Bulk Provisioning of LSP Requests with Shared Path Protection in a PCE-based WDM Network

Jawwad Ahmed, Cicek Cavdar, Paolo Monti, Lena Wosinska
Royal Institute of Technology (KTH), Stockholm, Sweden
Email:{jawwad, cavdar, pmonti, wosinska}@kth.se

***Abstract*— The Path Computation Element (PCE) is a network entity utilized for network path computation operations, especially useful in optical networks based on wavelength division multiplexing (WDM). In the PCE paradigm, the communication between a node and the PCE is specified by the Path Computation Element Communication Protocol (PCEP). According to PCEP protocol, multiple LSP (Label Switched Path) requests can be *bundled* together before being sent to the PCE in order to reduce the control overhead. Multiple bundles received by the PCE can then be provisioned at once as a single *bulk*. Enabling bulk provisioning of LSP requests at the PCE in a concurrent manner can bring significant improvements in terms of higher network resource utilization and control plane overhead reduction. However, these advantages come at a cost of a longer connection setup-time and of an instantaneous increase in the network load, which may lead to a degradation of the network performance, e.g. blocking probability.**

In this study pros and cons of bulk provisioning are explored in shared path protection (SPP) by comparing sequential and concurrent path computation strategies. An efficient meta-heuristic named GRASP_SPP_BP (Greedy Random Adoptive Search Procedure for Shared Path Protection with Bulk Provisioning) is proposed for concurrent provisioning of primary and shared backup path pairs. GRASP_SPP_BP minimizes the backup resource consumption while requiring minimal path computation time. The presented results demonstrate that, in a SPP network scenario, a significant reduction in the PCEP control overhead, network blocking probability and backup resource consumption can be achieved via LSP bulk provisioning at the PCE with the proposed GRASP_SPP_BP approach.

I. INTRODUCTION

Currently, a GMPLS (Generalized Multiprotocol Label Switching) based control-plane is one of the most promising solutions for the automated setup and tear down operations of LSPs (Label Switch Paths) in IP over Wavelength Division Multiplexing (WDM) networks. Path computation operations require a lot of processing power, and may be subject to multiple constraints such as wavelength continuity, physical impairments, and QoS requirements (e.g., delay, bandwidth, and load balancing). On the other hand, the GMPLS based control plane is distributed in its nature and each node in the network may or may not have the necessary processing power to perform the path computation operations. To circumvent this problem, a Path Computation Element (PCE) [1] based network can be deployed, where the PCE serves as a centralized entity specialized in solving complex multi-constrained LSP path computation/optimization problems. The PCE concept was validated in an experimental network [2], and its performance was compared with a conventional, GMPLS-based, distributed path computation approach. Results demonstrated efficient resource utilization and good network scalability of the PCE-based approach. A more detailed experimental evaluation of a PCE-based versus a distributed approach can be found in [3].

In the PCE-based networks, communication between a node (with a PCC (Path Computation Client) module) and the PCE is specified by the Path Computation Element Communication Protocol (PCEP) [4]. The PCEP protocol allows to *bundle* (i.e., pack) multiple LSP requests in a single PCReq message before being transmitted to the PCE for path computation to reduce control overhead. Similarly, multiple LSP path computation replies can be grouped together in a single PCRep message before they are sent back to the PCC [5] [6]. In addition, at the PCE, LSP requests from different bundles can be merged to form a single *bulk* before being fed to a concurrent path optimization algorithm. *Bulk provisioning* and concurrent optimization of multiple LSP requests at the PCE, *(i)* produces substantial improvements in terms of overall network optimization (i.e., concurrent versus sequential optimization), and, thanks to the PCEP bundling option, *(ii)* reduces the overall PCEP control plane overhead. However, these advantages come at the cost of an increased connection setup-time, where the main contribution is the time required for the bulk formation at the PCE. A first assessment of this trade-off was presented in [7] where multiple LSP request bundles are grouped in bulks at the PCE and concurrently provisioned. Results showed that, by choosing an appropriate bulk dimension a significant reduction in blocking probability can be achieved without a noticeable increase of the LSP setup-time. However, this study addressed only marginally the survivability problem and how protection resources can be provisioned in an efficient way for a given bulk.

Survivability, on the other hand, is a very important feature in communication networks and various protection schemes have been proposed in the literature [8]. One protection scheme that performs well in terms of resource utilization is Shared Path Protection (SPP). In a network scenario with SPP one backup path is provided for the working path of each

This work was supported by the Network of Excellence "Building the Future Optical Network in Europe" (BONE), funded by the European Commission through the 7th ICT-Framework Programme.

