

A Dynamic Bulk Provisioning Framework for Concurrent Optimization in PCE-Based WDM Networks

Jawwad Ahmed, Cicek Cavdar, *Member, IEEE*, Paolo Monti, *Member, IEEE*, and Lena Wosinska, *Senior Member, IEEE*

Abstract—A centralized network control and management plane, such as the one based on a path computation element (PCE), is highly beneficial in terms of resource optimization in wavelength division multiplexing optical networks. Benefits of centralized provisioning are even more evident when connection requests are provisioned in *batches*, i.e., they allow a better use of network resources via concurrent optimization. In this study, a dynamic *bulk* provisioning framework is presented with the objective of optimizing the use of network resources that also presents, as an additional benefit, the ability to yield a reduction of the control plane overhead. The rationale behind the proposed framework is based on a mechanism in which the PCE client is allowed to bundle and simultaneously send multiple labeled switch path (LSP) requests to the PCE where, in turn, several bundles can be concurrently processed together as a single bulk. From the network deployment perspective, a PCE-based network architecture is proposed to practically realize this approach. For dynamic bulk provisioning of optical LSP requests, a time-efficient integer linear programming (ILP) model (LSP_BP_ILP) is presented to minimize the request blocking, the network resource consumption, and the network congestion. In addition, a heuristic based on a greedy randomized adaptive search procedure (GRASP), namely LSP_BP_GRASP, is also proposed as a scalable alternative. The presented results demonstrate significant advantages of the proposed PCE bulk provisioning framework based on concurrent optimization in terms of reduced blocking probability and control overhead when compared with conventional dynamic connection provisioning approaches processing a single connection request at a time.

Index Terms—Bulk provisioning, concurrent optimization, dynamic provisioning, generalized multiprotocol label switching (GMPLS), greedy randomized adaptive search procedure (GRASP), integer linear programming (ILP), path computational element (PCE), wavelength division multiplexing (WDM).

I. INTRODUCTION

MANY of the next-generation network (NGN) applications [1] (e.g., grid, cloud computing, and e-science) require high bandwidth pipes, in many cases corresponding to the bandwidth of a whole wavelength channel in all-optical

networks, to be available in a dynamic fashion between the end nodes. In all-optical networks, generalized multiprotocol label switching (GMPLS) [2] and automatically switched optical network (ASON) [3] are the dominant standards for an automated end-to-end connection establishment. Given the bandwidth hungry nature of NGN applications, these connections require the capacity of an entire wavelength along the path (referred to as a lightpath) and they are provisioned in the form of optical labeled switch paths (optical LSPs) [2]. In the remainder of this paper, use of the term LSP always refers a lightpath (i.e., an optical LSP) unless stated otherwise. On the other hand, this study is also applicable to multiprotocol label switching (MPLS) where LSPs can be of subwavelength granularity. Both ASON and GMPLS-based control planes natively support distributed path computation at ingress nodes [4]. This distributed philosophy, although good in terms of network scalability, can lead to a suboptimal routing solution [5], and may not be able to ensure strict QoS requirements for the provisioned requests. This is the reason why the use of a centralized management and control might be preferable in order to offer resource optimization in networks supporting NGN applications. In such a scenario, one option is to have path computation operations performed at a centralized entity such as a path computation element (PCE) [6], followed by the reservation of resources via a signaling protocol, i.e., resource reservation protocol with traffic engineering extensions (RSVP-TE).

There are a number of choices for the implementation of a PCE-based approach. For example, the work in [7] considers an architecture where PCE and network management system (NMS) are colocated and the NMS is responsible for relaying the communication between ingress nodes and PCE. In this case, the ingress nodes do not need to have a working path computation client (PCC) module. A different PCE-based architecture is considered in [8] where each ingress node runs a PCC module for direct interaction with the PCE, mainly for path computation purposes, via the path computation element communication protocol (PCEP) protocol. This eliminates one extra layer of communication, with still all services and QoS policies deployed at the PCE (i.e., the NMS can be a part of PCE).

Most of the existing research work on PCE is validated using either static or incremental traffic conditions [5], [7], [9], [10]. In static traffic conditions, traffic demands are known in advance based on a forecast. In the incremental case, on the other hand, connection request arrivals might be stochastic in nature, the same as in a dynamic provisioning scenario, but after being set up an established connection stays in the network indefinitely,

Manuscript received December 02, 2011; revised February 10, 2012; accepted March 11, 2012. Date of publication April 18, 2012; date of current version May 16, 2012.

The authors are with the School of Information and Communication Technology (ICT), Royal Institute of Technology (KTH), SE-100 44 Stockholm, Sweden (e-mail: jawwad@kth.se; cavdar@kth.se; pmonti@kth.se; wosinska@kth.se).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JLT.2012.2195296

i.e., it is never terminated. The research work on PCE can also be differentiated based on the *traffic models*. For all traffic conditions, two groups of traffic models can be defined: *scheduled* and *unscheduled*. In a scheduled traffic model, the holding time information is assumed to be known, whereas it is not known (or not considered) in an unscheduled one. In a scheduled traffic model, setup and tear-down times of a connection request are allowed to slide in a larger time window [11]. For this reason, they are called sliding scheduled traffic demands. However, most of the works on sliding window traffic consider only static traffic conditions [12]–[14]. Moreover, none of these approaches looks into the possible research challenges in the control and management plane architectures. In this study, we propose a framework for a centralized architecture of network control and management, suitable for dynamic traffic conditions and flexible setup times which also fall in the category of sliding window-based traffic models.

The benefits of a centralized approach are particularly evident when the connection requests are provisioned in batches, i.e., a better usage of network resources and an enhanced QoS control [9], [10]. However, there is little research on batch provisioning under dynamic traffic conditions, in particular in a PCE-based WDM environment. In [9], an architecture that is suitable for concurrent provisioning of connection requests for grid services is presented. In [10], the authors propose a flexible architecture for a PCE-based network, which supports batch provisioning of grid services focusing on the reduction of the LSP setup time (of packet layer LSPs) and on the optimization of resources. However, both studies consider incremental traffic conditions only, and a single aggregation level of connection requests is used for the batch provisioning.

The works presented in [8] consider a dynamic traffic model and addresses a PCE-based scenario where LSP request aggregation takes place at the ingress nodes to form *bundles*. The focus is to reduce the control plane overhead of the PCEP protocol by bundling connection requests. In [15], this idea is further enhanced by employing two levels of aggregation for the LSP requests. One level is at the ingress nodes, where LSP requests are aggregated in bundles, and the other at the PCE, where multiple bundles originating at different nodes are used to form *bulks*.

In this paper, a novel provisioning framework for a PCE-based network architecture along with an appropriate signaling for dynamic bulk provisioning of LSP requests is proposed for the first time. The contribution of the presented study is threefold. First, a flexible bulk provisioning policy is considered for resource optimization using a time-efficient ILP model. In order to be used under dynamic traffic conditions, this model not only minimizes the resource usage (as it is the case for the conventional, i.e., static, ILP formulations) but also tries to minimize directly the number of rejected connection requests, i.e., blocking probability, while trying to avoid resource congestion for the possible future connection arrivals that are not known in advance. This is done to reduce the probability that these future connections will be rejected/blocked, a problem that is peculiar of the dynamic provisioning paradigm. Along with the ILP model, this paper also proposes an alternative and scalable routing and wavelength assignment (RWA) metaheuristic based on greedy randomized adaptive search procedure (GRASP) [16]. The choice of the GRASP

metaheuristic was motivated by the combinatorial nature of the bulk provisioning problem. The second contribution of this study is the setup time flexibility. In the proposed framework, the time window for the setup time of a connection request is flexible and can slide based on the waiting time values that can be set at both the ingress node and at the PCE. Finally, as a third contribution, this paper presents a realistic modeling of the proposed PCE-based network architecture and associated key signaling protocols (i.e., PCEP and RSVP-TE). The values of performance metrics such as RWA computation time and LSP setup time are provided to give a better understanding of the tradeoffs involved in terms of time delays and performance. To the best of our knowledge, signaling overhead of the PCEP protocol has not been quantified in any other study related to PCE-based bulk/batch provisioning.

The presented results, based on extensive simulations using a continental optical backbone network, highlight the benefits of the proposed provisioning framework in terms of reduction of both the connection blocking probability and the control plane overhead, when compared to conventional, i.e., sequential, provisioning approaches. Both these advantages come at the cost of an increased LPS setup time, but this tradeoff can be controlled by a careful choice of the bundling and bulking parameters. As the presented results show, the proposed framework is flexible enough so that a user not so concerned with the reduction of the control plane overhead can still obtain good blocking performance at the expense of a small setup time increase.

