# A Link State Advertisement (LSA) Protocol for Optical Transparency Islands

Shovan Das, Reza Roshani Tabrizi, Paolo Monti, Marco Tacca, and Andrea Fumagalli

## Technical Report UTD/EE/02/2006

**April 2006, revised August 2006**

# A Link State Advertisement (LSA) Protocol for Optical Transparency Islands

Shovan Das, Reza Roshani Tabrizi, Paolo Monti, Marco Tacca, and Andrea Fumagalli
The Open Networking Advanced Research (OpNeAR) Lab
Erik Jonsson School of Engineering and Computer Science
The University of Texas at Dallas, Richardson, TX 75080, USA
Email:{skd043000,rxr058100,paolo,mtacca,andreaf}@utdallas.edu

*Abstract*— **Plug and play optical (PPO) nodes can be used to ease the deployment of optical networks. Once plugged, PPO nodes provide all-optical circuits between client nodes to alleviate the electronic processing bottleneck of high speed networks. PPO nodes must self-adjust to changes of the optical physical topology and fiber propagation characteristics, and provide wavelength routing functionalities to client nodes.**

**This paper presents a protocol, the TI-LSA protocol, for physical topology discovery at the PPO node layer, e.g., it may be used to advertise available optical resources and changing conditions of the optical physical layer. The protocol is based on the link state advertisement (LSA) principle and modified to take advantage of the transparency island (TI) properties in the optical data plane.**

**As discussed in the paper, the proposed TI-LSA protocol is a scalable solution to the problem of topology discovery and update in PPO networks when the optical transparency island size is relatively small.**

## I. Introduction

Today technology in fiber optics communications has the potential to facilitate end-to-end data exchange in the multi-gigabit transmission range [1]. Optical circuits, or lightpaths [2], can be established in the network to provide transparent channels between end node pairs [3]. Electronic processing of transmitted data along a lightpath is not required, thus avoiding a potential electronic processing bottleneck when high transmission rates are required. This is often referred to as optical transparency.

In some areas, the deployment of optical networks may be facilitated by the use of self-configurable plug and play optical (PPO) nodes [4]. Similarly to wireless ad hoc solutions [5], PPO networking could allow fast and ad hoc deployment of optical resources to best fit application requirements. It could also simplify the complex procedures for the design, installation, and maintenance of today optical networks, as no-human intervention would be required to perform these tasks.

The key components of the PPO node are: $(i)$ an on-board miniature optical transmission laboratory, or mini-lab, $(ii)$ real-time transmission models, and $(iii)$ a service channel interface for network management and control. The real-time transmission models are used at the PPO node to process the measurements produced by the on-board optical mini-lab. Their objective is to estimate the maximum transmission rate and span that are permitted when creating lightpaths from the PPO node to other PPO nodes available in the network. Lightpath rate and span are both limited by the physical constraints of the optical medium. In practice, the PPO node is able to establish lightpaths to reach only a subset of other PPO nodes, and these nodes are said to belong to the PPO node *transparency island* (TI) [6], [7]. Note that each PPO node has its own TI, and that some TI(s) may overlap. The use of TI makes it possible to naturally limit both the scope of the PPO node action and the required amount of local control information at the PPO node.

Once connected, a PPO node must cooperate with other already existing PPO nodes, and determine how to use available optical resources, such as fibers and wavelengths, to provide requested lightpaths to the connected clients, e.g., end users, routers. To do so, the PPO node must 1) discover resources and detect changes in the optical data plane, and 2) solve the routing and wavelength assignment (RWA) problem [8]. Finding a solution to the former problem and assessing the solution scalability is the focus of this paper.

Discovery of resources and detection of changes in networking are often accomplished using link state advertisement (LSA) protocols. A well know example is the open shortest path first (OSPF) protocol [9], [10], whereby link state information entries are flooded across the network. To provide a scalable solution, flooding of LSA entries is constrained within areas, which are defined when the network is designed. Areas do not change in time, and each network node belongs to one area only[1], i.e., areas do not overlap. Therefore, LSA protocols based on areas or subnetworking do not fit the PPO ad hoc networking requirements, as nodes can be added and removed many times during the network lifetime. For this reason, the TI-LSA protocol is introduced.

The TI-LSA protocol is based on the OSPF LSA flooding principle, which is adapted to take advantage of the PPO node TI. The number of flooded TI-LSA entries is limited by constraining the advertisement within the optical reach of the PPO node, i.e., the boundaries of the PPO node TI. In other words, each TI-LSA entry reaches the PPO nodes that require that information, without unnecessarily flooding

This research was supported in part by NSF Grants No. ANI-0435393 and the Italian Ministry of University (MIUR) (contract # RBNEO1KNFP).

[1] With the exception of area border gateways.

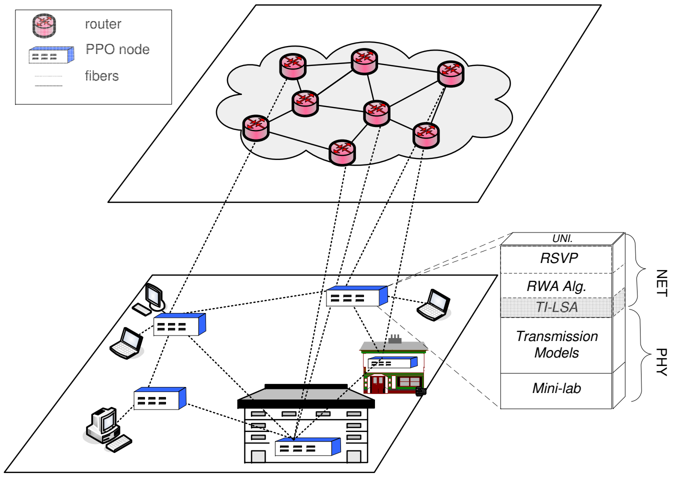

Fig. 1. PPO node enabled network

other PPO nodes that will not make use of that entry. While the TI may resemble the concept of optimal partitioning of domains into OSPF areas [11], [12], a substantial difference between the two is that areas are manually predetermined, while TI(s) are self detected. The TI(s) may change over time as they automatically adapt to the changes of both the optical network topology and the fiber transmission characteristics, e.g., affected by temperature and aging. And so does the constrained flooding of the TI-LSA entries. While delivering to each PPO node all the TI-LSA entries that may be required to intelligently use the available optical resources, the TI-LSA protocol permits to realize networks with a number of PPO nodes that is virtually infinite, thanks to its TI constrained flooding.

A particular challenge in designing the TI-LSA protocol is that TI(s) of distinct PPO nodes may be different and may overlap partially. Some substantial protocol changes are then required when compared to the conventional single area LSA (SA-LSA) protocols, e.g., the OSPF protocol implemented using a single area. The payoff, as discussed in the paper, is that by taking into account the PPO node TI the proposed TI-LSA protocol constitutes a scalable solution to the problem of topology discovery and update when the TI size is relatively small. This is indeed the case in optical networking when sophisticated and costly signal regeneration techniques are not an option, e.g., 3R [13]. This observation suggests that PPO networking may be well suited in the access and metro area, where both inexpensive equipment on the one hand, and ad hoc deployment on the other may be premium features.