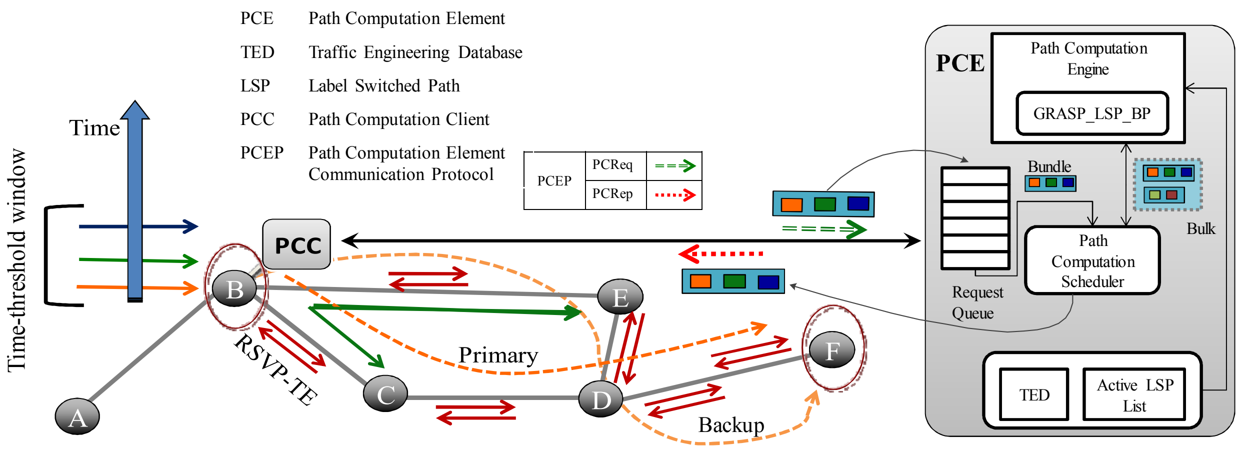

Fig.1. Proposed PCE-based Network Architecture for Dynamic Bulk Provisioning of LSP Requests

provisioned request. The peculiarity of this scheme resides in the fact that multiple backup paths can share the same wavelength resource as long as their respective working paths are *failure-disjoint*. More specifically a wavelength in a protection path can be shared only if its associated primary path is link-disjoint with the primary paths of the remaining protection paths already assigned to that wavelength. Therefore, this technique can guarantee 100% network survivability in single link failure scenario and is very efficient in backup wavelength sharing [9]. The higher the shareability level, the better becomes the overall resource usage in the network, but at the same time the more challenging it becomes to solve the path provisioning problem.

The main contribution of this paper is to evaluate the performance of the PCE-based bulk provisioning in a network scenario with SPP. The rationale for such a study lies behind the intuition that the wavelength sharability levels may greatly benefit from a concurrent path optimization strategy. With this purpose in mind a flexible GRASP (Greedy Random Adoptive Search Procedure) [10] based concurrent meta-heuristic, namely GRASP_LSP_SPP (GRASP for Shared-Path-Protection with Bulk Provisioning), is proposed along with a suitable PCE-based bulk provisioning network architecture. Simulation results demonstrate that there are significant performance gains to be achieved in terms of network resource optimization when the proposed GRASP_SPP_BP is used in the path computation phase in a bulk provisioning scenario. The proposed heuristic is flexible and individual parameters can be tuned to meet specific performance requirements. Furthermore, it is shown that even moderate-sized bulks can be processed efficiently without any noticeable increase in LSP setup-time and/or path computation time.

The paper is organized as follows. The proposed PCE-based network architecture and LSP bulk creation approach is described in section II while the proposed GRASP_LSP_SPP algorithm is presented in section III. In section IV performance results are presented to evaluate the benefits of concurrent processing of LSP requests in a network scenario with SPP, when bulk provisioning approach is utilized together with the proposed GRASP_LSP_SPP heuristic. Finally conclusions are drawn in section V.

II. PROPOSED PCE-BASED NETWORK ARCHITECTURE

The proposed network architecture based on the dynamic bulk provisioning process is depicted in Fig. 1. As mentioned before PCEP protocol is used for the communication between a PCC and a PCE and it allows sending multiple LSP requests in a single PCReq message, and similarly computed LSP replies from the PCE can also be bundled together in a single PCRep message, before sending to the PCC. This feature is exploited in the presented study to reduce PCEP control overhead.