The outline of this paper is the following. The bulk provisioning problem is formally defined, and a suitable PCE-based network architecture to support this approach is proposed in Section II. In Section III, an efficient ILP model, i.e., LSP_BP_ILP, is provided for optical LSP requests provisioning in a dynamic PCE-based networking scenario. In addition, an efficient GRASP-based metaheuristic is proposed in Section IV which can be used as an alternative to the optimal LSP_BP_ILP solution. A detailed performance study is conducted and presented in Section V, where both LSP_BP_ILP and LSP_BP_GRASP approaches are considered. Finally, conclusions are drawn in Section VI.

II. CONTROL PLANE SIGNALING IN BULK PROVISIONING

The proposed PCE-based network architecture for dynamic bulk provisioning of LSP requests is shown in Fig. 1. Only optical LSP requests are considered in the current study. However, it is worth mentioning that the PCE-based bulk provisioning framework presented here is equally applicable for other GMPLS/MPLS-based networks as well where LSP requests may have subwavelength granularity. PCEP [17] protocol is used for the communication between the PCC and the PCE as indicated in Fig. 1. Two important messages defined for PCEP in the context of this study are 1) the PCReq message, used to send the path computation requests from the PCC to the PCE, and 2) the PCRep message, which returns the computed path (for a particular LSP) from the PCE back to the PCC. PCEP protocol allows for sending multiple LSP requests in a single PCReq message. Similarly, multiple path computation responses at the PCE can also be bundled together in a single PCRep message to be sent back to the PCC. This feature is exploited in the presented study to reduce PCEP control overhead. At the PCE, the bulk provisioning algorithms

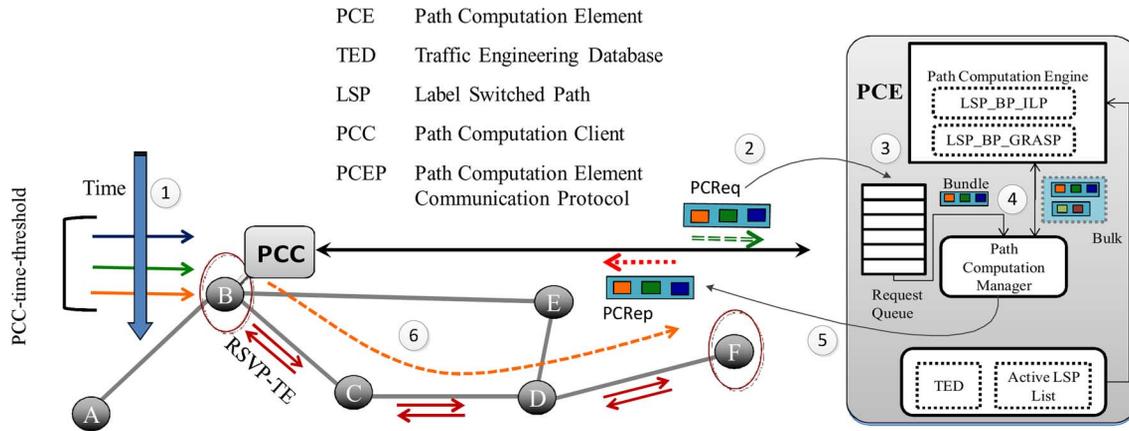


Fig. 1. Proposed PCE-based network architecture for dynamic bulk provisioning of LSP requests.

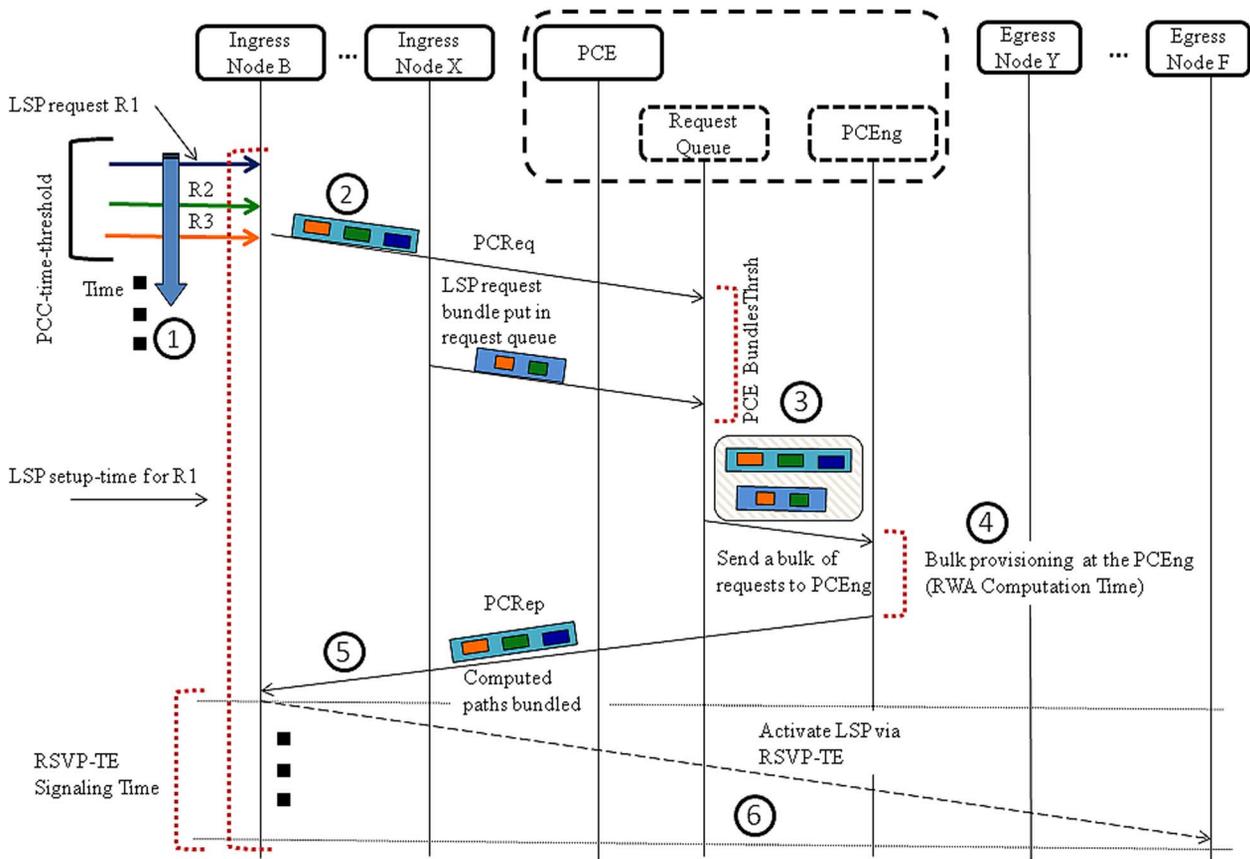


Fig. 2. Time-line view of the signaling for bulk provisioning of LSP requests.

reside in the path computation engine (PCEng). LSP requests are first placed in the request queue (RQ) to be encapsulated by the path computation manager and fed to the PCEng together with the network status information which is saved in the traffic engineering database and in the active LSP list.

The signaling procedure for bulk provisioning of LSP requests is based on PCEP protocol as shown in Fig. 2. As presented in both Figs. 1 and 2, three LSP requests arrive at the ingress (source) node B within a predefined *PCC-time-threshold* (step 1). After the *PCC-time-threshold* for bundling expires, all three LSP requests are bundled in a single PCReq message and transmitted to the PCE for path computation (step 2). Another

option would be to employ a connection-threshold approach where each ingress node (PCC) waits for a specific number of requests to arrive before bundling them together in a single PCReq message. However, a time-based threshold ensures an upper bound on the waiting time for the LSP requests at the ingress nodes before being sent to the PCE.

