Adaptive SLA-Aware Algorithms for Provisioning Shared Mesh Optical Networks

Alireza Nafarieh, William Robertson, William Phillips, and Shyamala Sivakumar

***Abstract*—The paper proposes a novel SLA-aware mechanism to improve the performance of the networks with huge traffic volume of high priority connection requests with long holding time for optical shared mesh networks. The proposed mechanism consists of two parts, a novel re-provisioning algorithm which buffers and further processes the potentially blocked high priority connection requests, and a new time-aware path constraint which takes advantage of availability and holding-time as two crucial SLA connection parameters. The proposed mechanism benefits from dynamic service level agreement negotiation between a customer and service providers. The simulation results show reduced blocking probability, increased availability satisfaction rate, decreased resource overbuild, and better resource utilization to preserve the high priority class of traffic compared to other SLA-aware algorithms and protection schemes in shared mesh optical networks.**

***Index Terms*—SLA-aware re-provisioning algorithm; dynamic service level agreement negotiation; maximum path availability algorithm; time-aware maximum path availability.**

I Introduction

Survivable wavelength division multiplexing (WDM) mesh optical networks play a crucial role on serving recent tremendous growth in the Internet traffic demand. The increasing demand on quality of service (QoS)-based traffic which carry huge amount of high-priority traffic class requires new traffic-engineering strategies along with provisioning algorithms and protection schemes to be developed. The new strategies typically take the service level agreement (SLA)-aware algorithms into account to maintain a satisfying level of QoS for the requested connection with regard to the parameters requested in SLA. Connection availability is one of the most important QoS parameters specified in SLA between customers and service providers over survivable WDM mesh networks. In addition to connection availability, connection holding time is another connection request characteristics which can play an important role in developing priority-aware algorithms for preserving high priority requests.

The paper contribution follows three main characteristics, proposing a new time-aware path metric, introducing a novel re-provisioning algorithm based on the proposed metric, and applying a huge volume of high-priority traffic with long duration to the introduced algorithm.

As discussed in [1], [2], based on SLA contracts or bandwidth-leasing markets between network operators and customers, it is reasonable to assume that the connection holding time can be known in advance while the algorithm is serving online type of traffic in which the algorithm has no knowledge about the coming request. As the first part of the contribution in this paper, we take advantage of a combination of the connection availability, the connection holding-time, and the maximum path availability [3] of the request to introduce a novel path metric. The proposed re-provisioning algorithm presented in this paper as the second part of the contribution benefits from the new path metric to better serve high-priority connection requests. To achieve this goal, we have assumed that the period during which a connection is valid, holding-time, is known *a priori*.

The second part of the contribution is to introduce a novel re-provisioning mechanism by which some of those high-priority connection requests which were blocked in other SLA-aware algorithms or protection schemes are preserved. The paper focuses on specific type of traffic which is of dynamic type and mainly high priority class with long duration while the majority of the existing algorithms either take small portion of the traffic as high priority into account or consider short connection holding times. The proposed re-provisioning mechanism is typically a dynamic provisioning algorithm including a time-aware buffering mechanism which is working based on the new metric presented in this paper. The re-provisioning algorithm employs three algorithms, an algorithm to calculate the maximum path availability reviewed from previous work [3], an algorithm to find the matrix of time-aware maximum path availability of each connection, and an algorithm by which the potentially blocked connections are buffered and served based on the connections holding time. The paper compares the performance of the proposed algorithm to either standard or other existing SLA-based or priority-aware algorithms.

The paper is organized as follows: in Section II, the related and previous work on connection holding-time based and priority-aware algorithms are discussed. Section III introduces the re-provisioning algorithm, a new path constraint used by the proposed algorithm, and involved algorithms in the proposed algorithm. Section IV talks about the simulation environment and the performance analysis of the proposed algorithm. Section V presents conclusion and future work.

Manuscript received May 14, 2012; revised July 20, 2012. This work was supported in part by the Internetworking Engineering Program of Dalhousie University.

A. Nafarieh is with the Internetworking Engineering Department of Dalhousie University, Halifax, NS, Canada (e-mail: ali.nafarieh@dal.ca).

W. Robertson is with the Internetworking Eng. Department of Dalhousie University, Halifax, NS, Canada (e-mail: billr@dal.ca).

W. Phillips is with the Engineering Mathematics Department of Dalhousie University, Halifax, NS, Canada (e-mail: w.phillips@dal.ca).

S. Sivakumar is with the Department of Information Systems of SMU, Halifax, NS, Canada (e-mail: shyamala.sivakumar@smu.ca).

