Noname manuscript No.
(will be inserted by the editor)

Layer-Preference Policies in Multi-layer GMPLS Networks

Péter Fodor · Gábor Enyedi · Gábor Rétvári · Tibor Cinkler

Received: date / Accepted: date

Abstract We address the problem of routing Label Switched Paths (LSPs) in multi-layer networks based on the Generalized MultiProtocol Label Switching (GMPLS) paradigm. In particular, we pursue policies for choosing the appropriate layer to host a new LSP request, as we find that such layer-preference policies have significant impact on network performance. We discuss several simple layer-preference policies and we reveal why these simple policies ruin network performance in the long run. Consequently, we develop an efficient heuristics, the *Min-phys-hop* routing and wavelength assignment algorithm, to govern the selection of the best layer of a multi-layer network in which to host new LSP requests. We discuss the applicability of this algorithm with respect to the state-of-the-art GMPLS standards, above all, the GMPLS routing extensions to OSPF-TE. By extensive simulations, we justify that the Min-phys-hop algorithm produces close-to-optimal blocking and resource consumption under almost all possible selections of input parameters, and this is regardless of the wavelength and Optical-Electrical-Optical (OEO) conversion capability present in the network.

Keywords Multi-Layer Traffic Engineering · GMPLS

1 Introduction

The term Generalized MPLS (GMPLS) signifies the architecture, in which a number of switched network transport layers are stacked onto each other and are operated under the authority of a unified control function.

Traditionally, different technological layers of multi-layer networks were operated by isolated control planes with no, or very limited information exchange between the control planes responsible for the different layers. This model is called the *Overlay model*, since the upper layer is simply overlayed on top of the lower layer without the two being aware of each other in any regards. This model was later extended to allow for limited information exchange between control planes. The resultant control architecture is called the *Augmented model*. With the advent of GMPLS, it became possible to completely separate the control plane from the data plane, which opened the way to introduce all the technological layers to under the authority of a unified control plane. This huge integration of vastly different network technologies is made possible by the abstraction of the notion of "labels": basically any quantity of traffic flow that can be differentiated, de-multiplexed and switched individually within the actual network layer is treated as a Label Switched Path (LSP) in GMPLS, like for instance a time slot in a time-division multiplexed infrastructure or a wavelength channel on an optical fiber.

On the one hand, the abstraction of LSPs makes it possible to monitor and control the entire stack of network layers by a common control infrastructure, thus advancing the convergence of new and legacy technologies and the seamless interconnection of heterogeneous networks. From the standpoint of routing and Traffic Engineering, on the other hand, the integrated view of the network (the so called *Peer model*) implies that the routing entity has combined resource and topology information from all the network layers, which facilitates for attaining better network efficiency than it is possi-

Péter Fodor, Gábor Enyedi, Gábor Rétvári, Tibor Cinkler
Department of Telecommunications and Media Informatics,
Budapest University of Technology and Economics,
2 Magyar tudósok körútja, H-1117 Budapest, Hungary
Tel.: +36-1-4634391
Fax: +36-1-4631763
E-mail: (fodorp, enyedi, retvari, cinkler)@tmit.bme.hu

ble under the strict separation of control functionalities, enforced by the conventional overlay model.

It is a design decision made early in the course of defining the GMPLS control plane that the standards suite does not specify explicitly the exact routing algorithm to be used to set up LSPs. The GMPLS standard only describes the environment, the functional model and the mode of operation of a hypothetical GMPLS routing algorithm. Accordingly, in a GMPLS-based multi-layer network architecture the task of the routing algorithm can be posed as follows (the so called *Constraint-based routing* problem): given the virtual graph representing the physical network infrastructure, the already established lower-layer LSPs subject to grooming and the switching capability of the network nodes, find a path for a new LSP request from the given source interface to the given destination interface, subject to a number of operational constraints like, e.g., the type of applicable protection, required bandwidth, etc. Observe that the constraint-based routing problem in multi-layer networks is more complex than in traditional, single layer networks, since it is not confined to the conventional task of finding an appropriate forwarding path that fulfills the constraints, but now it is also up to the routing entity to decide in which layer to serve the request. Since the network consists of two or more technological layers stacked on top of each other, and any technological layer is eligible to host a new request, the routing algorithm has to decide whether to set up a new lower layer LSP, which later can be subjected to host further upper layer LSPs, or to groom the request into a sequence of already existing LSPs (or a mixture of the two options). As shall be shown, the rule for choosing the right layer to set up new LSPs (the so called *layer-preference policy*) has crucial impact on the performance of the network. Therefore, studying and evaluating layer-preference policies, within the technological context defined by the GMPLS paradigm, stands in the focus of main interest in this paper.

After a quick survey of the literature on layer-preference (Section 2), we define two simple layer-preference policies and we discuss the respective pros and cons. We find that neither of these simple policies is adequate to govern layer-preference as they cause adverse performance degradation in the long run. Therefore, in Section 3 we propose a novel heuristic called the *Min-phys-hop* routing and wavelength assignment algorithm. We argue that this algorithm is not only efficient but it is also highly practical, because it readily lends itself to be implemented in contemporary GMPLS networks. Consequently, in Section 4 we discuss the deployability of the Min-phys-hop algorithm, taking into account the conventions imposed by the GMPLS standards and the technological restrictions imposed by operational network devices of our days. To further stress that the proposed algorithm is really viable in practice, in Section 4.1 we sketch two reference scenarios offering a seamless deployment path towards a full-fledged GMPLS-enabled network architecture. The final part of the paper, Section 5, is devoted to evaluate and compare layer-preference policies. We define a unified framework for this purpose and we present the results of comprehensive simulation studies, which provide evidence that layer-preference is a critical factor in the emergent performance of a multi-layer network and that the Min-phys-hop algorithm is fairly efficient in this regard. Finally, we conclude the paper in Section 6.

The GMPLS framework is designed for massively multi-layer networks. Wherever possible, such general networks with more than two layers will be considered throughout the paper. However, since the most popular setup contains only two layers, namely an IP/MPLS layer on top of a (Dense) Wavelength-Division Multiplexing (DWDM) optical infrastructure, sometimes we shall restrict ourselves to this very two-layer setup. When not stated otherwise, the lower layer we shall call the optical layer, its LSPs we shall call lightpaths, and the term LSP will be usually meant to denote IP/MPLS connections.

2 Backgrounds

The GMPLS architecture is described in large detail in [2], [3] and [4]. The signaling framework can be found in [5] while the routing model is described in [6]. Implementation-specific considerations can be found in [7] (particularly concerning OSPF-TE, the Open Shortest Path First routing protocol-Traffic Engineering extensions as described in [8]) and [9] (the same for IS-IS-TE, the Intermediate-System-to-Intermediate-System routing protocol Traffic Engineering extensions).

A good introductory material on routing and wavelength assignment algorithms can be found in [10] and multilayer traffic engineering (MLTE) is reviewed in [11]. Dynamic MLTE schemes are a well-researched area, for a good introduction the reader is referred to [12] and [13]. It must be noted, however, that specifically layer-preference policies, the main question we investigate here, has never been explicitly addressed in the literature, this problem only gets some marginal treatment. In particular, in [1], [14], [15] so called *grooming policies* are identified, which govern the way a layer is selected to host a new LSP. A grooming policy, for instance, would be defined as “insert the LSP to the direct lightpath from the source to the destination if one is available, otherwise establish a new direct lightpath, and if both attempts fail, use a combination of existing and new lightpaths". The problem with grooming policies is that they are rigid in the sense that they are unable to express mixed layer-preference policies, taking into account the implied resource consumption (like "set up a new lightpath of length at most two hops, and if this attempt fails, use a mixture of existing and new lightpaths"). An early attempt to define such mixed-strategies is described in [16]. The most important findings of these works is that although certain layer-preference policies work well under specific circumstances (like in a lightly loaded network or one with unlimited wavelength conversion), there does not seem to exist a universally optimal static policy. This paves the way for a mixed layer-preference policy, as shall be discussed in the next section.