At the PCE, this bundle is placed in the PCE RQ (step 3). Here, a counter, i.e., *PCE BundlesThresh*, is employed to ensure that a specific number of PCReq messages are collected before the path computation phase starts. In the scenario in Fig. 2, the *PCE BundlesThresh* is assumed to be two. So, after two PCReq messages have arrived in RQ, all the LSP requests are com-

bined in a single list to form a *bulk* (with 5 LSP requests in this case) for concurrent bulk provisioning at the PCE (step 4) in the PCEng. To bound the delay that might be experienced by some bundles placed in RQ during the bulk formation, an additional *cutoff timer* is introduced and activated when aggregation of bundles for each bulk initiates. This cutoff timer is considered as a *hard constraint* and only those bundles that arrived before the expiration of this timer are considered to form a bulk, even if the bulk size is smaller than the PCE BundlesThresh.

Bulk provisioning is defined as the concurrent provisioning of multiple requests from different source nodes. When bulk provisioning at the PCEng is over, all the replies are packed again in PCRep messages where all the responses for LSPs originating at the same ingress nodes are grouped in a single PCRep message to be transmitted to their respective ingress nodes (in this case node B) in step 5. After the ingress nodes receive the computed paths, the RSVP-TE signaling procedure is triggered to activate the computed paths (step 6) for data transmission.

As shown in Fig. 2, the LSP setup time is defined as the time elapsed from a LSP request arrival (at the ingress node) to the time instance in which the connection is set up, i.e., the LSP request is successfully signaled in the network and ready for data transmission. The setup time includes the PCC-time-threshold, the path computation time in the PCEng, the PCC-PCE communication time, the queuing time, and the time necessary for reserving the computed path through network (i.e., the RSVP-TE signaling time).

If a set of LSP requests is required to be processed concurrently, then, according to the PCEP protocol [17], this has to be specified in the PCReq message. This is called *synchronized path* computation in which the synchronization vector (SVEC) objects are introduced in the PCReq messages, and an association is made between the SVEC objects in each transmitted PCReq message via the SVEC list [18]. All the LSP requests in SVEC list are required to be treated as synchronized (for concurrent optimization) by the PCE. In case the path computation fails only for a subset of requests included within the same SVEC, then a NO-PATH is returned exclusively for those requests.

III. LSP BULK PROVISIONING PROBLEM

In this section, we formally define the LSP bulk provisioning problem in the PCE (LSP_BP). We propose an efficient ILP formulation (LSP_BP_ILP) that optimally solves the problem for a given physical topology, number of available wavelengths on each link, routing of already allocated LSP requests, and a set of new LSP requests to be established. The LSP_BP_ILP finds the routes for each LSP request with the objective of minimizing the request blocking together with capacity consumption of the network. Note that, in this study, only optical LSPs (i.e., light-paths) are considered.

A. Notation and Problem Formulation

Physical topology of the network is represented by a directed graph $G(N, E)$, where N corresponds to the set of nodes in the network, and E is the set of fiber links. W denotes the maximum number of supported wavelengths on each link. Source and destination nodes of an LSP request c are denoted by s and d , respectively. Note that c denotes a request in the LSP demand set, i.e., $c \in D$ from s to d while (x, y) denotes the nodes of a physical link, i.e., $(x, y) \in E$ and w denotes a wavelength, with

$w \in W$ in all the following formulations. In order to describe the mathematical model and problem formulation, the following notations are introduced for variables and given parameters.

Given

- 1) $G(N, E)$: Physical topology consisting of set of nodes (N) and set of links (E).
- 2) W : Maximum # of wavelengths supported on each link.
- 3) W_{xy} : Number of free wavelengths on link (x, y) .
- 4) D : Set of LSP requests to be provisioned in the PCE where each request c is denoted by λ_c from a source s to a destination d and $\lambda_c \in D$.

Variables

- 1) L_{xy} : Number of wavelengths to be used by LSPs on link (x, y) based on current solution of the processed bulk.
- 2) $F_{xy,w}^c$: Binary variable equal to 1 if LSP request c from s to d passes through physical link (x, y) .
- 3) A_c : Binary variable equal to 1 if LSP request c is successfully provisioned.
- 4) A_c^w : Binary variable equal to 1 if LSP request c uses wavelength w .
- 5) M : Load of the maximally loaded link in the network.

Objective 1

$$\begin{aligned} \text{Minimize } & \alpha \cdot \left(|D| - \sum_{\forall c} A_c \right) + \\ & \beta \cdot \sum_{\forall (x,y)} (W - W_{xy}) L_{xy} + \gamma \cdot M. \end{aligned}$$

Objective 2

$$\begin{aligned} \text{Minimize } & \alpha \cdot \left(|D| - \sum_{\forall c} A_c \right) + \\ & \beta \cdot \sum_{\forall (x,y)} L_{xy} + \gamma \cdot M. \end{aligned}$$

Objective 3

$$\begin{aligned} \text{Minimize } & \alpha \cdot \left(|D| - \sum_{\forall c} A_c \right) + \\ & \beta \cdot \sum_{\forall (x,y)} (W - W_{xy}) L_{xy}. \end{aligned}$$

Objective 4

$$\text{Minimize } \alpha \cdot \left(|D| - \sum_{\forall c} A_c \right) + \beta \cdot \sum_{\forall (x,y)} L_{xy}.$$

Constraints

$$\sum_{\forall x} F_{xy,w}^c - \sum_{\forall x} F_{yx,w}^c = \begin{cases} 0, & y \neq s, d \\ -A_c^w, & y = s, \\ A_c^w, & y = d \end{cases} \quad \forall w, \forall c, \forall y \quad (1)$$

$$\sum_{\forall w} F_{xy,w}^c \leq 1, \quad \forall c, (x, y) \quad (2)$$

$$\sum_{\forall w} A_c^w \leq A_c, \quad \forall c \quad (3)$$

$$\sum_{\forall c} F_{xy,w}^c \leq 1, \quad \forall w, (x, y) \quad (4)$$

$$A_c \leq \lambda_c, \quad \forall c \quad (5)$$

$$\sum_w \sum_c F_{xy,w}^c = L_{xy}, \quad \forall (x, y) \quad (6)$$

$$L_{xy} \leq W_{xy}, \quad \forall (x, y) \quad (7)$$

$$F_{xy,w}^c \leq W_{xy}^w, \quad \forall c, \forall w, \forall (x, y) \quad (8)$$

$$M \geq L_{xy} + (W - W_{xy}), \quad \forall (x, y). \quad (9)$$

Description

Four different objective functions are considered in the proposed ILP model in order to evaluate the effect of the network congestion control parameters on the overall performance in a dynamic bulk provisioning scenario. All the objectives are shown as a weighted sum with different weights denoted by α, β , and γ where $\alpha \gg \beta \gg \gamma$ in order to give priority to the minimization of the first term. In Objective 1, the first term aims at minimizing the LSP blocking rate, the second tries to minimize the weighted link loads (Weighted LL) in order to control congestion, and the last term ensures load balancing by minimizing the load of the maximally loaded link (M). Objective 1 is denoted as Weighted LL + M from now on. In Objective 2, minimization of the link loads are considered without any weighting factor in the second term (LL + M). In Objective 3, only the weighted link loads are used as the secondary objective (Weighted LL). Finally, in Objective 4, only the total wavelength consumption is considered (LL) without any congestion control.

Constraint (1) represents the flow conservation and the wavelength continuity constraint (WCC) for routing request λ_c from node s to d . A_c keeps track of the successfully routed connections. Constraints (2) and (3) ensure that each LSP can only use one wavelength on a link and constraint (4) ensures that each w can be used by only one LSP. The successful provisioning variable is bound by whether an LSP request c exists in constraint (5). The wavelength link load on link (x, y) is defined in constraint (6) while each link load is bound by the link capacity in constraint (7). Constraint (8) serves for eliminating the flow decision variables when w on link (x, y) is not available because of currently routed LSPs in the network. The load of the maximally loaded link is defined in constraint (9) which helps load balancing as a tertiary objective (for Objectives 1 and 2).