DOI: 10.7763/IJIEE.2013.V3.369

II RELATED WORK

Previous work in [3], [4] have introduced priority-aware algorithms that take advantage of proposed path metric, maximum path availability. The algorithms try to improve the performance of the high priority requests with regard to blocking probability and resource utilization comparing to conventional shared-mesh protection schemes [5], [6] and the SLA-aware algorithm [7]. In [4], a static priority-aware pre-provisioning algorithm has been proposed based on the SLA parameters negotiation for shared-mesh WDM networks. Although the pre-provisioning algorithm in [3] benefits from static SLA parameters negotiation, no dynamic mechanism for dissemination of path availability information has been considered. Likewise in [3], the proposed algorithms have applied for small number of high-priority requests with short durations.

The conventional shared mesh algorithm presented in standards [5], [6] takes advantage of constraint-based shortest path algorithm for path calculation. Although link availability is one of commonly used parameters in SLA negotiations, the algorithm does not consider either the availability of the links or holding-time of the connections as the constraints in their path calculations.

The core idea presented in [8], [9] consists of exploiting the knowledge of the holding time of connection requests to minimize resource overbuild in form of backup capacity and hence achieve better resource-usage efficiency. The algorithm proposed in [8] improves sharing of backup resources by applying the holding time directly in the path computation process. In [9], a new holding time based algorithm has been proposed by which all existing connections that have not been affected by failures during their lifetime are allocated a new SLA availability target which is a function of holding-time.

SLA-based algorithms working based on connection holding-time have been discussed in [10]-[12] as well. As authors in [10] propose, if the SLA requirement of a connection is violated, one more wavelength on each backup link of the availability-downgraded connection is newly assigned. The new cost function considers the capacity of the backup paths so that the availability requirements are satisfied. In [11], [12], SLA-based algorithms have been proposed to accommodate customer-specific requirements. These algorithms assume that certain periods may need extra protection. The mechanisms in [11], [12] propose that a connection can be protected with different SLA requirements at different times over the entire holding time of the connection.

Unlike the above mentioned algorithms, the algorithm proposed in this paper does not affect the routing and wavelength assignment process and does not change the process of the primary and backup path calculations, but takes advantage of holding time as *a priori* knowledge to determine when the best time is to serve and re-provision the already buffered high-priority requests. The proposed algorithm in this paper considers the traffic of a number of high priority requests with long durations. In addition, in this paper, it is assumed that the SLA requirement of the connection is not changing over the holding-time of the connection.

III PROPOSED ALGORITHM

A. Adaptive Re-Provisioning SLA-Aware (ARSA) Algorithm

Fig. 1 shows the modules involved in the ARSA algorithm. Connection requests are served one by one from the connection requests matrix (CRM). The request is applied to time-aware maximum path availability (TMPA) and maximum path availability (MPA) algorithms to be served either as an original request or as a buffered request for further processing. The TMPA algorithm uses the previously established connections matrix (ECM) as an input and interacts with the MPA algorithm [3] for the calculation of the proposed path metric. Then the buffering high priority connection (BHC) algorithm is applied by which CRM is updated with a modified arrival time for the connection request. The elaborated diagram on how the ARSA algorithm works is shown in Fig. 2.

After each request is processed, the graph topology and wavelength usage matrices are updated. Each high priority request is established, blocked, or buffered for further processing. The low-priority traffic is handled by routing and wavelength assignment module with no further processing. The n^{th} connection request is in form of C_n *(s, d,* A_r*, p,* $T_{arrival}$*)* with the requested parameters: source, *s*, destination, *d*, availability, A_r, the priority level, *p*, and the arrival time, $T_{arrival}$, respectively.

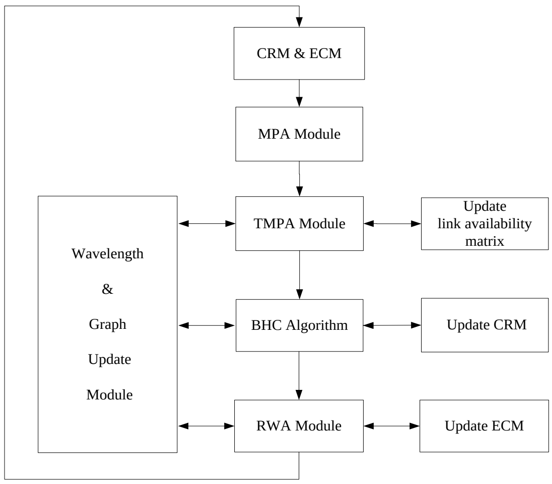

Fig. 1. ARSA algorithm modules interaction

Algorithm 1 introduces the ARSA algorithm. When a new high-priority connection is requested, ARSA calculates MPA matrix by running MPA Algorithm discussed in [3] to find out whether the Gold request meets the best offer made by the service provider. If it does, ARSA applies RWA module to the request. If the Gold connection requirements are not met, ARSA runs Algorithm 2 to modify the request and buffer the connection through TMPA and BHC modules. When a Gold request has been buffered, ARSA employs BHC module using Algorithm 3 to serve the high-priority request. Since the ARSA algorithm has been designed to better serve high-priority requests, it does not apply any further processing to low-priority requests other than regular RWA module.

