ji} or θ_{ji}. Hence, any request that arrived in either of these intervals *will* belong to $V_i(t_o)$. Thus, G_i and L_i simplifies to:

$G_i(t_o)$ = No request arrived at N_i in μ_{ji}.
$L_i(t_o)$ = No request arrived at N_i in θ_{ji}.

We define the event $M_i(t_o) \triangleq \{B_i(t_o) < \gamma\}$. Hence,

$$G_i(t_o) \cap \bar{L}_i(t_o) \cap M_i(t_o)$$
$$= \left\{A_i(t_o) > t_o - \alpha, D_i(t_o) > t_o + \beta, B_i(t_o) < \gamma, V_i(t_o) \neq \phi\right\}$$

Also $P\left(G_i(t_o) \cap \bar{L}_i(t_o) \cap M_i(t_o)\right)$ = $P(G_i(t_o) \cap M_i(t_o))$ – $P(G_i(t_o) \cap L_i(t_o) \cap M_i(t_o))$. We now solve for the individual terms of this expression as follows.

$P(G_i(t_o) \cap M_i(t_o))$
$=P($*No request arrived at* N_i *in* $\mu_{ji}) \cap (B_i(t_o) < \gamma)$

$$= P\left(\bigcup_{n=0}^{\lfloor\gamma\rfloor} \bigcup_{Z_n} \bigcap_{h=1}^{|S|} B_{hi}(t_o) = n_h \textit{ with arrivals limited to } \theta_{ji}\right)$$

where, $Z_n = \left\{\left(n_1, n_2, \ldots, n_{|S|}\right) \middle| \sum_{h=1}^{|S|} n_h = n\right\}$

$$= \sum_{n=0}^{\lfloor\gamma\rfloor} \sum_{Z_n} \prod_{h=1}^{|S|} \left(\frac{e^{-\lambda_{hi}\left(|\mu_{ji}| + |\theta ji|\right)} \left(\lambda_{hi} |\theta ji|\right)^{n_h}}{n_h!}\right) \quad (9)$$

where, |.| denotes the cardinality operator, *i.e.* $|\mu_{hi}| + |\theta_{hi}|$ = $\min\{t_o, \Delta_h\}$ and $|\theta_{ji}|$ = $\max\{0, \min\{\alpha, \delta_{ji} – \beta, \delta_{ji}\}\}$. Further, $P(G_i(t_o) \cap L_i(t_o) \cap M_i(t_o))$

$$= \prod_{j=1}^{h} e^{-\lambda_{hi} \times \min\{t_o, \Delta_h\}} = \exp\left(-\sum_{h=1}^{|S|} \lambda_{hi} \times \min\left\{t_o, \Delta_h\right\}\right) \quad (10)$$

Using (9) and (10), we solve for the second term in (7):

$$P\left(A_i(t_o) > t_o - \alpha, D_i(t_o) > t_o + \beta, B_i(t_o) < \gamma \middle| V_i(t_o) \neq \phi\right)$$

$$= 1 - \frac{1 - \sum_{n=0}^{\lfloor\gamma\rfloor} \sum_{Z_n} \prod_{h=1}^{|S|} \left(e^{-\lambda_{hi}\left(|\mu_{ji}| + |\theta ji|\right)} \left(\lambda_{hi} |\theta ji|\right)^{n_h} / n_h!\right)}{1 - \exp\left(-\sum_{h=1}^{|S|} \lambda_{hi} \times \min\{t_o, \Delta_h\}\right)} \quad (11)$$

Similarly, we solve for the first term in (2) as follows:

$$P\left(A_i(t_o) > t_o - \alpha, B_i(t_o) < \gamma \middle| V_i(t_o) \neq \phi\right)$$

$$= 1 - \frac{1 - \sum_{n=0}^{\lfloor\gamma\rfloor} \sum_{Z_n} \prod_{h=1}^{|S|} \left(e^{-\lambda_{hi}\left(|\mu'_{ji}| + |\theta'_{ji}|\right)} \left(\lambda_{hi} |\theta'_{ji}|\right)^{n_h} / n_h!\right)}{1 - \exp\left(-\sum_{h=1}^{|S|} \lambda_{hi} \times \min\{t_o, \Delta_h\}\right)} \quad (12)$$

where, $\mu'_{ji} = \left[\max\left\{0, t_o - \Delta_h\right\}, \min\left\{t_o, \max\left\{t_o - \alpha, t_o - \delta_{ji}\right\}\right\}\right]$, $\theta'_{ji} = \left[\min\left\{t_o, \max\left\{t_o - \alpha, t_o - \delta_{ji}\right\}\right\}, t_o\right]$ and thus $|\mu'_{ji}| + |\theta'_{ji}| = \min\left\{t_o, \Delta_h\right\}$, $|\theta'_{ji}| = \max\left\{0, \min\left\{\alpha, \delta_{ji}\right\}\right\}$. Further, substituting (11)-(12) in (7), we get,

$$F_{\sigma,\psi,B_i}(\alpha, \beta, \gamma)$$
$$\triangleq P\left(\sigma_i(t_o) < \alpha, \psi_i(t_o) < \beta, B_i(t_o) < \gamma \middle| V_i(t_o) \neq \phi\right)$$
$$= \sum_{n=0}^{\lfloor\gamma\rfloor} \sum_{Z_n} \prod_{h=1}^{|S|} \frac{e^{-\lambda_{hi}\left(\min\{t_o, \Delta_h\}\right)} \times \left(\lambda_{hi} \times \left(|\theta'_{ji}| - |\theta_{ji}|\right)\right)^{n_h}}{n_h! \times \left(1 - e^{-\sum_{h=1}^{|S|} \lambda_{hi} \times \min\{t_o, \Delta_h\}}\right)} \quad (13)$$

Probability Distribution Function (pdf) of utility of a node: Using (8), we compute the *pdf* of the utility of N_i as,

$$F_{bid_i(t_o)}(a) = P\left(bid_i(t_o) < a \middle| V_i(t_o) \neq \phi\right)$$
$$= \int_0^\infty \int_0^{\frac{(1-a)x}{a}} \frac{\delta^2}{\delta x \delta y} \left(F_{\sigma,\psi,B_i}\left(x, y, ahB_{\max}\right)\right) dydx \quad (14)$$

B. Probability of success in the first cycle

We denote the probability of node N_i winning the next time-slot at time t_o for transmission amongst all the competing nodes in the LT, as $P^i_{succ}(t_o)$, *i.e.* probability of success of N_i at time t_o. For each node N_i, we define the maximum utility amongst all other nodes as $O_i(t_o) = \max_{i'=1,\ldots,N, i' \neq i}\left\{bid_{i'}(t_o)\right\}$. Given that the utilities of different nodes over the network are independent, the distribution function of $O_i(t_o)$ is:

$$F_{O_i(t_0)}(y) = \prod_{i'=1, i' \neq i}^{N} F_{bid_{i'}(t_o)}(y)$$

The probability of success of N_i at time t_o is given by:

$$P^i_{succ}(t_o) = P\left(bid_i(t_o) \geq O_i(t_o)\right) = \int_0^1 \int_0^x f_{bid_i(t_o), O_i(t_o)}(x, y)\, dydx \quad (15)$$

The distributions of $bid_i(t_o)$ and $O_i(t_o)$ are independent since the computation of utility at a node are affected only by factors *local* to the node like arrival rates of services, buffer occupancies, delay stringencies at the node and are not affected by other nodes over the network. Hence, $f_{bid_i(t_o), O_i(t_o)}(x, y) = f_{bid_i(t_o)}(x) \times f_{O_i(t_o)}(y)$. We obtain:

$$P^i_{succ}(t_o) = \int_0^1 f_{bid_i(t_o)}(x) \times F_{O_i(t_o)}(x)\, dx \quad (16)$$

C. Generic Scheduling Cycle

We now extend the above analysis to include the generic scheduling cycle. We assume that if a node has won the slot l, then at the end of this slot, its buffer is emptied *except* for the packets that arrive during transmission during l. This assumption allows us to say that if all slots have the same duration, (unit time) then we need the *pdf* of the utility at t=1,2,3,…,l,… , given that the changes are only at the slot boundaries. This assumption introduces a theoretical anomaly (for accounting the packets during the slot *l-1*), but through simulations we have observed that this anomaly is negligible and can be ignored.

For a particular node N_i define ζ_i as the time interval from an arbitrary point in the past to the present time-slot (t_o) with the following properties.