IV. GRASP WITH BULK PROVISIONING (LSP_BP_GRASP)

In this section, a metaheuristic approach based on GRASP [16] is proposed which is called LSP bulk provisioning using GRASP (LSP_BP_GRASP). GRASP is an iterative process which consists of two phases, a construction phase in which a feasible solution is built, and a local search phase during which a local optimum is searched in the neighborhood of the feasible solution. The best overall solution is saved. In the construction phase, a solution is built iteratively one element at a time. At each iteration, the next element to be added to the solution is selected randomly from a restrictive candidate list (RCL) that is sorted based on a greedy function. This function measures the benefit of selecting an element. Note that the heuristic is adaptive because the benefits associated with each element in RCL are updated at each iteration to reflect the changes brought in by the selection of the previous element in the construction phase. The probabilistic component comes from selecting one of the best candidates in the RCL (but not necessarily the top

Algorithm 1. LSP_BP_GRASP

Input: $G(N, E)$, W , D a set of requests to be provisioned where $\lambda_c = \{s, d\}$ and $\lambda_c \in D$, $MaxItr$

Output: A set $bestSol$ of provisioned requests constituting best solution

Initialization: $RCL_List = \emptyset, bestSol = \emptyset, A = \emptyset$

```

1. while current iteration <= MaxItr
2.   while D is not empty //compute a solution
3.     RCL_Size ← ⌊D / 2⌋ //dynamic RCL size
4.     while size of RCL_List < RCL_Size
5.       if D is empty then break
6.       for each LSP request λc in D
7.         Allocation cost ← Allocation_Cost(λc)
8.         if a request λc cannot be allocated
9.           Remove it from D
10.        Select λc with the minimum allocation cost
11.        Add it to RCL_List
12.        Remove it from D
13.      Remove a random connection λc from RCL_List
14.      Allocate(λc) //perform RWA
15.      Add to A //build current solution
16.    Compute objective value for current solution in A
17.    if objective value for the current solution < bestSol
18.      bestSol ← A
19.    if current iteration < MaxItr //if not last
20.      De-allocate n random connections λc from A
21.      where n is a random number
22.      Add these connections back to D
23.  return bestSol

```

element) and adding it to the partial solution that is built so far. Once a complete solution is built, the local search allows finding better solutions in the neighborhood of the current one.

LSP_BP_GRASP is presented in three different parts. Algorithm 1 shows the main steps, while Algorithms 2 and 3 define the supporting procedures called from Algorithm 1. All the connection requests that need to be provisioned in a given bulk provisioning phase are put in a list (referred to as D in the pseudocode) to form a bulk and to be fed to LSP_BP_GRASP.

In the pseudocode, $MaxItr$ specifies the maximum number of iterations used to generate and improve the solution (maximize the set of provisioned requests). A provisioned LSP _{i} comprises of k links l_1, l_2, \dots, l_k for a path P_i while $W_{resv,i}$ represents the total number of newly reserved wavelengths used by P_i for a provisioned LSP _{i} . At the end of each iteration, set A contains each LSP _{i} that has been provisioned. So, set A contains the current solution at the completion of a given iteration. In order to compare results with the LSP_BP_ILP, we take into account Objective 3 which can be stated as: $\text{Min } \alpha \times (|D| - |A|) + \beta \times (W - W_{xy}) \times L_{xy}$, where $\alpha \gg \beta$.

Algorithm 1 constructs a feasible solution in Steps 1–9 and loops through a number of iterations equal to $MaxItr$ to improve it in the succeeding iterations. Each solution is built iteratively until D becomes empty (step 2). RCL_Size in Step 3 refers to the size of the restricted candidate list (RCL) which is set dynamically based on the input size, and RCL_List refers to the list itself. RCL_List is built iteratively in steps 4–8. When RCL_List is populated, a random connection is moved to set A and it contributes to the building of a partial (intermediate) solution. The whole procedure is repeated again in an iterative manner until a feasible solution is constructed. In step 11, this solution is saved if the objective is best so far.

Algorithm 2. Allocate**Input:** $G(N, E)$, W , $\lambda_c = \{s, d\}$, current network state S **Output:** TRUE if λ_c is provisioned otherwise FALSE

1. Load pre-computed K -shortest paths for s and d of the given λ_c
2. //assign path using WLCR if allocation possible
 $P_i \leftarrow \text{WLCR}(K\text{-SP})$
3. **if** P_i is not null
 - Do wavelength assignment for selected P_i
using First-Fit
 - return** TRUE
4. **else return** FALSE

Algorithm 3. Allocation_Cost**Input:** $G(N, E)$, W , $\lambda_c = \{s, d\}$, current network state S **Output:** allocation cost $W_{resv, i}$ if λ_c is provisioned

1. **Allocate**(λ_c) //a temporary allocation
2. $W_{resv, i} \leftarrow$ calculate no. of reserved wavelengths by P_i
3. //release all wavelength resources associated with this allocation
De-allocate current request λ_c
4. **return** $W_{resv, i}$

In step 12, if the current iteration is not the last iteration, then n random connections are deallocated (removed from A and moved back to D), where n is selected randomly based on a uniform distribution. Note that $MaxItr$, RCL_Size , and n are three important parameters that influence the running time of the LSP_BP_GRASP.

In Algorithm 2, the precomputed k -shortest paths are loaded in memory in step 1. A suitable path P_i is assigned in step 2 using the weighted least congested routing (WLCR) algorithm [19]. If a path can be computed, then the procedure returns true in step 3 after performing wavelength assignment on first-fit basis otherwise false is returned in step 4.

Algorithm 3 is called by Algorithm 1 to compute the allocation cost of a given λ_c . A *temporary allocation* (in local snapshot of the network state) is performed for the current λ_c in step 1. A count of the newly allocated wavelengths which need to be reserved by the P_i are stored in $W_{resv, i}$ in step 2. Step 3 deallocates the temporarily allocated λ_c . Total allocation cost $W_{resv, i}$ for the current λ_c is then returned in step 4.

To calculate the time complexity of the LSP_BP_GRASP, it can be observed that first part in Algorithm 1 is computationally dominant (steps 1–9). An upper bound for this part is $(MaxItr) \times (|D|) \times (|D|/2) \times ((|D| + |D| - (|D|/2 + 1))/2)$ from steps 1, 2, 4 and 5, which is equal to $(MaxItr) \times (|D|^3)$ by ignoring terms with lower powers of $|D|$ and constant multipliers in the resulting formula. So, the time complexity of the LSP_BP_GRASP is $O(|D|^3)$ for any fixed value of $MaxItr$. Note that to initially calculate a precomputed set of K -shortest paths for each of the $v \times (v - 1)$ node pairs using Bellman–Ford algorithm requires a time complexity of $O(K \times v \times e)$.

V. PERFORMANCE EVALUATION

The performance evaluation of the proposed LSP_BP_ILP and LSP_BP_GRASP for the LSP bulk provisioning problem is conducted in a Java-based discrete event-driven simulator [20]. Since a transparent WDM network scenario is assumed in the current study, only optical LSPs (i.e., lightpaths) are being considered here. The National Science Foundation (NSF) network topology is used for simulations which comprises 14 nodes and 20 bidirectional links (see Fig. 3) with 16 wavelengths capacity

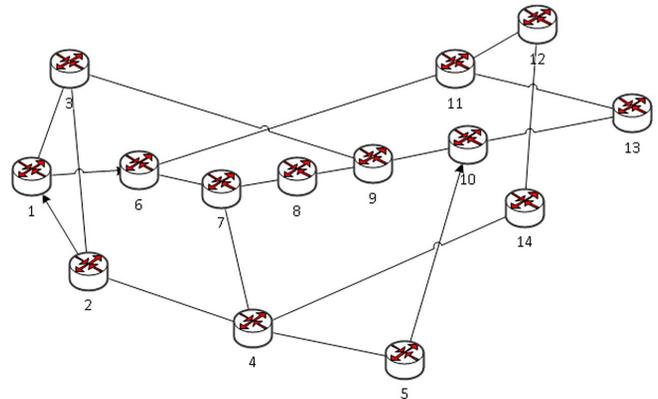


Fig. 3. 14-Node NSF topology.

(in each direction). The WCC is enforced for path computation. Two scenarios are considered for LSP request processing: 1) *Sequential*, where all the requests in a bulk are processed at the PCE in a sequential manner (i.e., one by one in the order of their arrival) and 2) *Concurrent*, where connection requests are processed in a combinatorial manner (i.e., based on LSP_BP_ILP and/or LSP_BP_GRASP).

For both LSP_BP_ILP and LSP_BP_GRASP, the values for α , β , and γ are set to 1000, 1, and 0.1, respectively. For LSP_BP_GRASP, the $MaxItr$ parameter is set to 100. For the sequential scenario, routes are computed using the WLCR algorithm [19]. LSP requests arrive to the network following a Poisson distribution, and the holding time is assumed to be exponentially distributed. An RSVP-TE based signaling procedure is employed for resource reservation after the path computation phase. For the signaling phase, the switch configuration time T_s at each node is assumed to be 50 ms and the node processing delay T_p is assumed to be 20 ms [21]. The considered performance parameters are blocking probability, average link utilization, PCEP control overhead, average LSP setup time, and RWA computation time (i.e., time required for RWA at the PCE). In addition to the PCEP signaling control overhead, the estimation of the PCEP control overhead includes TCP, IP, and Ethernet overhead, assuming that the control plane is implemented over Ethernet.

