SPARK: A Smart Parametric Online RWA Algorithm

Francesco Palmieri, Ugo Fiore, and Sergio Ricciardi

Abstract: **The large potential bandwidth available in wavelength-division multiplexed optical networks makes this technology of crucial importance for satisfying the ever increasing capacity requirements of the next-generation Internet. In this scenario, the routing and wavelength assignment (RWA) problem that concerns determining the optical paths and wavelengths to be used for connection establishment in a wavelength-routed network, is still one of the most important open issues. In this paper we propose a new online dynamic grooming-capable RWA heuristic scheme working on wavelength division multiplexing (WDM) networks as a multistage selection process. The proposed algorithm is transparent with respect to the presence of wavelength converters, achieves very low connection rejection ratios with minimal computational complexity and is appropriate for the modern multilayer optical circuit and wavelength switched networks with sparse wavelength conversion capability.**

Index Terms: **Lightpath, optical networks, routing and wavelength assignment (RWA), wavelength division multiplexing (WDM).**

I. INTRODUCTION

We are moving toward a society which requires that we have access to information at our fingertips when, where, and in whatever format we need it. The information is provided to us through our global mesh of communication networks, the Internet, whose current implementation does not have the capacity to support the future demands for higher bandwidth. Fiber-optic and wavelength division multiplexing (WDM) technologies can be considered our best option to achieve fast and reliable communications, due to their potentially limitless capabilities: huge bandwidth (nearly 50 terabits per second), low signal distortion, low power requirement, and low cost. As these technologies are deployed into backbone networks, a new optical routing paradigm, operating at the wavelength layer and based on native per channel switching is rapidly emerging as the candidate control-plane solution for next-generation optical infrastructures. Wavelength routed all-optical networks can be seen as dynamic network infrastructures, where a number of transparent optical paths are established between given source and destination pairs on specific connection request basis. Establishment of such a path consists of choosing a route and wavelengths, providing a logical link between its end nodes. Since wavelengths are a limited resource on each link, the control plane must keep track of the current allocations and judiciously deal with oncoming demands. This is referred to as the routing and wavelength assignment (RWA) problem, whose objective will be to minimize the number of wavelengths used, and to maximize the number of optical paths successfully set up ensuring that no two requests sharing a network link are assigned same wavelength (*clash constraint*). The RWA problem, that can be naturally formulated as an integer linear programming (ILP) problem has been shown to be NP-complete [1]. Furthermore, the complementary problem of routing several lower rate, sub-wavelength, requests packaged onto a single wavelength by using time division multiplexing (TDM) so that the combined traffic rate will be no more than the available wavelength bandwidth, is called dynamic traffic *grooming* and the related optimization problem has also been proved to be NP-hard [2]. The integrated RWA with dynamic traffic grooming brings forth the advantage of having a unified control plane paradigm handling resources both at the IP and the WDM optical layer. Many heuristic methods have been developed to deal in a computationally feasible time with such a challenging problem [2], [3]. In this scenario, we propose a simple, smart, and effective two-stage wavelength routing algorithm based on an on-line dynamic grooming scheme that, for each new connection request, finds a set of feasible lightpaths fulfilling QoS requirements, and bases its final choice on a novel heuristic. We call this algorithm *SPARK*, that stands for smart parametric adaptive RWA algorithm based on K-shortest paths. Our new algorithm, as compared with the most widely known state-of-the-art reference techniques such as minimum hop algorithm (MHA) [4], shortest widest path (SWP) [5], or maximum open capacity allocation (MOCA) [6], [7], that is the classical minimum interference routing paradigm transposed into the optical domain, achieves a significantly lower blocking probability, a better resource utilization, and load balance, and exhibits a lower computational complexity. SPARK can handle converter-free networks as well as networks where some nodes are frequency converters, and can account for all the complexity, performance and resource-limitation constraints implicit in the various flavors of optical switching devices. More specifically, it explicitly penalizes all lightpaths that require wavelength conversion—this may be particularly useful if we have a cost associated with wavelength conversion (for example, if wavelength conversion would occur in the electronic domain, the cost may be very high). Finally, it is based on a totally flexible network model, supporting heterogeneous WDM equipment, in which the number and type of lambdas can vary on each link, and provides a fully dynamic path selection scheme in which the grooming policy is not predetermined but may vary, along with the evolution of the network traffic.

II. BACKGROUND

This section briefly introduces some of the basic concepts that will be useful to better explain the proposed RWA scheme, by presenting its architectural building blocks, ideology and the theory behind it and sketching the current research scenario.

A. RWA in Wavelength-Routed Networks

According to the WDM paradigm, the optical transmission

Manuscript received May 13, 2007.

The authors are with the Federico II University of Napoli, via Cinthia 5, 80126-Napoli, Italy, email: {fpalmier, ufiore, sricciar}@unina.it.

spectrum is carved up into a number of non-overlapping wavelength bands, each supporting a single communication channel operating at whatever protocol or rate one desires (protocol and bit-rate transparency). Thus, by allowing multiple WDM channels to coexist on a single fiber, we can tap huge fiber bandwidth, with the corresponding challenges being the design and development of appropriate network architectures, control plane protocols, and algorithms. Next generation WDM networks will be based on configurable switching nodes, operating transparently at the wavelength layer, to set up and tear down all-optical end-to-end communication channels, called *lightpaths* that can traverse multiple physical links and essentially create a virtual topology on top of the physical topology. Information sent via a lightpath does not require any opto-electronic conversion at intermediate nodes, greatly reducing delay and latency phenomena. In the RWA problem, connection requests can be known a priori (static case), or arrive unexpectedly with unknown holding times (dynamic case). In presence of totally dynamic connection requests, each lightpath must be computed on-line at the arrival of each request based on the current network state, so when coping with this class of problems, we always need special support from control plane protocols to obtain up-to-date information about the global network layout in terms of link status and resource availability. Generally, dynamic on-line RWA algorithms aim to satisfy as many lightpath setup requests as possible, that in other words is minimizing the total connection rejection probability in the entire network. All the heuristic RWA algorithms [8] that operate dynamically must take into account the tradeoff between complexity and performance. In fact, such an adaptive routing behavior requires in any case a relatively longer setup delay and a higher control overhead. Moreover, when we operate transparently in the optical domain, the lightpath selection process in addition to the clash constraint can be subject to the *wavelength continuity* constraints. That is, if no wavelength conversion device is available, the same wavelength must be assigned along the entire route.