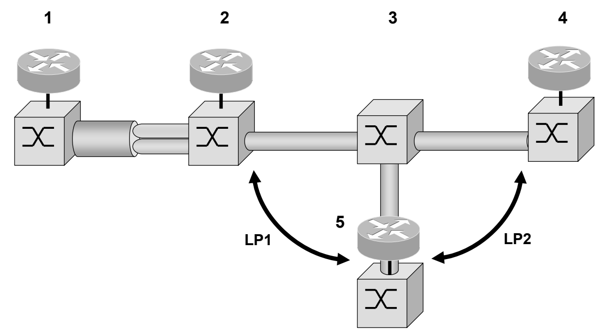


Fig. 1 Virtual graph representation of a sample network scenario. There are two wavelength channels at each link: fat pipes represent optical links and thin pipes within optical links are wavelength channels. There are two lightpaths, LP1 and LP2, both consuming a single wavelength channel on intermediate optical links.

3 Layer-Preference Policies

First, we demonstrate the problem of constraint-based routing in multi-layer networks and the importance of layer-preference through a simple example. Consider the physical network topology depicted in Figure 1. There are four combined IP/(D)WDM router devices (1, 2, 4 and 5) and an Optical-CrossConnect (OXC) node (node 3) in the network. These devices have different switching capabilities: while an OXC device can only switch between optical links, an IP/(D)WDM device can initiate and terminate lightpaths and is able to switch individual wavelengths as well (through multiplexing/demultiplexing wavelength channels through the IP layer). Suppose that, at the point in time when we look at the network, two lightpaths have already been established: LP1 from node 2 to node 5 (via OXC 3) and LP2 from node 5 to node 4 (again via OXC 3), both offering enough capacity to admit a new request. Further suppose that a new GMPLS connection request has to be routed between nodes 1 and 4.

In the traditional overlay model, the IP/MPLS layer (the client layer) requests a new direct lightpath between the source and the destination node from the (D)WDM layer (the server layer) and the connection request is tunneled into the new lightpath. However, allocating a new lightpath is a tedious task and it may very well be impossible, if any of the intermediate nodes run out of spare wavelength channels. In our example, we can only instantiate one connection between node 1 and 4 under the overlay model, since the first connection consumes the remaining spare wavelength channels between node 2 and 4, and there is no room to accommodate the second request along a direct lightpath.

In contrast to the overlay model, the peer model allows for routing a request to a path that is obtained as the concatenation of new and already existing lightpaths. That is, if no direct lightpath can be established we can still route the connection through a newly created lightpath between nodes 1 and 2, plus the concatenation of LP1 and LP2. At intermediate nodes, the GMPLS LSPs are demultiplexed, and the packets of the new request are groomed into the next lightpath. The additional flexibility offered by the peer model over the overlay model usually leads to substantial boost in the profitability and the useful throughput of the optical network [1]. Therefore, we mostly concentrate on the peer model henceforth.

In a GMPLS network using the peer model, the routing entity has integrated knowledge on the configuration and resource availability of all the technological layers that make up the protocol stack, and therefore it has the authority to select the layer at which new requests will be routed at. We call the set of rules governing the choice of the preferred layer in which to accommodate route requests as the *layer-preference policy.*

Given an incoming request, the first task a routing algorithm performs is to check whether there exists a direct lightpath with sufficient capacity between the source and the destination. If such direct lightpath exists, then the new LSP is inserted into this lightpath right away. If no such lightpath exists, however, then one can choose between two simple, contradictory layer-preference policies: *routing in the lower layer* and *routing in the upper layer.* The first policy favors setting up direct low level LSPs for incoming requests, while the latter one prefers grooming upper layer LSPs into lower layer LSPs whenever possible.

In the rest of this section, we discuss these layer-preference policies in more detail. We will show that there is not a single one-fits-all solution and that favoring any single layer of the IP/(D)WDM architecture comes with its very own special advantages and drawbacks. Then, we define a new heuristics, the Min-phys-hop algorithm, aimed at overcoming the limitations of simple layer-preference policies.

3.1 Routing in the Lower Layer

One obvious choice is to push *routing into the lower layer*, that is, to serve a new request in the bottommost layer in the stack that can handle it. In a two-layer IP/(D)WDM network, this would amount to always instantiate a new direct lightpath for a new LSP request, and only attempt to reuse existing lightpaths once setting up a new one fails due to the lack of appropriate resources at the optical layer.

Unfortunately, routing in the lower layer causes the frequent setting up of lower-layer LSPs, which, by nature, tend to have an abundance of capacity but are tedious to establish and tear down, and might consume expensive resources due to frequent optical-electronic-optical (OEO) transformations along the path. More regrettably, however, this policy causes an adverse saturation phenomenon: As the network is filled up with traffic from different source and destination nodes, a huge number of direct but hardly ever used lightpaths will be established. At some point the network runs out of spare wavelength channels, and there remains no other choice to accommodate a new LSP request than to use a lengthy combination of existing lightpaths (since no new ones can be built), which will certainly cause suboptimal routing in the long run. Since this phenomenon shows exceptionally strong resemblance to the fragmentation of memory and disk blocks in computers, we call it *wavelength fragmentation* [17]. For a demonstration of wavelength fragmentation, see the simulation results in Section 5.3.

3.2 Routing in the Upper Layer

A way to avoid wavelength fragmentation is to push *routing into the uppermost layer* possible, that is, to reuse existing lower-layer LSPs to host new upper layer LSPs as long as it is possible, and only apply to lower layers when it is absolutely unavoidable. The problem here is that a lower-layer LSP, represented as a direct link in the virtual graph (a so called TE-link in the GMPLS terminology), does not offer any tangible information for the routing algorithm as to how much real physical resource it uses and what does it cost (in terms of optical transmitters/receivers, electronic resources, etc.) to groom the new LSP into it. This often tricks the traditional shortest path routing algorithm to choose exceedingly long and costly paths, which, when viewed from the physical layer, may even contain *physical level loops*.

To understand the emergence of loops, consider the sample scenario depicted in Figure 1. Here, routing in the upper layer yields that, for a request from node 2 to node 4, the sequence of LP1 and LP2 will be assigned as a forwarding path. However, we observe that the path in the optical layer taken by the packets of this connection contains a loop, since node 3 will be hit twice. Once on the way through LP1 and yet another time on the way through LP2. One of the most important results of this paper is the revelation that such routing loops pose a significant obstacle in optimizing a peer-architecture. We shall point out that routing loops have uniquely adverse effect on the routing performance, even though, as shall be shown below, routing loops are a natural concomitant phenomenon inherent to the peer model (and, in some cases, to the overlay model as well).

Theorem 1 *Consider a peer model virtual topology, in which nodes represent switching devices and links represent either wavelength channels or lower-layer LSPs in the form of TE links. Now, the decision question whether there exists a path that does not contain loop(s) at the lower layer is NP-complete.*

Proof The transformation is straightforward from the *Path with Forbidden Pairs problem* (GT54) [18].

3.3 The Min-phys-hop Algorithm

As it turns out, it is completely hopeless to devise a routing algorithm that can always avoid creating loops. Instead, one must resort to viable heuristics. Therefore, we developed a novel heuristic, which we call the *Min-phys-hop algorithm*. The heuristic is based on the idea that in order to a path to be efficient, it should traverse as few physical nodes as possible. For this, we label each link in the virtual graph that describes the integrated knowledge on the network layer stack by the physical length of the LSP it represents, and choose the least-cost path in the resultant weighted graph. This way, short, direct lightpaths will always be preferred over exceedingly long LSPs and loops are avoided as long as possible. For a formal description of the Min-phys-hop algorithm specialized to two-layer IP-MPLS/(D)WDM networks, see Figure 2.