A stateless PCE [22] is assumed since the current lightpath configuration state does not need to be maintained for the provisioning of LSP requests. A *baseline* scenario is considered for benchmarking purposes in which LSP requests are provisioned one by one with no bundling at the source node and no bulking at the PCE. This is in contrast to the sequential and the concurrent schemes where the proposed bulk provisioning approach is utilized. The mean LSP holding time is fixed to a unit time (1.0 min), while the LSP request arrival rate varies from 150 to 170 arrivals/min, to represent different network loads. For the presented results, each experiment was repeated eight times, and 100 000 connection requests were generated for each experiment. The confidence interval for all the plotted results is 5% or less, with a 95% confidence level except for very low values of the blocking probability. The simulation platform comprises a Red Hat Enterprise Linux workstation fitted with 12 GB of memory and dual Intel Xeon CPUs (four cores per CPU) clocked at 2.0 GHz. For solving the proposed

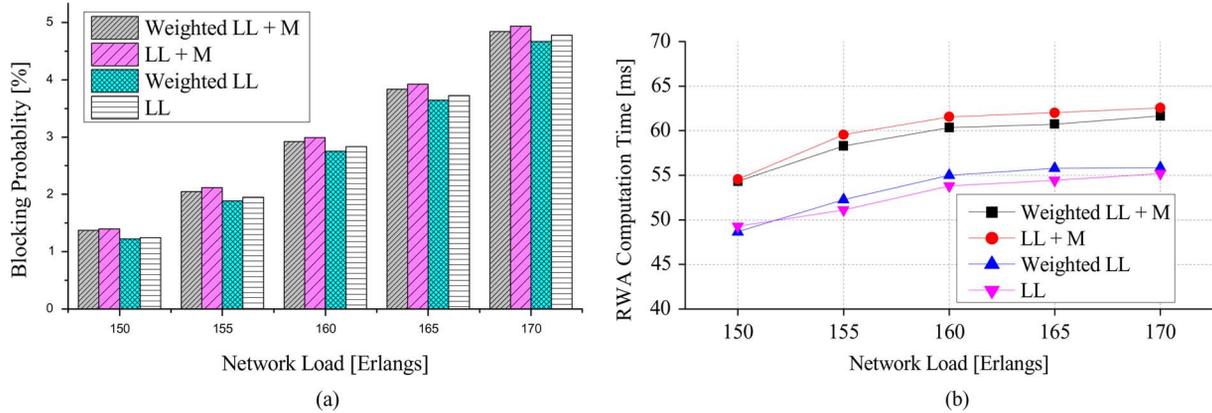


Fig. 4. Performance effect of congestion control objectives in LSP_BP_ILP. (a) Blocking probability versus network load. (b) RWA computation time versus network load.

ILP_LSP_BP model at the PCE, Gurobi Optimizer 3.05 [23] is utilized through Java application programming interface.

A. Performance Effect of Congestion Control Objectives in LSP_BP_ILP

The first set of results in Fig. 4(a) and (b) shows a performance comparison for different approaches corresponding to the objective functions that are defined in Section III for LSP_BP_ILP. These results are obtained with a PCC-time-threshold value of 30 s and the PCE BundlesThresh parameter was set to 5. It can be observed from Fig. 4(a) that differences in terms of blocking probability are minimal, despite of that the best performing strategy in terms of blocking probability is the one using the Objective 3, i.e., to minimize the weighted link loads (Weighted LL). The main reason is that Weighted LL provides a more *fine-tuned* congestion control when computing paths (i.e., it considers the congestion on all the links using weighted link loads instead of just taking into account the most loaded link in the objective). Even minimizing wavelength consumption, i.e., minimizing LL shows good performance. The worst performance in terms of blocking probability is experienced with Objective 2 (i.e., LL + M) because this objective function only tries to minimize the congestion on the maximally loaded link, without balancing the load for all the links in the network (i.e., minimize the variance of all the link loads in the network) which may lead to higher average link utilization.

RWA computation times for the four different objective functions of LSP_BP_ILP are presented in Fig. 4(b). The computation time is noticeably lower (e.g., $\approx 18\%$ at load of 150 Erlangs) for LL and Weighted LL approaches when compared to the objective functions that involve minimizing the load of the maximally loaded link (M).

Note that for increasing loads, the arrival rate of connection requests also increases. This in turns results in bigger bundles of connection requests being formed at the source nodes to be sent to the PCE. This has an impact on the RWA computation time that is increasing with high load values for all the four alternative objective functions of LSP_BP_ILP.

After analyzing the performance of the congestion control parameters in LSP_BP_ILP, it can be concluded that introducing M into the objective function does not bring any gains in terms of

blocking probability performance in a dynamic scenario. On the other hand, it increases the RWA computation time noticeably. For this reason, Weighted LL (Objective 3) is considered for the remaining performance evaluation of LSP_BP_ILP presented in this section.

B. Blocking Probability

Results for blocking probability as a function of the network load are presented in Fig. 5 for two different PCC-time-threshold values, i.e., 10 s in Fig. 5(a) and 30 s in Fig. 5(b). In all experiments, the value of the PCE cutoff timer is set to be equal to the PCC-time-threshold in order to have the PCE cutoff timer compatible with the average setup time of a connection. This means that in Fig. 5(a), the PCE cutoff timer is set to 10 s, and in Fig. 5(b) to 30 s.

It can be observed in Fig. 5(a) that LSP_BP_ILP is able to noticeably reduce the blocking probability, when compared to the other approaches. The performance improvement of LSP_BP_ILP is increasing at higher load values. This is because, for these values of the load, the number of LSP requests in one bulk increases leading to better chances of exploiting the benefits of concurrent optimization. Interestingly, the blocking probability results for the LSP_BP_ILP are better than the ones of the Baseline approach. This can be explained by looking at the value of the PCC-time-threshold. At 10 s, the effect of traffic burstiness created by bundling connection requests at the source node is still not so pronounced, and the beneficial blocking probability effects of concurrent optimization at the PCE are able to compensate for it.

In Fig. 5(b), the value of the PCC-time-threshold is higher, i.e., 30 s. As a consequence, the number of LSP requests bundled at each source node is bigger. This increases the burstiness of the traffic, thus decreasing the chance of finding available resources for multiple LSPs at the same time. Note that, for the results in Fig. 5(b), the value of PCE BundlesThresh is equal to one. This means that each bulk considered for concurrent optimization contains connection requests originating all at the same source node. As shown in the figure, this is a limiting factor for the potential benefits in terms of blocking probability of a concurrent optimization. Therefore, the blocking probability for the baseline is lower than the other provisioning approaches. This is not true anymore when the set of connection requests optimized