Algorithm 1. ARSA algorithm

Input: CRM, C_n (s, d, A_{rCn}, p), n=1
Output: Graph and wavelength matrices, updated CRM

1. Serve the n^{th} connection request from $CRM_{(n)}$
2. Calculate $MPA_{(s,d)}$ using MAP Algorithm [3]
3. If $MPA_{(s,d)}=0$
   Block the connection
   Else *continue*
4. If $A_{rCn} \geq MPA_{(s,d)}$
   Go to 7
   Else
   Find the primary/backup paths of the connection using the RWA module
5. Update wavelength and graph matrices
6. Remove the connection from CRM
7. If p=Gold
   Calculate $TMPA_{(s,d)}$ through Algorithm 2
   Else
   Block the connection
8. Serve and buffer the connection through the BHC module applying Algorithm 3
9. $n \leftarrow n+1$ & go to step 1

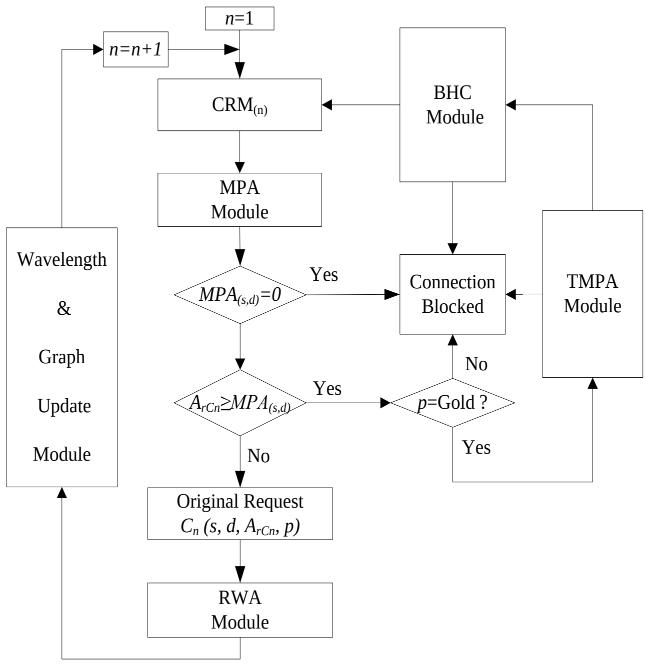

Fig. 2. ARSA Algorithm block diagram.

B. *TMPA Module*

Algorithm 2 shows how the TMPA and MPA matrices are calculated in TMPA module. The $MPA_{(s,d)}$ which can be disseminated around the network by the dynamic SLA negotiation mechanism [13] carries the release time of the connection. The release time is the time during which the value of MPA is valid for the connection. The TMPA matrix contains the MPA values of the connections of a specific pair of source and destination and their associated release times.

MPA_0 is the available MPA for the current served connection request, $C_0(s, d, A_{r0}, p, T_{0arrival}, T_{0holding})$, at the current moment, T_0. $ECM_{(s,d)}$ is the matrix of all previously established connections, including Gold and Silver requests, for a certain pair of source and destination. The *MPA* values of the *TMPA* matrix, MPA_{Cj}, are the potential maximum path availability for the given pair of source and destination when the links forming the primary and backup paths of associated connections, C_j, from the first to the j^{th} row of the TMPA matrix are released. The Algorithm 2 calculates the $MPA_{m \times m}$ matrix [3] for a network topology of m nodes. $MPA_{(s,d)}$ calculated in MPA algorithm is the maximum offered path availability for a certain source-destination pair in the n^{th} connection request, and has been discussed in detail in previous work [3]. The parameters advertised in the dynamic SLA mechanism are the availability of the links forming the graph. However, proper SLA negotiation [13] needs the information about the availability of all possible paths for any pair of source and destination.

It is assumed that there is an automatic mechanism for SLA parameters negotiation between service providers and customers to propagate MPA information all over the network. The protocols used for dynamic SLA negotiation have been discussed in [13] in detail. The authors in [13] show how MPA information is disseminated over intra/inter-domain of a network of including multiple autonomous systems.