Before describing the TI-LSA protocol, the envisioned PPO node network architecture is defined in the next section.

## II. The PPO Node Network Architecture

Fig. 1 depicts a network with both client (routers) and PPO nodes. Via a user-network interface (UNI), client nodes may request PPO nodes to create lightpaths to form a desired virtual topology. Upon reception of a lightpath request to connect two client nodes, the PPO node must first determine if it possible to establish that lightpath and meet the transmission rate requirement specified by the client. To perform this task, the PPO node must be aware of the physical topology, fiber transmission impairments (e.g., power loss, polarization mode dispersion) and available wavelengths. Note that changes of the physical topology may be frequent as PPO nodes are plugged and unplugged dynamically as needed. Once the physical topology, fiber transmission impairments and available wavelengths are known to the source PPO node, conventional RWA algorithms can be applied to choose both path and wavelength to establish the requested lightpath. Conventional reservation protocols (e.g., RSVP [14], [15]) may then be used to establish the requested lightpath and allow client nodes to exchange packets directly over that lightpath.

The PPO node sub-layers are shown in Fig. 1. First, the PPO node must characterize and monitor the key transmission parameters of the outgoing fiber links connecting to neighboring PPO nodes. This task is performed in cooperation with the neighboring PPO nodes, using the on-board optical mini-lab [4] and a Hello protocol [9]. The mini-lab measures the fiber key transmission parameters and detects their changes over time. The Hello protocol detects fiber cuts, fiber repairs. Fiber changes may positively or negatively affect the optical signal quality in the data plane. Based on their impact on the signal quality they are classified as `perf-up`, `link-up`, `link-down`, or `perf-down` events. When their impact on the signal quality cannot be determined, they are classified as `perf-unknown` events.

To gain a more comprehensive view of the physical topology the fiber measurements and detected fiber link changes are flooded to other PPO nodes in the form of TI-LSA entries. The TI-LSA entries may also be used to flood information about availability of wavelengths. Each PPO node combines the TI-LSA entries received from the other PPO nodes to build its own link state database. Based on its link state database and by using real-time transmission models [16], the PPO node can swiftly determine whether a requested lightpath can be established.

As already mentioned, in this architecture, the PPO node is only required to deal with lightpaths that can be established while producing acceptable optical signal quality at the receiver. In practical terms, lightpaths that span across too many fibers (without signal regeneration) may not be created as their resulting signal quality does not yield the desired bit error rate. This observation leads to the conclusion that the PPO node link state database may be limited to represent a subset of the entire physical topology, i.e., its own TI.

Formal definitions of TI and description of the TI-LSA protocol are given next.

## III. Transparency Island Definition

Fig. 2 provides a sketch of the optical TI(s) of two PPO nodes. The TI reach is dependent on the employed transmission rate and might vary over time due to changes of both the topology, and the fiber parameters. Plugging of new PPO nodes

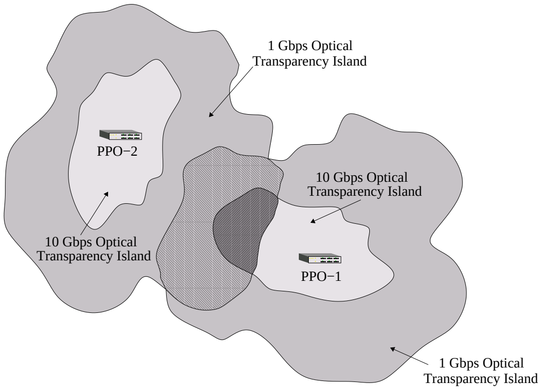

Fig. 2. Optical transparency islands

and/or deploying of a new fibers may enable some PPO nodes to reach remote PPO nodes that were previously unreachable. Alternatively, unplugging of a PPO node and/or a fiber cut may deprive some PPO nodes of optical resources, thus reducing the size of the TI. A formal definition of the TI is given next.

The PPO node network is modeled as a graph $G(\mathcal{N},\mathcal{A})$, where $\mathcal{N}$ is the set of PPO nodes in the network and $\mathcal{A}$ is the set of uni-directional links connecting the PPO nodes. A PPO node with ID $i$ is denoted as $N_i$ and a link connecting PPO nodes $N_i$ and $N_v$ is denoted as $l_{(i,v)}$. Let $C_{(i,j)}^{(r)}$ be the set of all simple paths that can be used to establish a lightpath[2], operating at transmission rate $r$, connecting PPO node $N_i$ to $N_j$. Set $C_{(i,j)}^{(r)}$ defines a subgraph of $G(\mathcal{N},\mathcal{A})$. Note that lightpaths containing loops are not allowed. Let $TI_i^{(r)}$ be the transparency island associated with node $N_i$, for lightpaths operating at transmission rate $r$. $TI_i^{(r)}$ is a subgraph of $G$, defined by:

$$TI_i^{(r)} = \cup_{j\in\mathcal{N}} \left( C_{(i,j)}^{(r)} \right). \qquad (1)$$

$TI_i^{(r)}$ contains all PPO nodes and links that can be used to establish a lightpath at transmission rate $r$ that originates at PPO node $N_i$. Note that incoming links $l_{(j,i)}$ cannot be in $TI_i^{(r)} \forall r, i$, as lightpaths containing loops are not allowed.

## IV. TI-LSA Protocol

This section describes the TI-LSA protocol used to gather link state information at every PPO node, i.e., building and maintaining $TI\_DB_i^{(r)}$ at PPO node $N_i$ for one single rate[3] $r$. $TI\_DB_i$ is a subgraph of $G(\mathcal{N},\mathcal{A})$ and it is the transparency island associated with node $N_i$ that is built based on, possibly inaccurate, link state information at PPO node $N_i$. $TI_i$ is the transparency island of PPO node $N_i$ when $N_i$ has complete and accurate knowledge of the PPO layer topology.

The design of the TI-LSA protocol is based on the following assumptions:

- PPO node $N_i$ is the *owner node* of its outgoing links, i.e., $N_i$ is in charge of advertising the status of its outgoing links,
- the owner node detects the status of its outgoing link in a finite duration of time, i.e., each outgoing link status change is detected by the combined use of a Hello protocol [9] and measurements produced by the on-board optical mini-lab,
- links $l_{(i,v)}$ and $l_{(v,i)}$ are subject to the same performance changes, e.g., the two links go down or come up at once,
- all transmitted control messages are received without error within a finite duration of time,
- all exchanged control messages are processed within a finite duration of time,
- a lightpath can be established when the bit error rate performance is satisfactory,
- it is assumed that the wavelength chosen for the lightpath does not affect the lightpath performance, i.e., if a lightpath can be established on a simple path $p \in C_{(i,j)}^{(r)}$ using a wavelength, then it can be also established using any other wavelength along the same simple path $p$,
- if a lightpath can be established on a simple path $p \in C_{(i,j)}^{(r)}$, then any lightpath routed using a sub-paths of $p$ can be established too.