B. Wavelength Conversion

Enforcing the wavelength continuity constraint usually results in a high connection rejection ratio, and a common approach to alleviating this constraint is the adoption of wavelength converters. We can define a node with wavelength conversion capability as a node capable of full-range wavelength conversion (a node capable of limited-range wavelength conversion) if a wavelength channel on any input line of the node can be converted to any wavelength channel on any output line (if a wavelength channel on any input line of the node can only be converted to several particular wavelength channels on any output line) [13]. Furthermore, a WDM network is referred to as a network with full wavelength conversion if each node of the network has the capability of wavelength conversion [12]. On the other hand, a WDM network is referred to as a network with sparse wavelength conversion if only a proper sub-set of the network nodes have wavelength conversion capability. Since nowadays the wavelength converters are still expensive devices, it is practically infeasible to equip each network node with a wavelength converter. Therefore, most research efforts focus mainly on the networks with sparse wavelength conversion [12], [14], [15]. In a WDM network with sparse wavelength conversion, the dynamic RWA problem needs deliberate study to take full advantage of the gain from conversion [12], [16], [17]. However, the conventional dynamic routing algorithms may not work well in the environment with sparse or full wavelength conversion [16].

C. Traffic Grooming

Current state-of-the-art optical fibers are capable of carrying tens to hundreds of wavelengths, each, in turn, supporting enormous bandwidth capacity (in the order of 10 to 40 Gbps), while a great deal of network traffic relatively remains at low rate. Consequently, it is imperative to combine through grooming several low-rate requests into high-bandwidth lightpaths to efficiently use the available bandwidth resource and enhance network connectivity. This implies that, in addition to expensive wavelength conversion devices that can operate entirely at the optical layer, a certain number of electro-optical conversion devices are required to support traffic grooming. Strictly speaking, at least two electro-optical conversion devices are needed to perform grooming on a lightpath, namely, one at the source node and another at the destination node, since groomed traffic require a first conversion to be added (by TDM) onto a lightpath at the source node and another conversion to be dropped (by demultiplexing) from the corresponding lightpath at the destination node. Different decisions about what traffic flows should be groomed, and when, reflect different objectives in term of network resource utilization, and are referred to as grooming policies. Clearly, the grooming problem, like the RWA one, can be handled statically or dynamically, and separately or fully integrated to the RWA logic itself. In [9] the static grooming optimization problem is formulated as an ILP for a network where every node has full switching capability (space, wavelength, and time slot) but the grooming fabric is limited. The authors in [10] observe that in a WDM network with sparse conversion capability, high-bandwidth connections are more likely to be blocked than low-bandwidth connections, and propose an admission control scheme to ensure fairness between bandwidth granularity classes in terms of blocking probability. An auxiliary layered graph (one access layer, one lightpath layer, and a wavelength layer for each wavelength) is built in [2] and [11] and shortest-path route computation is applied on the auxiliary graph to determine how a connection request should be served. In such a graph, "vertical" links, connecting different layers, model different operations such as grooming and conversion, and can thus be assigned appropriate weights to model different grooming policies. In the work specifically dedicated to dynamic grooming [11], the same authors evaluate four different grooming policies. These policies mainly concentrate on priorities between the different operations, so that, e.g., the concatenation of multiple existing lightpaths may have precedence over the establishment of a new lightpath between the source and destination nodes. An "adaptive" grooming policy is then presented, integrated in a flexible dynamic RWA scheme, which is the combination of the two best-performing policies among the four analyzed.

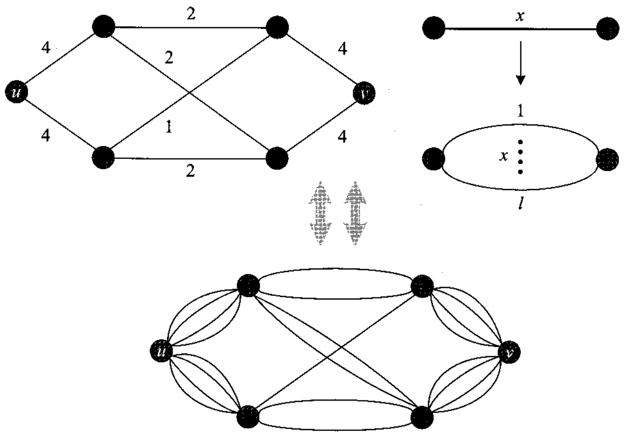


Fig. 1. Generating the working multi-graph.

III. THE INTEGRATED RWA ALGORITHM

Since the integrated RWA problem is NP-complete, and hence the optimal solution cannot be determined in less than exponential times, that for large networks is unacceptable, we need an efficient and effective heuristic solution driving to an almost optimal solution in polynomial times. Accordingly, we propose a very simple algorithm that operates online, running at each request of a dedicated connection with specific QoS requirements (typically bandwidth capacity) between two network nodes s and d. We make the assumption that each connection is bidirectional and consists in a specific set of traffic flows that cannot be split between multiple paths. The connection can be routed on one or more (possibly chained) existing lightpaths between s and d with sufficient available capacity or on a new lightpath dynamically built on the network upon the existing optical links. Grooming decisions are taken instantaneously reflecting an highly adaptive strategy that dynamically tries to fulfill the algorithm's network resource utilization and connection serviceability objectives. The proposed two-stage integrated RWA scheme is structured in a pre-selection phase where a parametric (k) number of candidate paths satisfying the connection constraints are found and a final selection stage where the optimum path among the candidates determined in the previous phase is chosen according to properly crafted heuristics regulated by a second parameter (*kHop*).

A. The Network Model and Initialization Stage

We consider an optical network consisting of n nodes interconnected by l bi-directional optical links (fibers), represented as a multigraph $G = (V, E)$, where each edge corresponds to an individual channel (wavelengths). Each fiber f can support up to λ_f channels, and there can be more than one fiber connecting the same pair of nodes. The multigraph construction process is sketched in Fig. 1 below.