Algorithm 2. TMPA algorithm

Input: CRM, $C_0(s, d, A_{r0}, p, T_{0arrival}, T_{0holding})$, k=1
Output: $TMPA_{(s,d)}$ matrix, $T_{(i,j)}$, $MPA_{(i,j)}$

1. Build the ECM matrix of associated (s,d) pair

$$\forall j \in \{1,2,\ldots,m\} \;\&\; \forall\, C_j \in CRM: \qquad ECM_{(s,d)} = \begin{bmatrix} C_{1(s,d)} & T_{1(s,d)} \\ C_{2(s,d)} & T_{2(s,d)} \\ \vdots & \vdots \\ C_{j(s,d)} & T_{j(s,d)} \\ \vdots & \vdots \\ C_{m(s,d)} & T_{m(s,d)} \end{bmatrix}$$

where m is the number of established connections per s-d pair, $C_{j(s,d)}$ is the j^{th} connection established from s to d, and $T_{j(s,d)}$ is the associated release time of $C_{j(s,d)}$

2. If $ECM_{(s,d)}$=*empty*
   Block the connection
   Else *continue*
3. Sort the columns of ECM based on the release times

$$T_{release} = T_{holding} + T_{arrival}$$

4. If k=1
   $TMPA(1) = [MPA_{C_{0(s,d)}} \quad T_{0(s,d)}]$
   Else *continue*
5. Save current wavelength, graph, and link availability matrices in new matrices
6. Release $C_{1(s,d)}$ to $C_{k(s,d)}$ from the $ECM_{(s,d)}$ matrix (associated primary and backup paths of the 1^{st} to k^{th} connections (rows) of the $ECM_{(s,d)}$ matrix)
7. Update new wavelength, graph, and link availability matrices
8. Calculate the MPA of k^{th} connection from s to d, $MPA^k_{(s,d)}$, through MPA Algorithm [3]
9. Calculate the k^{th} row of the *TMPA* matrix:

$$TMPA^k_{(s,d)} = [MPA^{C_j}_{(s,d)} \quad T_j]$$

10. If $A_{r0} \leq MPA^k_{(s,d)}$
    Serve the connection in the BHC module employing Algorithm 3
    Else *continue*

11. If k < size(ECM)
    $k=k+1$ & go to 5
12. Repeat steps 1-11 to build the matrix $\mathrm{TMPA}_{(s,d)}$:

$$TMPA_{(s,d)} = \begin{bmatrix} MPA_{(s,d)}^{C_0} & T_0 \\ \vdots & \vdots \\ MPA_{(s,d)}^{C_j} & T_j \\ \vdots & \vdots \\ MPA_{(s,d)}^{C_k} & T_k \end{bmatrix}$$

13. Repeat steps 1-12 to build the entire matrix TMPA

$$\forall s \;\&\; d \in \{1,2,\dots,m\}: \quad TMPA_{mxm} = [TMPA_{(s,d)}]$$

C. BHC Module

Algorithm 3 discusses the steps involved in the buffering high-priority connection (BHC) module. It is assumed that the j^{th} connection request arriving at T_0 is a Gold request whose requirements have not been met and will potentially be blocked by the RWA module if no further processing is applied to it. Threshold buffering period of the queued connection is presented as T_{th}. As observed in Fig. 3, it is assumed that T_0/N_{HP} Gold connections have been requested in the time interval T_0. The following equations show how threshold buffering period is calculated and on which parameters relies.

Fig. 3. Arrival interval of the next possible Gold request after establishing C_0.

$$\begin{cases} T_{th} = T_0 + \nu * T_{sd(avr)} \\ T_{sd(avr)} = \dfrac{T_0}{N_{HP}} \\ N_{HP} = \xi * N_{sd} \quad\quad 0 < \xi \le 1 \end{cases} \quad \rightarrow \quad T_{th} = (1 + \frac{\nu}{\xi * N_{sd}})T_0$$

where T_0 is the time at which the connection C_0 is being processed, T_{th} is the threshold time during which the request will be queued, N_{HP} is the number of the established high-priority connections between either s-d or d-s pairs in the time interval T_0, N_{sd} is the total number of the requests between either s-d or d-s pairs in the time interval T_0, ν is an integer, ξ shows the percentage of high priority requests, and $T_{sd(avr)}$ is the average period after which a high-priority request between either s-d or d-s pairs may show up after T_0.