Here, it is also important to understand the notion of *dependent* path computation. A set of path computation requests are said to be dependent if their computation cannot be performed independent of each other (i.e. path diversity requirements). Note that, since path protection capabilities are provided in the considered scenario, LSP requests in a bundle are required to be dependent according to the PCEP protocol specification [4]. For this reason the SVEC (Synchronization Vector) object is introduced in the PCReq message, and L flag is turned on in the SVEC object to force link-diversity for the computed primary and backup paths. In addition, an association is introduced between the SVEC objects in each transmitted PCReq message via a technique called *SVEC list* [11]. All the LSP requests in a SVEC list are required to be treated as *synchronized* (for concurrent optimization) by the PCE.

At each ingress node bundling is enabled via a time-threshold based approach where the PCC bundles LSP requests in a single PCReq message after a timer expires (Fig. 1). An alternative connection-threshold based approach may also be employed where each ingress node (PCC) waits for a specific number of requests to arrive before bundling is applied. However, time based threshold ensures an upper bound on the waiting time for the LSP requests on the ingress nodes before sending them to the PCE. On the PCE side these LSP request bundles are put in a Request Queue (RQ).

In addition, a counter, i.e., PCE Bundle Threshold (BundleThrsh#), is employed at the Path Computation Scheduler (PCS) to ensure that a specific number of PCReq messages are collected to form a *bulk* of LSP bundles before they are forwarded to Path Computation Engine (PCG) (Fig.1). When the path computation phase is over, the LSP path computation responses are bundled into PCRep messages (based on their respective source node) and sent back with the computed paths by the PCS. Since a SPP scenario is considered, a list of the currently active LSPs apart from the TED (Traffic Engineering Database) is required by the PCG to compute backup paths.

III. SPP WITH BULK PROVISIONING (GRASP_SPP_BP)

This section presents the proposed Greedy Random Adoptive Search Procedure (GRASP) based provisioning algorithm, i.e., GRASP_SPP_BP. The algorithm is demonstrated in three different parts. Algorithm 1 shows the main steps while Algorithm 2 and 3 define the supporting procedures called by Algorithm 1. All the requests that need to be provisioned in a given bulk provisioning phase are put in a list (referred to as D in the pseudo code) to form a bulk and to be fed to the GRASP_SPP_BP.

A. Problem Statement

Input

Input parameters of the GRASP_SPP_BP algorithm can be stated as follows: $G(v,e)$ represent the current network topology where v represents the set of nodes in the network and e is the set of edges. W represents the maximum number of wavelengths supported per fiber. D is the set of LSP requests to be provisioned. $MaxItr$ specifies the maximum number of iterations to generate and improve the solution (set of provisioned requests) and X denotes the number of provisioned LSPs in a given iteration. A provisioned LSP i comprises of k links $l_1, l_2 \cdots l_k$ for the primary path P_{Pri} and *m* links $l_1, l_2 \cdots l_m$ for the secondary path P_{Sec}. Let W_{pi} and W_{si} represent the total number of newly reserved primary and backup (shared) wavelengths by P_{Pri} and P_{Sec} respectively for a provisioned LSP i then we define the $Sum_{pri} = \sum_{i=1}^{X} W_{pi}$ and $Sum_{\sec} = \sum_{i=1}^{X} W_{si}$. Backup wavelengths W_{si} may be shared by multiple backup paths if they satisfy the shareability criteria.

Output

A denotes the set of provisioned LSPs each with primary and backup path pair where $|A| \le |D|$.

Objective

Minimize $\alpha * (|D| - |A|) + \beta * Sum_{pri} + \gamma * Sum_{\sec}$ for the generated solution where $\alpha > \beta > \gamma$. The first term in the objective with the highest weight α tries to maximize the total number of LSPs provisioned by the algorithm while the second term minimizes the weighted sum of the wavelengths used by the primary paths in the generated solution. The last term minimizes the weighted sum of the wavelengths used by the backup paths in the generated solution, i.e., to encourage shareability.

Description

Algorithm 1 goes through a number of iterations $MaxItr$ to build an initial solution and improve it in the succeeding iterations (step 1). 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) (RCL_List in the pseudo code) which is decided dynamically based on the bulk size. RCL_List is built iteratively in a greedy manner based on the allocation cost (step 6). When RCL_List is fully populated a random connection is moved to set A to build a partial (intermediate) solution. The whole procedure is re-

Algorithm 1. GRASP-LSP_SPP

Input: $G(v,e)$, W, D a set of requests to be provisioned where $D = \{R_1, R_2, ... R_n\}$ and $R_i = \{s,d\}$, $MaxItr$
Output: A set of provisioned requests A