1. At time(t_o–ζ_i), the buffer at N_i was empty.
2. In [t_o–ζ_i, t_o], N_i has not won any time slots.

Then, using (14), the cumulative distribution function (*cdf*) of the utility of N_i in its first cycle will be

$$F^I_{bid_i(t)}(x) == \int_0^\infty \int_0^{\frac{(1-a)x}{a}} \frac{\delta^2}{\delta x \delta y}\left(F_{\sigma,\psi,B_i}\left(x,y,ahB_{\max}\right)\right)dydx \tag{17}$$

Due to the assumption of empty buffers at the end of a successful slot l, all the requests in the buffer at the present instant arrive only during and/or after the slot l. We treat the buffer at N_i as being empty at the start of slot l. Hence,
P(bid_i(t_o)<a | *Slot w was the last slot won by* N_i)

$$= \begin{cases} F^I_{bid_i(t_o)}(a), & if\ \ w=0 \\ F^I_{bid_i(1)}(a), & if\ \ w=t_o \\ F^I_{bid_i(t_o-w+1)}, & \forall w: 1 \le w \le t_o, if\ \ w \ne t_o \end{cases} \tag{18}$$

The steady state of the system is studied using a Markov model with the assumption that each node has won a slot at least once within a sufficiently *long* time duration. The state of the system at the junction between two successive time-slots is represented by the N-tuple (κ_1,κ_2,…,κ_N), where κ_i represents the time elapsed since the end of the last slot won by N_i. Since the utility for transmission in slot l are advertised at the beginning of slot (l – 1), the present state of the system is dependent on the past *two* states (present and previous slots).

Consider the utility of a slot starting at (l+1) *i.e.* (l+2) th slot. If the time interval κ_i and winning node of the (l+1)th slot are known, then (14) gives the distribution of the utilities at each node. Using (16), we state the probability with which N_i will win the (l+2)th slot as:

$$P\left(N_i \text{ wins the } (l+2)^{th} \text{ slot} \,\middle|\, \begin{matrix} \text{State at } l \text{ is } (\kappa_1,\kappa_2,\dots,\kappa_N) \\ \text{State at } (l+1) \text{ is } (\kappa_1',\kappa_2',\dots,\kappa_N') \\ N_{i'} \text{ won the } (l+1)^{th} \text{ slot} \end{matrix} \right)$$

$$= \int_0^1 f^I_{bid_i(\kappa_i+1)}(x) \prod_{i'=1,i'\ne i}^N F^I_{bid_{i'}(\kappa_{i'}+1)}(x)dx$$

$$= P\left\{ \begin{matrix} \text{State at } (l+1) \text{ is } (\kappa_1',\kappa_2',\dots,\kappa_N') \\ \& \ N_i \text{ wins the } (l+2)^{th} \text{ slot} \end{matrix} \,\middle|\, \begin{matrix} \text{State at } l \text{ is } (\kappa_1,\kappa_2,\dots,\kappa_N) \\ \& \ N_{i*} \text{ wins the } (l+1)^{th} \text{ slot} \end{matrix} \right\} \tag{19}$$

This gives a hint to the states of the Markov chain. Theoretically, $\kappa_i \in [0,\infty)$. However, we do not need an infinite number of states, since $F^I_{bid_i(t)}(x)$ is independent of t for $t > \max_{1\le h \le |S|}\{\Delta_h\}$. This means that for all κ_l>M=$\max_{1\le h\le |S|}\{\Delta_h\}$, κ_l=min{κ_l, M}.

D. Markov Chain Formulation

The Markov chain is defined in the finite state space Ʊ. We define the states in Ʊ, as a 2-tuple consisting of the winner of the current time-slot and information about the duration passed since each node won a slot in the past. A state in this chain is thus an ordered pair {v, a}, where v=(κ_1,κ_2,…,κ_N) and a∈{1,2,…,N} and has the following properties.

1. ∃! *i'∈{1,2,…,N}*, $\kappa_{i'}$*=0*, *i.e.* exactly one of the components is 0.
2. ∀*i∈{1,2,…,N}*, *i≠i'*, κ_i*∈{1,2,…,M}*, *i.e.* all other components are integers.

As N_i *wins a particular time-slot, in the subsequent slot its* κ_i*=0, and for all other nodes, their respective value of* $\kappa_{i'}$ *gets incremented by 1.*

This observation leads to the transformation $\Im$: Ʊ→Ʊ as $\Im(\{\kappa_1,\kappa_2,\dots,\kappa_N\},a) = \{\{\kappa_1',\kappa_2',\dots,\kappa_N'\},a'\}$, where a∈{1,2,…,N} and,

$$\kappa_i' = \begin{cases} 0, \ if\ \ i=a \\ \min\{\kappa_i+1, M\}, \ \ otherwise \end{cases} \tag{20}$$

One-step (one-time-slot) transition probabilities from {v_1,a_1} to {v_2,a_2} are now defined. For the two states that are not connected by the transformation $\Im$, the transition probability is 0, *i.e.* P({v_1,a_1}→{v_2,a_2})=0, whereas, for connected states due to the transformation, the transition probabilities are:

$$P\left(\{v_1,a_1\} \xrightarrow{\Im} \{v_2,a_2\}\right) = \int_0^1 f^I_{bid_i(\kappa_i+1)}(x) \prod_{i'=1,i'\ne i}^N F^I_{bid_{i'}(\kappa_{i'}+1)}(x)dx \tag{21}$$

E. Steady state analysis of the Markov Chain

Consider the state q≡{{0,M,M,…,M},1}, which denotes the state reached after M consecutive wins of N_1. State q, is reachable from all the states in Ʊ, and is thus recurrent. Starting from state q, there is a non-zero probability that the system will return to the same state in one time-step (or time-slot). Hence, q is aperiodic. Let $\wp$ denote the set of all states reachable from q. Since q is recurrent, $\wp$ will be an irreducible subset of Ʊ. Also, since one state in $\wp$ is recurrent and aperiodic, all its states are recurrent and aperiodic [19]. Further, because $\wp$ is finite all its states are recurrent non-null aperiodic [20]. To study the state of the system in the limit as t→∞, we only need to consider the set of all recurrent states which we claim is $\wp$.

Claim 1: A state in this system (as described above) is recurrent if and only if it belongs to $\wp$.

Proof: For any state $a \in \bar{\wp}$ (complement of $\wp$), there is a non-zero probability that it will reach q, *i.e.* $p_{a,q}$>0. From q, the chances of returning to a are 0 as $a \notin \wp$. Hence, the probability of eventually returning to a, from a is not 1, *i.e.* $p_{a,a} \ne 1$. Hence, a is not recurrent. ■

In the transition matrix we consider states only from $\wp$ and evaluate its steady-state probabilities. All the states in $\bar{\wp}$ are transient and thus can be ignored for a steady state analysis. Note that $\wp$ is recurrent, non-null and aperiodic; and every state in $\wp$ is mutually reachable from every other state via q. Hence, the stationary state probabilities are the limiting probabilities and they are independent of the initial state [20]. Below we state an explicit characterization of the states in $\wp$.

Claim 2: Any state in $\wp$ is of the form {v=(κ_1,κ_2,…,κ_N),a} where a∈{1,2,…,N} and *v's* properties are:

1. ∃! i∈{1,2,…,N}, κ_i=0.

2. $\forall v=1,2,\ldots,M-1$, $|\{i \mid i\in\{1,\ldots,N\},\kappa_i=v\}| \le 1$.
3. All other components of υ must be exactly M.

Proof: $\wp$ is the set all states that are reachable from q by the transformation $\Im$. Since q has only one element that is 0, any state reached from q by the above transformation will have at most a single element whose value is 1. It then follows that all states in $\wp$ have at most one element with value 1. As $\wp$ is closed under $\Im$ and all states in $\wp$ have at most a single 1, any state in $\wp$ can have at most one value of 2 and this process carries on up to (M–1). ■

Claim 3: Any state of the form defined above in Claim 2 belongs to $\wp$.

Proof: Given a state $f=\{\upsilon=\{\kappa_1,\kappa_2,\ldots,\kappa_N\},a\}$ satisfying properties 1-3 as defined above in Claim 2, we prove by induction that this state is reachable from q by a sequence of transformations $\Im$. We apply induction on the number of components in υ that are neither 0 nor M.

Base Case: Let κ_j be the only component of υ that is not 0 and not M and let $\kappa_i=0$. Starting from q, we can reach f:

$$\begin{aligned} q &\xrightarrow{\Im\left(N_1 \text{ wins the next slot}\right)} \{\{0,M,M,\ldots.,M\},i\} \\ &\xrightarrow{\Im^M\left(N_i \text{ wins the next } M \text{ slots}\right)} \{\{M,M,\ldots,\kappa_i=0,\ldots,M\},j\} \\ &\xrightarrow{\Im\left(N_j \text{ wins the next slot}\right)} \{\{M,M,\ldots,\kappa_j=0,\kappa_i=1,\ldots,M\},i\} \\ &\xrightarrow{\Im^{\kappa_j-1}\left(N_j \text{ wins } (\kappa_j-1) \text{ slots}\right)} \{\{M,M,\ldots,\kappa_j-1,\kappa_i=0,\ldots,M\},i\} \\ &\xrightarrow{\Im\left(N_i \text{ wins the next slot}\right)} \{\{M,M,\ldots,\kappa_j,\kappa_i=0,\ldots,M\},a\} \end{aligned} \tag{22}$$

Here, if $\kappa_j=1$, $\Im^{\kappa_j-1}=\Im^0$ represents the identity transformation.