The algorithm looks for the minimum amount of time in the TMPA matrix at which the connection requirements are met. That is, the algorithm will investigate how long the request should be buffered to have better chance to be established. If the requested availability of the Gold request is higher than the $MPA_{(s,d)}$, the BHC module checks the TMPA matrix row by row to find which row satisfying the connection requirements has smaller release time. When the BHC module finds the TMPA matrix's row which optimally suits the connection requirements, it modifies the request's arrival time and buffers it in the CRM matrix as a connection request with a new arrival time.

Algorithm 3. BHC module

Input: $\mathrm{ECM}_{(s,d)}$, $C_j(s, d, A_r, p, T_{arrival})$, $TMPA_{(s,d)}(mxm)$
Output: $C'_j(s, d, A_r, p, T'_{arrival})$

1. For all values k={1,…,m}:
   Check the $TMPA$ matrix
   Find the k^{th} row of the $TMPA$ matrix in which:
   $MPA^k_{(s,d)} \ge A_{rCj}$
2. If $T_k < T_{th}$
   $T'_{arrival} = T_k + \varepsilon$
   Update CRM with the modified connection request:
   $C'_j(s, d, A_r, T'_{arrival})$
   Else Block C_j
3. For all values of $MPA^k_{(s,d)}$ in $TMPA_{\mathrm{mxm}}$:
   If $A_{rCj} > MPAk_{(s,d)}$
   Block C_j

D. RWA and Updates Modules

The RWA and wavelength update modules have been discussed in [3] in detail. The routing scheme which is used in the RWA module to determine the primary and backup paths is adaptive routing [14]. Before a path computation algorithm, *Dijkstra's*, is applied to the request to find the primary path, the cost of the links of the graph is modified by the primary path cost function presented in [3]. Based on the cost function, if there is no bandwidth available on the link, the link is removed from the graph; otherwise the cost of the link is a function of the link availability. Following this step the wavelengths are assigned to the path based on a per-link basis, and the wavelength usage matrix is updated in this step. If the path computation finds no path from s to d, the request is blocked. The wavelength assignment of the primary paths follows the First-Fit (FF) algorithm [15]. In the FF technique, the wavelengths are numbered and the lowest numbered free wavelengths are selected.

Before the RWA module calculates the backup path, the cost of the links of the graph are changed one more time based on the backup path cost function presented in [3]. As this equation denotes, to setup a backup path, the algorithm looks for the paths with the highest available bandwidth and lowest number of shared paths on each link. The algorithm checks if the backup path can share any wavelength considering link-disjointness constraint. Then, it follows FF technique to allocate a wavelength to the links forming the path. After calculating the primary path and finding totally link-disjoint primary-backup paths pair, the wavelength and graph update module modifies the graph topology matrix by removing the links forming the primary path. If the request is of connection release type, in addition to the graph and wavelength matrices, the TMPA matrix is also updated.

IV Performance Analysis

A. Simulation Environment

The topology selected for the simulation is NSFNet shown in Fig. 4. The links have wavelength conversion capability with 8 wavelengths per link. The links have the same distance. The link availabilities are uniformly distributed between 0.99 and 0.9995. Connection availability requests are uniformly distributed between two classes of traffic: Gold class with the availability of 0.9999, and Silver class with the availability of 0.999. To simulate an environment with a large number of high-priority requests, we have considered the percentage of high-priority requests a constant value of ξ=50% for analysing the effect of connection duration on network performance discussed in Subsection B, and variable values ranging from ξ=10% to 80% for analysing the effect of number of high priority requests discussed in Subsection C. The connections arrival is a Poisson process with constant arrival rate of β=40. The holding time of the connections follows an exponential distribution with a mean value ranging from μ=50 to 500. For the sake of simplicity, we have assumed that v is large enough. The primary and the backup paths are considered to be completely disjoint and the simultaneous failure of primary links is highly unlikely. The total number of connection requests over entire simulation period is 10^5.

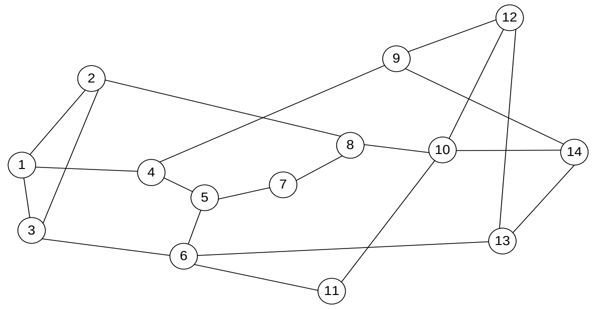

Fig. 4 NSFNet network topology

In this paper, the dynamic traffic type has been selected for performance analysis, and the availability satisfaction ratio (ASR), the blocking probability (BP), and the average number of link-wavelength per connection (AWPC) of the ARSA algorithm are compared with other standard and existing algorithms. The performance of the ARSA algorithm is compared with the schemes in which there are either no automatic SLA negotiations like SSPP [5] and [6], or SLA-aware algorithms such as SPA presented in [4] and the algorithm presented in [7], known in this paper as SLA-aware.