1. **while** current iteration <= $MaxItr$
2. **while** D is not empty //compute a solution
3. $RCL\_Size \leftarrow \lceil D / 2 \rceil$ //dynamic RCL size
4. **while** size of RCL_List < RCL_Size
   if D is empty then **break**
5. **for** each LSP request R_i in D
6. Allocation cost ← **Allocation_Cost(R_i)**
7. **if** a request R_i cannot be allocated
   Remove it from D
8. Select R_i with the minimum allocation cost
   Add it to RCL_List
   Remove it from D
9. Remove a random connection R_i from RCL_List
   Allocate(R_i) //perform RWA
   Add to A
10. Compute *objective* value for current solution
11. **if** *objective* value for the current solution < best solution
    Best solution ← current solution
12. **if** current iteration < maximum iterations //if not last
    De-allocate n random connections R_i from A
    where n is a random number
    Add these connections back to D
13. **return** A

Algorithm 2. Allocate

Input: $G(v,e)$, W, $R_i = \{s,d\}$, current network state S
Output: TRUE if R_i is provisioned otherwise FALSE

1. Load pre-computed K-shortest paths for s and d of the given R_i
2. //assign primary path using WLCR if allocation possible
   P_{Pri} ←**WLCR(K-SP)**
3. Load K-shortest paths K-SP_disjoint for the s and d of the given R_i which are link-disjoint to the Primary Route
4. //assign secondary path using the WLCR if allocation possible
   P_{Sec} ←**WLCR(K-SP_Disjoint)**
5. **if** $P_{Pri} \wedge P_{Sec}$ is not null
   Do wavelength assignment for selected P_{Pri} and P_{Sec} using First-Fit
6. **return** TRUE
7. **else return** FALSE

Algorithm 3. Allocation_Cost

Input: $G(v,e)$, W, $R_i = \{s,d\}$, current network state S
Output: allocation cost W_{resv} if R_i is provisioned

1. **Allocate(R_i)** //a temporary allocation
2. $W_{Pri,\,resv}$ ← calculate newly reserved wavelengths by P_{Pri}
3. $W_{Sec,\,resv}$ ← calculate newly reserved wavelengths by P_{Sec}
4. $W_{resv} \leftarrow W_{Pri,\,resv} + W_{Sec,\,resv}$
5. //release all wavelength resources associated with this allocation
   De-allocate current request R_i
6. **return** W_{resv}

peated until a complete solution is built. In step 11 this solution is saved if the value of the objective is the best so far.
Step 12 checks if the total number of iterations is reached. If not then n random connections are selected to be re-computed. These connections are 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 GRASP_SPP_BP in addition to the input size $\left|D\right|$.

In Algorithm 2, k pre-computed shortest paths are loaded in memory in step 1. A suitable primary path P_{Pri} is assigned in step 2 using the WLCR (Weighted Least Congested Routing) algorithm [12]. k link-disjoint shortest paths are loaded for P_{Pri} in step 3. In step 4 WLCR is used again to assign a suitable backup path P_{Sec} for the selected primary path P_{Pri}. If both primary and secondary paths can be computed then the procedure returns TRUE in step 6 otherwise a FALSE is returned.

Algorithm 3 is called by Algorithm 1 to compute the allocation cost of a given Ri. A *temporary* allocation is performed for Ri in step 1. The number of newly allocated wavelengths needed for P_{Pri} and P_{Sec} is stored in $W_{Pri,\,resv}$ and $W_{Sec,\,resv}$ during step 2 and step 3, respectively. $W_{Pri,\,resv}$ and $W_{Sec,\,resv}$ are summed up in step 4, and step 5 frees the resources used to temporarily allocate Ri. The total allocation cost W_{resv} for the current Ri is then returned in step 6.

To calculate time-complexity of the GRASP_LSP_SPP it can be observed that first part in Algorithm 1 is computationally dominant (step1-9). An upper bound for this part is $(MaxItr) * (\left|D\right|) * (\left|D\right| / 2) * ((\left|D\right| + \left|D\right| - (\left|D\right| / 2) + 1) / 2)$ from step 1, 2, 4 and 5, which equates to $MaxItr * (\left|D\right|^3)$ by ignoring terms with lower powers of $\left|D\right|$ and constant multipliers in the resulting formula. So, time-complexity of the GRASP_LSP_SPP is $O(\left|D\right|^3)$ for any fixed value of $MaxItr$. Note that to initially calculate pre-computed K-shortest paths for each of the $v * (v-1)$ node pairs using Bellman-Ford algorithm requires a time-complexity of $O(K * v * e)$.