Consider the network scenario depicted in Figure 1 and suppose that the task is (again) to set up an LSP from node 2 to node 4. Now, using the Min-phys-hop algorithm amounts to set the weight of both LP1 and LP2 to 2 (that is, the number of optical links the lightpath traverses), the weight of all the other links to 1

The Min-phys-hop algorithm

INPUT: A graph $G(V, E)$ describing the peer model of the network, a source node s and a destination node d.
ALGORITHM:

1. Construct the edge weights:

   $\omega_{OP} = 1$

   $\omega_{LP} = \#$phys hops the lightpath traverses

2. Compute the shortest weighted path in $G(V, E)$ over the link weight set defined by ω from s to d.

Fig. 2 The Min-phys-hop routing algorithm

and find the minimum cost path. This means that, in this specific example, a new direct lightpath will be set up from node 2 to 4, which means that no routing loops will be created in physical layer.

4 Deploying the Min-phys-hop routing algorithm in GMPLS networks

The purpose of the Min-phys-hop algorithm is to perform routing and wavelength assignment in multi-layer networks and govern LSP grooming. As such, for it to be really useful in practice, it must fit perfectly into the GMPLS framework. In this section, we show that the Min-phys-hop algorithm lends itself readily to be implemented and deployed in contemporary GMPLS networks.

The GMPLS framework is a remarkably feature-rich one, embracing a vast number of different network technologies, routing models and modes of operation. Before evaluating the applicability of the Min-phys-hop algorithm for GMPLS, we need to review a number of important technological questions.

Although the GMPLS paradigm has been extended recently to be able to handle inter-domain LSPs spanning multiple Autonomous Systems (ASs) [19], below we only concentrate on an intra-domain scenario. We shall assume that the routing entity holds complete and (relatively) up-to-date information on the topology and resource availability in its local AS. This assumption is in line with the rest of the literature and the present state-of-the-art in GMPLS technology.

The two most important routing protocol infrastructures of the GMPLS protocol suite are OSPF-TE-GMPLS (The GMPLS extensions to the Open Shortest Path First routing protocol-Traffic Engineering extensions, [7]) and IS-IS-TE-GMPLS (The GMPLS extensions to the Intermediate-System-to-Intermediate-System routing protocol-Traffic Engineering extensions, [9]). Since the functionality provided by these protocols is more or less identical from the viewpoint of GMPLS, we shall consider the Min-phys-hop algorithm only in terms of OSPF-TE-GMPLS. Naturally, all of our findings are equally valid to IS-IS-TE-GMPLS as well.

The GMPLS standards do not specify the exact location of the routing entity within the network: routing might be distributed amongst cooperating Interior Gateway Protocol (IGP) entities throughout the network, or it might be centralized in the so called Path Computation Elements (PCE, [20]) located anywhere within, or even outside the domain. Below, we shall deal with both of these scenarios. Next, we overview the questions, which help us to answer to how the Min-phys-hop algorithm fits into the GMPLS framework.

The first question we ask is whether the mode of operation of the algorithm fits into that of GMPLS. In the usual context of constraint-based routing, LSP setup requests arrive one-by-one at the routing entity, which then carries out calculations to find an appropriate path, subject to constraints included in the request. This mode of operation is called *on-demand* routing, and it is the default mode of GMPLS routing. The direct opposite is *route precomputation*: here, paths are precomputed for all possible route requests, and incoming requests are served from this precomputed routing table. This is the basic mode of operation of IP networks, and it is expected that some form of route precomputation will find its way into GMPLS networks as well (e.g., to serve the uppermost IP layer). While the Min-phys-hop algorithm perfectly serves the needs of on-demand routing, it is still important to investigate whether it supports precomputation as well, and if yes, then to what extent.

The answer is generally yes, though with limitations. Under the hood, the Min-phys-hop algorithm is nothing more than labeling the edges with the physical length of the underlying objects and performing shortest path computations over the resultant graph. But shortest path algorithms have for long manifested an obvious choice for route precomputation, so for the first sight there does not seem to be any difficulty here. The problem is that in architectures following the peer model, where the entire stack of all network layers is exposed to the routing algorithm, it is allowed to initiate lower layer LSP setups upon servicing an upper layer LSP request. However, this might change the topology of the virtual graph (e.g., in an IP-MPLS over (D)WDM setup, when a lightpath is established, the corresponding wavelength edges should be dropped from the virtual graph), and there is no way to make this change visible to other LSP requests being under precomputation. Therefore, the Min-phys-hop algorithm is only usable for precomputation when the lower layers are not allowed to change during the calculation of the routing table. Such a setup basically accounts for an overlay-modeled network, where there is no integration and sharing of routing information between the layers. As a summary, we can state that the Min-phys-hop algorithm is only usable for route-precomputation in an overlay-based architecture.

For the Min-phys-hop algorithm to be usable in the context of GMPLS, it is essential that the underlying routing protocol machinery, the GMPLS extensions to the OSPF-TE routing protocol in our case, make all the data available that is necessary to execute the algorithm. The most important data needed for Min-phys-hop is – apart from the virtual graph constructed from the Traffic Engineering Database (TED) describing the network and the switching capabilities of network nodes – the length of the lower-layer LSPs in terms of the number of physical hops they traverse.

Unfortunately, as of the present state-of-the-art, neither OSPF-TE nor the GMPLS extensions include the information on the physical length of TE-links in the Link State Advertisements (LSAs) generated to describe these elements. OSPF-TE [8] adds the following set of data to the ones defined in the original OSPF standard to describe TE-links:

- Link type: the type of the link, either point-to-point or multi-access
- Link ID: to uniquely identify the other end of the link
- Local interface IP address: the IP address of the local interface corresponding to this link
- Remote interface IP address: the IP address of the neighbor's interface corresponding to this link
- Traffic engineering metric: link metric for traffic engineering purposes; different than the standard OSPF link metric and assigned by a network administrator
- Maximum bandwidth: maximum bandwidth that can be used on this link
- Maximum reservable bandwidth: maximum bandwidth that may be reserved on this link; may be greater than the maximum bandwidth in which case the link may be oversubscribed
- Unreserved bandwidth: amount of bandwidth not yet reserved on this link
- Administrative group: bit mask assigned corresponding to the administrative group (Class or Color) assigned to the interface

Additionally, the GMPLS extensions for OSPF-TE standard adds some further enhancements to the TE properties of GMPLS TE links. Encoding of this information in OSPF is specified in [7]:

- Support for Unnumbered Links: unique link identifier if the corresponding interfaces do not have separate IP addresses
- Link Protection Type: protection capability for the link (Unprotected, 1+1, 1:1, etc.)
- Shared Risk Link Group Information: unique SRLG identifier(s) describing the SRLG(s) the link belongs to
- Interface Switching Capability Descriptor: to identify the switching, multiplexing and de-multiplexing capabilities of the interfaces connected to the link

Unfortunately, the physical length of the TE-links is absent from the set of properties used to describe a lower-layer LSP in both OSPF-TE and OSPF-TE-GMPLS. Hence, there is no straightforward way to encode this information into the virtual graph and thus the Min-phys-hop algorithm has no ways to differentiate between the resource usage of TE-links. The only possibility is to allocate the OSPF-TE property "Traffic engineering metric" to this purpose. That is, GMPLS Label Switch Routers (LSRs) that originate or terminate a LSP encode the physical length of the LSP in the "Traffic engineering metric" of the LSA generated to describe that LSP. This LSA is then appropriately flooded throughout the network by OSPF-TE, conveying the required information to all LSRs in the domain.