B. *The Effect of Connections' Duration*

In Fig. 5, BP denotes the percentage of blocked connection requests over all arriving requests. Fig. 5 compares blocking probability of the ARSA algorithm with SSPP [5], [6], SLA-aware [7], and SPA [4] algorithms. As shown in Fig. 5, ARSA has better blocking probability performance while the other existing algorithms have almost the same blocking rates for connections with long durations. Fig. 6 investigates AWPC by showing the average number of the allocated wavelengths per connection. As shown in Fig. 6, the ARSA algorithm shows almost 13% decrease in AWPC comparing to other algorithms.

ASR, in Fig. 7 and Fig. 8, represents the percentage of provisioned connections (either Gold or Silver traffic) whose availability requirements are met over all provisioned connections. As noticed in Fig. 7, the ARSA algorithm improves availability satisfaction ratio for Gold requests (ASR-gold) by 2.5% on average. In addition, the improvement in Gold requests does not degrade the availability satisfaction ratio of Silver (ASR-silver) requests as it is shown in Fig. 8, so that the ARSA algorithm either has the same or higher ASR for Silver class of traffic than the other algorithms. That is, to preserve high priority traffic, the proposed algorithm does not scarify other classes of traffic.

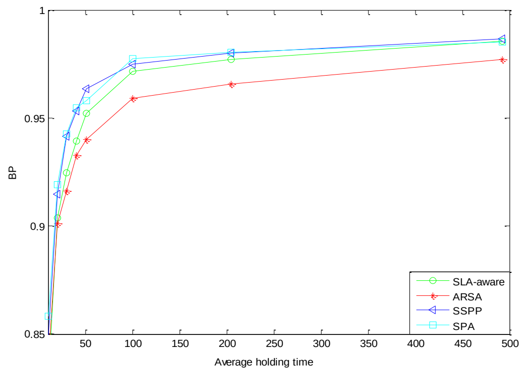

Fig. 5. The effect of connections duration variation on blocking probability.

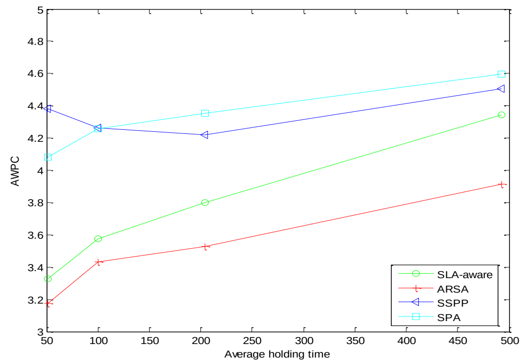

Fig. 6. The effect of connections duration variation on average number of the allocated wavelengths per connection.

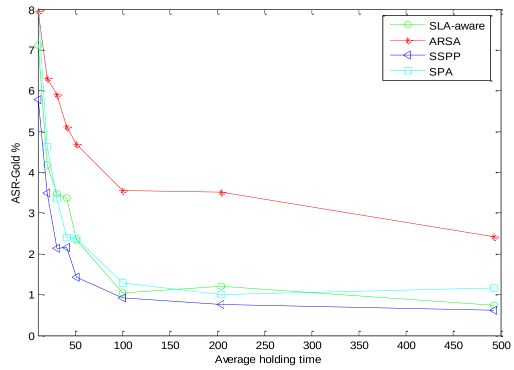

Fig. 7. The ARSA availability satisfaction rate for Gold requests compared to other algorithms.

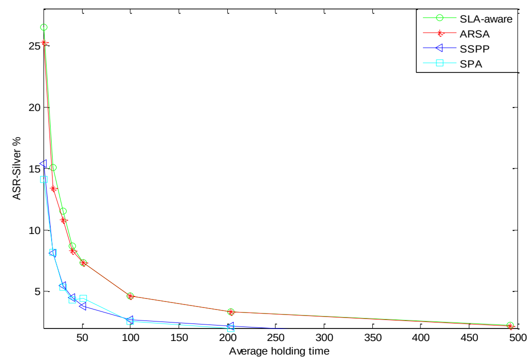

Fig. 8. The ARSA availability satisfaction rate for Silver requests compared to other algorithms.