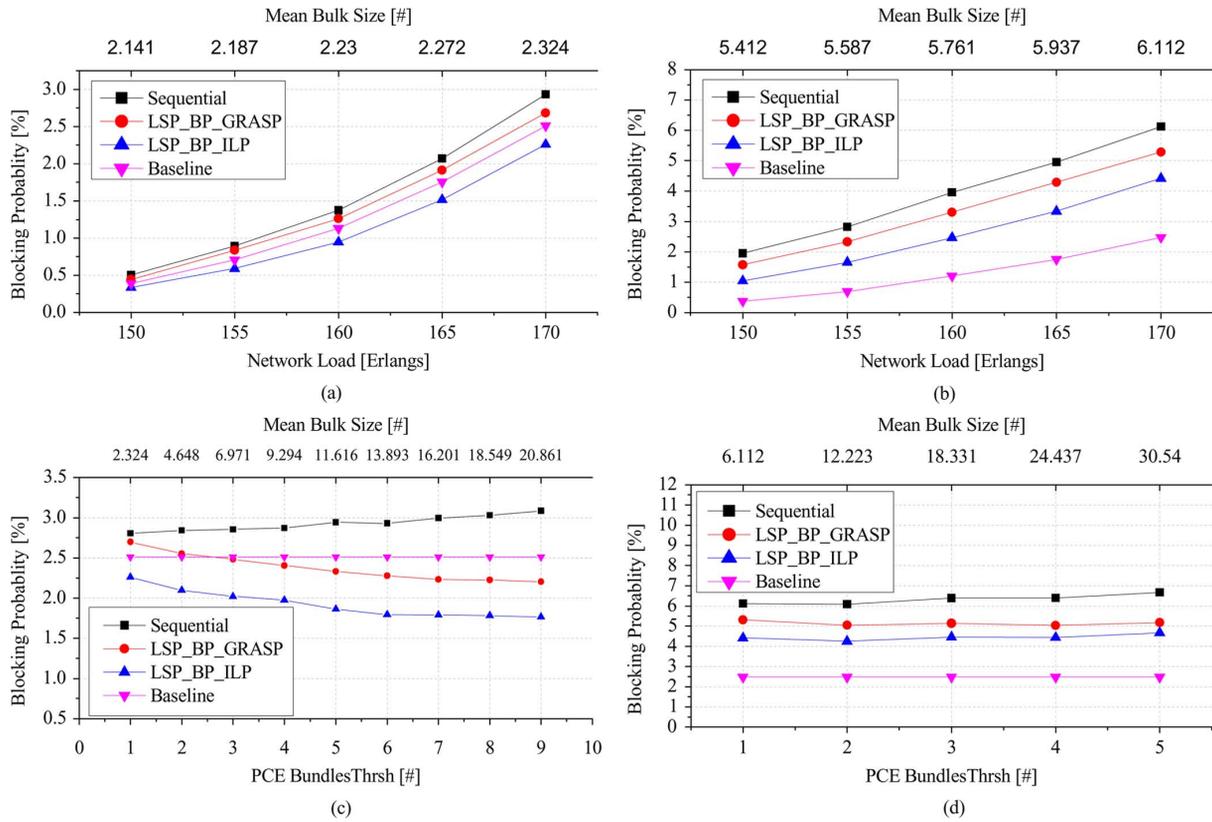


Fig. 5. Blocking probability versus network load and PCE BundlesThresh for different PCC-time-thresholds. (a) PCC-time-threshold = 10 s and PCE BundlesThresh = 1. (b) PCC-time-threshold = 30 s and PCE BundlesThresh = 1. (c) Network Load = 170 Erlangs and PCC-time-threshold = 10 s. (d) Network Load = 170 Erlangs and PCC-time-threshold = 30 s.

concurrently is coming from a variety of source nodes, i.e., the value of PCE BundlesThresh is sufficiently high, as shown next.

The value of blocking probability as a function of PCE BundlesThresh is shown in Fig. 5(c) and (d) for a PCC-time-threshold and PCE cutoff timer both equal to 10 and 30 s, respectively. The value of offered load is fixed at 170 Erlangs. The blocking probability tends to decrease with increasing values of the *mean bulk size* for both the LSP_BP_ILP and the LSP_BP_GRASP. These results confirm the earlier claim that higher values of PCE BundlesThresh allows to process requests from a more diverse set of ingress nodes, hence a better optimization of resources. The opposite effect can be observed for the sequential approach because of the limitation of sequential processing of multiple LSP requests. Finally, it is worth to note that the connection blocking of the GRASP-based heuristic is bounded from above by the performance of the sequential strategy, and from below by the performance of the ILP-based solution.

C. PCEP Control Overhead

The values of PCEP control overhead as a function of the network load for two different PCC-time-threshold values, i.e., 10 and 30 s, are presented in Fig. 6(a), and (b), respectively. The value of the PCE cutoff timer is set to 10 and 30 s, respectively. PCEP control overhead reduction is significant ($\approx 30\%$) as shown in Fig. 6(a) even with a low PCC-time-threshold for all bulk provisioning schemes when compared to the baseline case. It can be observed that the control overhead gradually increases for higher values of load for all schemes. This happens because

with higher loads, the arrival rate of connection requests also increases which results in growing PCEP control overhead per unit time as shown. In Fig. 6(b), we can see that the PCEP control overhead can be reduced even further by employing a higher PCC-time-threshold for all bulk provisioning approaches.

D. LSP Setup Time

The value of the average LSP setup time as a function of the network load is presented in Fig. 7 for PCC-time-threshold values of 10 s [see Fig. 7(a)], and 30 s [see Fig. 7(b)]. The value of the PCE cutoff timer is set to 10 and 30 s, respectively.

It can be observed from Fig. 7(a) that the LSP setup time is slightly higher for LSP_BP_ILP and LSP_BP_GRASP as compared to baseline and it is very close to the sequential approach. Setup time is mostly dominated by the PCC-time-threshold employed at the source nodes. The figure also shows that the setup time difference between the two proposed concurrent approaches, i.e., LSP_BP_ILP and LSP_BP_GRASP, is almost negligible and that LSP_BP_GRASP performs the same as sequential, which proves that the proposed ILP model for concurrent RWA computation is an attractive and viable choice to be deployed in real PCE-based WDM network scenarios. It can also be seen that the LSP setup time is dominated by the PCC-time-threshold. If PCC-time-threshold is set to zero, then the LSP setup time will only be dependent on the signaling and path computation time of the RWA schemes in the PCE if the reduction in the PCEP control overhead is not a priority.

Fig. 7(b) shows that in this specific scenario, the difference between LSP_BP_ILP and LSP_BP_GRASP is a bit more no-

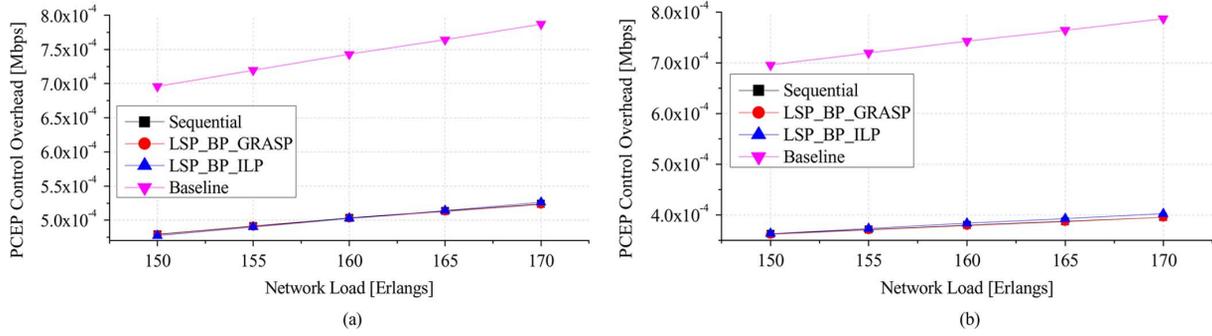


Fig. 6. PCEP control overhead versus network load for different PCC-time-thresholds. (a) PCC-time-threshold = 10 s and PCE BundlesThrsh = 1, (b) PCC-time-threshold = 30 s and PCE BundlesThrsh = 1.