IV. Performance Evaluation

In order to evaluate the performance of the proposed GRASP_LSP_BP algorithm a Java based discrete event-driven simulator is used on the NSF network topology with 14 nodes and 20 bidirectional links (Fig. 2). Each link in the network corresponds to one fiber with 16 wavelengths, and the Wavelength-Continuity-Constraint (WCC) is enforced for path computation. Two scenarios are considered for SPP LSP request processing: (i) sequential, when all the requests in a PCReq message are processed at the PCE in a sequential manner (i.e., in the order of arrival) and (ii) concurrent, when these requests are processed in a combinatorial manner (i.e., based on the proposed solution). For the GRASP_SPP_BP case $MaxItr$ = 100 and values for α, β and γ are set to 1000.0, 1.0 and 1.0 respectively. For the sequential case both primary and secondary routes are computed using the WLCR [12] algorithm and First-Fit is used as the wavelength-assignment algorithm. The arrival rate (λ) of the LSP requests follows a Poisson distribution and the service time ($1/\mu$) is assumed to be exponentially distributed. A RSVP-TE based signaling protocol is employed for resource reservation after the path computation phase. For the signaling phase the switch configuration time (T_S) at the nodes is assumed to be 50 ms and the node processing delay (T_P) is assumed to be 20 ms. The performance parameters considered during the evaluation phase are: blocking probability (BP), avg. link utilization, PCEP control overhead, average LSP setup-time, resource overbuild (ratio of the length of the backup path to the primary path), average hop count per LSP, and RWA computation time (i.e. time duration between arrival of an LSP request to the PCG and provisioning of that request). The estimation of the PCEP control overhead also includes: TCP, IP and Ethernet overhead, assuming that the control-plane is implemented over Ethernet.

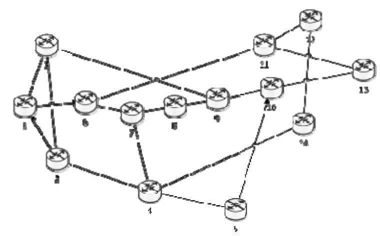

Figure 2: 14-Node NSF Topology

LSP setup-time defines the time interval from when an LSP request enters the system (at the ingress node) to the time instance in which the LSP request is successfully signaled in the network and ready for data transmission. It includes path computation, communication, queuing time and time necessary for reserving the computed path through the network (i.e., signaling time). A stateful PCE (SF-PCE) [13] is assumed (i.e., maintaining the current lightpath configuration state) for SPP path computation. A baseline scenario is also used for benchmarking purposes in which LSP requests are provisioned one by one with no bundling. The mean LSP service-time $1/\mu$ is fixed at 100 s, while the arrival-rate λ is assumed to be 1/110 arrivals per second per node pair, resulting in a total-load in the network equal to 82.0 Erlangs. The simulation platform is a Red Hat Enterprise Linux workstation with 12 GB of memory and a dual Intel Xeon CPUs (4 cores per CPU) clocked at 2.0 GHz.

The PCE BundleThrsh# is set to 1 for the first set of results, i.e., from Fig. 3(a) to Fig. 3(e). Fig. 3(a) shows a significant reduction of BP by GRASP_SPP_BP compared to WLCR even at a time-threshold value of 5 s and the gain continues to grow for higher values of the time-threshold. BP tends to slightly increase after a time-threshold of 20 s because larger bundle sizes increase the instantaneous network load as the network traffic from a specific source tends to become burstier. However, at lower time-thresholds (e.g., lower than 20 s), BP drops because this effect is weaker here as compared to the gain provided by the concurrent optimization. For the WLCR case the trend is similar but with a much steeper curve. Fig. 3(a) also shows the mean bundle size which is the average number of LSP requests processed by the PCE in each bulk provisioning phase. It grows slowly with increasing time-threshold values. The rate of growth of the mean bundle size is dependent on the arrival rate.