Induction Hypothesis: Let any state have the above properties, and exactly n of its components not equal to 0 and M be reachable from q.

Induction Step: Let f have exactly (n+1) components that neither 0 nor M. Further, let the index of the smallest and the smallest non-zero component in υ be u and y respectively. Then consider the state $w=\{\upsilon'=\{\kappa_1',\kappa_2',\ldots,\kappa_N'\},u\}$, where the components κ_i''s of υ' are defined as,

$$\kappa_i' = \begin{cases} M, \text{ if } \kappa_i = 0 \text{ or } M. \\ \kappa_i - \kappa_y, \text{ otherwise.} \end{cases} \tag{23}$$

w satisfies properties 1-3 of claim 2, and has exactly n components that are neither 0 nor M. By the induction hypothesis, w is reachable from q. We can now reach f from w as follows:

$$q \rightarrow w \xrightarrow{\Im^{\kappa_y-1}\left(N_u \text{ wins next } (\kappa_y-1) \text{ slots}\right)} \{\tilde{\upsilon},u\} \overset{\Im}{\rightarrow} f \tag{24}$$

Thus, $f\in\wp$. Hence the proof. ■

Let **P** be the transition probability matrix amongst the states in $\wp$. Then by [20], the normalized eigenvector of **P** *w.r.t.* eigenvalue 1 will be unique and its components will be the probability that the system will be found in a particular state $\{\upsilon,a\}$ after a sufficiently large time, *i.e.* $\lim_{t\to\infty}P(\{\upsilon,a\})$. The success probability of node N_i is then given by:

$$P_{succ}^{a} = \lim_{t\to\infty} P\left(N_a \text{ wins the bid}\right) = \sum\nolimits_{\upsilon\in\wp} P\left(\{\upsilon,a\}\right) \tag{25}$$

F. Average Delay

The average delay experienced by N_i to win a time-slot for transmission is given by $\Delta_{avg} = T_S / P_{succ}^{i}$, where P_{succ}^{i} denotes the time ensemble value of $P_{succ}^{i}(t)$.

We now develop a stochastic model whose aim is to compute the probabilities of existence of *a* LT and *n*-LTs using the algorithm described in Section III. A request f_{ij} is assigned to LT k as a result of the following 6 cases:

1. On arrival, request f_{ij} is assigned to an existing LT k.
2. Case 1 does not occur, and f_{ij} is assigned to an existing LT k as a result of SID.
3. Case 1 and Case 2 do not occur and f_{ij} is assigned to k as a result of DID.
4. Case 1-3 do not occur, and LT k was exclusively created to accommodate f_{ij} in the network.
5. f_{ij} is assigned to k as a result of ILFR.
6. f_{ij} is assigned to k as a result of NWLR.

From the above 6 conditions and their definitions in Section III, the probability of request f_{ij} assigned to k is:

$$\begin{aligned} P\left(f_{ij}\in k\right) &= P\begin{pmatrix} f_{ij} : \Delta\bar{U}\left(k\cup f_{ij}\right) > \Delta\bar{U}\left(k'\cup f_{ij}\right), k'\in K(t) \\ \& \; LB_{Thresh} \le \bar{U}\left(k\cup f_{ij}\right) < UB_{Thresh} \; and \; N_i, N_j \in k \end{pmatrix} \\ &+ P\left(f_{ij} : \xrightarrow{SID} k\right) + P\left(f_{ij} : \xrightarrow{DID} k\right) + P\left(f_{ij} : \xrightarrow{LT \text{ Creation}} k\right) \\ &+ P\left(f_{ij} : \xrightarrow{ILFR} k\right) + P\left(f_{ij} : \xrightarrow{NWLR} k\right) \end{aligned} \tag{26}$$

Note: The above equation assumes $\xrightarrow{X}$ as the functional operator, where X is the function being performed (e.g. NWLR or SID etc.). Also, the operator $\Delta : \Delta\bar{U}\left(k\cup f_{ij}\right) = \bar{U}\left(k\cup f_{ij}\right) - \bar{U}(k)$. We now proceed to compute the probabilities of the terms in (26). First, preliminary results are presented in equations (27)–(34).

We evaluate the probability of the event N_i that is an element of LT k. N_i is any node chosen from any of the N nodes in the network. But for N_i to be in k, it has to be amongst the $n(k)$ nodes. For an N-node network,

$$P\left(N_i \in k\right) = \tfrac{n(k)}{N} = \pi_k \tag{27}$$

Further, probability that both N_i and N_j belong to LT k is equal to the probability that each of them individually belongs to k, and assuming independence, we have:

$$P(N_i\in k, N_j\in k) = P(N_i\in k).P(N_j\in k) = (\pi_k)^2 \tag{28}$$

Given nodes N_i and N_j in LT k, there exists an equal chance of N_i being upstream or downstream of N_j, hence,

$$P(\|N_i\oplus N_j\|_k=1) = P(N_i \text{ is upstream of } N_j \text{ on } k) = 1/2 \tag{29}$$

The request f_{ij} is assumed to be Poisson distributed. Thus,

$$P\left(f_{ij} = x\right) = e^{-\lambda}\lambda^x / x!$$

The *pdf* of utility of a LT k, *i.e.* $\bar{U}(k)$.

$$P\left(\bar{U}(k) = x\right) = P\left(\frac{\sum_{\forall f_{ij}\in k} f_{ij}}{\left(n(k)-1\right)C} = x\right) = \frac{e^{-m\lambda_{ik}^{h}}\left(m\lambda_{ik}^{h}\right)^{xC\left(n(k)-1\right)}}{x!} = H(x,k)$$

where, $m = \left|\left\{f_{ij} : f_{ij} \rightarrow k\right\}\right|$. We now find the *cdf* of $\bar{U}(k)$ as

$$P\left(\bar{U}(k) \le x\right) = \int_0^x P\left(\bar{U}(k) = y\right)dy = \int_0^x H(y,k)\,dy \tag{30}$$

Note: The path between nodes N_i and N_j is free on w_k if N_i and N_j belong to LT k which is established on w_k. Thus:

$$P\left(w_k : \overline{\overline{path_{N_i\sim N_j}}} = 1\right) = P\left(N_i, N_j \in k \text{ and } w_k \mapsto k\right) \tag{31}$$

A LT is established on any available wavelength with equal probability. Hence if w is the number of wavelengths:

$$P\left(w_k \mapsto k\right) = \frac{T_{\max}}{w \times \left(\sum_{f_{ij}: f_{ij} \in k, k \in K(t)} f_{ij}\right)} = M\left(k,t\right) \tag{32}$$

The denominator in (32) denotes load, and using [21], we say that the load is loosely proportional to the number of wavelengths used and hence (32) follows. In the above equation, T_{max} is the maximum traffic the system can accommodate. Therefore, from (31)-(32) we have,

$$P\left(w_k : \overline{\overline{path_{N_i \sim N_j}}} = 1\right) = \left(\pi_k\right)^2 \times M\left(k,t\right) \tag{33}$$

We now compute the probability that a segment S_{ij} is common to LTs k and k'. Amongst all randomly chosen LTs consisting of N_i and N_j, only half of them will consist of requests from N_i to N_j (assuming uniform distribution). Given two identical LTs with similar nodes as members, but on different wavelengths, either of them is equally likely to be setup. There will be $\beta_{ij}/2$ LTs consisting of segment S_{ij}, where β_{ij} represents the total number of LTs existing in the network at the given instant, passing through N_i and N_j. If any two LTs are chosen from this set, they share $S_{ij} : P\left(S_{ij} \in k, S_{ij} \in k'\right) = {}^{\beta_{ij}/2}C_2 / {}^{L}C_2$ where $L=(2w(N-1))$ is the total number of LTs possible.

Consider the probability that there exists a segment S_{ij} such that one of its end nodes is $N_i \in k$, and the other end is $N_j \in k'$. This implies that we have to compute the probability that given k, we find all possible LTs k' that can support S_{ij}. To do so, we assume that N_j is the convener of k', and k' can occupy any of the available wavelengths (32). Further, there are $(N-n(k)-1)$ possible LTs (of varying number of nodes) that can begin from N_j, and hence:

$$P\left(\exists S_{ij} : N_i \in k, N_j \in k'\right) = w\left(N - n\left(k\right) - 1\right) \sum_{f_{ij}: f_{ij} \in k, k \in K(t)} f_{ij} / T_{\max} \tag{34}$$

With these basic relations we now compute the probability of occurrence of each module shown in Fig. 3.