The unique purpose of the Min-phys-hop routing algorithm is to select paths so that the induced usage of the valuable network resources is minimized. This is reflected (as evidenced by the simulation studies discussed later) in the reduction on the number of physical-level loops and in the average length of the paths. However, in a realistic network setting there might arise further requirements and constraints imposed on the returned path, other than simplistic minimization of network resources, including:

- Minimum bandwidth: all links of the path should offer at least the specified amount of bandwidth in the "Unreserved bandwidth" link descriptor
- Maximum acceptable delay
- Class or color: restrict the path to a specific administrative class of links
- Minimum acceptable protection: restrict the path to exclusively to e.g. 1+1 or 1:1 protected links
- Adaptation: LSPs of specific adaptations and payload structures can be requested, like, for example, a VC-3 Synchronous Digital Hierarchy (SDH) circuit
- Interface Switching Capability: since a GMPLS network might span various network layers, it is possible to confine the selected path into a particular network layer

Since any of these constraints might be rightfully imposed either in itself or combined with some other one, it is essential to review how the Min-phys-hop algorithm can handle constraints on the selected paths and how it mixes with traditional constraint-based routing algorithms.

There are in essence two approaches to constraint-based routing. There exist certain constraints that can be satisfied as easily as filtering the links in the virtual graph on which path selection is carried out. For instance, finding a path fulfilling a certain minimum bandwidth requirement can be done by simply removing all links of capacity lower than the requirement from the virtual graph and returning any paths in the pruned virtual graph. Not just that these *bottleneck type* of constraints are easy to handle, but they also mix quite well (meaning that it is straightforward to fulfill two or more bottleneck type constraints at the same time: just filter all the links violating any one of the imposed constraints). Unfortunately, *additive type* of constraints (like e.g., delay or administrative cost) are much harder to consider. These constraints are called additive because the quantity describing a particular path equals the sum of the quantities describing its links. The problem is that additive type of constraints do not mix well: selecting a path subject to two or more additive type constraints at the same time is NP-hard. This means that the Min-phys-hop algorithm (which involves its very own additive metric in the constraint-based routing calculation: the physical length of the TE-links) is not suitable to compute delay-constrained paths, because the number of additive type of metrics to be considered would be two, rendering the path selection problem NP-hard. On the other hand, practically any of the remaining constraints are easy to incorporate into the Min-phys-hop algorithm, like minimum bandwidth, minimum protection type, etc., since these are all bottleneck type constraints.

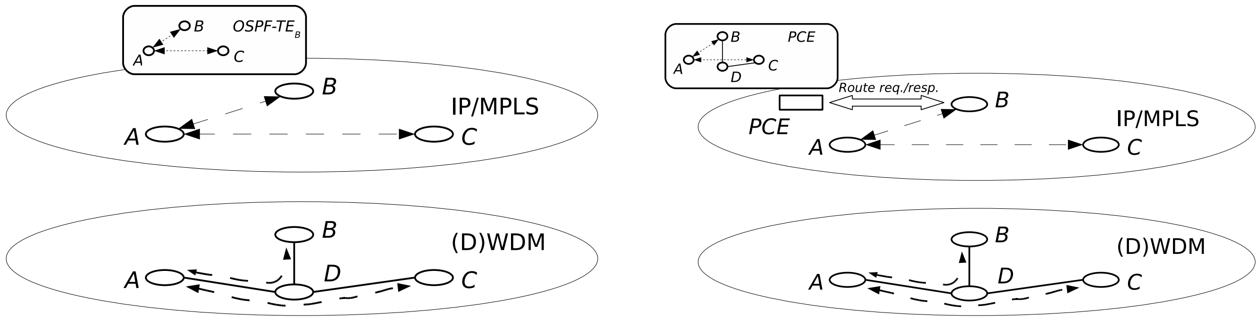
Next, it is important to examine, how the Min-phys-hop algorithm supports networks consisting of more than two layers stacked on top of each other, like e.g., a Packet-Switch Capable layer (e.g., MPLS) on top of a Time-Switch Capable layer (e.g., SDH) on top of a Lambda-Switch Capable layer (e.g., (D)WDM). This is permitted and, to a large extent, fostered by the GMPLS framework. Thus, the question naturally emerges: how does the Min-phys-hop algorithm handle LSP-hierarchies as described in [21]? Since the metric defined by the physical length parameter is stackable – that is, the physical length of a higher-layer LSP is the sum of the physical length of the lower-layer LSPs it consists of, which are again labeled by the sum of the still-lower-layer LSPs –, the Min-phys-hop algorithm correctly generalizes to GMPLS networks incorporating more than 2 switching capabilities. Note that, however, we do not address this scenario in our simulations.

Finally, it is important to examine how the Min-phys-hop algorithm performs in real GMPLS networks. Since we do not have an appropriate-sized GMPLS test bed at our disposal to test the algorithm on, we need to confine ourselves to simulation studies. This is the main topic of the rest of this paper, but first we sketch some likely scenarios in which the Min-phys-hop algorithm may find its use in GMPLS networks.

4.1 Scenarios for deployment

After comprehensive evaluations, it seems that the Min-phys-hop algorithm readily fits into the GMPLS framework. It only uses routing information that is made available by OSPF-TE-GMPLS to it, it supports multiple layers, LSP hierarchies and both on-demand routing and precomputation. Consequently, it seems plausible to consider deploying it in GMPLS networks. Below, we sketch two potential deployment scenarios.

The most likely deployment path towards GMPLS stands in the gradual upgrading of today's IP-MPLS over (D)WDM networks towards a complete GMPLS stack by introducing the (D)WDM layer to under the authority of the unified GMPLS control plane [22]. As the first step of this process, it is expected that an overlay-modeled control architecture is implemented instead of a full-fledged peer architecture. In such a network architecture (see Fig. 3(a)), routing is distributed amongst the IGP entities residing on LSRs across the routing domain. These OSPF-TE-GMPLS protocol entities see an overlay model of the network stack (in which lightpaths from the (D)WDM layer are represented as TE-links but additional (D)WDM network layer infrastructure is invisible) and, based on this virtual graph, precompute a full SPF tree to all IP prefixes available in the domain using the Min-phys-hop algorithm. Once there is no path available to a destination prefix (either because there is no connectivity to the prefix or the capacity at the corresponding lightpaths is exhausted), the management plane solicits the routing entity responsible for the (D)WDM layer to establish a new lightpath towards these prefixes. These requests are served by the conventional routing machinery of the (D)WDM layer. In this model, integrating the legacy networking technologies into GMPLS is done only half-way: the forwarding planes are handled commonly via the notion of abstract GMPLS labels (so the label space is shared), but the routing functionality is still unshared. We see that the Min-phys-hop algorithm fits perfectly into such a scenario, although



(a) In early deployments, routing is distributed amongst OSPF-TE protocol entities seeing an overlay model of the network.

(b) In an advanced GMPLS architecture, a centralized PCE module is responsible for constraint-based routing, which maintains a peer model of the network.

Fig. 3 Deployment scenarios for the Min-phys-hop algorithm in IP-MPLS/(D)WDM networks.

more than two layers, constraint-based routing and full-fledged Traffic Engineering is generally not available.