An edge x and its status may be represented by a 6-tuple $(u_x, v_x, f_x, \lambda_x, g_x, r_x)$ where u_x and v_x are the extremes, f_x is an index associated to the physical link or fiber number, λ_x corresponds to the logical channel number or wavelength index on the fiber f_x, g_x, and r_x are respectively the total and the residual wavelength capacity in bandwidth units. Note that our fiber links may support a different number of wavelengths and that wavelengths on different fibers may have diverse maximum capacities g_x, according to the WDM equipment involved. Each node in the network can be a lambda-edge router with several WDM interfaces and electronic wavelength conversion capability or a pure WXC with or without wavelength conversion capability. Nodes are represented by a couple (v, π_v) where v is the vertex number in the graph and π_v is a wavelength conversion capability attribute that indicates the possibility to violate the wavelength continuity constraint by choosing, when building a lightpath traversing v, one of the available connected wavelengths. The conversion capability attribute has been specifically defined to implicitly represent the complexity and expensiveness, in performance terms, of the conversion devices, so that:

$$\pi_v = \begin{cases} 1, & \text{transparent WXC,} \\ 2, & \text{opaque WXC,} \\ 4, & \text{lambda switching router,} \\ 8, & \text{router.} \end{cases} \tag{1}$$

We may weigh edges according to the nodes they are attached to; we use the (2) to take into account the type of conversion device that the edge v links:

$$\Theta(x) = \prod_{v \in V(x)} \pi_v \tag{2}$$

where x is an edge, $V(x)$ is the set of the two nodes that are extremes of the edge x and π_v is the conversion capability of the node v defined in (1).

Each time a new lightpath needs to be established between an ingress-egress pair of nodes in our multi-graph we modify it by removing (actually by zeroing the available capacity) the graph edges traversed by the lightpath (corresponding to the wavelength used) and by adding a direct edge x, called *cut-through* edge, with capacity g_x set to $g_{\min}$ and residual capacity r_x set to $g_{\min} - b$, where b is the fraction of the link bandwidth required by the lightpath and $g_{\min}$ is the minimum global wavelength capacity between all the edges belonging to the lightpath. A *cut-through* edge can be used in any path selection operation and thus can participate in one or more LSPs as a single virtual edge (a single hop at the IP layer) terminated on lambda-edge router nodes, but will not be used when building any other lightpath in the optical domain. When an established lightpath is torn down because the last connection occupying it is ended, the cut-through edge is removed and the edges in the extended graph corresponding to the underlying physical links are set back with full capacity. This schema allows to flexibly model nearly any network topology and to consider node conversion capabilities, wavelength availability and residual bandwidth per logical link at the IP layer. The multiplexing or demultiplexing of several connections at any granularity is possible on grooming-capable switches and on lambda edge nodes, in which wavelength conversion is performed in the electronic domain, whereas pure optical nodes could have the ability to switch at wavelength granularity only. Each new connection request can be routed over a direct lightpath modeled as a single cut-through edge in our multi-graph, or over a sequence of lightpaths (a multi-hop path at the IP level, where each hop can be a lightpath), if it crosses lambda-edge or routers as well.

B. First Stage: Metric and Candidate Paths

The first stage of our algorithm computes a list of k feasible cycle-free paths, in increasing order of cost, between the source and the destination of the connection to be routed, constrained by its QoS requirements. At present, the fastest loop-free K-SPF algorithm has been proposed by Katoh-Ibaraki-Mine [19]. Here k is a configurable parameter that can be used to limit the number of feasible paths that should be considered in the following step, thus controlling the depth of the analysis process according to a performance/precision compromise. The K-SPF algorithm used has been properly modified to meet the specified bandwidth requirements of each new request and to enforce the continuity constraint so that when traversing origin or converter nodes we are totally free in selecting any outgoing edge (i.e., any wavelength), whereas with all the other nodes we can only select an outgoing edge corresponding to the same wavelength associated to the incoming one.

Any source-based routing algorithm uses link weights to compute the best path. The weighting function is of fundamental importance for the overall success of the algorithm. In our proposal, the above pre-selection process is driven by a weighting function w taking into account, for each edge x, the available (dynamic) residual bandwidth r_x, and the (static) global capacity g_x. It is intuitive that a *good* weighting function should be inversely proportional to both the residual and the maximum capacities. However, the two factors do not need to contribute equally. Moreover, we defined four desirable properties as the basic requirements for our weighting function to build an optimal edge metric.

$$\forall x \in E : r_x = 0 \Rightarrow w(x) \approx \infty. \tag{3}$$

That is, the weight associated to an edge with no available residual bandwidth should be as high as possible since a fully saturated edge cannot belong to any new lightpath.

$$\forall x, y \in E : (g_x = g_y) \wedge (r_x > r_y) \Rightarrow w(x) < w(y). \tag{4}$$

Considering two edges with the same global and different residual capacities, the weight associated to the highest loaded one must be higher than the other. This property privileges the choice of edges with highest residual capacity.

$$\forall x, y \in E : (r_x = r_y) \wedge (g_x > g_y) \Rightarrow w(x) < w(y). \tag{5}$$

Given two edges with same residual and different global capacities, the weight associated to the one with highest global capacity must be lower than the other. This is due to the consideration that edges with lower global capacities are more prone to saturation and usually have lower chances to recover bandwidth in time from connection teardown.

$$\begin{gathered}\forall x, y \in E : \\ \left(\tfrac{r_x}{g_x} = \tfrac{r_y}{g_y}\right) \wedge ((g_x \neq g_y) \vee (r_x \neq r_y)) \\ \Rightarrow w(x) \neq w(y).\end{gathered} \tag{6}$$

Every two edges with the same residual/maximum capacity ratio and different residual or global capacity values must have different associated weights. This avoids assigning the same weight to two edges with the same saturation ratio but with different residual bandwidth. Furthermore, each edge is also weighted according to a statically assigned cost c_x that may be used to characterize the QoS properties of different wavelength channels (such as delay, transmission properties, etc.) and/or the transmission impairments introduced by the physical layer. Each edge x is finally weighted according to the function $\Theta(x)$ that takes into account the kind of its extreme nodes, as defined in (1) and (2). Obviously, the weighting function should be inversely proportional to the global and residual capacity of the edge; indeed it should be proportional to the cost c_x and to the function $\Theta(x)$:

$$w(x) \propto \frac{1}{g_x},\ w(x) \propto \frac{1}{r_x},\ w(x) \propto c_x,\ w(x) \propto \Theta(x). \tag{7}$$

Besides, we used the $\log(g_x)$ instead of g_x, in order to lessen its relative importance in the product, as r_x is more important than g_x in assigning weights. We chose the weighting function (9), which satisfies all the four properties and respect the proportionality stated in (7). Starting from the above assumptions, the edge weighting function $w(x)$ is defined as:

$$w : E \rightarrow \Re \tag{8}$$

$$w(x) = \frac{\Theta(x)\ c_x}{r_x \log(g_x)}. \tag{9}$$

Each time a connection is successfully routed in the network, edges' weights are recomputed along the lightpath in order to reflect the decreased residual bandwidth, thus taking into account the *new* network status.