Probability of SID occurring

A LT k will encounter SID if there exists a request f_{ij} that arrived but could not be added to any existing LT (by the NMS). Our objective is to dimension k to include node N_j, creating k' whose end-node is N_j. Hence,

$$P\left(k \overset{SID}{\rightarrow} k'\right) = P\left(\begin{matrix} \exists f_{ij} : \left(f_{ij} \textit{ could not be added to } k \in K\left(t\right)\right) \\ \textit{and } \left(f_{ij} \textit{ was added to k by SID}\right) \end{matrix}\right) \tag{35}$$

Equation (19) has two components, of which the first component is solved as follows:

$P(f_{ij}$ could not be added to any LT $k \in K$(t))

$$= \prod_{\forall k \in K(t)} \left[\tfrac{5}{2} - \left(\pi_k\right)^2 - \sum_{y=0}^{UB_{Thresh}} H\left(y, k \cup f_{ij}\right)\right] \tag{36}$$

Equation (36) follows the definition of SID in Section III.B. That is f_{ij} cannot be added to any existing LT k: (1) if the source and destination nodes of the flow do not simultaneously belong to any LT or, (2) even if they belong, then the source node is downstream of the destination or, (3) if the utility of the LT when combined with the new request f_{ij} exceeds UB_{Thresh}. The 2nd component of (35) is calculated as:

$P(f_{ij}$ *was added to k by* SID)

$$= \sum_{y=0}^{UB_{Thresh}} H\left(y, k \cup f_{ij}\right) \times \pi_k \left(1 - \pi_k\right) \left(\pi_k\right)^2 \times M\left(k,t\right) \tag{37}$$

Hence, from (35)-(37) we have,

$$P\left(k \overset{SID}{\rightarrow} k'\right) = \prod_{\forall k \in K(t)} \left[\tfrac{5}{2} - \left(\pi_k\right)^2 - \sum_{y=0}^{UB_{Thresh}} H\left(y, k \cup f_{ij}\right)\right]$$
$$\times \sum_{y=0}^{UB_{Thresh}} H\left(y, k \cup f_{ij}\right) \times \left(\pi_k\right)^2 \times \left(M\left(k,t\right) - 1\right) \times \pi_k \tag{38}$$

Probability of DID occurring

The probability for DID to occur is:

$$P\left(k \overset{DID}{\rightarrow} k'\right) = P\left(\begin{matrix} \exists f_{ij} : \left(f_{ij} \text{ could not be added to } k \in K\left(t\right)\right) \ \& \\ \left(f_{ij} \text{ could not be added to } k \in K\left(t\right) \text{ by SID}\right) \\ \text{and } \left(f_{ij} \text{ was added to } k \in K\left(t\right) \text{ by DID}\right) \end{matrix}\right) \tag{39}$$

The summation of the probabilities of request f_{ij} being added/not-being-added to LT k by SID = 1 (events being mutually exclusive and exhaustive, given that f_{ij} is eventually added to k). Using (39) we have:

$P(f_{ij}$ could not be added to $k \in K(t)$ by SID)

$$= 1 - \sum_{y=0}^{UB_{Thresh}} H\left(y, k \cup f_{ij}\right) \times \pi_k \left(1 - \pi_k\right) \left(\pi_k\right)^2 \times M\left(k,t\right) \tag{40}$$

Using similar arguments as for (22), we have, $P(f_{ij}$ was added to $k \in K(t)$ by DID)

$$= \sum_{y=0}^{UB_{Thresh}} H\left(y, k \cup f_{ij}\right) \times \pi_k \left(1 - \pi_k\right) \left(\pi_k\right)^2 \times M\left(k,t\right) \tag{41}$$

Thus using (35)-(36) & (40)-(41), (39) can be re-written as

$$P\left(k \overset{DID}{\rightarrow} k'\right) = \prod_{\forall k \in K(t)} \left[\tfrac{5}{2} - \left(\pi_k\right)^2 - \sum_{y=0}^{UB_{Thresh}} H\left(y, k \cup f_{ij}\right)\right]$$
$$\times \left(1 - \sum_{y=0}^{UB_{Thresh}} H\left(y, k \cup f_{ij}\right) \times \left(\pi_k\right)^2 \times M\left(k,t\right) \times \pi_k \left(1 - \pi_k\right)\right)$$
$$\sum_{y=0}^{UB_{Thresh}} H\left(y, k \cup f_{ij}\right) \times \left(\pi_k\right)^2 \times M\left(k,t\right) \times \pi_k \left(1 - \pi_k\right) \tag{42}$$

Probability of creation of a light-trail for an arriving request

We now compute the probability that a LT is created.

$$P\left(\begin{matrix} k \text{ is} \\ \text{created} \end{matrix}\right) = P\left(\begin{matrix} \exists f_{ij} : \left(f_{ij} \textit{ could not be added to } k \in K\left(t\right)\right) \\ \& \left(f_{ij} \textit{ could not be added to } k \in K\left(t\right) \textit{ by SID}\right) \\ \& \left(f_{ij} \textit{ could not be added to } k \in K\left(t\right) \textit{ by DID}\right) \end{matrix}\right) \tag{43}$$

Using similar arguments as in (40) and (41) we have, $P(f_{ij}$ could not be added to $k \in K(t)$ by DID)

$$= 1 - \sum_{y=0}^{UB_{Thresh}} H\left(y, k \cup f_{ij}\right) \times \pi_k \left(1 - \pi_k\right) \left(\pi_k\right)^2 \times M\left(k,t\right) \tag{44}$$

Using (35)-(36), (39)-(40) & (44), we have (43) as,

$$P\left(k \text{ is created}\right) = \prod_{\forall k \in K(t)} \left[\tfrac{5}{2} - \left(\pi_k\right)^2 - \sum_{y=0}^{UB_{Thresh}} H\left(y, k \cup f_{ij}\right)\right]$$
$$\times \left(1 - \sum_{y=0}^{UB_{Thresh}} H\left(y, k \cup f_{ij}\right) \times \left(\pi_k\right)^2 \times M\left(k,t\right) \times \pi_k \left(1 - \pi_k\right)\right)^2 \tag{45}$$

Probability of creation of a light-trail by NWLR

We compute the probability of node N_i being the originator node as:

$$P\left(N_i = orig\left(t\right)\right) = P\left(i = \arg\max_{\substack{\forall j: N_j \in k, k \in K(t), \\ k \text{ is metastable}}} \left\{\frac{B_{jk}(t)/CT_S}{P_{succ}^{jk}(t)}\right\}\right)$$
$$= \prod_{\substack{\forall j: N_j \in k, k \in K(t), \\ k \text{ is metastable}}} P\left(\frac{B_{ik}(t)}{P_{succ}^{ik}(t)} > \frac{B_{jk}(t)}{P_{succ}^{jk}(t)}\right) \tag{46}$$

We then compute the set of all possible LTs that can be part of $prey_k(t)$ by defining a set $G_k(t)$ as:

$$G_k(t)=\begin{cases} k':conv(k')\in k,\overline{\overline{conv(k')\oplus end(k')}}=1 \text{ or} \\ end(k')\in k,\overline{\overline{conv(k')\oplus end(k')}}=1 \text{ or} \\ \left|S_{end(k')conv(k')}\right|=1,\overline{\overline{end(k')\oplus conv(k')}}=1 \text{ or} \\ \left|S_{end(k)conv(k)}\right|=1,\overline{\overline{end(k)\oplus conv(k')}}=1 \end{cases} \tag{47}$$

where, $\overline{\overline{N_i\oplus N_j}}=1$ means N_i is upstream of N_j on the fiber/wavelength in the direction of the originator light-trail, and 0 otherwise. We then obtain:

$$P\left(k':k'\in G_k(t),k'\in K(t)\right)=\frac{\sum_{f_{ij}\to k:k\in K(t)} f_{ij}(1+\mu)}{N(N-1)wC} \tag{48}$$

The above equation (with parameter μ) is stated to give the probability that k' is a prey LT. In (48), we consider the set of all possible prey LTs for k, *i.e.* $G_k(t)$ and *scale* this set with the network load. Summation over f_{ij} divided by $N(N-1)wC$ gives the utilization factor in the network, which is reduced by a fraction, when load is provisioned through LTs. This reduction in utilization is due to the guard-bands between successive connections. Parameter μ is defined as the *spread-factor* and is the loss of efficiency due to guard-bands. Hence, (48) gives us the probability of a LT k' to exist at time t in the set $G_k(t)$. Let $\Omega_k(t)$ be a set with binary elements whose size is given by:

$$2^{\sum_{j=1}^{|G_k(t)|} {}^{n(k_j)}C_2}\times\sum_{j=1}^{|G_k(t)|} {}^{n(k_j)}C_2=2^{\ddot{x}}\times\ddot{x}$$