As the next, long term aim of GMPLS, not just the forwarding plane functionality but routing control as well is expected to become unified. This is possible by building a complete peer model of the entire GMPLS technological stack in which every piece of network element, interface, switching capability and forwarding channel is represented (see Figure 3(b)). Since building and managing such an expensive network representation might impose too much burden on the LSRs, a so called Path Computation Element (PCE) can be installed in the network to carry out routing calculations on behalf of the LSRs in the domain [20]. In a PCE-based architecture, an LSR willing to set up a new LSP makes a routing request to the PCE responsible for the domain. The PCE learns routing information (for instance, through participating in the flooding process of the IGP), builds a Traffic Engineering Database (TED) from the collected information and constructs a graph describing the entire network stack, in which all elements are marked with the amount of free capacity available at that element. Additionally, for the purposes of executing the Min-phys-hop algorithm, TE-links are labeled with the physical length of the underlying LSP or the aggregate length of the underlying LSP hierarchy. This quantity is made available by OSPF-TE-GMPLS in the "Traffic engineering metric" TE-link description attribute. Using the virtual graph, the PCE computes a suitable path subject to the constraints communicated by the initiating LSR to the PCE and returns that path to the LSR. It is also possible to keep the LSR and the PCE synchronized, either to inform the latter of the success or the failure of setting up the LSP or to notify the former on the availability of better paths that might have become usable meanwhile. A PCE based architecture is advantageous, not just because it helps relieve LSRs from the burden of routing, but also because the dedicated hardware of the PCE makes it possible to invoke more sophisticated and more complex routing algorithms, executed in an on-demand fashion. Again, the Min-phys-hop algorithm seems a perfect match to perform constraint-based routing at the PCE.

5 Simulation studies

In the previous section, we concluded that the Min-phys-hop algorithm lends itself readily to be deployed in GMPLS networks. The motivations for doing so are multi-faceted: the algorithm is simple and easy to implement since it is, under the hood, a simple least-cost path algorithm with special link weights. Additionally, the original motivation to develop the Min-phys-hop algorithm was to find a trade-off between two contradictory routing policies on choosing the appropriate layer to host LSP requests, namely, whether to push routing towards the uppermost technological layers in the GMPLS stack or rather to serve requests in the bottommost layer possible. Finding such a trade-off is important, because we found that both of these policies cause adverse phenomena, namely wavelength fragmentation and/or routing loops.

In this section, we support these claims by empirical evidence: we present the results of extensive simulations to demonstrate how those adverse phenomena emerge and that, by using the Min-phys-hop algorithm, we can avoid these pitfalls and generally achieve better network performance. First, we show a simple means to model limited wavelength and optical-electrical-optical (OEO) conversion capabilities in the simulated networks and we discuss a way to represent layer-preference policies, designed to capture all the important characteristics of layer-preference policies discussed earlier in this paper. Then, we turn to discuss the simulation results.

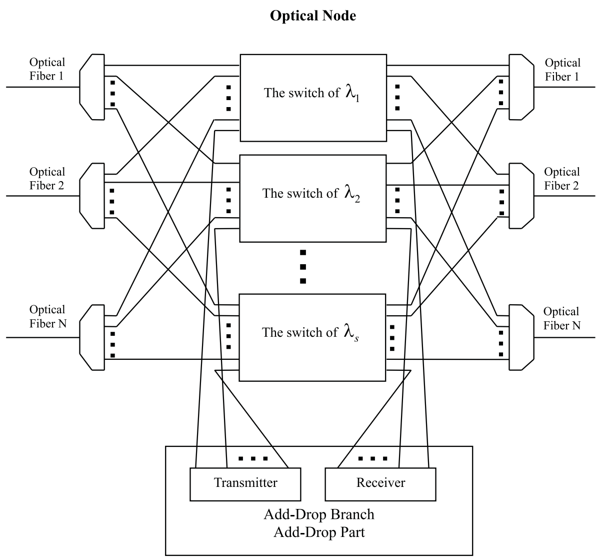


Fig. 4 Typical Optical CrossConnect (OXC) device

5.1 Optical device model

In our simulation studies, we concentrated on two-layer IP-MPLS/(D)WDM networks. The main constituent of such a network is an Optical-CrossConnect (OXC) device. Figure 4 depicts the structure of a typical $N \times N$ OXC of our days. It has N input and N output ports, S wavelengths at each incoming and outgoing fiber and it can switch any particular λ_i wavelength from any incoming port to the same λ_i wavelength on any outgoing port. Additionally, this OXC can drop (and add) exactly S channels by introducing the corresponding wavelengths to optical receivers (transmitters) for further electronic processing. Electronic processing is also the way for wavelength conversion in this device, that is, there is no optical domain wavelength conversion available in the OXC. Also note that a certain λ_i wavelength can be dropped from exactly one incoming port and it is not possible to drop the same λ_i wavelength from two or more incoming ports at the same time. The same applies to adding wavelengths to outgoing ports. This restriction will be important, because our model for the OXC, described in detail in the sequel, is designed deliberately to reflect this type of interior contention of today's OXC devices.

The model we used to represent limited OEO conversion capability is depicted in Figure 5(a). There are N input and N output ports, however, since our graph model is in essence undirected, we did not differentiate between incoming and outgoing interfaces. Additionally, all the S wavelengths at the connected fibers are represented by individual wavelength edges of capacity C_{WL}. The electronic point, which corresponds to the "Add Drop branch" in Figure 4, is represented by the point E and the optical receivers (OE conversion) and optical transmitters (EO conversion) are modeled by capacitated edges from the wavelength edges to the electronic point. The capacity equals $M \times C_{\mathrm{WL}}$, where M manifests restricted OEO conversion capability. For $M = 0$ there is no electronic layer and no OEO conversion, and setting M to infinity means that there is unlimited OEO conversion.

For $M = 1$ the model accurately reflects the OXC device of Figure 4 with all its capabilities and limitations. More specifically, our model correctly encodes the restriction that one particular λ_i can only be dropped (and added) from just one incoming port (to one outgoing port) at the same time. Note also that our model is remarkably flexible in the sense that it is able to express many more optical switching equipments, not just the OXC device above. In particular, Figure 5(b) shows the model of an OXC device without electronic layer (that is, the "Add Drop branch" is absent from the device) and Figure 5(c) depicts the model for an OXC with unlimited wavelength conversion capability in the optical domain.

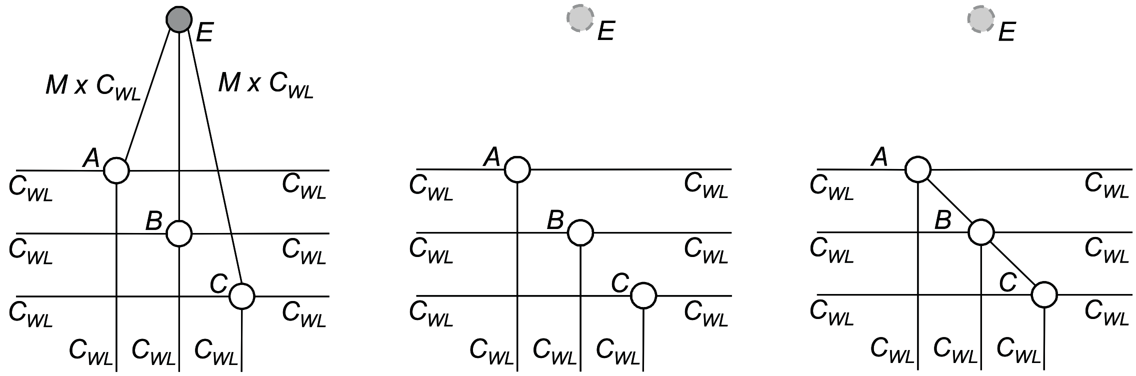
5.2 Representing Layer-Preference Policies