Recall that the objective of the TI-LSA protocol is to inform $N_i$ of any status change that may occur on the links that belong to $TI_i$, i.e., the TI-LSA protocol must guarantee that $TI\_DB_i = TI_i$ within a finite duration of time. This objective is accomplished by transmitting messages that contain link state update(s), i.e., LSU-ENTRY(s). An LSU-ENTRY is a field in the messages exchanged that contains information regarding one link. For example, an LSU-ENTRY regarding link $l_{(i,v)}$ may contain the fiber power loss and current dispersion profile, the available wavelengths, etc. By the simple fact of receiving an LSU-ENTRY about link $l_{(i,v)}$, PPO node $N_j$ can conclude that $N_i$ and $N_v$ are active and running. Several LSU-ENTRY(s) (local or remote) can be grouped together by the PPO node to form a single advertisement message defined as LSU-PACKET. The PPO node can also remove any unnecessary LSU-ENTRY from a LSU-PACKET to reduce the signaling overhead.

A number of events can trigger PPO node $N_i$ to flood the status of one or more of its outgoing links to all the neighboring PPO nodes. These events are:

- `perf-up`, `link-up`, `perf-unknown`, i.e, a link may improve its performance, come up or change its performance in an unknown way, respectively,
- `perf-down`, `link-down`, i.e., a link can deteriorate its performance, or go down, respectively.

Events `perf-up`, `link-up`, and `perf-unknown` are grouped together, as they require the TI-LSA protocol to perform the same set of procedures. The same apply to events `perf-down` and `link-down`. Note that the plugging (unplugging) of a PPO node is detected in the form of a set of `link-up` (`link-down`) events.

[2] It is assumed that if a lightpath can be established, then its performance in terms of transmission impairments is satisfactory.

[3] For the first version of the TI-LSA protocol, only one data rate is supported. For simplicity, in the remainder of the paper, the index $r$ is not used, e.g., $TI\_DB_i^{(r)}$ is denoted by $TI\_DB_i$.

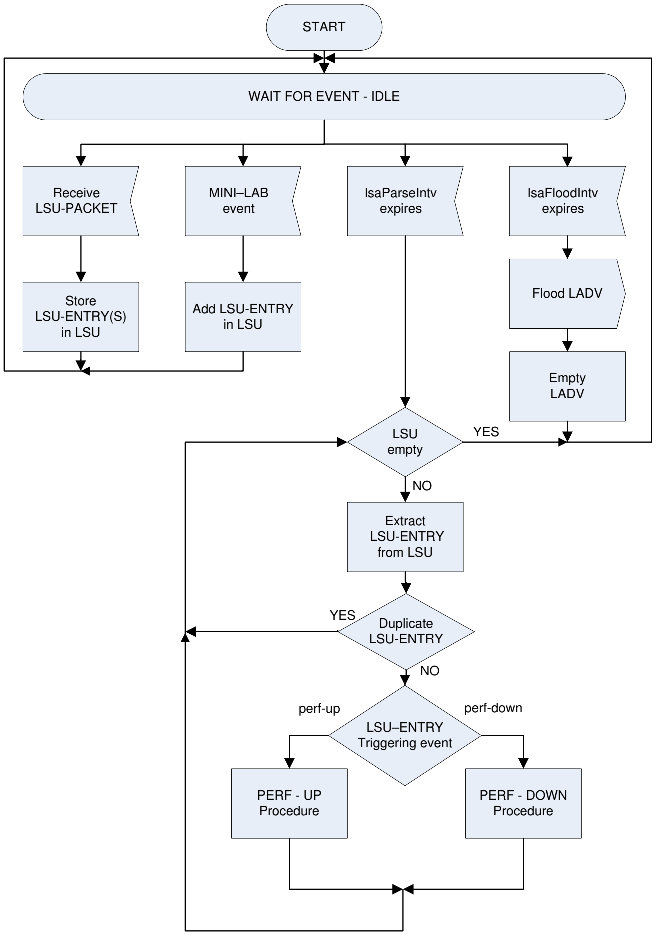

Fig. 3. TI-LSA protocol flowchart

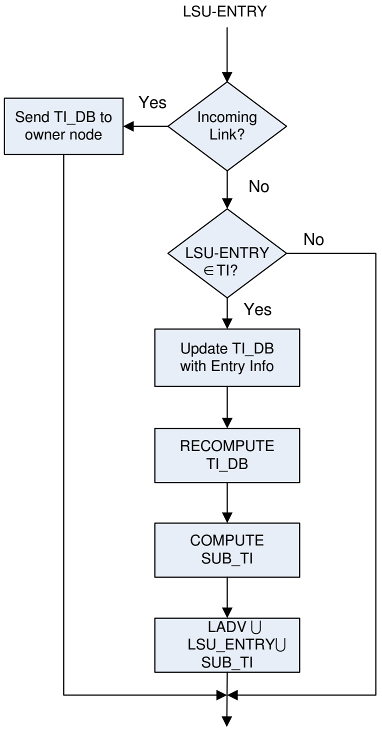

Fig. 4. `perf-up`, `link-up`, and `perf-unknown` flowchart

The description that follows applies to PPO node $N_i$. The flowchart of the TI-LSA protocol is shown in Fig. 3. Upon reception of an LSU-PACKET at an incoming interface, every LSU-ENTRY in the LSU-PACKET is processed within a finite time. Before processing the received LSU-ENTRY(s), the PPO node waits for a counter (`lsaParseIntv`) to expire. The `lsaParseIntv` counter is used to make it possible to receive multiple LSU-PACKET(s) and to process all their LSU-ENTRY(s) at once. All received LSU-ENTRY(s) are temporarily stored in set LSU. An LSU-ENTRY can also be created and stored in the LSU in response to an event generated by the mini-lab and Hello message exchange.

The first step when processing a received LSU-ENTRY is to verify that the LSU-ENTRY is not a duplicate, i.e., the LSU-ENTRY was not already received. This can be done by assigning a sequence number to the LSU-ENTRY. If the LSU-ENTRY is not a duplicate, the second step is to check which event triggered the flooding, e.g., `perf-up` or `perf-down`, etc. Depending on the triggering event, the LSU-ENTRY is handled following one of two possible procedures (Figs. 4 and 6). The outcome of these procedures is to determine whether or not the LSU-ENTRY has to be included in the local $TI\_DB_i$ and further advertised to neighboring PPO nodes. If the LSU-ENTRY has to be further flooded, it is stored into the link advertisement (LADV) database. When a counter (`lsaFloodIntv`) expires, all LSU-ENTRY(s) in LADV are moved into an LSU-PACKET and flooded at once.