C. Second Stage: Choosing The Best Path

Once the K-SPF algorithm has found, in the first stage, the k least-cost paths satisfying the connection's bandwidth constraints, we have to choose the *best* path among them according to our selection criteria, aiming at minimizing the network's blocking probability, which is the number of rejected connection requests. Let us now examine the path scoring function, needed to differentiate among the available preselected paths. The underlying idea consists in privileging those paths whose edges have more *free* bandwidth than others, also taking into account the length of the paths and the possible optical wavelength conversions and optical-electro-optical (O-E-O) conversions along paths. We assign a "*goodness*" *score* to each of the k shortest paths according to the *scoring* function f_S which evaluates the characteristics of the path p, defined as:

$$f_S : P \rightarrow \Re. \tag{10}$$

The characteristics evaluated by the scoring function for each path p_i, $i = 1, 2, \cdots, k$ are:

- Residual tebandwidth $r_x(p_i)$ of each edge x of the path p_i;
- residual bandwidth $r'_x(p_i)$ that would be left on each edge x of path p_i *if* the request was routed on the path;
- number of hops (i.e., *length*) of the path p_i: $l(p_i)$;
- cumulative weighted number of wavelength and O-E-O conversions along the path p_i: $\xi(p_i)$.

The last factor cumulatively reflects the cost of traversing particular network elements such as optical wavelength converters or lambda-edge routers together with the cost of the conversion operation eventually performed. Its value will be influenced, and proportionally incremented, by each lambda or O-E-O conversion, or device traversal as in the following (10):

$$\xi(p) = \sum_{v \in V_p} \lambda_v(p) \tag{11}$$

where $V(p)$ is the set of nodes traversed in the path p and $\lambda v(p)$ is the weight associated to the possible conversion operation that may be performed along the path p at node v, defined as:

$$\lambda_v(p) = \begin{cases} 1, & \text{no conversion on pure WXC,} \\ 2, & \text{totally optical conversion,} \\ 4, & \text{opaque conversion,} \\ 8, & \text{no conversion on converter node.} \end{cases} \tag{12}$$

This is very useful to discourage, as possible, conversion operations, according to their inherent weight, implicitly enforcing the continuity constraint in the optical core, and also greatly helps in avoiding, the use (or misuse in case of no conversion through a conversion-capable node) of such expensive devices in lightpath determination. Note that only at this stage we know whether a conversion occurs or not, thus (11) is defined and can be evaluated only in the scoring function and not in the edge weighting function (9) where no path has been found yet. Obviously, we impose that f_S depends directly on the residual bandwidths before the connection $r_x(p_i)$ and after the connection $r'_x(p_i)$ and, inversely, on the length of the path and on the weighted number of conversions. So, we can write:

$$\begin{aligned} f_S(p_i) &\propto r_x(p_i) \text{ for each edge } x \text{ of } p_i \\ f_S(p_i) &\propto r'_x(p_i) \text{ for each edge } x \text{ of } p_i \\ f_S(p_i) &\propto \tfrac{1}{l(p_i)} \\ f_S(p_i) &\propto \tfrac{1}{\xi(p_i)}. \end{aligned} \tag{13}$$

First, the scoring function computes for each edge the fraction of remaining free bandwidth after the possible route of the connection request. Then, it computes an arithmetic mean of these percentages between all the edges and weights the result by the conversion factor:

$$\frac{1}{\xi(p_i)\, l(p_i)} \sum_{j=1}^{l(p_i)} \frac{r'_j(p_i)}{r_j(p_i)}. \tag{14}$$

This value represents a first score for the path. Note that (14) takes into account not only each link's residual bandwidth *before* and *after* the connection, but implicitly also the connection's QoS requirement on bandwidth capacity as the implicit difference (and thus the *ratio*) between $r'_x(p_i)$ and $r_x(pi)$. Now, we consider the length of the path. Let $l_{\min}$ be defined as follows:

$$l_{\min} = \min_{i=1,2,\ldots,k} l(p_i). \tag{15}$$

We penalize a path p only if $l(p) > l_{\min}$ and the longer p is, the more it is penalized. The penalty is regulated by means of a tunable parameter *kHop* and it is given by:

$$kHop \left| 1 - \frac{l_{\min}}{l(p_i)} \right|. \tag{16}$$

The higher *kHop* is, the more the path is penalized for its length, making the algorithm behave similarly to MHA. Typical values for *kHop* range between 20 and 60. The quantity (16) is subtracted from the first score of the path (14), thus obtaining the final formulation of the scoring function.

$$f_S(p_i) = \frac{1}{\xi(p_i)\, l(p_i)} \sum_{j=1}^{l(p_i)} \frac{r'_j(p_i)}{r_j(p_i)} - kHop \left| 1 - \frac{l_{\min}}{l(p_i)} \right|. \tag{17}$$

The computation of the scoring function is done for each of the k shortest paths, and the path eventually chosen is the one with the highest goodness score. This is the best path found that will be used to route the connection request. On the edges of this path only, the weight function is recalculated to reflect the decreased residual bandwidth.

IV. PUTTING ALL TOGETHER: THE INTEGRATED RWA ALGORITHM AT WORK

In detail, as a new request arrives, the control plane on each node, starting from the originating one, runs our source-based RWA algorithm and triggers the proper path setup actions:

- It should determine if the request can be routed on one of the available lightpaths, by TDM it with other already established connections, or a new lightpath is needed on the optical transport core to join the terminating (edge) nodes. In presence of multiple options between new feasible and already established lightpats, the cost (9) and path selection (17) functions, together with the grooming and the lambda conversion costs, applied on the existing lightpaths and on the wavelength links that can be used to set up new lightpaths, dynamically determine the least-cost routing for the request, on the current network status basis. For example, if two lightpaths between s and d exist, both with sufficient available capacity, the tie is resolved in favor of the least-cost lightpath. Such a policy guarantees maximum lightpath utilization and automatically achieves, until possible, effective dynamic grooming assuming that the link state database is properly updated. Note that our algorithm has the flexibility to provide parallel lightpath set-up (i.e., two lightpath between the same pair of nodes) if the existing lightpath has overcome the saturation threshold established by the scoring function.
- In any case the source node sends a request along the existing path or the determined new lightpath by using an available signaling scheme;
- all nodes in the path, when they receive the request, run the SPF algorithm, calculate the new network topology and actually establish the requested lightpath and reserve bandwidth resources.