The set $\Omega_k(t)$ represents all the feasible combinations of requests at nodes in the prey LTs. The total combination of flows is the number of columns in $\Omega_k(t)$, given by $\sum_{j=1}^{|G_k(t)|} {}^{n(k_j)}C_2=\ddot{x}$, and hence the total number of possible combination of requests is given by: $2^{\ddot{x}}$. Therefore:

$$\Omega_k(t)=\begin{cases}1, & \textit{if } f_{ij}\to k'', \\ 0, & \textit{otherwise.}\end{cases}$$

This is illustrated as:

$$\Omega=\begin{bmatrix}0&0&0&1\\0&0&1&0\\ .&.&.&.\\1&1&1&1\end{bmatrix}_{2^{\ddot{x}}\times\ddot{x}} \tag{49}$$

Each row in (49) now represents a feasible combination of requests from the prey LTs. For each LT $k\in G_k(t)$, ${}^{n(k)}C_2$ flows are possible. Hence, the total request in a possible m^{th} combination is:

$$r_m^{\Omega_k(t)}=\sum_{i,j:f_{ij}\to k',k'\in G_k(t)} f_{ij} \tag{50}$$

The absolute utility of that combination of requests is:

$$\bar{U}(m)=\bar{r}_m^{\Omega_k(t)}=\frac{\bar{r}_m^{\Omega_k(t)}}{n(k'')\times C} \tag{51}$$

Since each (m^{th}) row (denoted by $r_m^{\Omega_k(t)}$) corresponds to a combination of requests, (over all the possible prey LTs), we compute the particular combination that leads to maximum utility: $\arg\max_m\left\{\bar{r}_m^{\Omega_k(t)}\right\}=k''$. This argument denotes the specific row that leads to the maximum utility and thus partially satisfying the rationality property. Therefore,

$$P\left(k\xrightarrow{NWLR}k''\right)=P\left(\bar{U}(k'')=\max_m\left\{\bar{r}_m^{\Omega_k(t)}\right\}\right) \tag{52}$$
$$\times P(LT \text{ k contains the originator}).P(\lambda_{k'} \text{ exists})$$
$$\times P\left(\bar{U}(k'')+\sum_{k'\in prey(t),f_{ij}\in k',f_{ij}\in prey_{flow}(t)}\bar{U}(k'-f_{ij})>\sum_{k'\in prey(t)}\bar{U}(k')\right)$$

However, to *completely* satisfy rationality we compute the utility associated with a flow moving from either the LT k (that contains the originator), or the prey k' to the new LT k''. For this, we must consider if the probability of success of a flow as it moves from one LT to another, is greater than the probability of success of the request in the original LT. Hence, instead of computing the row that gives the maximum utility, we now select all those rows that satisfy validity (flows benefitted by NWLR) and then give the maximum utility. We define q as a column-vector of size $2^{\ddot{x}}\times 1$, whose element is:

$$q_m=\begin{cases}1,\forall f_{ij}:f_{ij}\in k,k=\arg\bar{r}_m^{\Omega_k(t)},P_{succ}^{ik''}>P_{succ}^{ik} \text{ or } P_{succ}^{ik''}>P_{succ}^{ik'} \\ 0,\quad \text{otherwise}\end{cases} \tag{53}$$

With the above definitions we can now compute the probability that NWLR occurs to create LT k'', from an originator in k. The new LT k'' is created on wavelength $w_{k''}$, assuming that $w_{k''}$ is available between the convener and end nodes of k''. For every request that moves from k or k' to k'' we apply validation property – probability of success of the node that houses the flow increases with the association with k'' and the request itself is assigned to k or k'. Hence,

$$P\left(k\xrightarrow{NWLR}k''\right) \tag{54}$$
$$=P\left(N_i\in k,N_i=orig(t)\right).P(w_{k''} \text{ exists}).P\left(\bar{U}(k'')=\max_m\left\{\bar{r}_m^{\Omega_k(t)}\right\}\right)$$
$$.\left[P\left(P_{succ}^{ik'}>P_{succ}^{ik''}\right).P\left(f_{ij}\in k'\right).P\left(k'\in G_k(t)\right)+P\left(P_{succ}^{ik}<P_{succ}^{ik''}\right).P\left(f_{ij}\to k\right)\right]$$

where, $P\left(w_{k''}:\overline{\overline{path_{conv(k'')\sim end(k'')}}}=1\right)=(\pi_k)^2\times M(k,t)$ and

$$P\left(N_i\in k,N_i=orig(t)\right)=\pi_k.\prod_{\substack{\forall j:N_j\in k,k\in K(t),\\ k \text{ is metastable}}}P\left(\frac{B_{ik}(t)}{P_{succ}^{ik}(t)}>\frac{B_{jk}(t)}{P_{succ}^{jk}(t)}\right)$$

We then find the probability that the utilization of the combination of the flows that corresponds to k'' is the highest amongst all the other $\bar{r}_m^{\Omega_k(t)}$ combinations. Thus:

$$P\left(\bar{U}(k'')=\max_m\left\{\bar{r}_m^{\Omega_k(t)}\right\}\right)$$
$$=P\left(\frac{\sum_{f_{ij}:f_{ij}\to k''}f_{ij}}{(n(k'')-1)C}\geq\frac{\sum_{f_{ij}:f_{ij}\to k,k\in r_l^{\Omega_k(t)}}f_{ij}}{(n(k'')-1)C},\forall l\in\left\{1,\ldots,2^{\ddot{x}}\right\}\right) \tag{55}$$

Subsequently, we find the probability that the success of a node in the new LT is greater than the success of the node in the ingress LT and this is given as: $P\left(P_{succ}^{ik'}<P_{succ}^{ik''}\right)=P\left(\sum_{v\in\wp'}P(\{v,i\})<\sum_{v\in\wp''}P(\{v,i\})\right)$. The probability that a prey LT is part of the set $G_k(t)$ is:

$P(k'\in G_k(t))$

$$=P\left(conv(k')\in k\right).P\left(\overline{\overline{conv(k')\oplus end(k')}}=1\right)$$
$$+P\left(end(k')\in k\right).P\left(\overline{\overline{conv(k')\oplus end(k')}}=1\right)$$
$$+P\left(\left|S_{end(k')conv(k')}\right|=1\right).P\left(\overline{\overline{end(k')\oplus conv(k')}}=1\right)$$

$$+P\left(\left|S_{end(k)conv(k)}\right|=1\right).P\left(\overline{\overline{end(k)\oplus conv(k')}}=1\right)$$

Simplifying we get,

$$=\tfrac{3}{2}\pi_k\left(n(k)+n(k')-\tfrac{5}{3}\right) \qquad (56)$$

The last term in (54) computes the probability that request f_{ij} is assigned to LT k' (similar treatment as in (26)).

$$P\left(f_{ij}\to k'\right)=P\begin{pmatrix} f_{ij}:\Delta\bar{U}\left(k'\cup f_{ij}\right)>\Delta\bar{U}\left(k''\cup f_{ij}\right),k''\in K(t) \\ \& \ LB_{Thresh}\le\bar{U}\left(k'\cup f_{ij}\right)\le UB_{Thresh} \text{ and } N_i,N_j\in k' \end{pmatrix}$$
$$+P\left(f_{ij}:\xrightarrow{SID}k'\right)+P\left(f_{ij}:\xrightarrow{DID}k'\right)+P\left(f_{ij}:\xrightarrow{ILFR}k'\right)$$
$$+P\left(f_{ij}:\bar{U}\left(k''\cup f_{ij}\right)\notin OR,\forall k''\in K(t),k':w_{k'}:\overline{\overline{path_{N_i\sim N_j}}}=1\right)$$
$$+P\left(f_{ij}:\xrightarrow{NWLR}k'\right) \qquad (57)$$

Similarly, probability that f_{ij} is assigned to k is computed as in (57). The individual probabilities of the terms in (57) are presented in (28), (30), (33), (38), (42), (54) and (59).