In order to model different layer-preference policies in our simulations, we transform these policies into special rules for setting link weights in the virtual graph. First, we build the virtual graph representation of the integrated IP-MPLS/(D)WDM network topology [23] using the extended OXC-model above and then we use Dijkstra's shortest path routing algorithm to compute paths over specially assigned link weights. In particular, the weights of the lightpath links (ω_{LP}) – these links stand for already established lightpaths – and wavelength links (ω_{OP}) is chosen as 1, and the weight of the rest of the links is set to a very small positive constant. Then, the layer-preference policy is manifested in the course of path selection by rescaling the link weights by a configurable α parameter as follows:

$$\omega_{\mathrm{LP}} \leftarrow \frac{1}{\alpha}\omega_{\mathrm{LP}}$$

$$\omega_{\mathrm{OP}} \leftarrow \frac{1}{1-\alpha}\omega_{\mathrm{OP}}$$

Observe that setting $\alpha = 0$ pushes routing into the lower (D)WDM layer, because the weight of lightpath links is set to infinity in this case. Contrariwise, setting $\alpha = 1$ yields that it is cheaper to accommodate a new LSP on a series of already established lightpaths. Moreover, all other settings of α between 0 and 1 represent different trade-offs between the two layer-preference policies, which is not possible within previ-



(a) OXC node with limited wavelength conversion capability

(b) OXC node without optical transmitters/receivers

(c) OXC node with full optical domain wavelength conversion

Fig. 5 OXC device model with 3 input-output ports and 3 wavelength channels ($N = 3$, $S = 3$)

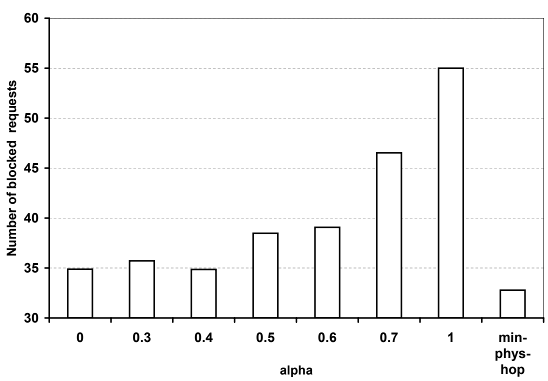


Fig. 6 Number of failed connections for difference setting of α and the Min-phys-hop algorithm..

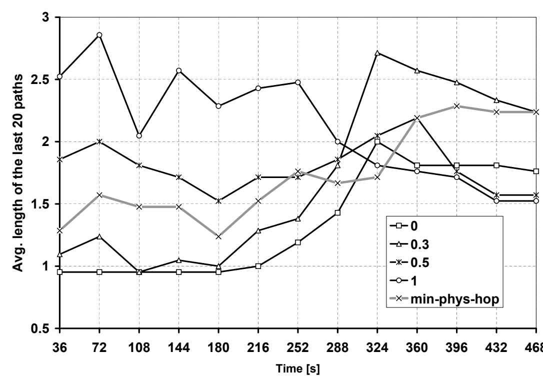


Fig. 7 Average length of the last 20 paths at every time instance, for different settings of α and the Min-phys-hop algorithm.

ous models ([1], [14], [15], [16]). Finally, there remains to implement the Min-phys-hop algorithm in our simulations, but this is easy: simply let $\omega_{\mathrm{OP}} = 1$ and set ω_{LP} to the number of hops the corresponding LSP traverses (see Fig. 2).

5.3 Evaluation of Simple Layer-preference Policies

Below, we demonstrate the adverse phenomena, namely wavelength fragmentation and routing loops, caused by simple layer-preference policies. We also include the Min-phys-hop algorithm in the simulation results to show that this algorithm is immune to such phenomena.

In the course of our simulations, we filled up the KL-network [23] with long lived connection setup requests one-by-one and observed the evolution of the network. The request source and destination pairs were

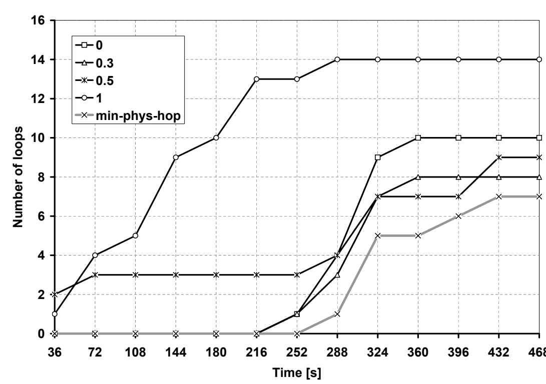


Fig. 8 Number of loops in the physical network for different settings of α and the Min-phys-hop algorithm.

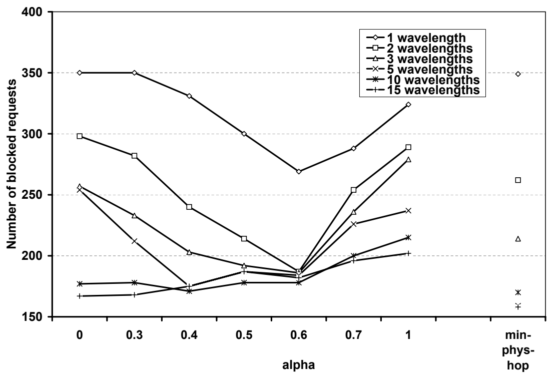


Fig. 9 Number of failed connections under different configurations of wavelength number and wavelength capacity for different settings of α and the Min-phys-hop algorithm.

selected according to independent uniform distributions and the request size was uniformly 1 unit. All optical edges could carry a total number of 3 wavelengths, and each such channel was of 10 units of capacity. For the sake of simplicity, nodes were of unlimited OEO conversion capability. We executed 500 simulation rounds, which was enough to saturate the network.

Figure 6 shows the number of failed connection requests as the function of α at the end of the simulation. The most important observation is that routing in the upper layer ($\alpha = 1$) is inferior to routing in the lower layer in our simulations, as it blocks almost twice as many connection requests. This is clearly attributed to the fact that routing in the upper layer is exceptionally prone to creating routing loops, which yields overly long paths and hence the network gets saturated prematurely. The results in Figure 7 (showing at every time instance the average length of the last 20 selected paths) and Figure 8 (depicting the number of physical-level loops as the function of time) further substantiate this observation.

It seems that, in this specific example, it is much more beneficial to encourage routing in the optical layer. The lower the value of α the lower the average length of the paths and the number of routing loops all the way to the point that with $\alpha = 0$, no loops are created until the 220th time step. However, at this point the vast majority of free wavelengths is used up and a dramatic increase in the path length and the number of loops can be observed. This is because only (often very long) paths through existing lightpaths remain available. This is the phenomenon we called as wavelength fragmentation in Section 3.1. We note that after some peak value the length of the paths begins to drop, because, due to the fragmentation, no paths of continuous wavelength edges can be found and only requests between adjacent nodes succeed.

We observe that the Min-phys-hop algorithm implements a decent trade-off between the two policies. Its reluctance to avoid forming physical-level loops remains unparalleled by the other routing policies. Only when there is practically no loop-free path over the physical network it selects loopy paths. Simultaneously, it creates short paths, which yields fewer blocked calls.

One might argue that our observations are specific to the selected configuration. Therefore, we repeated the simulations for different settings of the number of wavelengths and their capacity (ensuring that the product always remains 30), though, the adverse effects of wavelength fragmentation and routing loops were always clearly manifested. Figure 9 shows the number of blocked calls at the end of the simulations in grand comparison. We observe that if there is only one (but of relatively large capacity) wavelength channel at each link, then it becomes notoriously hard to optimize the network. This is because the virtual topology is adjusted in extremely large steps in this case. Therefore, one might prefer to route the requests in the upper layer and avoid the intervention at the optical layer as much as possible. However, as the number of wavelength channels decreases so the adverse effects of routing loops increases, and routing in the optical layer becomes more beneficial. In almost all cases, the Min-phys-hop algorithm manifests a highly attractive trade-off between the two routing policies.