The signaling scheme for triggering a path or lightpath setup and reserving the needed bandwidth, fiber or wavelength resources along the path is very similar to the TE-RSVP protocol [4] used by MPLS. To make a reservation request, the source needs the path and the bandwidth that it is trying to reserve. The request is sent by the source along with path information. At every hop, the node determines if adequate bandwidth is available in the onward link. If the available bandwidth is inadequate, the node rejects the requests and sends a response back to the source. If the bandwidth is available, it is provisionally reserved, and the request packet is forwarded on to the next hop in the path. If the request packet successfully reaches the destination, the destination acknowledges it by sending a reservation packet back along the same path. As each node in the path sees the reservation packet, it confirms the provisional reservation of bandwidth. In addition, it also performs the required configuration needed to support the incoming traffic such as setting up labels in an MPLS label switching node, or reconfiguring the lambda switching internal devices (such as MEMS) in a transparent optical wavelength switching system. In order to accept/reject an incoming request, every node must have knowledge of the available and reserved bandwidth and wavelengths on each outgoing link. This implies that every node needs to run a distributed control-plane protocol that keeps up-to-date information about the complete network topology and available resources. More precisely, a periodic link-state advertisement scheme must convey all the link state information to every node in the network, ensuring the complete synchronization between all the nodes' network status views. Since the amount of per-link state information is very small, any appropriate link state scheme like those employed by OSPF [5] can be adequate for this purpose. However, we point out that the Dijkstra-based path selection scheme, should meet certain conditions:

- A link cannot reserve more traffic than it has capacity for;
- shorter paths are preferred because they consume fewer network resources;
- critical resources, e.g., residual bandwidth in bottleneck links, should be preserved for future demands.

The last two conditions reflect that what we really seek is to keep the connection blocking probability (or in other words the rejection ratio) as low as possible, or equivalently to increase the network utilization. The two-stage algorithm used in our RWA proposal is structured in a pre-selection phase where a parametric (*k*) number of candidate paths satisfying the connection constraints are found and a final selection stage where the optimum path among the candidates determined in the previous phase is chosen according to properly crafted heuristics regulated by a second parameter (*kHop*). It should be noticed that in its behavior SPARK considers several factors of *optimality*:

- The residual bandwidths $r_x(p_i)$ and $r'_x(p_i)$ on each edge x of the path p_i before and after servicing a connection;
- the global capacity $g_x(p_i)$ of each edge x of the path p_i;
- the length of the path p_i, $l(p_i)$;
- the bandwidth request (implicit in the scoring function);
- the number of wavelength conversion and number of optical-electro-optical conversion (cost/minimization of expensive conversion devices);
- the network status and resource usage with implicit knowledge of the connection routed in the network.

These factors are considered at different stages of the algorithm, as they become available in the dynamic RWA environment. The weight function is evaluated *offline* with respect to the connection requests, and thus it takes into account only the *available* information about the residual bandwidth r_x and the global bandwidth g_x of each edge x of the multigraph. As we noticed, every time a new connection is routed in the network, the weight function is obviously recalculated on the only edges of the new lightpath in order to reflect the decreased residual bandwidth. In this way the weight function considers the status of the network, that is the knowledge of each link status (residual and global capacity) and, implicitly, the knowledge of the connections already routed in the network as it knows the residual bandwidth of each edge which is the basic information to know for a connection. As we have seen, when a new connection request arrives, the K-SPF algorithm finds the k least-cost paths according to the weight function (see Section III-B) and the scoring function is calculated on each of the k paths (see Section III-C), considering the *new* information that come with the knowledge of the connection request. The new information known are: The bandwidth request, the list of k feasible paths p_i, and for each of the candidate paths p_i:

- The number of wavelength converters and conversions in the path p_i;
- the number of lambda-edge routers and O-E-O conversions on each path p_i;
- the residual bandwidths $r_x(p_i)$, $r'_x(p_i)$ and length of each path p_i.

Note that the cost function is defined over the set of edges (8), whereas the scoring function is defined over paths (10) and reflects our intent of achieving an almost optimal balancing in the network by privileging short paths offering more free bandwidth. That is why we have to select among the k shortest candidate paths the one with the maximum score according to the function (17), instead of simply selecting *the* shortest path (according to the cost function).

V. COMPLEXITY ANALYSIS

Consider our multigraph with n nodes and m edges. Computing the set of k feasible paths for a specified source–destination pair requires in the worst case $O(k(m + n \log n))$ with the Katoh *et al.* [19] K-SPF algorithm. The second stage computes the scoring function for each of the k paths found. The goodness score calculation involves the computation of the decrease in available bandwidth for each edge in the path. The maximum length of a cycle-free path in a graph with n nodes is $n - 1$, so the second stage requires $O(kn)$. Hence, the worst case runtime is $O(k(m+n \log n))$, since the K-SPF complexity is the dominating factor between the two stages. Such complexity is surprisingly lower that that of the original SPF algorithm, $O(n^2)$, or a little higher than that of the faster SPF improved by using a priority queue with a Fibonacci heap in the implementation [13], i.e., $O(m + n \log n)$, but however significantly lower than that of naïve MOCA, $O(n^3 m \log(n^2/m))$ optimized with the Goldberg max-flow algorithm [20]. Note also that our network representation reduces space complexity, with respect to the layered graph approach conventionally used in dynamic RWA algorithms. Using up to λ wavelengths on each edge, the layered representation with C converter nodes will require $\lambda n + 2$ nodes (λ layers, each dedicated to a single wavelength, plus the two additional nodes to serve as ingress and egress) and $\lambda m + 2\lambda + C(\lambda - 1)$ edges (converters can be modeled by cross-layer edges that connect each layer to the λ-adjacent layer—a

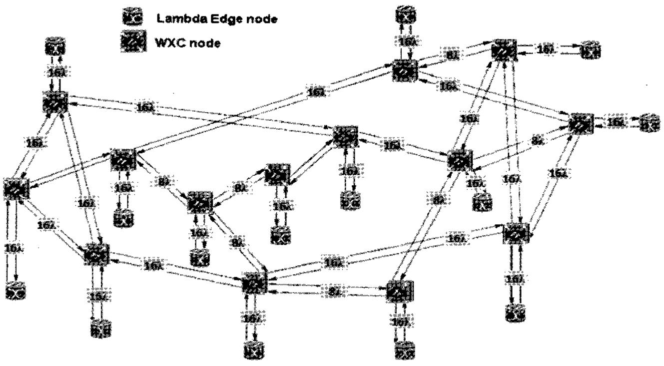


Fig. 2. NSFNet test topology.