Fig. 4 shows the flowchart of the procedure used when the LSU-ENTRY is triggered by one of the following events: `perf-up`, `link-up`, and `perf-unknown`. Let the LSU-ENTRY be associated with link $l_{(k,v)}$. First, $N_i$ must determine whether the LSU-ENTRY is associated with an incoming link, i.e., $v = i$. If so, $N_i$ must send its entire transparency island database ($TI\_DB_i$) to the owner of the link, i.e., PPO node $N_k$. Otherwise, if $l_{(k,v)}$ belongs to $TI_i$[4] then both $l_{(k,v)}$ and $N_v$ are added to $TI\_DB_i$. Then, $SUB\_TI^{(i)}_{(k,v)}$ is computed as follows. $SUB\_TI^{(i)}_{(k,v)}$ is a subgraph of $TI\_DB_i$ and it is defined as the set of links and nodes that may be used to establish lightpaths starting at $N_i$, using $l_{(k,v)}$, and terminating at $N_j$, $\forall N_j \in \mathcal{N}$. The LSU-ENTRY(s) associated with all the links in $SUB\_TI^{(i)}_{(k,v)}$ are then added to LADV for flooding. Note that this step is peculiar of the TI-LSA protocol and it is not required in the conventional OSPF. An example is used next to clarify the importance of this step.

Fig. 5 shows the importance of calculating $SUB\_TI$ and adding all LSU-ENTRY(s) associated with links in $SUB\_TI$ to LADV with an example. In the example, it is assumed that lightpaths cannot exceed 4 hops due to signal quality requirements. Thus a generic link $l_{(u,v)}$ belongs to $TI\_DB_i$

[4] The actual algorithm used to determine whether the LSU-ENTRY is to be included in $TI\_DB_i$ is outside the scope of this paper.

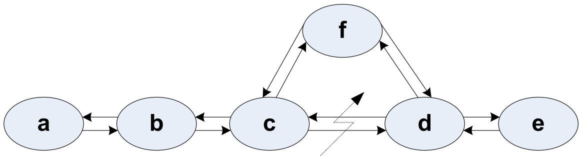

Fig. 5. $SUB\_TI$ example

if it is possible to establish at least one lightpath (which does not require more than 4 hops) from $N_i$ to $N_v$ using $l_{(u,v)}$. Assume that links $l_{(c,d)}$ and $l_{(d,c)}$ are up only after time $t_0$. Before $t_0$ we have:

- $TI_a = TI\_DB_a = \{l_{(a,b)}, l_{(b,c)}, l_{(c,f)}, l_{(f,d)}\}$
- $TI_b = TI\_DB_b = \{l_{(b,a)}, l_{(b,c)}, l_{(c,f)}, l_{(f,d)}, l_{(d,e)}\}$
- $TI_c = TI\_DB_c = \{l_{(b,a)}, l_{(c,b)}, l_{(c,f)}, l_{(f,d)}, l_{(d,e)}\}$
- $TI_d = TI\_DB_d = \{l_{(b,a)}, l_{(c,b)}, l_{(f,c)}, l_{(d,f)}, l_{(d,e)}\}$

At time $t_0$, $l_{(c,d)}$ becomes operational and triggers the following events[5]. When $N_d$ detects the status change of its incoming link $l_{(c,d)}$, it sends an LSU-ENTRY for every link in $TI\_DB_d$ to $N_c$. When $N_c$ receives the LSU-ENTRY(s), $l_{(d,f)}$ is the only link that is added to $TI\_DB_c$. Then $N_c$ computes $SUB\_TI^{(c)}_{(c,d)} = \{l_{(c,d)}, l_{(d,f)}, l_{(d,e)}\}$. For each link in $SUB\_TI^{(c)}_{(c,d)}$ an LSU-ENTRY is stored in LADV and flooded using an LSU-PACKET by $N_c$ to its neighbors. When $N_b$ receives the LSU-PACKET, the only link that must be added to $TI\_DB_b$ is $l_{(d,f)}$. $N_b$ computes $SUB\_TI^{(b)}_{(c,d)} = \{l_{(b,c)}, l_{(c,d)}, l_{(d,f)}, l_{(d,e)}\}$. Once again, for each link in $SUB\_TI^{(b)}_{(c,d)}$ an LSU-ENTRY is stored in LADV and flooded using an LSU-PACKET by $N_b$ to its neighbors. When $N_a$ receives the LSU-PACKET, both $l_{(d,f)}$ and $l_{(d,e)}$ are added to $TI\_DB_a$. Notice now what would happen if a procedure similar to the OSPF flooding mechanism is used in the example instead, i.e., the LSU-ENTRY associated with a link is flooded only if the link is not already present in the PPO node database. When $N_b$ receives the LSU-PACKET from $N_c$, it floods an LSU-PACKET containing information about $l_{(d,f)}$, but not about $l_{(d,e)}$, because the latter is already in $TI\_DB_b$. As a result, it is impossible for $N_a$ to correctly build $TI\_DB_a$.

Fig. 6 shows the flowchart of the procedure used when the LSU-ENTRY is triggered by one of the following events: `perf-down` and `link-down`. Let the LSU-ENTRY be associated with link $l_{(k,v)}$. Notice that, whenever there is a `perf-down` and `link-down` type, $TI\_DB_i$ may at most loose some links (and nodes). Therefore, it is not necessary for $N_i$ to compute $SUB\_TI_i$. First, $N_i$ checks whether an entry associated with link $l_{(k,v)}$ is present. If not, no further action is required. Otherwise, $N_i$ recomputes and updates $TI\_DB_i$, and stores an LSU-ENTRY associated with $l_{(k,v)}$ in LADV.

## V. Correctness of the TI-LSA Protocol

This section demonstrates the correctness of TI-LSA protocol. Additional assumptions are needed:

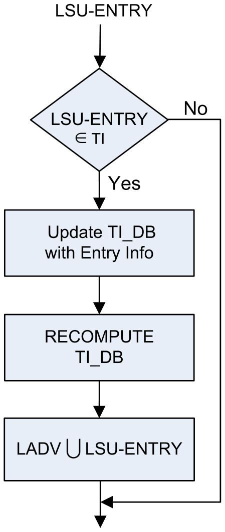

Fig. 6. `perf-down`, `link-down` flowchart

- the size of the network $G(\mathcal{N}, \mathcal{A})$ is finite,
- there is a finite number of link state changes up to time $t_0$,
- the physical topology, i.e., $G(\mathcal{N}, \mathcal{A})$, is the same for the data plane (for establishing lightpaths) and for the control plane (for exchanging LSU-PACKET(s)),
- each PPO node is able to determine whether a link associated with a received LSU-ENTRY should be included or removed from its transparency island.

*Theorem 1*: If there is a finite number of link state changes in the network up to time $t_0$ then, at a finite time after $t_0$, each PPO node $N_i$ receives all the necessary information and is able to correctly determine $TI\_DB_i$, i.e., the TI-LSA protocol is correct. To prove the above theorem few lemmas are used.