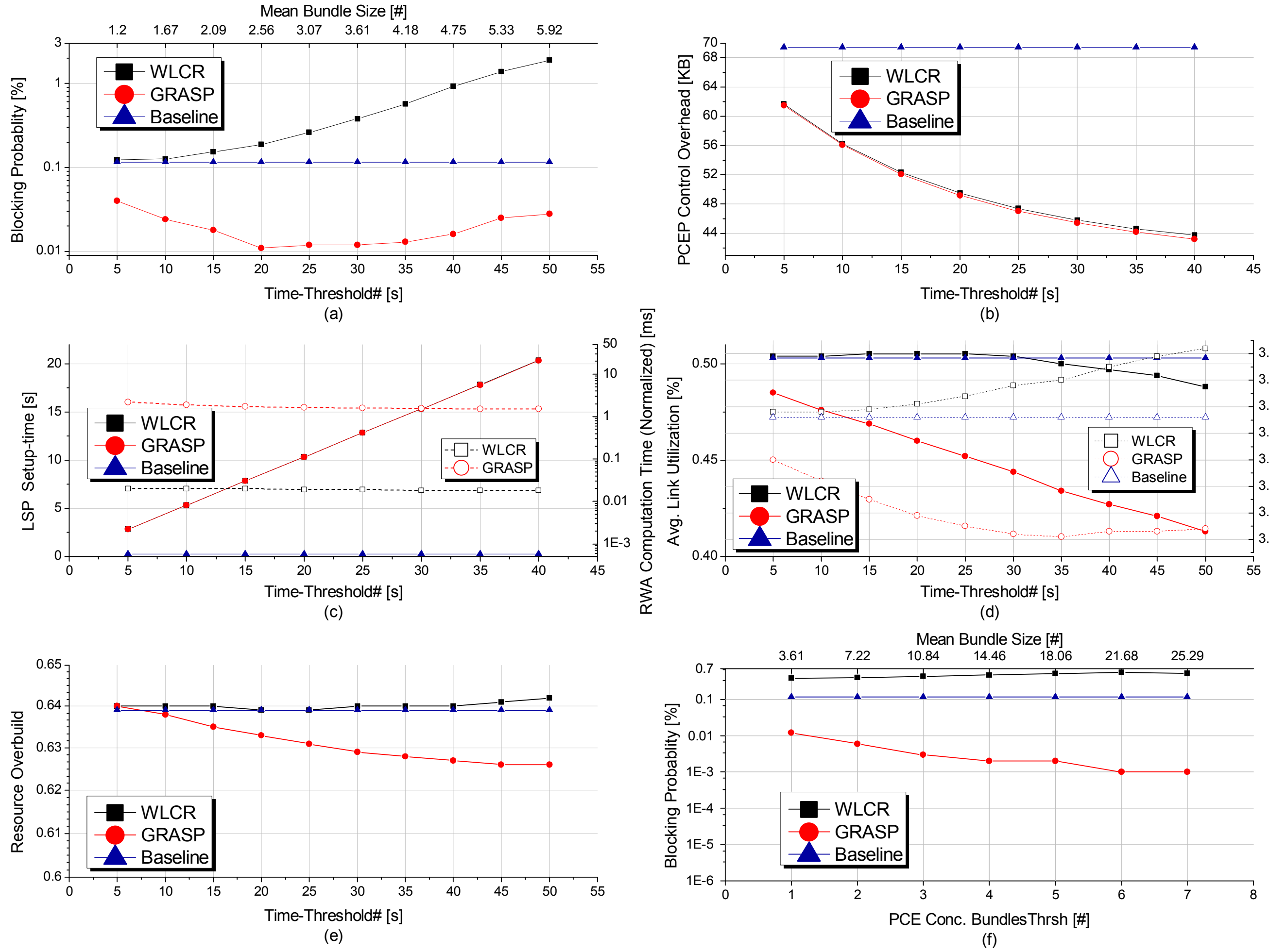

Fig. 3(a): Blocking Probability vs. Time-threshold & Mean Bundle Size, Fig. 3(b): PCEP Control Overhead vs. Time-threshold, Fig. 3(c): Avg. LSP Setup-time & RWA Computation Time vs. Time-threshold, Fig. 3(d): Avg. Link Utilization & Avg. Hop Count vs. Time-threshold, Fig. 3(e): Resource Overbuild vs. Time-threshold, Fig. 3(f): Blocking Probability vs. PCE BundleThrsh & Mean Bundle Size