Table 1. Simulations performed and parameters used.

—	NSFNet / Geant2	Random generated network topologies
Number of connections	varying from 0 to 10000 (step 1000)	varying from 0 to 10000 (various steps)
Random generated bandwidths (OC-unit)	{1, 3, 12, 24}	{1, 3, 12, 24, 48, 192}
k	varying from 3 to 8	
kHop	varying from 20 to 60	
Number of simulations	each simulation repeated 10 times	

wavelength conversion spanning multiple frequencies will thus entail many such edges in sequence) whereas the equivalent multigraph will require only n nodes and λm edges.

VI. PERFORMANCE EVALUATION AND RESULTS

In order to evaluate the performance of the SPARK algorithm we implemented a simple and flexible discrete-event optical network simulation environment, totally written in Java, supporting SPARK and several other RWA algorithms, such as MHA, SWP, and MOCA, with basic wavelength assignment paradigms such as first-fit and wavelength conversion capability. Those algorithms were selected, because they are well-known reference methods, and the assumptions they impose over the network are compatible with ours—or can be made such with minor modifications—therefore allowing the comparison to be significant.

Simulations have been performed on several topologies, either random-generated or well-known such as NSFNet [21] and Geant2 [22]. These networks have been modeled as non-oriented graphs in which each bidirectional edge has a non-negative capacity, ranging from 1 to 768 OC-units and each optical cross-connect node is attached by one client lambda edge node generating Poisson traffic, as shown in Fig. 2 below sketching the NFSNet test topology. The connection requests have been randomly distributed over all the pairs of edge network nodes. Bandwidth demands have been taken to be uniformly distributed between 1 and 24 optical carrier units in Geant2 and between 1 and 12 units in NSFNet according to the actual bandwidth availability offered by the topologies. We were interested in how algorithms would react to progressive network saturation, therefore our experiments were conducted under the assumption that connections were permanent, so that the blocking ratio raises steadily.

All the results presented have been taken from many simulation runs on the above networks with a number of persistent connection requests varying from 0 to 10000. Simulations have been performed on an HP DL380 dual processor (Intel Xeon 2.5 Ghz) server running FreeBSD 4.10 operating system and Sun Java 1.4.2 runtime environment. The SPARK k and *kHop* parameter values used in our simulations are reported in Table 1.

The proper choice of the k and *kHop* values is an important tuning step for the achievement of good performance, as it can be seen from the results presented in the following. This happens because when the algorithm can select the best one from a larger number of paths for a given source destination pair, the consequential path allocation policy naturally results in a more balanced network usage. On the other side, the k factor affects the execution time (unlike the *kHop* parameter), and consequently has to be chosen as the best possible compromise between execution time and performance. Anyway, in our extensive simulations, we found that quite low values of k are sufficient to get the best results (typical values are $k = 3, 4$). The improvement in the results obtained with greater values does not justify the extra computational effort. Anyway, in general, as the meshing degree increases, and thus more solutions are available, higher k values led to more interesting results. The *kHop* parameter affects the behavior of the scoring function, allowing us to control the choices (and so the performance) of the algorithm. Higher values of *kHop* make the algorithm prefer short paths; lower values make it prefer probably longer but less expensive paths (with respect to the cost-weight mapping done by the weighting function). We can tune the value of the *kHop* parameter for the specific network topology we are working with (it may be sized on the diameter of the network), in order to achieve the best performance. During our experiments, it resulted clear that on some topologies the MHA algorithm behaves quite good: this is the case, for example, of NSFNet (see Fig. 2). On such networks, we may use high values of *kHop* to make SPARK behave more likely MHA, meanwhile preserving the first part of scoring function evaluations (17), and thus obtaining the best results, shown in Fig. 3. Besides, *kHop* may be taken high also if the objective is to get small latency times thanks to a low number of hops of the path. We also observed that if only the shortest path heuristic is employed, such as in MHA, the paths composed of multiple active lightpaths are more likely to be picked. The reason is that the cut through arcs which usually bypass several optical links tend to be shorter. The most significant performance metric observed in our experiments is the path-setup rejection ratio. Clearly, a smaller rejection ratio indicates a better resource usage, and hence a more balanced network utilization in the medium and long term. To gain more confidence in the results, each run has been repeated 10 times and the average performance metric values have been calculated. Compared to MHA, SWP, and MOCA, SPARK performs significantly better both at low and high loads, since it is able to overcome some of their most common drawbacks by taking into account the overall unbalancing and blocking effects (see Figs. 3 and 4). This is due to the increase in network unbalancing since only some

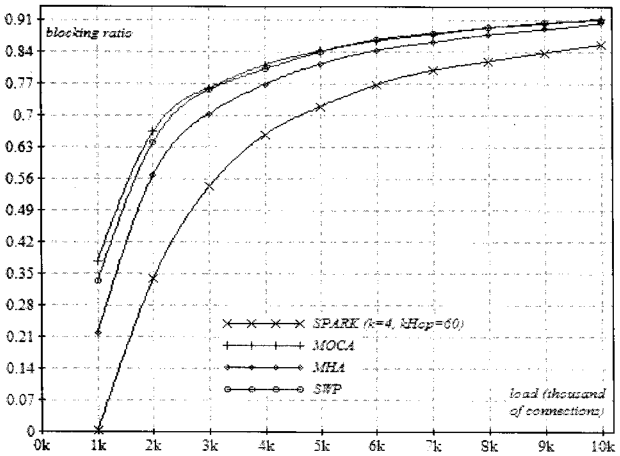


Fig. 3. Average rejection ratio on NSFNet (30 nodes, 43 fibers, 536 edges, 8/16 λ per fiber, and 1/2 fiber per pair of nodes).