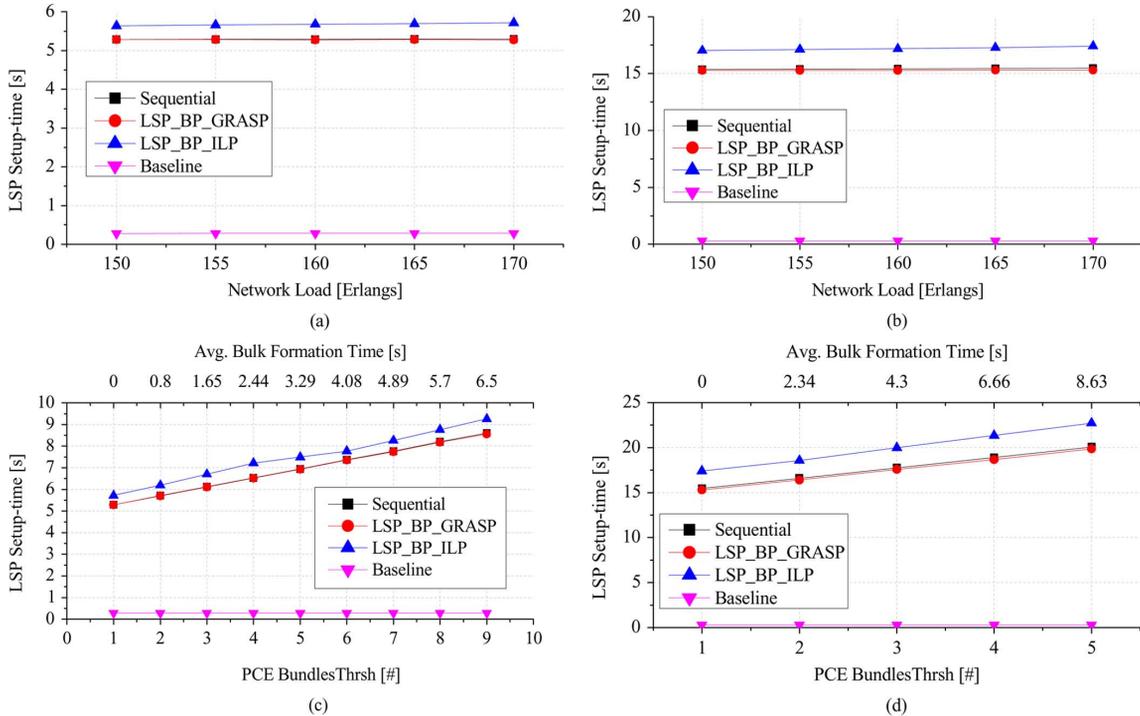


Fig. 7. LSP setup time versus network load and PCE BundlesThrsh for different PCC-time-thresholds. (a) PCC-time-threshold = 10 s and PCE BundlesThrsh = 1. (b) PCC-time-threshold = 30 s and PCE BundlesThrsh = 1. (c) Network load = 170 Erlangs and PCC-time-threshold = 10 s. (d) Network load = 170 Erlangs and PCC-time-threshold = 30 s.

ticeable because of the bigger bulk sizes that on average need to be processed at the PCE.

The value of LSP setup time as a function of PCE BundlesThrsh is shown in Fig. 7(c) and (d) for a PCC-time-threshold and PCE cutoff timer both equal to 10 and 30 s, respectively. Note that the values shown on the top x axis show average time spent at the PCE to build a bulk of the required size (i.e., the average bulk formation time based on PCE BundleThrsh). Fig. 7(c) indicates that higher values of PCE BundlesThrsh causes the LSP setup time to gradually increase since a longer waiting time is required to form larger bulks at the PCE (i.e., queuing delay in RQ) for a PCC-time-threshold of 10 s. In Fig. 7(d), we observe a similar behavior as in Fig. 7(c), although values of the LSP setup time are higher because of the higher value of the PCC-time-threshold that is being considered here.

E. Average Link Utilization

The average link utilization as a function of the network load is presented in Fig. 8. PCC-time-threshold is equal to 10 s in Fig. 8(a) and to 30 s in Fig. 8(b). The value of the PCE cutoff timer is set to 10 and 30 s, respectively.

The proposed LSP_BP_ILP approach makes the most efficient use of wavelength resources in the network [see Fig. 8(a)] having the lowest blocking probability when the PCC-time-threshold value is 10 s [see Fig. 5(a)]. However, at higher load, curves for both LSP_BP_ILP and LSP_BP_GRASP start to converge as wavelength resources start to become saturated. The worst performance is shown by the sequential approach which is almost at the same level as baseline in terms of link utilization at different load values. Efficient resource usage by LSP_BP_ILP is a direct consequence of computing

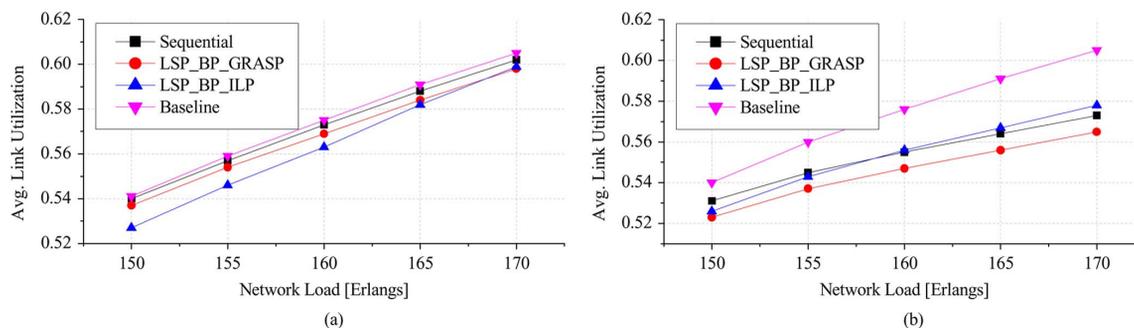


Fig. 8. Average link utilization versus network load for different PCC-time-thresholds. (a) PCC-time-threshold = 10 s and PCE BundlesThrsh = 1. (b) PCC-time-threshold = 30 s and PCE BundlesThrsh = 1.

short routes for most of the requests while at the same time avoiding congestion in the network (as mentioned previously, Weighted LL allows congestion control at a more fine-tuned level).

The situation changes when a PCC-time-threshold value of 30 s is used [see Fig. 8(b)]. With this threshold, LSP_BP_GRASP shows the lowest average link utilization, at the expense of worse blocking as compared to LSP_BP_ILP [see Fig. 5(b)], while LSP_BP_ILP and sequential approach are at the same level for different load values.

It can be observed from the presented set of results that most benefits in terms of network resource optimization can be gained if an appropriate PCC-time-threshold value is selected as well as an appropriate value of PCE BundlesThrsh parameter at the PCE. It is worth re-emphasizing that the reason for this behavior is that a value of PCE BundlesThrsh (>1) allows the bulks formed at the PCE to contain requests from a more diverse set of source nodes which enhances the degree of resource optimization that can be achieved by LSP_BP_ILP and LSP_BP_GRASP. On the other hand, a suitable value of PCC-time-threshold at the source nodes allows reducing PCEP control overhead.

VI. CONCLUSION

In this study, a flexible PCE-based bulk provisioning framework is presented where a PCE-based network architecture is proposed for concurrent optimization of LSP requests which results in significant resource optimization (i.e., blocking probability and average link utilization) and control overhead reduction at the expense of an increase in LSP setup time. With this objective in mind, a novel mathematical model (LSP_BP_ILP) and a low overhead GRASP-based metaheuristic (LSP_BP_GRASP) are proposed for concurrent optimization of the optical LSP requests at the PCE in a dynamic bulk provisioning scenario.

For an effective use, the presented framework requires a careful choice of the value of the PCC-time-threshold as well as of the PCE BundlesThrsh parameter, i.e., the bundling and the bulking parameter, respectively. If the PCE control overhead need to be reduced by 30%, the PCC-time-threshold needs to be at least 10 s. The blocking probability can be reduced up to 22% by concurrent optimization (ILP), with a PCE-BundlesThrsh value of 9. On the other hand, by selecting carefully the PCC-time-threshold and the PCE-BundlesThrsh, both PCEP control overhead and blocking probability can be reduced by 30% and 22%, respectively. It is worth noting

that the proposed framework is flexible. If the PCEP control overhead reduction is not a priority, the PCC-time-threshold can be set to zero. In this case, the LSP setup time will be dependent on the RWA calculation time at the PCE and on the signaling delay in addition to PCE BundlesThrsh value. Value of the cutoff timer at the PCE should also be carefully selected to avoid long LSP setup times on one hand, while, on the other, still allowing to form bulks of the required size (based on the value of the PCE BundlesThrsh).

The feasibility of the proposed LSP_BP_ILP model in terms of computational time is also confirmed by the numerical results in terms of LSP setup time and RWA computation time when compared to the GRASP and the sequential-based solutions. Performance (i.e., blocking probability and computation time) of different objective functions for the ILP is also compared, and it is shown that the best performance can be accomplished by the Weighted LL approach both in terms of reducing blocking probability and computation time.