*Lemma 1*: When a PPO node $N_i$ receives an LSU-ENTRY associated with $l_{(u,v)}$ at time $\hat{t}$ through a simple path $p$ from $N_u$, it must have received by then other LSU-ENTRY(s), which make $N_i$ aware of all the links along simple path $\bar{p}$ to node $N_u$. Path $p$ is a simple path from node $N_u$ to $N_i$, $\bar{p}$ is a simple path from $N_i$ to $N_u$ traversing the same set of PPO nodes as $p$, but in the reverse order.

*Lemma 2*: Within a finite time after $t_0$, every PPO node $N_i$ receives at least an LSU-ENTRY about every link in $TI_i$

*Lemma 3*: Within a finite time after $t_0$, PPO node $N_i$ receives at least one LSU-ENTRY about $l_{(u,v)}$ and $N_i$ has information in $TI\_DB_i$ about all the links along the best performing lightpath from $N_i$ to $N_v$ via $l_{(u,v)}$. Here, best performing lightpath refers to the lightpath with the best optical signal quality.

Lemma 1, 2, 3, and Theorem 1 are proved next.

*1) Proof of Lemma 1:* Fig. 7 is used to prove Lemma 1. Assume that PPO node $N_i$ receives the LSU-ENTRY associated with link $l_{(u,v)}$ at time $\hat{t_2}$ through simple path $p_2$. Then according to Lemma 1, $N_i$ must have already received LSU-ENTRY(s) about all links along path $\bar{p_2}$, i.e.,

[5] For the sake of simplicity, events that are not relevant to the example are ignored.

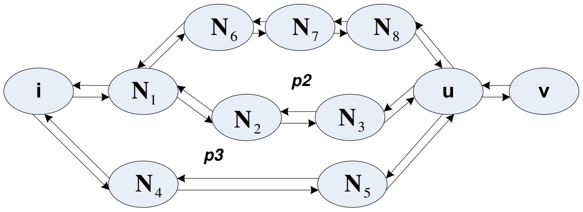

Fig. 7. TI-LSA protocol example

$l_{(i,N1)}, l_{(N1,N2)}, l_{(N2,N3)}, l_{(N3,u)}$ at time $\hat{t_2}$ or before time $\hat{t_2}$.

This can be easily seen by analyzing how the LSU-ENTRY associated with $l_{(u,v)}$ is advertised. First, the PPO owner node $N_u$ floods an LSU-PACKET containing an LSU-ENTRY associated with $l_{(u,v)}$. When the LSU-PACKET is received by PPO node $N_{N3}$, two cases are possible. In the first case, $N_{N3}$ has already flooded an LSU-PACKET containing an LSU-ENTRY associated with $l_{(N3,u)}$ towards $N_{N2}$. In the second case, $N_{N3}$ has not yet flooded an LSU-PACKET containing an LSU-ENTRY associated with $l_{(N3,u)}$ towards $N_{N2}$. In the latter case, the LSU-ENTRY(s) associated with both $l_{(u,v)}$ and $l_{(N3,u)}$ are stored in LADV. When timer `lsaFloodIntv` expires, the content of LADV is used to form an LSU-PACKET that is flooded and sent to $N_{N2}$. Therefore, PPO node $N_{N2}$ receives the LSU-ENTRY(s) associated with both $l_{(u,v)}$ and $l_{(N3,u)}$ at the same time. Similar observations can be made about the other PPO nodes along $p_2$.

*2) Proof of Lemma 2:* Lemma 2 ensures that the LSU-ENTRY associated with $l_{(u,v)}$ is received by $N_i$ within a finite amount of time if $l_{(u,v)}$ belongs to $TI_i$. First, consider a single link change in the network. Later this assumption can be relaxed to consider an arbitrarily large (but finite) number of link changes in the network. To prove this lemma two scenarios must be considered. The first scenario is when a link is added to $TI\_DB_i$ due to a `perf-up` or `link-up` or `perf-unknown` event. The second scenario is when a link is deleted from $TI\_DB_i$ due to a `perf-down` or `link-down` event.

*Scenario I*: If $l_{(u,v)}$ belongs to $TI_i$ but $N_i$ is not aware of it yet, i.e., $l_{(u,v)}$ is not in $TI\_DB_i$, then $N_i$ should receive an LSU-PACKET containing an LSU-ENTRY associated with link $l_{(u,v)}$, which enables $N_i$ to correctly update $TI\_DB_i$.

*Proof of scenario I*: Scenario I is proved through mathematical induction.

**Initial Step.** Consider the case where $N_i$ is one hop away from $N_u$. If $l_{(u,v)}$ belongs to $TI_i$ but PPO node $N_i$ is not aware of it yet, only two cases are possible. Either $l_{(u,v)}$ or $l_{(i,u)}$ has undergone one of the following events: `perf-up`, `link-up`, or `perf-unknown`. If $l_{(i,u)}$ undergoes one of `perf-up`, `link-up`, or `perf-unknown` events, the PPO owner node $N_i$ will detect the change and flood the LSU-ENTRY associated with $l_{(i,u)}$ to all its neighbors. $N_u$ is one of the neighbors. $N_u$ receives the LSU-ENTRY and detects that it is associated with one of its incoming links. Then, according to the TI-LSA protocol, it sends an LSU-PACKET containing LSU-ENTRY(s) about all the links in $TI\_DB_u$ to $N_i$. Note that if $l_{(u,v)}$ is in $TI_i$, then it must be in $TI_u$ too. Through this LSU-PACKET, $N_i$ becomes aware of $l_{(u,v)}$. If $l_{(u,v)}$ undergoes one of `perf-up`, `link-up`, or `perf-unknown` events, then $N_u$ will detect the change using either the on-board micro lab or the Hello message exchange. $N_u$ then recomputes $TI_u$, and it adds $l_{(u,v)}$ to $TI\_DB_u$ if it is not already there. Then, according to the TI-LSA protocol, $N_u$ computes $SUB\_TI^{(u)}_{(u,v)}$. By definition, $l_{(u,v)}$ is part of $SUB\_TI^{(u)}_{(u,v)}$. The LSU-ENTRY(s) of all the links in $SUB\_TI^{(u)}_{(u,v)}$ are flooded by $N_u$ to all its neighboring node, including $N_i$. In either cases, $N_i$ gets the LSU-ENTRY associated with $l_{(u,v)}$.