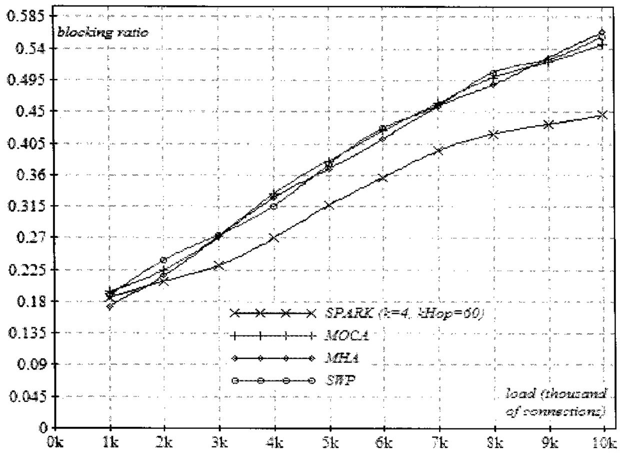


Fig. 4. Average rejection ratio on Geant2 (45 nodes, 97 fibers, 1421 edges, 2/4/8/16 λ-per fiber, and 1/2/3/4 fibers per pair of nodes).

links belonging to the "best" lightpaths are overloaded.

Both in NSFNet and in Geant2 we can clearly see the very interesting results achieved by SPARK. In NSFNet (Fig. 3) SPARK performs much better than other algorithms, exhibiting a very low rejection ratio at all loads. We can also observe the behavior of SPARK and other algorithms when the network approaches its saturation point. In Geant2 (Fig. 4), while the other algorithms stay close to each other along a quite linear trend of blocking ratio growth, SPARK attains lower rejection ratios, globally improving the resource utilization and thus minimizing the overall blocking factor of the network.

Note that the NSFNet topology is quite uniform in the distribution of nodes and link resources, thus the results follow a regular trend; instead, the Geant2 topology is more complex, having a higher meshing degree and more heterogeneous WDM equipment such as links and network elements. The performance gap at high loads is more evident on the Geant2 network, due to its more various network topology that can be exploited by SPARK to achieve better results.

Table 2. Gains of SPARK on NSFNet (simulations from Table 1).

Gains percentage (and average number of extra connections) of* SPARK *with respect to the other algorithms			
connection requests	***MOCA***	***MHA***	***SWP***
1000	99, 2 (373, 6)	98, 5 (215, 2)	99 (330, 2)
2000	48, 9 (650, 2)	40, 4 (459, 3)	47 (602, 8)
3000	28,6 (652, 7)	22, 9 (482, 5)	28, 2 (640, 9)
4000	19 (616, 1)	14, 4 (555)	18 (577)
5000	14, 4 (608, 3)	11, 5 (469, 1)	14, 3 (601, 3)
6000	11, 2 (580, 7)	9 (455, 5)	11, 5 (597, 9)
7000	9 (553, 4)	7, 2 (432, 3)	9, 2 (569, 7)
8000	8, 2 (589, 5)	6, 6 (465, 9)	8, 3 (593, 9)
9000	7, 1 (575, 2)	5, 8 (460, 9)	7, 3 (592, 2)
10000	6, 2 (570, 9)	5, 3 (475, 7)	5, 9 (537)

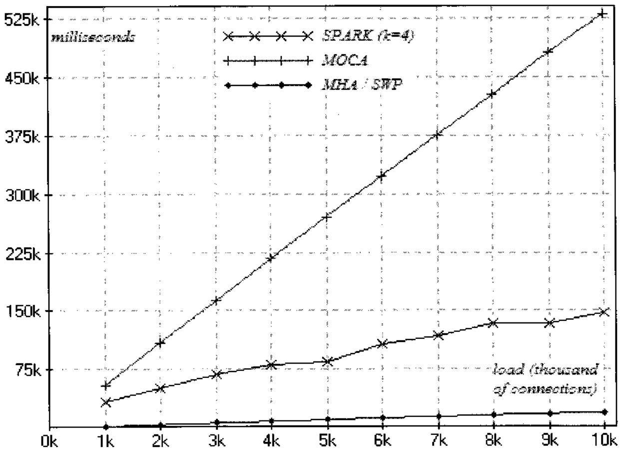


Fig. 5. Average elapsed times on Geant2 (45 nodes, 97 fibers, 1421 edges, 2/4/8/16 λ-per fiber, and 1/2/3/4 fibers per pair of nodes).

Similar results have been also obtained with the random generated networks, in which SPARK has always behaved globally better than the other algorithms. As we can see from table 2, on NSFNet SPARK has a gain in terms of number of connections routed comprised between 5.3% (at 10000 connection requests with more than 475 extra connections routed by SPARK) and 99.2% (at 1000 connection requests with more than 373 extra connections routed by SPARK) with respect to the other algorithms.

Together with the very interesting results in blocking ratios, SPARK operates significantly faster than MOCA, and moderately slower than MHA and SWP as we can see from Fig. 5 where the average elapsed times on Geant2 are shown.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a novel and complete dynamic RWA scheme, called SPARK, easily integratable in state-of-the art routing and signaling protocols and technologies, conceived to work on both pure optical and hybrid electro-optical switching and routing devices, transparently handling grooming of lower rate connections, and capable to operate in presence of wavelength conversion devices. Our new algorithm, despite of its very low computational complexity, achieves a better load balance and results in a significantly lower blocking probability as compared with many widely used routing algorithms, as verified by an extensive simulation study. These features, together with the high operating flexibility and configurability due to its native parametric design, and flexible network modeling framework, make the SPARK algorithm an ideal candidate to be implemented in modern industrial optical control plane frameworks. SPARK not only is a parametric—hence *tunable*—algorithm, but it also represents a reusable structure that can be used to obtain different versions of SPARK itself. There are several respects under which SPARK may be modified for future works and investigations. The weighting function may be adapted to take into account also other features besides global and residual bandwidths, like, for example, the physical length of fibers, the average signal latency time on the fiber or the charge fee for the use of links, thus minimizing other objectives as well as the existing ones. The path scoring function may be further modified in order to give different importance to *critical* links (a critical link may be a link whose residual bandwidth dramatically decreases after the route of the connection request on it), also if SPARK already implicitly handles critical links through its inherent traffic engineering and resources balancing behavior. Future works on SPARK may also concentrate on the tuning of the *kHop* parameter based on the particular network topology the algorithm is working with. Another research line may be making SPARK differentiate nodes that generate more traffic than others (the scoring function may be properly modified to take into account also the ingress and the egress nodes) as they are more likely than other nodes to generate the future connection requests. All these possible future works are quite simple to carry out thanks to the clear, modular and simple structure of both the SPARK algorithm and its main components.