C. The Effect of Number of High Priority Connections

Fig. 9 and Fig. 10 investigate how the proposed algorithm behaves when the percentage of Gold requests increases from 10% to 80%. The blocking probability of high-priority requests, BP-gold, has been considered as the number of the blocked Gold requests over the total number of the Gold requests. Fig. 9 shows that as ξ increases, BP-gold of the ARSA and SLA-aware algorithms decreases since both are priority-aware algorithms. However, ARSA better preserves high-priority requests as ξ increases from 10% to 80%, and drops blocking probability for 3.5% on average while SLA-aware algorithm drops it just for 1.6% on average. That is, the other algorithm less affect the blocking rate of high-priority requests when the number of such requests increases. Likewise in Fig. 10, the ARSA algorithm better preserves high-priority requests as it increases ASR of Gold requests with long connection duration, ASR-gold, for almost 4% when the percentage of high priority request, ξ, varies from 10% to 80%. However, SLA-aware algorithm cannot preserve Gold requests with long duration connections and it shows ASR-gold performance degrades when the connections duration becomes larger.

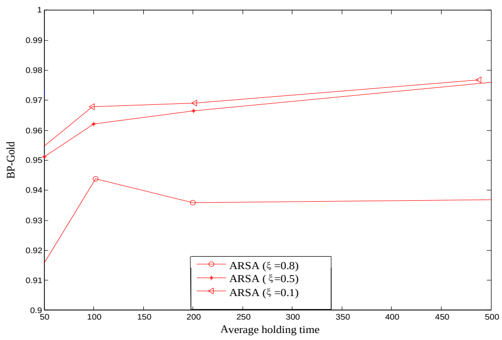

Fig. 9. The effect of changes in the number of high-priority requests on blocking probability of the ARSA mechanism compared to other algorithms.

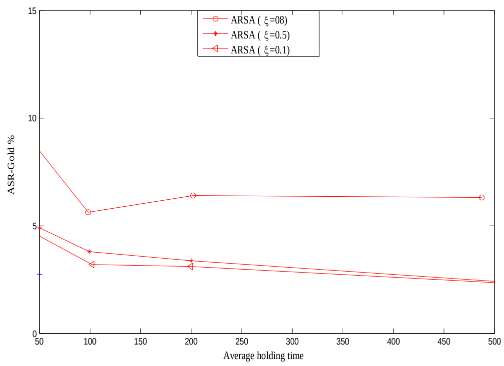

Fig. 10. The effect of changes in the number of high-priority requests on availability satisfaction rate of the ARSA mechanism compared to the SLA-aware algorithm.

V CONCLUSION

In this paper, an SLA-aware algorithm, adaptive re-provisioning SLA-aware algorithm, has been introduced for shared mesh survivable WDM networks. The proposed algorithm takes advantages of a dynamic negotiation of SLA parameters which can help customers to have a better picture of the entire network with respect to the path availabilities. The proposed algorithm has been well developed to cope with the other existing algorithms' shortcomings in better serving a large number of high-priority connection requests with fairly long durations. To achieve this goal, a novel path constraint, time-aware maximum path availability, has been introduced. The algorithms involving in calculation of the proposed path metric have been discussed.

The TMPA path constraint benefits from two important SLA connection parameters, requested availability and holding time. This metric helps the introduced re-provisioning algorithm in this paper to buffer the potentially blocked high-priority requests and serve them in a timely manner rather than blocking them. Simulation results show that the ARSA algorithm reduces the blocking probability of the high priority requests, increases availability satisfaction rate, better preserves high priority connection requests, reduces the average number of allocated wavelengths per connection, and decreases resource overbuild compared to conventional and existing SLA-aware algorithms.

REFERENCES

[1] M. Tornatore, C. Ou, J. Zhang, A. Pattavina, and B. Mukherjee, "PHOTO: an efficient shared-path protection strategy based on connection-holding-time awareness," *Journal of Lightwave Technology*, vol. 23, 2005, pp. 3138–3146.

[2] L. Song and B. Mukherjee, "New approaches for dynamic routing with availability guarantee for differentiated services in survivable mesh networks: the roles of primary-backup link sharing and multiple backup paths," in *Proc. Int. Conf. Globe Communications, GLOBECOM*, pp. 1–5, 2006.

[3] A. Nafarieh, S. Sivakumar, W. Phillips, and B. Robertson, "Dynamically provisioned priority-aware algorithms in shared mesh optical networks," in *Proc. Int. ICST Conf. on Heterogeneous Networking for Quality, Reliability, Security and Robustness, QShine*, 2010

[4] A. Nafarieh, W. Phillips, B. Robertson, and S. Sivakumar, "Statically pre-provisioned priority-aware algorithm for shared-mesh optical networks," *International Conference on Transparent Optical Networks, ICTON*, pp. 1-4, 2010.