**Inductive Step.** Consider PPO node $N_i$ $k \geq 1$ hops away from $N_u$. Then, at least one of the links between $N_i$ and $N_v$ has undergone one of the `perf-up`, `link-up`, or `perf-unknown` events. Let $l_{(m,j)}$ be such a link. Assume that $TI\_DB_i = TI_i$, i.e., $N_i$ has received LSU-ENTRY(s) associated with at least $l_{(m,j)}$ and $l_{(u,v)}$. According to the TI-LSA protocol $N_i$ must compute $SUB\_TI^{(i)}_{(m,j)}$. Links $l_{(m,j)}$ and $l_{(u,v)}$ must belong to $SUB\_TI^{(i)}_{(m,j)}$ as well as links along at least one simple path from $N_j$ to $N_u$. Therefore, LSU-ENTRY(s) of all links in $SUB\_TI^{(i)}_{(m,j)}$, and in particular $l_{(m,j)}$ and $l_{(u,v)}$ are stored in LADV, and eventually flooded to all neighboring PPO nodes with an LSU-PACKET. Let $N_s$ be one neighbor PPO node that receives the LSU-PACKET. $N_s$ is $k+1$ hops away from $N_u$. In conclusion, if this case works for $k$ hops, then it must also work for $k+1$ hops. Using the principle of mathematical induction scenario I of Lemma 2 is proved.

*Scenario II*: If $l_{(u,v)}$ once belonged to $TI_i$ but, as a result of a `perf-down` or `link-down` event, it is no longer part of $TI_i$, then $N_i$ must receive an LSU-ENTRY associated with $l_{(u,v)}$. This in turns enables $N_i$ to correctly update $TI\_DB_i$.

*Proof of scenario II*: Scenario II is proved through mathematical induction.

**Initial Step.** PPO nodes $N_i$ which are one hop away from $N_u$ receive the LSU-ENTRY associated with the `perf-down` or `link-down` event of $l_{(u,v)}$. In fact the PPO owner node $N_u$ detects the change in $l_{(u,v)}$ and according to the TI-LSA protocol floods the LSU-ENTRY to its immediate neighbors.

**Inductive Step.** Consider PPO node $N_i$ $k \geq 1$ hops away from $N_u$. Assume that $TI\_DB_i = TI_i$, i.e., $N_i$ has received an LSU-ENTRY associated with $l_{(u,v)}$. According to the TI-LSA protocol $N_i$ must recompute $TI\_DB_i$, store the LSU-ENTRY associated with $l_{(u,v)}$ in LADV, then move the LSU-ENTRY in the LSU-PACKET flooded to neighboring PPO nodes. Let $N_s$ be one neighbor PPO node that receives the LSU-PACKET. $N_s$ is $k+1$ hops away from $N_u$. In conclusion, if this case works for $k$ hops, then it must also work for $k+1$ hops. Using the principle of mathematical induction scenario I of Lemma 2 is proved.

If there is more than one link change, each change can be considered individually. Also note that, since a PPO node takes finite time to compute its TI and propagation and queuing delay in the network are also finite, PPO nodes will receive and flood LSU-ENTRY(s) in finite time. Therefore, using the proof of scenario I and of scenario II Lemma 2 is proved also for the case of multiple link changes.

*3) Proof of Lemma 3:* The proof of Lemma 3 follows easily from Lemma 1 and Lemma 2. Notice that the knowledge of the best performance path is a sufficient but not necessary condition to correctly compute $TI_i$ of PPO node $N_i$. As a matter of fact, knowledge of at least one simple path that can be used to establish a lightpath from PPO node $N_i$ and every other PPO node in $TI_i$ suffices. The knowledge of the best performance path can be used to speed up the decision process on whether a given link should be added to $TI_i$ or not. If the lightpath from $N_i$ to $N_v$ along the best performance path cannot be established (due to low signal quality) then $l_{(u,v)}$ does not belong to $TI_i$ for sure.

*4) Proof of Theorem 1:* Recall that Lemma 2 ensures that $N_i$ receives the LSU-ENTRY(s) about all the links in $TI_i$. Lemma 3 ensures that at some point in time, when an LSU-ENTRY about a link is received by a PPO node, the PPO node is already aware of the path to establish the best performance lightpath via that link. Then, the combination of Lemma 1, Lemma 2, and Lemma 3 proves Theorem 1.

## VI. Complexity of the TI-LSA Protocol

First, the complexity of the TI-LSA protocol is analyzed in terms of number of LSA-ENTRY(s) flooded in the network after link $l_{(u,v)}$ changes status. For the worst case analysis we can assume that, at each PPO node $N_i$, $TI_i$ is same as $G(\mathcal{N},\mathcal{A})$, and $SUB\_TI$ is $TI_i$. The link change can be triggered by a `perf-up`, `link-up`, `perf-unknown` or `perf-down`, `link-down` event. In the case of `perf-up`, `link-up`, or `perf-unknown` event, PPO node $N_i$ checks whether $l_{(u,v)}$ is already in $TI_i$. If $l_{(u,v)}$ is already in $TI_i$, an LSU-ENTRY associated with $l_{(u,v)}$ along with $SUB\_TI^{(i)}_{(u,v)}$ is stored in LADV and then flooded in an LSU-PACKET. The number of LSU-ENTRY(s) in the LSU-PACKET is therefore equal to the number of links in $SUB\_TI^{(i)}_{(u,v)}$, i.e., $|\mathcal{A}|$-number of incoming links of $N_i < |\mathcal{A}|$. In the worst case, the LSU-PACKET is flooded once on every link of the network. Therefore, the total number of LSU-ENTRY(s) flooded in the network is bounded from above by $(|\mathcal{A}| * |\mathcal{A}|) = (|\mathcal{A}|^2)$. In the case of `perf-down`, `link-down` event, PPO node $N_i$ recomputes $TI_i$ and stores the received LSU-ENTRY in LADV for flooding. In the worst case, the LSU-ENTRY is flooded once on each link. The total number of LSU-ENTRY(s) flooded in the network is bounded from above by $(|\mathcal{A}|)$. Considering both cases, the total number of flooded LSU-ENTRY(s) is bounded by $(|\mathcal{A}|^2)$.

The above analysis considers the worst case scenario. In practice, it is expected that at each PPO node $N_i$, $TI_i$ is a subgraph of $G(\mathcal{N},\mathcal{A})$, and $SUB\_TI^{(i)}_{(u,v)}$ is a subgraph of $TI_i$. Let $\hat{TI}$ be largest transparency island in the network in the number of links. The number of LSU-ENTRY(s) flooded in the network is then bounded from above by $(|\hat{TI}|^2)$. When link state information is advertised using OSPF [9] using a single area configuration, the number of LSU-ENTRY(s) flooded in the network is bounded from above by $(|\mathcal{A}|)$. Roughly speaking, the TI-LSA protocol is to be preferred over OSPF in terms of number of LSU-ENTRY(s) flooded in the network when $|\hat{TI}| \leq \sqrt{|\mathcal{A}|}$.

Now, the complexity of the TI-LSA protocol is analyzed in terms of time required to converge to a stable state after link $l_{(u,v)}$ changes status. The analysis is based on the assumption that: $(a)$ the transmission time along every link is the same, and $(b)$ the queuing plus processing time at each PPO node is the same. Whenever a change in the status of link $l_{(u,v)}$ is detected by the PPO owner node $N_u$, $N_u$ floods the LSU-ENTRY associated with $l_{(u,v)}$ on its outgoing links. Based on the above assumption[6], a PPO node $N_i$ first receives this LSU-ENTRY along the shortest path in terms of hops from $N_u$ to $N_i$. Therefore, the number of flooding steps is bounded by the number of hops along the shortest path. In the worst scenario, i.e., a linear topology, this is equal to $|\mathcal{A}|$. Therefore the number of steps required is bounded from above by $(|\mathcal{A}|)$.