REFERENCES

[1] I. Chlamtac, A. Ganz, and G. Karmi, "Lightpath communications: An approach to high-bandwidth optical WANs," *IEEE Trans. Commun.* vol. 40, pp. 1171–1182, 1992.

[2] H. Zhu, H. Zang, K. Zhu, and B. Mukherjee, "A novel generic graph model for traffic grooming in heterogeneous WDM mesh networks," *IEEE/ACM Trans. Netw.*, vol. 11, no.2, pp. 285–299, 2003.

[3] X. Niu, W. D. Zhong, G. Shen, and T. H. Cheng, "Connection establishment of label switched paths in IP/MPLS over optical networks," *Photon. Netw. Commun.*, vol. 6, pp. 33–41, 2003.

[4] D. Awduche, L. Berger, D. Gain, T. Li, V. Srinivasan, and G. Swallow, "Extensions to RSVP for LSP tunnels," in Internet Draft draft-ietf-mpls-rsvp-lsp-tunnel-04.txt, 1999.

[5] R. Guerin, D. Williams, and A. Orda "QoS routing mechanisms and OSPF extensions," in *Proc. IEEE GLOBECOM*, 1997.

[6] K. Kar, M. Kodialam, and T. V. Lakshman, "Minimum interference routing of bandwidth guaranteed tunnels with MPLS traffic engineering applications," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 12, pp. 921–940, 2000.

[7] K. Kar, M. Kodialam, and T. V. Lakshman, "Integrated dynamic IP and wavelength routing in IP over WDM networks," in *Proc. IEEE INFOCOM*, 2001.

[8] H. Zang, J. P. Jue, and B. Mukherjee, "A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks," *Opt. Netw. Mag.*, vol. 1, pp. 47–63, 2000.

[9] K. Zhu and B. Mukherjee, "Traffic grooming in an optical WDM mesh network," *IEEE J. Sel. Areas Commun.*, vol.20, pp. 122–133, 2002.

[10] S. Thaigaran and A. Somani, "Capacity fairness of WDM networks with grooming capabilities," *SPIE Opt. Netw. Mag.*, vol. 2, pp. 24–31, 2001

[11] H. Zhu, H. Zang, K. Zhu, and B. Mukherjee, "Dynamic traffic grooming in WDM mesh networks using a novel graph model," *SPIE Opt. Netw. Mag.*, vol. 4, no. 3, pp. 65–75, 2003.

[12] S. Subramaniam, M. Azizoglu, and A. K. Somani, "All-optical networks with sparse wavelength conversion," *IEEE/ACM Trans. Netw.*, vol. 4, no. 4, pp. 544–557, 1996.

[13] S. Ramamurthy and B. Mukherjee, "Fixed-alternate routing and wavelength conversion in wavelength-routed optical networks," *IEEE/ACM Trans. Netw.*, vol. 10, no. 2, pp. 351–367, 2002.

[14] J. Kleimberg and E. Kumar, "Wavelength conversion in optical networks," in *Proc. the tenth Annual ACM-SIAM Symp. on Discrete Algorithms*, 1999, pp. 566–575.

[15] B. Ramamurthy and B. Mukherjee, "Wavelength conversion in WDM networking," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 7, 1998.

[16] X. Chu, B. Li, and Z. Zhang, "A dynamic RWA algorithm in a wavelength-routed all-optical network with wavelength converters," in *Proc. INFOCOM*, 2003.

[17] E. Kaeasan and E. Ayanoglu, "Effects of wavelength routing and selection algorithms on wavelength conversion gain in WDM optical networks," *IEEE/ACM Trans. Netw.*, vol.6, no.2, pp. 186–196, 1998.

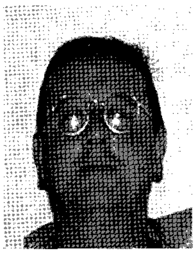
[18] H. R. Lewis and L. Denenberg, *Data Structures and Their Algorithms.* HarperCollins, New York, 1991.

[19] N. Katoh, T. Ibaraki, and H. Mine, "An efficient algorithm for *k* shortest simple paths," *Netw.*, vo. 12, pp. 411–427, 1982.

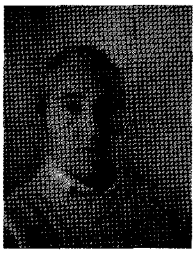
[20] A. V. Goldberg and R. E. Tarjan, "A new approach to the maximum flow problem," *J. ACM*, vol. 35, no. 4, 1988.

[21] NSFNet. [Online]. Available: http://www.nsf.gov/about/history/nsf0050/internet/launch.htm

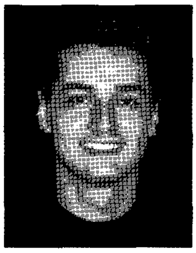
[22] The GEANT2 network. [Oline]. Available: http://www.geant2.net/



Francesco Palmieri holds two Computer Science degrees from Salerno University, Italy. Since 1989, he has worked for several international companies on a variety of networking-related projects, concerned with nation-wide communication systems, network management, transport protocols, and IP networking. Since 1997 he leads the network management/operation centre of the Federico II University, in Napoli, Italy. He has been closely involved with the development of the Internet in Italy in the last years, particularly within the academic and research sector, as a member of the Technical Scientific Committee and of the Computer Emergency Response Team of the Italian Academic and Research Network GARR. He is an active researcher in the fields of high performance/evolutionary networking and network security. He has published several papers in leading technical journals and conferences and given invited talks and keynote speeches.



Ugo Fiore has been with Italian National Council for Research at the beginning of his career. He has been working for more than 10 years in the industry, developing software support systems for telco operators. He is currently with the network management/operation centre of the Federico II University, in Napoli, Italy. His research interests focus on optimization techniques and algorithms aiming at improving the performance of high-speed core networks. He is also actively investigating security-related algorithms and protocols.



Sergio Ricciardi received the degree cum laude in Computer Science from the University of Naples Federico II, Italy, in 2006 and he is information engineer since 2007. His current activities concern wired and wireless networks, grid computing, information, and communication technologies. His research interests are mainly focused on constraint-based routing and topology management for optical networks.