5.4 Performance evaluation

So far, we have seen that the Min-phys-hop algorithm constitutes an appealing layer-preference policy, with low connection blocking and resource consumption. However, we have restricted ourselves to ideal networks, consisting of optical nodes of unlimited wavelength conversion capability. In this section, we broaden the scope of our simulations: we examine a real network topology consisting of realistic optical devices with limited OEO conversion capability. Our simulations are aimed at measuring the blocking probability (the effective measure of the goodness of the layer-preference policy) and the average path length and number of physical-level loops, as produced by the Min-phys-hop algorithm compared to the entire spectrum of layer-preference policies residing between pushing routing completely into the lower layer ($\alpha = 0$) and the higher layer ($\alpha = 1$).

The parameters of the simulations were chosen as follows: The topology we used was a real network, the

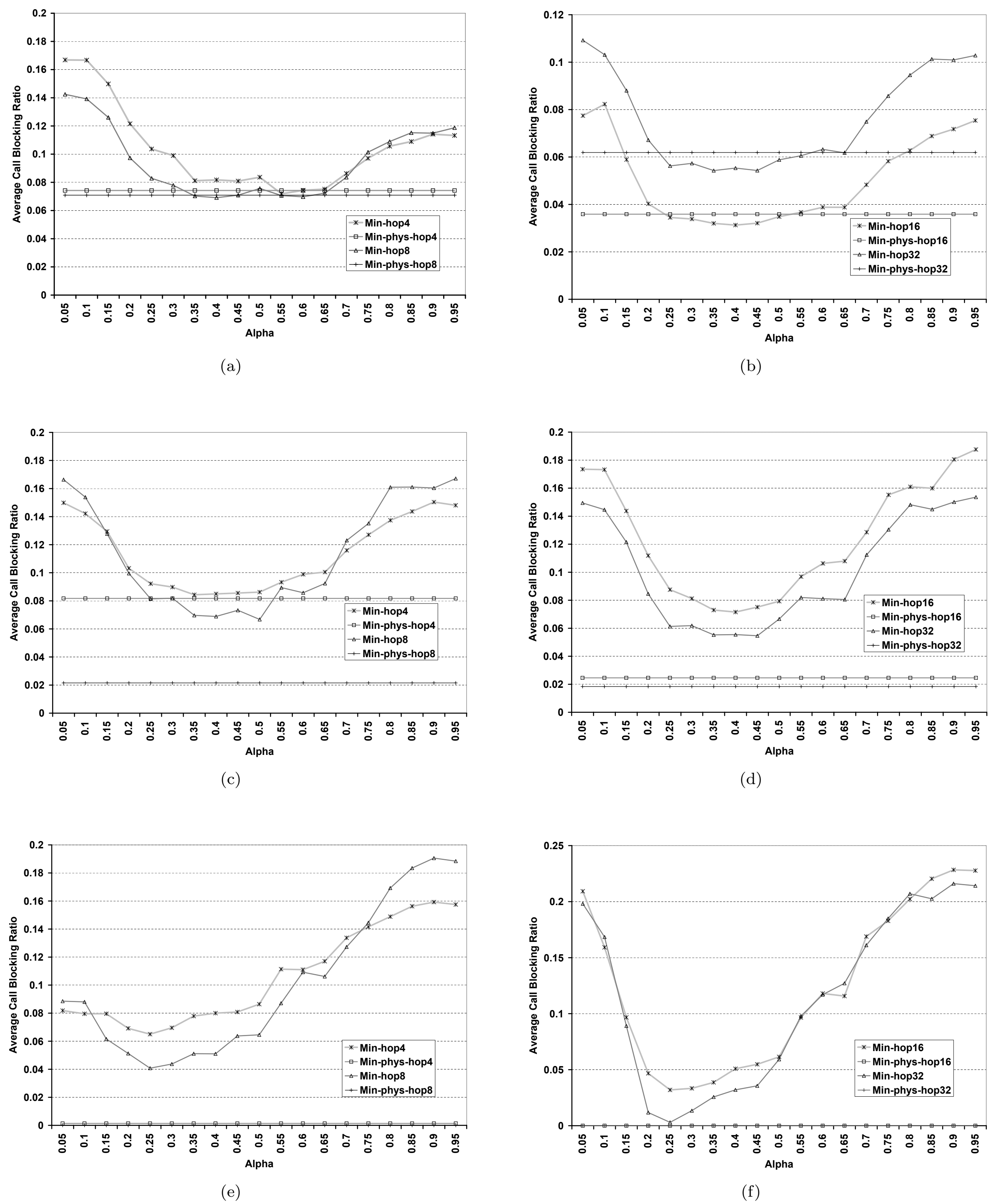
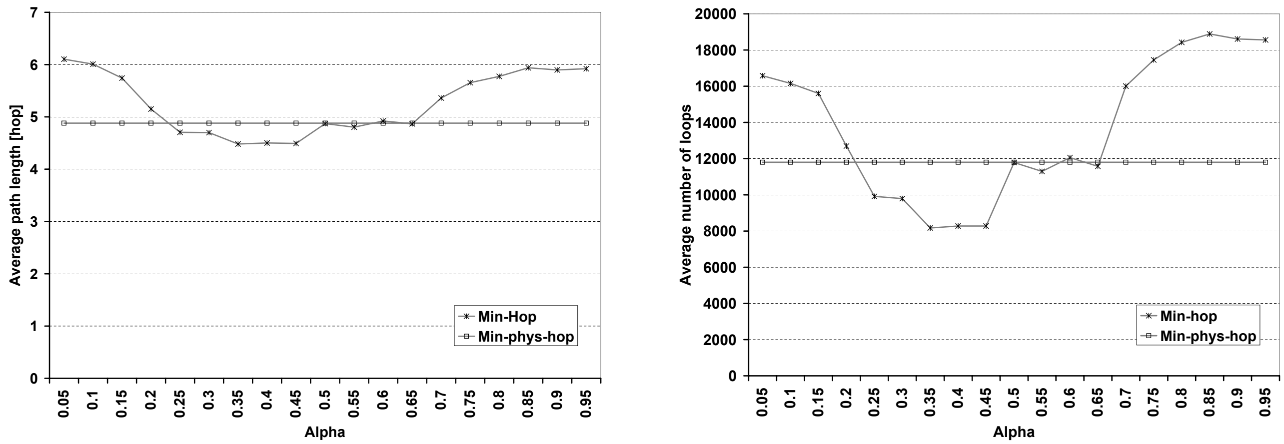


Fig. 10 Average call blocking ratio (CBR) produced by different layer-preference policies and the Min-phys-hop algorithm for various number of wavelengths per optical link, for networks of unlimited wavelength conversion (a), (b) and limited wavelength conversion (c), (d) (for $M = 1$) and (e), (f) (for $M = 2$). The number after the name of the path selection mechanisms represents the number of wavelengths per optical link in the network.



(a) (b)

Fig. 11 Average path length (a) and average number of physical level loops (b) produced by different layer-preference policies and the Min-phys-hop in a network of unlimited wavelength conversion.

28 node European reference network [24]; the number of wavelengths per optical link was varied between 2 and 32 (although, due to space limitations, we do not include all results) and the capacity of wavelength channels was chosen as 100 units. LSP requests were generated one-by-one, while the corresponding source and destination nodes were selected according to a uniform distribution over all pairs of nodes. Requests arrived according to independent Poisson processes for each source-destination pair and holding times were distributed exponentially with an expected value of 10 units. The average request arrival intensity and the bandwidth of wavelength channels were selected so that there are always at least 4 requests alive between a particular source and destination pair at the same time. The average request size was distributed uniformly between 24 and 26 units.