[5] J. Lang, B. Rajagopalan, and D. Papadimitriou, "GMPLS recovery functional specification," *IETF RFC4426*, 2006

[6] E. Mannie, and D. Papadimitriou, "Recovery (Protection and Restoration) Terminology for GMPLS," *IETF RFC4427*, 2006

[7] R. He, B. Lin, and L. Li, "Dynamic service-level-agreement aware shared-path protection in WDM mesh networks," *Journal of computer application*, vol. 30, 2007, pp.429-444.

[8] M. Tornatore, C. Ou, J. Zhang, A. Pattavina, and B. Mukherjee, "Efficient shared-path protection exploiting the knowledge of connection-holding time," in *Proc. Optical Network dsesign and Modeling Conference*, 2005, pp. 65-72.

[9] M. Tornatore, D. Lucerna, L. Song, B. Mukherjee, and A. Pattavina "Dynamic SLA Redefinition for Shared-Path-Protected Connections with Known Duration," presented at National Fiber Optic Engineers Conference, 2008.

[10] L. Song, J. Zhang, and B. Mukherjee, "Dynamic provisioning with reliability guarantee and resource optimization for differentiated services in WDM mesh networks," in *Proc. Optical Fiber Communication Conference*, vol. 3, pp. 3, 2003.

[11] M. Xia, C. U. Martel, L. Shi, M. Tornatore, and B. Mukherjee, "A Novel SLA for Time-Differentiated Resilience with Efficient Resource Sharing in WDM Networks," *IEEE ICC*, 2010

[12] X. Wei, L. Guo, X. Wang, Q. Song, and L. Li, "Availability guarantee in survivable WDM mesh networks: A time perspective," *Journal of Information Sciences*, 2008, pp. 2406-2415.

[13] A. Nafarieh, B. Robertson, W. Phillips, and S. Sivakumar "Dynamic SLA Negotiation Mechanism in Support of Priority-aware Algorithms in Shared Mesh Optical Networks," in *Proc. Int. Conf. on Intelligent Network and Computing, ICINC*, 2010.

[14] V. Sarangan and J. C. Chen, "Comparative study of protocols for dynamic service negotiation in the next generation internet," *IEEE Communications Magazine*, 2006, pp. 151-156.

[15] H. Zang, J. Jue, and B. Mukherjee, "A Review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks," *Optical Networks*, 2000, pp. 47-60.

Ali Nafarieh received his B.Sc. and M.Sc. in Electrical Engineering, and M.Eng. and Ph.D in Internetworking Engineering in 1996, 2001, 2007 and 2012 respectively. He is currently a postdoctoral fellow at Internetworking Engineering Program of Dalhousie University. He has served recently as a lecturer since September 2011and as a laboratory instructor since 2007 in Internetworking program. Dr. Nafarieh's research interests are QoS-aware routing mechanisms over optical networks, energy efficient WDM network architectures, green optical communications, and passive optical networks.

Bill Robertson received his B.Sc. and M.Sc.(Eng.) from Aberdeen University, Scotland. He worked in the Department of Telecommunications, Pretoria, and then at Cape Town University and Groote Schuur Hospital before joining Stellenbosch University, Department of Electrical & Electronic Engineering. He earned his Ph.D in 1986. He has held various academic positions and is currently Director of the Master of Engineering in Internetworking Program. Dr. Robertson's research interests are in signal processing and network communications.

William J. Phillips received the B.Sc. degree in Engineering Mathematics and the M.Sc. degree in Mathematics from Queen's University at Kingston and the Ph.D in Mathematics from the University of British Columbia. Dr. Phillips held visiting positions at the University of Guelph, Queen's University, Dalhousie University, and Saint Mary's University before joining the Department of Applied Mathematics at Technical University of Nova Scotia. He is currently Professor and Head of the Department of Engineering Mathematics at Dalhousie University (merged with the Technical University of Nova Scotia in 1997).

Shyamala C. Sivakumar obtained her B.Eng (Electrical) from Bangalore University, India in 1984. She obtained her M.A.Sc (Eng) and Ph.D from the Department of Electrical Engineering at the Technical University of Nova Scotia (now Dalhousie University), Canada in 1992 and 1997 respectively. Dr. Sivakumar joined Saint Mary's University in 2000. She is an Associate Professor of Computing and Information Systems at the Sobey School of Business, Saint Mary's University, Canada. She is also an Adjunct Associate Professor with the Department of Engineering Mathematics and Internetworking at Dalhousie University, Halifax. She has in excess of 35 peer-reviewed journal papers, book chapters and conference proceedings.