Probability of ILFR

We now compute the probability of a request moving from k to k' due to ILFR. To do so, we recall the definitions of the partial utility $pbid_{ijk}(t)$ and the average partial utility associated with k – $apbid_k(t)$ from Section III.D. The probability of ILFR occurring is then:

$$P\left(k\xrightarrow{Phishing}k'\right) \qquad (58)$$
$$=P\left(pbid_{ijk}(t)>apbid_k(t)\right).P\left(\bar{U}\left(k'\cup f_{ij}\right)+\bar{U}\left(k-f_{ij}\right)>\bar{U}(k')+\bar{U}(k)\right)$$
$$.P\left(\sum_{v\in\wp'}P\left(\{v,i\}\right)<\sum_{v\in\wp''}P\left(\{v,i\}\right)\right).P\left(\bar{U}\left(k'\cup f_{ij}\right)<UB_{Thresh}\right).\tfrac{1}{2}\left(\pi_k\right)^2$$

The probability of flow f_{ij} moving from k to k' due to ILFR is:

$$=P\left(\max\left\{\tfrac{B_{ijk}(t)}{B_{\max}},\tfrac{\sigma_{ijk}(t)}{\sigma_{ijk}(t)+\psi_{ijk}(t)}\right\}>\frac{\sum_{f_{ij}\in k}\max\left\{\tfrac{B_{ijk}(t)}{B_{\max}},\tfrac{\sigma_{ijk}(t)}{\sigma_{ijk}(t)+\psi_{ijk}(t)}\right\}}{\left|f_{ij}:f_{ij}\in k\right|}\right) \qquad (59)$$
$$.\sum_{v>0}\sum_{x,y,z}\left(x,k'\cup f_{ij}\right).H\left(y,k'-f_{ij}\right).H\left(z,k\right).H\left(x+y-z-v,k'\right)$$
$$.P\left(\sum_{v\in\wp'}P\left(\{v,i\}\right)<\sum_{v\in\wp''}P\left(\{v,i\}\right)\right).\sum_{y=0}^{UB_{Thresh}}H\left(y,k'\cup f_{ij}\right).\tfrac{\left(\pi_k\right)^2}{2}$$

where, $B_{ijk}(t)$ is the partial buffer occupancy due to request f_{ij} at N_i. Similarly, $\psi_{ijk}(t)$ and $\sigma_{ijk}(t)$ are the corresponding criticality and allowable limit values due to $B_{ijk}(t)$.

Probability of recourse

Probability that a request f_{ij} is referred to the recourse module is:

$$P\left(f_{ij}\xrightarrow{Recourse}\right)=P\left(f_{ij}\to k\right).P\left(\bar{U}(k)<LB_{Thresh}\right).P\left(t_{live}(k)>TTL\right)$$
$$+P\left(f_{ij}\to k\right).P\left(\bar{U}(k)>UB_{Thresh}\right)+P\left(t_{HP}+T_S>\psi_{ik}(t)\right).P\left(\psi_{ik}(t)>t_{HP}\right)$$
$$+P\left(1-\tfrac{B_{\max}\left(t_{HP}+T_S\right)}{C}<\tfrac{B_{ik}(t)}{B_{\max}}\right).P\left(\tfrac{B_{ik}(t)}{B_{\max}}<1-\tfrac{B_{\max}t_{HP}}{C}\right)$$

Each of the above probabilities has been computed earlier in (11), (26), (30) and (68).

Probability of destruction of a light-trail k

The probability that a LT k is destroyed is given by:

$$P\left(f_{ij}\xrightarrow{Destruction}\right)=P\left(k\xrightarrow{SID}\right)+P\left(k\xrightarrow{DID}\right)+P\left(\bar{U}\left(k'\cup f_{ij}\right)<LB_{Thresh}\right) \qquad (60)$$

The above depends on 3 terms, each of which is computed in equations (30), (38), (42).

Probability of creation of a light-trail

A LT may be created out of SID, DID, creation module or as a result of NWLR. The probability that a LT k is created:

$$P\left(k\in K(t)\right)=P\left(\xrightarrow{SID}k\right)+P\left(\xrightarrow{DID}k\right)+P\left(\xrightarrow{LT\text{ Creation}}k\right)+P\left(\xrightarrow{NWLR}k\right) \qquad (61)$$

Note that the individual components in (61) are given in (38), (42), (45) and (52).

Probability that a light-trail exists

A LT k may encounter one of the following four events in a time-slot t:

1. *Creat*: k is created in the t^{th} time-slot
2. *Dest*: k is destroyed in the t^{th} time-slot
3. ϑ: The last event k encountered was *Dest* and has not been created thereafter.
4. Λ: The last event that k encountered was *Creat* and has not been destroyed thereafter.

To denote the possible events that a LT undergoes, till the t^{th} slot, we define a matrix M_t of size ($2^t\times t$):

$$M_t(i,j)=\begin{cases} Creat, \text{ if } k \text{ has been created in the } j^{th} \text{ slot} \\ Dest, \text{ if } k \text{ has been destroyed in the } j^{th} \text{ slot} \\ \vartheta, \text{ if } k \text{ has been destroyed in some previous} \\ \text{slot and has not been created till the } j^{th} \text{ slot} \\ \Lambda, \text{ if } k \text{ has been created in some previous} \\ \text{slot and has not been destroyed till the } j^{th} \text{ slot} \end{cases} \qquad (62)$$

Note that each row of M_t represents a permutation of *possible* cases till the t^{th} slot, since in the first slot the only case that could happen was *Creat* or ϑ. Subsequently, in every slot, $t>1$, one of the two following events occur:

- *Creat* may only be followed by *Dest* or ϑ
- *Dest* may only be followed by *Creat* or ϑ
- ϑ may only be followed by *Creat* or ϑ
- Λ may only be followed by *Dest* or Λ

Hence, the total number of permutations (rows) that can occur till the t^{th} time-slot are $2\times2^{t-1}=2^t$. Now probability of an element in a row is given as:

$$P\left(r_{ma}^{M_t}\right)=\begin{cases} P\left(\xrightarrow{Creation}k\right), \text{ if } r_{ma}^{M_t}=Creat \\ P\left(k\xrightarrow{Destruction}\right), \text{ if } r_{ma}^{M_t}=Dest \\ 1-P\left(k\xrightarrow{Destruction}\right), \text{ if } r_{ma}^{M_t}=\vartheta \\ 1-P\left(\xrightarrow{Creation}k\right), \text{ if } r_{ma}^{M_t}=\Lambda \end{cases} \qquad (63)$$

Hence, the probability of a particular permutation of events, *i.e.* of the entire row $r_m^{M_t}$ is:

$$P\left(r_m^{M_t}\right)=\prod_{a=1}^{t}P\left(r_{ma}^{M_t}\right) \qquad (64)$$

From the probability of LT creation (61) and LT destruction (60), the probability of existence of a LT is:

$$P\left(k\in K(t)\right)=\sum_{m=1}^{2^t}P\left(r_m^{M_t}\right) \qquad (65)$$

Probability that given $\bar{k}$ (specific) light-trials exist

For a set of $\bar{k}$ LTs to exist, each independent LT must exist.

$$P\left(k_1,k_2,...,k_{\bar{k}} \in K(t)\right)=\prod_{i=1}^{\bar{k}} P\left(k_i \in K(t)\right) \tag{66}$$

Probability that any $\bar{k}$ light-trails exist

For any $\bar{k}$ LTs to exist, any feasible combinations of $\bar{k}$ out of L maximum possible LTs need to exist.

$$P\left(\left|K(t)\right|=\bar{k}\right)=\sum_{k_1,k_2,...,k_{\bar{k}}\in\varepsilon} k_1,k_2,...,k_{\bar{k}} \in K(t) \tag{67}$$

where, ε denotes the set of all feasible $\bar{k}$-combinations of LTs from the universal set of L LTs.

Probability that a light-trail exists beyond the first TTL slots

$$P\left(t_{live}(k)>TTL\right)=1-P\left(t_{live}(k)\le TTL\right)=1-\prod_{t=1}^{TTL} P\left(t_{live}(k)=t\right)$$
$$=1-\prod_{t=1}^{TTL} P\left(\text{LT } k \text{ is destroyed in } (t+1)^{th} \text{ timeslot}\right) \tag{68}$$

The second term in (68) is given in equation (60).

V. Autonomic Optical Network–Simulations

The preceding growth algorithm leads to a system that enables autonomic growth (without manual intervention).

We built a discrete event simulation (DES) model to measure performance over a WDM network that supports LTs. We assume an *N*-node network in both ring and mesh configurations and the algorithm is implemented to create LTs and provision traffic. The model is parameter-driven with traffic, "*OR*", holding times, load, permissible delays and topology being externally controllable. The network assumes 2-fibers between every pair of adjacent nodes and each fiber has 40 data and one control wavelength. Each data wavelength is assumed to have 1 Gbps capacity. Time is modeled in discrete slots with each time-slot of duration 333.3μs and a guard band of 16.66μs. At each node, we have buffers – one per wavelength, and each such buffer has a maximum capacity of 5Mb. Traffic is modeled as voice, video and data with maximum delays of 20, 15 and 50ms. We create sessions between pairs of nodes and these are unidirectional connections over which packets travel from the ingress to the egress. A session is characterized by an arrival process; with Poisson-modeled traffic simulating voice communication, while Pareto-modeled traffic simulates video and data. Packet size is exponentially distributed with maximum transmission unit (MTU) of 1492 bytes. Utilities are sent to the arbiter at the beginning of a time-slot.