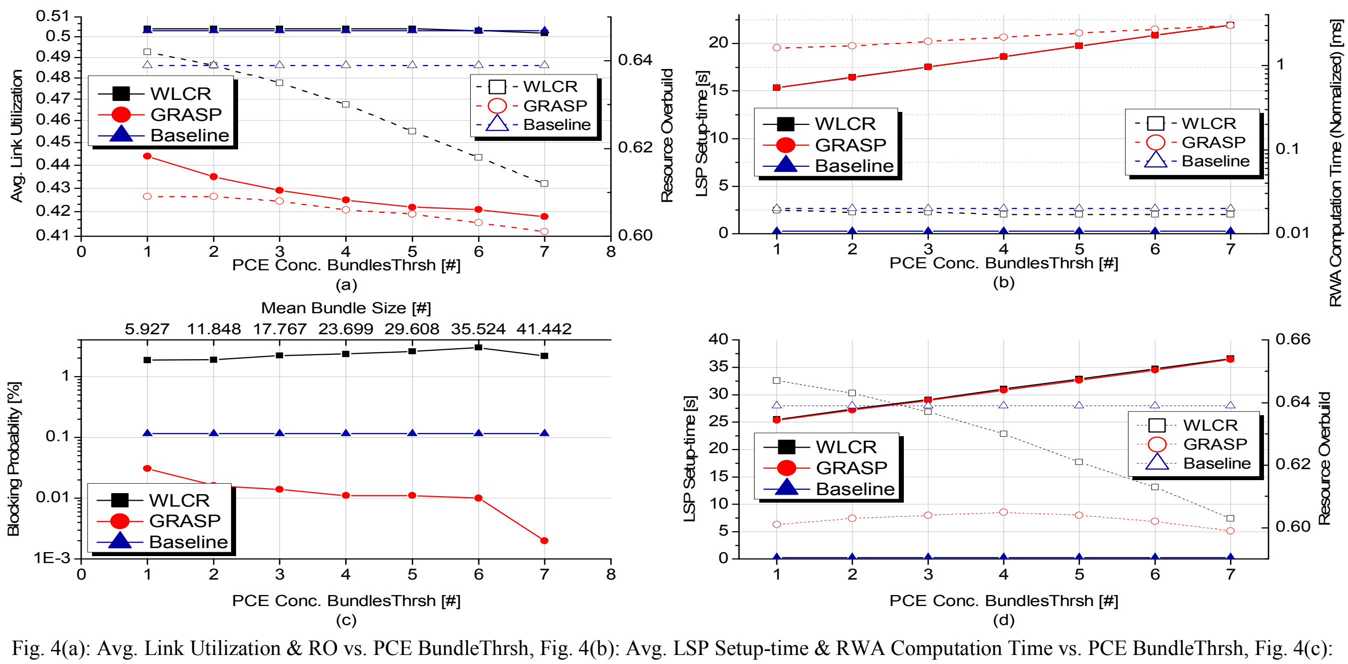

Fig. 4(a): Avg. Link Utilization & RO vs. PCE BundleThrsh, Fig. 4(b): Avg. LSP Setup-time & RWA Computation Time vs. PCE BundleThrsh, Fig. 4(c): Blocking Probability vs. PCE BundleThrsh & Mean Bundle Size, Fig. 4(d): Avg. LSP Setup-time & Resource Overbuild vs. PCE BundleThrsh

Fig. 3(b) shows that PCEP control overhead reduction is significant (about 35%) when the time-threshold value reaches 30 s. After this point the curves for both WLCR and GRASP_SPP_BP tends to flatten. Furthermore, the LSP setup-time (Fig. 3(c)) increases linearly with the time-threshold value for both GRASP_SPP_BP and the WLCR case. It also shows that the gap between the two cases is negligible which proves that proposed GRASP_SPP_BP for concurrent RWA computation is an attractive and viable choice to be deployed in real PCE-based WDM network scenarios. Note that although the actual RWA computation time per LSP (Fig. 3(c)) is significantly higher for the GRASP_SPP_BP case as compared to the WLCR, it is still negligible (≈ 2.0 ms) for each successful LSP request. Fig. 3(d) shows that GRASP_SPP_BP makes more efficient usage of wavelength resources in the network. The link utilization drops with the increase in the time-threshold value as bigger bundles allow better resource optimization. Fig.3(d) also shows that the average hop count for the paths computed by GRASP_SPP_BP is much lower as compared to the WLCR and it tends to drop for higher time-threshold values. This also explains the reason for the lower link utilization in the GRASP_SPP_BP case.

Fig. 3(e) shows the gain in terms of the resource overbuild as a function of the value of the time-threshold. It is shown that efficient backup capacity usage is achieved by GRASP_SPP_BP as the time-threshold increases.

In order to investigate the effect of the PCE BundleThrsh#, the value of the time-threshold at the PCC is set to 30 s in Fig. 3(f), 4(a) and 4(b) in the second set of results. Fig. 3(f) shows that BP tends to drop with increasing number of LSP requests in one bulk for the GRASP_SPP_BP because a higher values of BundleThrsh# allows to process requests from a more diverse set of ingress nodes. The opposite is true for WLCR because of its limitation in the sequential processing of LSP requests.

Benefits of GRASP_SPP_BP in terms of avg. Link utilization and resource overbuild can be seen in the Fig. 4(a). The gain increases with the growing number of bundles from different sources. However the LSP setup-time also continues to increase for both GRASP_SPP_BP and WLCR (Fig. 4(b)). Nevertheless the difference in terms of LSP setup time is still not noticeable.