REFERENCES

- [1] K. Knightson, N. Morita, and T. Towle, "NGN architecture: Generic principles, functional architecture, and implementation," *IEEE Commun. Mag.*, vol. 43, no. 10, pp. 49–56, Oct. 2005.
- [2] E. Mannie, "Generalized Multi-Protocol Label Switching (GMPLS) Architecture," IETF RFC 3945, Oct. 2004.
- [3] N. Larkin, "ASON and GMPLS—The battle of the optical control plane," Data Connection Ltd., Enfield, U.K., Aug. 2002.
- [4] Q. Song, I. Habib, and W. Alanqar, "On the performance evaluation of distributed dynamic routing in GMPLS optical networks," in *Proc. IEEE Global Telecommun. Conf.*, St Louis, Nov. 2005, pp. 2067–2071.
- [5] P. Castoldi, F. Cugini, L. Valcarenghi, N. Sambo, E. Le Rouzic, M. J. Poirrier, N. Adriolli, F. Paolucci, and A. Giorgetti, "Centralized versus distributed approaches for encompassing physical impairments in transparent optical networks," presented at the presented at the Opt. Netw. Des. Model. Conf., Athens, Greece, May 2007.
- [6] A. Farrel, J.-P. Vasseur, and J. Ash, "A Path Computation Element (PCE) Based Architecture," IETF RFC 4655, Aug. 2006.
- [7] T. Tsuritani, M. Miyazawa, S. Kashihara, and T. Otani, "Optical path computation element interworking with network management system for transparent mesh networks," in *Proc. Opt. Fiber Commun./Nat. Fiber. Opt. Eng. Conf.*, San Diego, CA, Feb. 2008, pp. 1–10.
- [8] J. Ahmed, P. Monti, and L. Wosinska, "LSP request bundling in a PCE-based WDM network," in *Proc. Opt. Fiber Commun./Nat. Fiber. Opt. Eng. Conf.*, San Diego, CA, Mar. 2009, pp. 1–3.
- [9] F. Cugini, S. Xu, H. Harai, F. Paolucci, L. Valcarenghi, and P. Castoldi, "Introducing path computation element (PCE) in optical grid networking," in *Proc., Eur. Conf. Opt. Commun.*, Brussels, Belgium, Sep. 2008, pp. 1–2.
- [10] L. Valcarenghi, P. Korus, F. Paolucci, F. Cugini, M. Kantor, K. Wajda, and P. Castoldi, "Experimental evaluation of PCE-based batch provisioning of grid service interconnections," in *Proc., IEEE Global Telecommun. Conf.*, Honolulu, HI, Nov. 2009.

- [11] T. Li and B. Wang, "Approximating optimal survivable scheduled service provisioning in WDM optical networks with iterative survivable routing," in *Proc. 3rd Int. Conf. Broadband Commun. Netw. Syst.*, San Jose, CA, Oct. 2006, pp. 1–10.
- [12] A. Jaekel and Y. Chen, "Demand allocation without wavelength conversion under a sliding scheduled traffic model," in *Proc. 4th Int. Conf. Broadband Commun. Netw. Syst.*, Raleigh, NC, Sep. 2007, pp. 495–503.
- [13] B. Wang, T. Li, X. Luo, Y. Fan, and C. Xin, "On service provisioning under a scheduled traffic model in reconfigurable WDM optical networks," in *Proc. 2nd Int. Conf. Broadband Commun. Netw. Syst.*, Boston, MA, Oct. 2005, pp. 13–22.
- [14] E. He, X. Wang, and J. Leigh, "A flexible advance reservation model for multi-domain WDM optical networks," in *Proc. 3rd Int. Conf. Broadband Commun. Netw. Syst.*, San Jose, CA, Oct. 2006, pp. 1–10.
- [15] J. Ahmed, P. Monti, and L. Wosinska, "Concurrent processing of multiple LSP request bundles on a PCE in a WDM network," in *Proc., Opt. Fiber. Commun./Nat. Fiber Opt. Eng. Conf.*, San Diego, CA, Mar. 2010, pp. 1–3.
- [16] T. A. Feo and M. G. Resende, "Greedy randomized adaptive search procedures," *J. Global Optimization*, vol. 6, pp. 109–133, 1995.
- [17] J.-P. Vasseur and J. L. Le Roux, "Path Computation Element (PCE) Communication Protocol (PCEP)," Nov. 2008.
- [18] I. Nishioka and D. King, "The Use of SVEC (Synchronization VECTOR) List for Synchronized Dependent Path Computations," Jun. 2010.
- [19] X.-W. Chu, B. Li, and Z. Zhang, "A dynamic RWA algorithm in a wavelength-routed all-optical network with wavelength converters," in *Proc. 22nd Annu. Joint Conf. IEEE Comput. Commun. Soc.*, San Francisco, CA, Apr. 2003, pp. 1795–1804.
- [20] J. Ahmed and S. Nawaz, "POSE: A new discrete event optical simulator for the Java platform," in *Proc., Nat. Conf. Emerging Technol.*, Karachi, Pakistan, Dec. 2004.
- [21] J. Teng and G. N. Rouskas, "A detailed analysis and performance comparison of wavelength reservation schemes for optical burst switched networks," *J. Photon. Netw. Commun.*, vol. 9, pp. 311–335, May 2005.
- [22] P. Castoldi, F. Paolucci, A. Giorgetti, F. Cugini, L. Valcarengi, S. Huang, and K.-I. Kitayama, "Management-augmented stateless PCE for wavelength routed optical networks," in *Proc. 10th Int. Conf. Transparent Opt. Netw.*, Athens, Greece, Jun. 2008, pp. 23–26.
- [23] Gurobi optimizer [Online]. Available: <http://www.gurobi.com>

Jawwad Ahmed received the Master's degree with a major in network technologies from the National University of Science and Technology (NUST), Islamabad, Pakistan. He is currently working toward the Ph.D. in photonics in the School of Information and Communication Technology (ICT), The Royal Institute of Technology (KTH), Stockholm, Sweden, with a specialization in optical networks.

His research interests include access networks, generalized multiprotocol label switching and path computational element-based optical networks design, interdomain routing, and discrete event simulation of communication networks.

Cicek Cavdar (M'10) received the Ph.D. degree from the Department of Computer Engineering, Istanbul Technical University, Istanbul, Turkey, in 2009.

During her Ph.D., she was a Researcher in the Networks Research Laboratory, University of California, Davis, from December 2005 to July 2008. She was an Assistant Professor in the Department of Computer Engineering, Istanbul Technical University, until April 2011. She is currently a Researcher at the Royal Institute of Technology, Stockholm, Sweden. Her research interests include design and analysis of optical networks with focus on survivability, multilayer resilience, energy efficiency, and elastic spectrum management.

Paolo Monti (M'01) received the Laurea degree in electrical engineering from the Politecnico di Torino, Torino, Italy, in 2001, and the Ph.D. degree in electrical engineering from the University of Texas at Dallas (UTD), Dallas, in 2005.

From 2006 to 2008 he was a Research Associate in the Open Networking Advance Research Laboratory (OpNeAR), UTD. In September 2008, he joined the Royal Institute of Technology, Stockholm, Sweden, where he is currently an Assistant Professor in the School of Information and Communication Technology (ICT) and a member of the Optical Networks Laboratory (ONLab). He has coauthored more than 50 papers published in international journals and presented in leading international conferences. His research interests include network planning, protocol design, performance evaluation, and optimization techniques for both optical and wireless networks.

Dr. Monti is serving on the Editorial Board of the Springer *Networking and Electronic Commerce Journal*, and is one of the Guest Editor of the Special issue on Green Networking and Computing of the *Journal of High Speed Networks*.

Lena Wosinska (SM'11) received the Ph.D. degree in photonics and Docent degree in optical networking from the Royal Institute of Technology (KTH), Stockholm, Sweden.

In 1986, she joined KTH, where she is currently a Professor in telecommunication in the School of Information and Communication Technology (ICT), heading Optical Networks Laboratory (ONLab). She has been involved in research on several EU projects and coordinates a number of national projects. Her research interests include fiber access networks, energy efficiency in optical networks, photonics in switching, optical network management, reliability, and survivability.

Dr. Wosinska has been involved in many professional activities including guest editorship of the Optical Society of America (OSA), Elsevier, and Springer journals, serving in Technical Program Committee of several international conferences, as well as reviewer for many journals and project proposals. During 2007–2009, she was an Associate Editor of the *OSA Journal of Optical Networking*, and since April 2009 has been serving on the Editorial Board of the *IEEE/OSA Journal of Optical Communications and Networking*. She was the Co-Chair of the International Society for Optical Engineers Asia-Pacific Optical Communications Conference 2008 and the IEEE/OSA/SPIE Asia Communications Photonics Conference (ACP) 2009 Subcommittee Network Architectures, Management and Applications (SC4). She was also the Chair of SC4 in the IEEE/OSA/SPIE ACP during 2010–2012. She was the General Chair of the IEEE International Conference on Transparent Optical Networks (ICTON) in 2011 and is currently the Chair of Optical Networks and Systems Symposium within the IEEE International Conference on Communication 2012. Since 2005, she has been the General Chair of the Workshop on Reliability Issues in Next Generation Optical Networks (RONEXT), which is a part of the IEEE ICTON.