The average ratio of blocked calls over all source-destination pairs for different number of wavelengths per optical link is depicted for unlimited wavelength conversion ($M = \infty$) in Figure 10(a) (for 4 and 8 wavelengths per link) and 10(b) (for 16 and 32 wavelengths per link) and for limited wavelength and OEO conversion in Figure 10(c) and 10(d) (for $M = 1$) and in Figure 10(e) and 10(f) (for $M = 2$). The case when there is no wavelength conversion is left for further study. Note that the request intensities and holding times were scaled measurement by measurement in order to assure that the range of call blocking rates stays sane. Therefore, it does not make sense to compare blocking ratios across simulations for different wavelength numbers or wavelength conversion parameters. The reason for this is that we only wanted to demonstrate that, for any choice of input parameters, the Min-phys-hop algorithm produces acceptable, quasi-optimal blocking ratio and resource usage and, as evidenced by the simulation results, this is exactly the case. Observe that, for a specific combination of wavelength number and M (OEO conversion capability), the Min-phys-hop algorithm usually attains the blocking ratio corresponding to the best choice of the layer-preference policy (that is, the setting of α that produces the minimal blocking ratio), and this is regardless of the OEO conversion capability available in the network. For networks with unlimited wavelength conversion, Min-phys-hop only approximates the optimum, but for limited wavelength conversion, where excessively long paths are even more costly in terms of optical transmitters and receivers, Min-phys-hop even outperforms that. However, it is also educational to observe that there does not seem to exist a universal one-fits-all α parameter, but instead, the best policy depends on the actual parameters of the network.

The diagrams describing the average path length (in Figure 11(a)) and the number of physical-level loops (in Figure 11(b)) demonstrate that not just that the Min-phys-hop algorithm produces low blocking, but it also achieves that near a relatively low resource consumption when compared to other layer-preference policies.

6 Conclusions

In this paper, we have developed the Min-phys-hop routing and wavelength assignment algorithm and examined the practical issues concerning its deployment in GMPLS networks. First, we introduced and analyzed layer-preference policies in multi-layer networks and we identified their respective disadvantages, wavelength fragmentation and the emergence of physical-level routing loops. Based on these observations, we developed the Min-phys-hop algorithm, a heuristic layer-preference policy, and we discussed the aptness of the algorithm to the state-of-the-art GMPLS standards, above all, the GMPLS routing extensions to OSPF-TE. We concluded that the Min-phys-hop algorithm presents itself as a viable choice for routing and wavelength assignment. In order to affirm this claim, we sketched two possible reference deployment scenarios. Finally, we showed that the Min-phys-hop algorithm is not only practical but quite efficient too: we developed a new graph model able to capture all the limitations and restrictions inherent to today's optical switching hardware and, using this model, we conducted comprehensive simulation studies. The results confirm that our algorithm reduces the number of blocked LSP requests and uses network resources more efficiently under almost all possible selections of input parameters, and this is regardless of the wavelength and OEO conversion capability present in the network. We also showed that there does not exist a universally optimal static layer-preference policy and that the Min-phys-hop algorithm realizes an adequate heuristics even considering the realistic limitations of contemporary network devices.

Acknowledgement

The work reported in this paper has been done within the framework of the European FP6 IST Project IP NOBEL II (www.ist-nobel.org)

References

1. K. Zhu, B. Mukherjee, "On-line approaches for provisioning connections of different bandwidth granularities in WDM mesh networks", in Technical Digest, *Optical Fiber Communications Conference (OFC 2002)*, Anaheim, CA, pp. 549-551, March 2002.
2. A. Banerjee, J. Drake, J. Lang, B. Turner, D. Awduche, L. Berger, K. Kompella, and Y. Rekhter, "Generalized multiprotocol label switching: An overview of signaling enhancements and recovery techniques", *IEEE Communications Magazine*, pages 144-151, July 2001.
3. E. Mannie, et al. "Generalized Multi-Protocol Label Switching (GMPLS) Architecture", RFC 3945, October 2004.
4. A. Banerjee, J. Drake, J. Lang, B. Turner, K. Kompella, and Y. Rekhter, "Generalized multiprotocol label switching: An overview of routing and management enhancements", *IEEE Communications Magazine*, pages 144–150, January 2001.
5. L. Berger, Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description", RFC 3471, January 2003.
6. K. Kompella, and Y. Rekhter, Eds., "Routing Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4202, October 2005.
7. K. Kompella, and Y. Rekhter, "OSPF Extensions in Support of Generalized Multi-Protocol Label Switching", RFC 4203, October 2005.
8. D. Katz, K. Kompella, and D. Yeung, "Traffic Engineering Extensions to OSPF Version 2", RFC 3630, September 2003.
9. K. Kompella, Y. Rekhter, Eds., "Intermediate System to Intermediate System (IS-IS) Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4205, October 2005.
10. H. Zang, J. P. Jue, B. Mukherjee, "A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks," Optical Networks Magazine, vol. 1, no. 1, Jan. 2000, pp. 47 - 60.
11. P. Iovanna, R. Sabella, M. Settembre, "A traffic engineering system for multilayer networks based on the GMPLS paradigm", *IEEE Network*, vol. 17, no. 2, pp. 28-37, Mar/Apr 2003.
12. R. Sabella, P. Iovanna, G. Oriolo, P. D'Aprile, "Strategy for Dynamic Routing and Grooming of Data Flows into Lightpaths in New Generation Network Based on the GMPLS Paradigm", *Journal of Photonic Network Communications*, vol. 7, no 2, pp. 131-144, March 2004.
13. H. Zang, J. P. Jue, L. Sahasrabuddhe, R. Ramamurthy, and B. Mukherjee, "Dynamic lightpath establishment in wavelength-routed WDM networks," *IEEE Communications Magazine*, vol. 39, no. 9, Sep. 2001, pp. 100 - 108.
14. H. Zhu, H. Zang, K. Zhu, and B. Mukherjee, "A novel generic graph model for traffic grooming in heterogeneous WDM mesh networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 11, pp. 285-299, April 2003.
15. H. Zhu, H. Zang, K. Zhu, and B. Mukherjee, "Dynamic traffic grooming in WDM mesh networks using a novel graph model," *in GLOBECOM 2002 - IEEE Global Telecommunications Conference*, vol. 21, pp. 2688-2692, Nov 2002.
16. J. Comellas, R. Martinez, J. Prat, V. Sales, and G. Junyent, "Integrated IP/WDM Routing in GMPLS-Based Optical Networks", *in IEEE Network*, vol. 17, no. 2, pp. 22-27, 2003.
17. G. Retvari, P. Fodor, J. Tapolcai, T. Cinkler: "Challenges of Multi-layer Traffic Engineering in GMPLS Networks ", *IEEE ICTON 2005*, 7th International Conference on Transparent Optical Networks, Barcelona, Spain, July 3-7, 2005
18. M. Garey and D. Johnson, "Computers and Intractability A Guide to the Theory of NP-Completeness". WH Freeman & Co, 1979.
19. A. Farrel, Ed., A. Ayyangar, JP. Vasseur, "Inter domain Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS) Traffic Engineering - RSVP-TE extensions", RFC5151, September 2007.
20. A. Farrel, J.-P. Vasseur, J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.
21. K. Kompella, and Y. Rekhter, "Label Switched Paths (LSP) Hierarchy with Generalized Multi-Protocol Label Switching (GMPLS) Traffic Engineering (TE)", RFC 4206, October 2005.
22. K. Shiomoto, Ed., "Framework for MPLS-TE to GMPLS migration", Internet Draft, draft-ietf-ccamp-mpls-gmpls-interwork-fmwk-03.txt, August 2007.
23. M. S. Kodialam and T. V. Lakshman, "Integrated dynamic IP and wavelength routing in IP over WDM networks," in INFOCOM, 2001, pp. 358–366.
24. R. Inkret et al.: *"Advanced Infrastructure for Photonic Networks: Extended Final Report of COST Action 266"*, Faculty of Electrical Engineering and Computing, University of Zagreb, 2003