PCC time-threshold value is fixed to 50 s in the third set of results as shown in Fig. 4.(c) and (d). BP in Fig. 4(c) shows the same behavior shown earlier for the time-threshold value of 30 s in Fig. 3(f). However absolute values for the BP are slightly higher in Fig. 4(c). It can be observed from both the second and the third set of results that most benefit can be gained in terms of network resource optimization if an appropriate time-threshold value is selected for the source nodes as well as an appropriate value of PCE BundleThrsh#. This is because a value of PCE BundleThrsh# (>1) allows the bulk formed at the PCE to contain requests from a more diverse set of source nodes which enhances the degree of optimization that can be achieved by the GRASP_SPP_BP.

LSP setup-time and resource overbuild values are shown in Fig. 4(d). It is evident that even for large bulk sizes (i.e. 35 and 41) the difference between the GRASP_SPP_BP and WLCR is negligible for the LSP setup-time. Resource overbuild saturates at around 0.60 for all plotted values of PCE BundleThrsh#. So, a good value of the BundleThrsh# is a trade-off between BP and LSP setup-time, and is dependent on the maximum tolerable LSP setup-time in a given scenario.

V. Conclusion

In this study a meta-heuristic referred to as GRASP_SPP_BP is proposed for dynamic bulk provisioning of LSP requests with shared path protection in a PCE-based WDM network. In addition, a suitable PCE-based network architecture is also proposed which is then used as the *proving ground* for the GRASP_SPP_BP. Based on the results it can be observed that there are significant gains in terms of blocking probability reduction via bulk provisioning of LSP requests at the PCE for the SPP. This improvement is made possible by the ability of the proposed heuristic to maximize the sharebility level of the provisioned backup paths. This result is confirmed also by the low value for the resource overbuild presented by the proposed GRASP_SPP_BP heuristic. Finally, it is shown that the RWA computation-time is negligible for the GRASP_SPP_BP and that it doesn't negatively affect the LSP setup-time even for large bulk sizes when compared to a sequential solution.

References

[1] A. Farrel, J.-P. Vasseur, and J. Ash, "A Path Computation Element (PCE) Based Architecture", IETF RFC 4655, Aug. 2006.

[2] F. Cugini, F. Paolucci, L. Valcarenghi, P. Castoldi, "Implementing a Path Computation Element (PCE) to encompass physical impairments in transparent networks", in Proceedings of OFC/NFOEC 2007, Anaheim, CA, USA, March 2007.

[3] P. Castoldi, F. Cugini, L. Valcarenghi, N. Sambo, E. Le Rouzic, M.J. Poirrier, N. Adriolli, F. Paolucci and A. Giorgetti, "Centralized vs. Distributed Approaches for Encompassing Physical Impairments in Transparent Optical Networks", in Proceedings ONDM 2007, Athens, Greece, May 2007.

[4] J.-P. Vasseur and J. L. Le Roux, "Path Computation Element (PCE) Communication Protocol (PCEP)", Internet draft, Nov. 2008.

[5] J. Ahmed, P. Monti, L. Wosinska, "LSP Request Bundling in a PCE-Based WDM Network", in Proceedings of OFC/NFOEC 2009, San Diego, CA, USA, March 2009.

[6] J. Ahmed, P. Monti, L. Wosinska, "Benefits of Connection Request Bundling in a PCE-based WDM Network", in Proceedings of NOC/OC&I 2009, Valladolid, Spain, June 2009.

[7] J. Ahmed, P. Monti, L. Wosinska, "Concurrent Processing of Multiple LSP Request Bundles on a PCE in a WDM Network", in Proceedings of OFC/NFOEC 2010, San Diego, CA, USA, March 2010.

[8] B. Mukherjee, Optical WDM Networks. Springer, Jan. 2006.

[9] C. Ou, J. Zhang, H. Zang, L.H. Sahasrabuddhe, and B. Mukherjee, "New and Improved Approaches for Shared-Path Protection in WDM Mesh Networks," Journal of Lightwave Technology, vol. 22, 2004

[10] T.A. Feo and M.G. Resende, "Greedy Randomized Adaptive Search Procedures," Journal of Global Optimization, vol. 6, 1995, pp. 109-133.

[11] Itaru Nishioka, Daniel King, "The use of SVEC (Synchronization VECtor) list for Synchronized dependent path computations", Inernet draft, June 2010.

[12] X.-W. Chu, et al., "A Dynamic RWA Algorithm in a Wavelength-Routed All-Optical Network with Wavelength Converters," IEEE INFOCOM 2003.

[13] P. Castoldi, F. Paolucci, A. Giorgetti, F. Cugini, "Management-augmented stateless PCE for Wavelength Routed Optical Networks", in Proceedings of ICTON 2008, Athens, Greece, June 2008.