The analysis above considered the worst case scenario. Once again, in practice, it is expected that at each PPO node $N_i$, $TI_i$ is a subgraph of $G(\mathcal{N},\mathcal{A})$, and $SUB\_TI^{(i)}_{(u,v)}$ is a subset of $TI_i$. The number of steps to achieve convergence is then bounded from above by $(|\hat{TI}|)$. When link state information is advertised using OSPF [9] using a single area configuration, the number of steps required to achieve convergence is bounded from above by $(|\mathcal{A}|)$. Roughly speaking, the TI-LSA protocol is to be preferred over OSPF in terms of convergence time when $|\hat{TI}| \leq |\mathcal{A}|$.

## VII. Emulation Results

The TI-LSA protocol was implemented and tested on the $\Omega$-$E$ network emulator. The $\Omega$-$E$ is a cluster of Linux based PC's, which can be configured to emulate the exchange of control messages in a desired network topology via Iptables functionality [17]. Each PC (the host PC) can accommodate one or more process, each process emulates one PPO node. Each outgoing (incoming) fiber link of a PPO node is realized as a virtual ethernet interface in the host PC.

The following assumptions are made when running the emulation. Every link in the topology is assumed to have the same length and made of the same fiber type. Each fiber carries one wavelength. It is assumed that the determinant factor for the lightpath performance is the optical power budget. The transmission power level and receiver sensitivity are assumed to be the same at all nodes. With these assumptions the lightpath maximum span is the same for all PPO node pairs and can be simply measured in terms of maximum number of hops (links). The maximum number of hops per lightpath determines the TI size and is varied during the study to assess its effect on the TI-LSA protocol performance.

Hello messages are sent every 5 s. If three consecutive hello messages are lost the corresponding link/node is considered

[6] Subject to $l_{(u,v)}$ belonging to $TI_i$.

to be down. The flooding interval (`lsaFloodIntv` timer defined in Section IV) at each PPO node is set to be equal to 30 s. The starting times for both the hello messages to be sent and the flooding mechanism are randomly chosen in the interval $[0, \texttt{lsaFloodIntv}]$. The random choice avoids synchronization among PPO nodes.

Performance results are collected for both the TI-LSA protocol and the single area LSA protocol (SA-LSA). With SA-LSA all the nodes in the network are set to be in the same area/domain. Results are averaged over four distinct emulation runs. Two network topologies are considered.

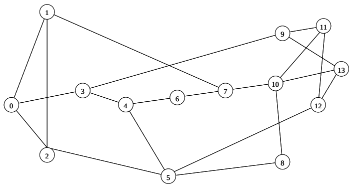

Fig. 8. NSFnet topology

In the first set of experiments, 14 processes, each emulating one PPO node, are connected to form the NSFnet topology (Fig. 8). A total of 42 unidirectional fiber links are created. Fig. 9 reports the total number of LSU-ENTRY(s) flooded during a complete start-up phase (i.e., all PPO nodes are turned on at once) till convergence is reached and every PPO node has completely built its own TI. The number of LSU-ENTRY(s) is shown as a function of the lightpath maximum span (measured in number of hops) — which determines the TI size. The SA-LSA protocol is not affected by the lightpath maximum span as all 14 PPO nodes belong to the same area. In the TI-LSA protocol the number of flooded LSU-ENTRY(s) increases with the lightpath maximum span. The number of LSU-ENTRY(s) grows until the maximum span is large enough to include all nodes and links in every PPO node TI. The curve flattens at that point as, for each PPO node $N_i$, $TI_i$ is $G(\mathcal{N}, \mathcal{A})$ with the exception of the incoming links of PPO node $N_i$. As expected the TI-LSA protocol is effective when the TI size is a fraction of the total network size. When the TI size is large, the number of LSU-ENTRY(s) in the TI-LSA can be significantly larger than the number of entries in the SA-LSA due to the $SUB\_TI$ flooding procedure of the former.

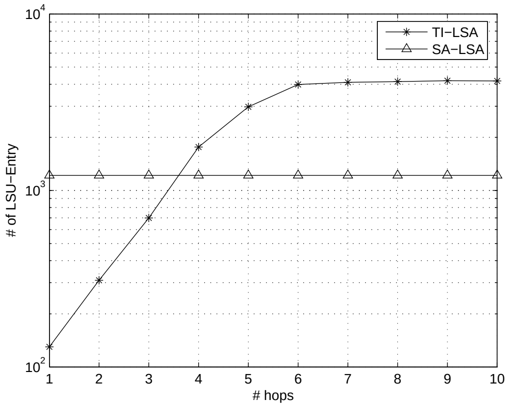

Fig. 9. NSFnet topology: number of LSU-ENTRY(s) vs. lightpath maximum span measured in number of hops

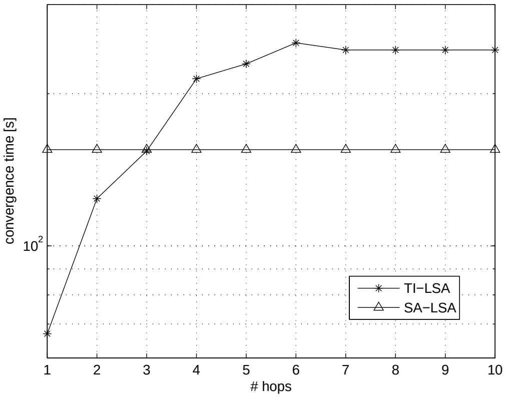

Fig. 10. NSFnet topology: convergence time vs. lightpath maximum span measured in number of hops

Fig. 10 reports the convergence time of both protocols as a function of the lightpath maximum span. This is the time required to transmit all the LSU-ENTRY(s) during the network start-up phase. Once again, the convergence time for the SA-LSA protocol is not affected by the lightpath maximum span. The convergence time in the TI-LSA protocol increases with the TI size, until it stabilizes when the TI size contains all nodes and links. At that point, the convergence time of the TI-LSA protocol is longer than the one of the SA-LSA protocol. This can be explained by the definition of TI (Section III). The TI is built considering lightpaths without loops. Therefore, at PPO node $N_i$, the incoming links $l_{(k,i)}$ do not belong to $TI_i$. This has an effect on the forwarding of LSU-ENTRY(s) in the TI-LSA protocol

In the second set of experiments, 79 processes, each emulating one PPO node, are connected to form the Pan American topology which contains 204 unidirectional links (Fig. 11).