Load is computed as the ratio of the total number of bits in the network, to the total capacity of the network divided by average hop-count of connections (*N/4* – for ring networks) and hence is normalized in the range [0,1].

We assume ring and inter-connected ring networks (mesh) – due to their prominence in the metro landscape with 8-20 nodes in different configurations. The minimum and maximum circumferences are 200km and 600km [22] respectively. Nodes are evenly distributed across the network. To simulate a mesh, we assume 3 interconnected rings, with each ring having 6, 6 and 8 nodes respectively. Routing is assumed to always follow shortest path, with the wrap-around path always reserved for protection [10, 14].

A. Number of light-trails

We first measure the number of LTs formed by the system. For comparison, we have four sets of results – analysis, simulation, building a linear program (LP) and a stochastic linear program (SLP). The results of the analysis are computed by "plugging" the values of the arrival rate into the expressions obtained in Section IV. The linear program reflects the *minimum* number of LTs needed to provision the traffic. The linear program was built in [23] and we feed to it traffic values obtained in the simulation. It is seen that the LP gives the lowest values. In [24] we built an SLP using Bender's decomposition [25] method. The idea is to *forecast* traffic based on probabilistic behavior of traffic (modeled as Poisson / Pareto distributions). Exact modeling is not possible and hence an approximation using confidence intervals (of 95.71%) was proposed in [24] which we adapt for this work. The difference between the LP and the SLP is that in the SLP there is a strong correlation between two successive instances of the optimal set of LTs – since there is a penalty associated with creating a new LT as opposed to keeping it over time.

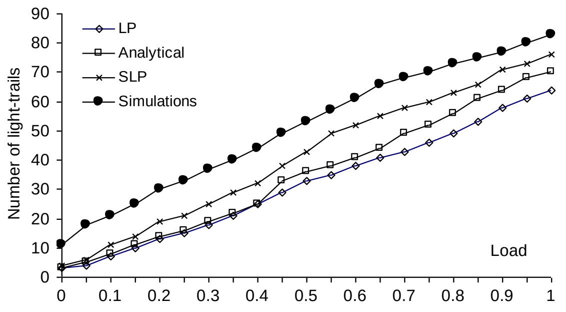


Fig. 5. Number of LTs as a function of normalized load.

Shown in Fig. 5 is a plot of the number of LTs required as a function of load for a 12-node single ring network. The LP provides a theoretical lower-bound and all the results are compared to this LP. However, if we factor inventory at the nodes, then the SLP is more important than the LP, as it results in less wasted transponders than the LP (due to the infrequent setup/tear-down of LTs in the SLP compared to the LP). The analytical results which provide a practical lower limit for the number of LTs are close to the LP. The *OR* is considered in the analytical results and but not in the LP leading to the difference between the two results.

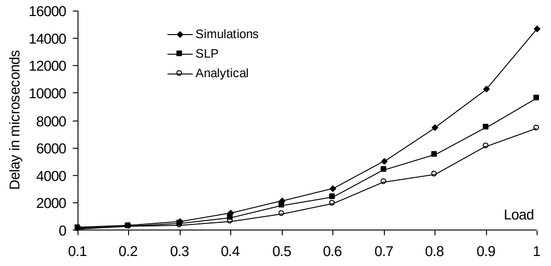


Fig. 6. Edge-to-edge delay using two-stage scheduling algorithm

B. Delay results

Our next parameter of interest is the average *edge-to-edge* delay (Fig. 6) seen through a network from the time a packet enters a trailponder at the ingress node until it leaves the trailponder at the egress node. Delay is upperbounded to 20ms – the maximum waiting time for a packet and is computed as the average over all services. The analytical results provide us the lower bound and these were obtained at an *OR* of 0.8. The simulation results diverge at medium to heavy loads (0.7-1). At unity load, the average delay is close to 15ms. The actual delay on data packets is observed in the 30-35ms but this is offset by voice (5-9ms) and video (4-10ms). The slot-size also has an impact on the delay profile, with an increase in the slot-size decreasing delay at medium loads and increasing it at high-loads. The delay results suggest the advantage of the two-stage algorithm and show that it betters the SLP by 15%. A decrease in the *OR* leads to an increase in delay, and hence multiple plots for *OR* are not shown. However, for *OR*<0.5, the analytical results are worse than the SLP results.

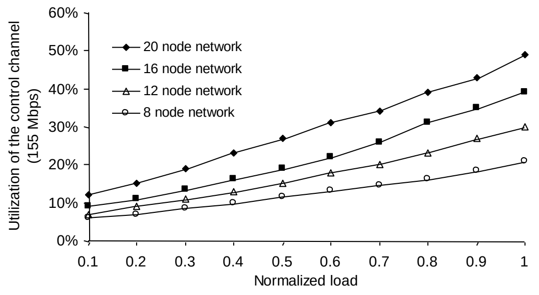


Fig. 7. Utilization of the control channel as a function of load.

C. *Complexity of the control channel*

Given the busy nature of message passing at the control layer, an obvious question is the complexity issue in the control channel. Our goal hence is to evaluate through simulations how the control channel performs as a function of load. We simulate the algorithm and measure control channel performance for different network configurations, thereby investigating if the results have similar profiles across load and across different topologies. Our baseline parameter for complexity in the control channel is the bandwidth of the control channel – fixed at 155Mbps or OC3 (as per the ITU-optical-supervisory-channel requirements). The control channel is electronically processed at every node, resulting in a 250-microsecond processing delay. A fixed amount of network overhead is assumed that varies with load (about 5Mbps at a load of 0.5; and 9Mbps at 0.8 load). Packets used by our algorithm are assumed to be 150 bytes in length, of which 48 bytes correspond to the header. These packets (like *bids, apbids, pbids*, dimensioning packets, set up packets, grant messages and others as mentioned in [3]) are encapsulated in STS3c frames using virtual concatenation. It is seen, in Fig. 7 that even at high loads and complex (mesh) networks the control channel utilization is only 48%. Of the 48% control channel utilization, the actual utilization of protocol packets is less than 20% (9% of the control bandwidth). For less complex networks the utilization only gradually increases with load. Note that the 20-node network has a topology of interconnected rings (mesh).

D. *Packet Drop*

Shown in Fig. 8 is packet drop ratio i.e. ratio of the number of packets dropped to all the packets that go through the network. Initially packet drop is almost a linear function of load. However, packet drop increases exponentially with the number of nodes. For these readings, the average LT sizes were 5 and 8 nodes. On further analysis, for lower loads packet drop happens due to time-out condition, while at higher loads, packet drop occurs due to buffer over-flow. Packet loss at high-loads is less than 1% and most of these are data packets which are recovered through TCP retransmissions.

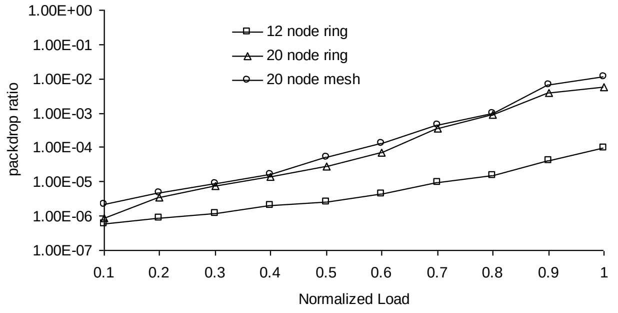


Fig. 8. Packet drop ratio.

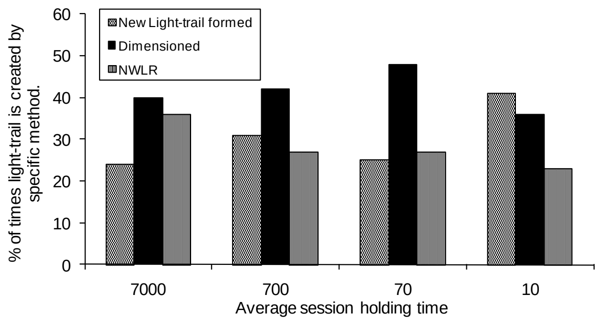


Fig. 9. How is a light-trail created?

E. *New Light-trail formed*

We show in Fig. 9 the percentage of times LTs are created due to (1) dimensioning, (2) new *successful* LTs created to meet a connection request and (3) NWLR. Note that new successful LT implies that a newly created LT is able to survive beyond the time-to-live (*TTL*). The abscissa in Fig. 9 shows traffic dynamism *i.e.* the average session holding time, where a session is defined as the average duration of $T_{ij} > 0$ (see equation (29)). For static traffic, dimensioning followed by NWLR are the preferred options; while for dynamic loads, we have to create new LTs often and NWLR is the least preferred. It is worth noting that NWLR occurrences reduce as dynamism increases. The *TTL* value is assumed to be 16 time-slots.