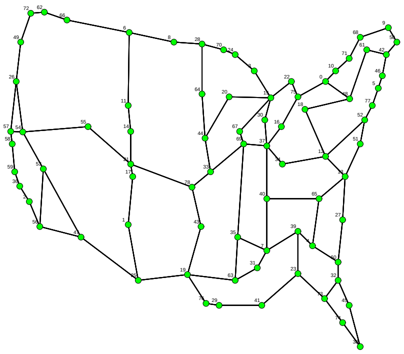

Fig. 11. Pan American topology

Figs. 12 and 13 report the total number of flooded LSU-ENTRY(s) and the convergence time during a complete start-up phase, respectively. Results are reported for both protocols and confirm the earlier claim that the TI-LSA protocol is beneficial when the TI size is small. In which case, the TI-LSA protocol performance is a function of the TI size, but is not affected by the overall network size.

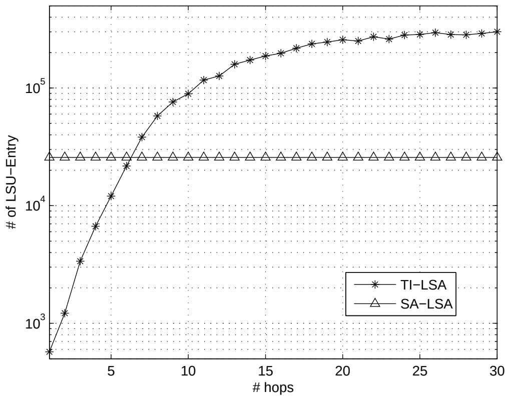

Fig. 12. Pan American topology: number of LSU-ENTRY(s) vs. lightpath maximum span measured in number of hops

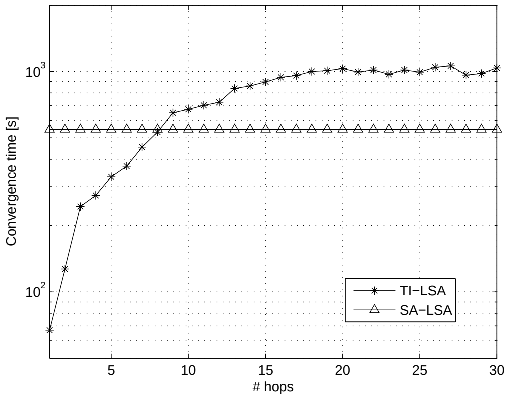

Fig. 13. Pan American topology: convergence time vs. lightpath maximum span measured in number of hops

## VIII. Conclusion

This paper presented an LSA protocol constrained to optical TI(s), which allows PPO nodes to discover resources and detect changes in the optical data plane and physical topology. The TI-LSA protocol is based on the OSPF flooding principle adapted to take advantage of the optical data plane TI(s). The number of flooded TI-LSA entries is limited by constraining the advertisement within the optical reach of the PPO node, i.e., the PPO node TI. In other words, each TI-LSA entry reaches the PPO nodes that require that information, without unnecessarily flooding other PPO nodes that will not make use of that entry. While the TI may resemble the area concept in the Internet, a substantial difference between the two is that areas are manually predetermined, while TI(s) are self detected. The TI(s) may change over time as they automatically adapt to the changes of both the optical network topology and fiber transmission characteristics. And so does the constrained flooding of the TI-LSA entries. While delivering to each PPO node all the TI-LSA entries that may be required to intelligently use the available optical resources, the TI-LSA protocol permits to realize networks with a number of PPO nodes that is virtually infinite, thanks to its TI constrained flooding.

A number of open challenges remain to be addressed and will have to be studied carefully. For example, solutions that provide end-to-end routing across multiple TI(s) must be identified as current inter-area routing solutions cannot work in conjunction with the TI-LSA protocol in a straightforward way.

## Acknowledgment

The authors would like to thank Nabor E. Mendoza-Diaz and Miguel Razo for their valuable technical contribution.

## References

[1] N. M. Froberg, S. R. Henion, H. G. Rao, B. K. Hazzard, S. Parikh, B. R. Romkey, and M. Kuznetsov, “The NGI ONRAMP Test Bed: Reconfigurable WDM Technology for Next Generation Regional Access Networks,” *IEEE/OSA Journal of Lightwave Technology*, vol. 18, no. 12, December 2000.

[2] I. Chlamtac, A. Ganz, and G. Karmi, “Lightpath Communications: a Novel Approach to High Bandwidth Optical WAN-s,” *IEEE Transactions on Communication*, vol. 40, no. 7, July 1992.

[3] C. A. Brackett, “Dense Wavelength Division Networks: Principles and Applications,” *IEEE Journal on Selected Areas in Communications*, vol. 8, August 1989.

[4] I. Cerutti, A. Fumagalli, R. Hui, P. Monti, A. Paradisi, and M. Tacca, “Plug and Play Networking with Optical Nodes,” in *Proceedings of IEEE 2006 International Conference on Transparent Optical Networks*, June 2006.

[5] S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, Eds., *Mobile Ad Hoc Networking*. Piscataway, NJ and New York, NY: IEEE Press and John Wiley & Sons, Inc., April 2004.

[6] P. E. Green, *Fiber Optic Networks*. Prentice-Hall, 1993.

[7] A. A. M. Saleh, “Islands of Transparency - an Emerging Reality in Multiwavelength Optical Networking,” in *IEEE/LEOS Summer Topical Meeting on Broadband Optical Networks and Technologies*, Monterey, CA, July 1998.

[8] H. Zang, J. Jue, and B. Mukherjeee, “A Review of Routing and Wavelength Assignment Approaches for Wavelength-Routed Optical WDM Networks,” *Optical Networks Magazine*, vol. 1, no. 1, January 2000.

[9] J. Moy, “RFC 2328 - OSPF Version 2,” April 1998.

[10] D. Katz, “RFC 3630 - Traffic Engineering (TE) Extensions to OSPF Version 2,” September 2003.

[11] R. Rastogi, Y. Breitbart, M. Garofalakis, and A. Kumar, “Optimal Configuration of OSPF Aggregates,” *IEEE/ACM Transactions on Networking*, vol. 11, no. 2, pp. 181–194, April 2003.

[12] A. Shaikh, D. Wang, G. Li, J. Yates, and C. Kalmanek, “An Efficient Algorithm for OSPF Subnet Aggregation,” in *ICNP ’03: Proceedings of the 11th IEEE International Conference on Network Protocols*. Washington, DC, USA: IEEE Computer Society, 2003, p. 200.

[13] H.-P. Nolting, “All-Optical 3R-Regeneration for Photonic Networks,” in *ONDM 2003: Proceedings of The 7th IFIP Working Conference on Optical Network Design & Modelling*, 2003.

[14] B. Braden, “RFC 2205 - Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification,” September 1997.

[15] D. O. Awduche, “RFC 3209 - RSVP-TE: Extensions to RSVP for LSP Tunnels,” December 2001.

[16] G. P. Agrawal, *Fiber-Optic Communication Systems*. John Wiley & Sons, Inc., 1997.

[17] “The Netfilter.org Project,” http://www.netfilter.org/.