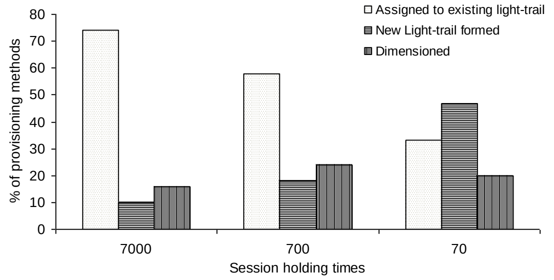


Fig. 10 Assignment of Connections.

Fig. 10 shows how a connection is provisioned (through a LT) as a function of dynamism (session holding time). The 3 possibilities for a new connection to be provisioned include – assigned to an existing LT, a new LT being formed for the connection or an existing LT dimensioned for the connection. Note that as the dynamism increases, the percentage of times an existing LT can be used for connection provisioning decreases, while the percentage of times we have to provision a new LT increases rapidly. Note that dimensioning instances increase in the beginning but decrease later. These measurements were taken at an *OR* of 0.8 and load averaged from 0.4 to 0.8.

F. *The OR-Paradox*

A parameter of interest that governs the performance of the simulation model is *OR*. As discussed in Section III, a higher choice of *OR* results in better efficiency but also increases packet drop ratio and hence we desire to find a workable *OR*. We provide in Fig. 11 and Fig. 12, variations of *OR* as a function of efficiency and packet drop ratio for high loads (0.7 to 0.95) and low loads (0.2 to 0.55). As can be seen there is a significant increase in packet drop ratio for both high and low loads as the *OR* changes from 0.8-0.9, while the efficiency in that region increases linearly. The only region where there is significant improvement in efficiency is when *OR* moves from 0.7 to 0.8, where an almost 15% increase in efficiency is seen. Hence an *OR* between 0.7 and 0.8 leads to an efficient solution.

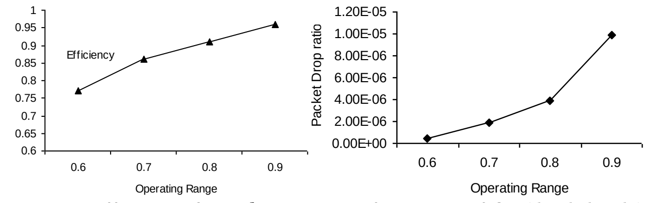


Fig. 11. Efficiency & Packet Drop as functions of OR(high loads).

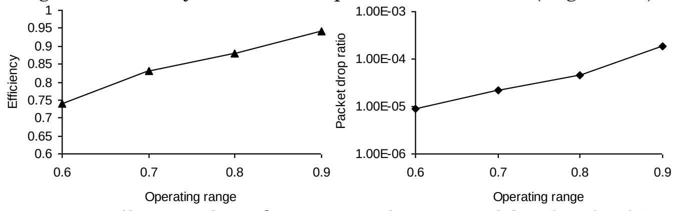


Fig. 12. Efficiency & Packet Drop as functions of OR (low-loads).

VI. Conclusion

In this paper, we have proposed an autonomic growth algorithm for control and design of LT networks. LTs have strong application in metropolitan networks because of their service bearing capabilities. The dynamic topology design problem has been considered by proposing a two-stage growth algorithm. The algorithm is based on bandwidth scheduling between nodes in an LT that leads to topology growth. We also provide a stochastic analysis of this algorithm. The algorithm is simulated and the results are compared to the analysis. The study provides an important aid in designing practical LT networks for providers using autonomic notions of control.

References

[1] K. Bode, "SBC: 18 million fiber users in three years", www.broadbandreports.com/shownews/55799.

[2] M. Chamania and A. Jukan: "A Survey of Inter-Domain Peering and Provisioning Solutions for the Next Generation Optical Networks," IEEE Communications Surveys and Tutorials, Vol. 11, No. 1, First Quarter 2009, pp. 33-51.

[3] A. Gumaste and I. Chlamtac, "Light-trails: A Novel Conceptual Framework for Conducting Optical Communications", IEEE Workshop on High-Performance Switching and Routing, Jun 2003.

[4] A. Gumaste and S. Q. Zheng, "Next Generation Optical Storage Area Networks: The Light-trails Approach", IEEE Commun. Mag., Mar 2005, Vol. 43 No. 3, pp. 72-79.

[5] P. Gokhale, T. Das and A. Gumaste, "Cloud Computing over Light-trail WDM Core Networks", 27th IEEE/OSA Optic Fiber Conference (OFC), TuJ2. March 2010.

[6] A. Gumaste, N. Ghani, P. Bafna, A. Lodha, A. Agrawal, T. Das and S. Zheng, "DynaSPOT: Dynamic Services Provisioned Optical Transport Test-bed – Achieving Multi-Rate Multi-Service Dynamic Provisioning using Strongly connected Light-trail (SLiT) Technology," IEEE/OSA Journal of Lightwave Technology, Jan 2008, Vol. 26, No. 1, pp. 183-195.

[7] A. Gumaste, J. Chandarana, P. Bafna, N. Ghani and V. Sharma, "On Control-Plane for Service Provisioning in Light-trail WDM Optical Networks," IEEE Intl. Conf. on Commun. (ICC) 2007.

[8] A. Gumaste, J. Wang, A. Karandikar and N. Ghani, "Multihop light-trails (MLT) – a solution to extended metro networks", IEEE Intl. Conf. on Commun. (ICC) 2009.

[9] S. Balasubramanian, A. K.Somani, and A. E. Kamal, "Sparsely Hubbed Light-Trail Networks", IEEE Intl. Conf. on Comp. Comm. Net. (ICCCN), October 2005 Boston, USA.

[10] X. Luo and B. Wang, "Integrated Scheduling of Grid Applications in WDM Optical Light-Trail Networks", IEEE/OSA J. of Lightwave Tech., Vol.27, No.12, pp.1785-1795, June15, 2009.

[11] Y. Ye, H. Woesner and I. Chlamtac, "OTDM Light-trail Networks," Intl. Conf. on Transparent Optical Networks (ICTON), July 2005.

[12] S. Balasubramanian and A. K. Somani, "A Comparative Study of Path Level Traffic Grooming Strategies for WDM Optical Networks with Dynamic Traffic", Invited Paper, IEEE Intl. Conf. on Comp. Comm. Net. (ICCCN) 2008.

[13] B. Mukherjee, "Optical WDM Networks", ISBN: 978-0-387-29055-3, Springer, 2006.

[14] J. Xing, H. Wang, and Y. Ji, "BLE protection scheme for light-trail WDM mesh networks" Proc. SPIE, 6784, pp. 67840X (2007)

[15] P. Palacharla, A. Gumaste, E. Biru and T. Naito, "Implementation of Burstponder Card for Ethernet Grooming in Light-trail WDM Networks," IEEE Int'l Conf. on Commun. (ICC) 2006.

[16] USPTO, A. Gumaste, US Patent #7,590,353: "System and Method for bandwidth allocation in an optical light-trail".

[17] USPTO, P. Palacharla, A Gumaste and S Kinoshita, US Patent #7,616,891, "System and method for transmission and reception of traffic in optical light-trails"

[18] B. Doytchinov, J. Lehoczky and S. Shreve, "Real-time queues in heavy-traffic with earliest-deadline-first queue discipline" Annals of Applied Probability, 2001, Vol.11, No.2, pp. 332–378.

[19] W. Feller, "An Introduction to Probability Theory and Its Applications", Vols. I, II, Wiley, New York, 1968.

[20] K. S. Trivedi, "Probability and Statistics with Reliability, Queuing, and Computer Science Applications", John Wiley and Sons, New York, 2001. ISBN: 978-0-471-33341-7.

[21] R.A. Barry and P. A. Humblet, "Models of Blocking Probability in All-Optical Networks with and Without Wavelength Changers", IEEE INFOCOM 1995, April 1995.

[22] Talk by Stuart Elby, Verizon, in Market Watch, OFC 2008, San Diego, CA, Feb 2008.

[23] A. Gumaste and P. Palacharla, "Heuristic and Optimal Assignment Techniques for Light-trail Ring WDM Networks", Elsevier Computer Communications Journal (CCJ) Oct. 2006.

[24] A. Lodha, A. Gumaste, P. Bafna and N. Ghani, "Stochastic Optimization of Light-trail Optical WDM Ring Networks using Bender's Decomposition", IEEE Conf. on High Speed Routing and Switching (HPSR), June 2007.

[25] R. Freund, "Benders' Decomposition Methods for Structured Optimization, including Stochastic Optimization," http://ocw.mit.edu/NR/rdonlyres/Sloan-School-of-Management/15-094JSpring-2004/B5BC04B2-A362-4A78-A67C-399948CB1776/0/benders_art.pdf