The 4th International Conference on Ambient Systems, Networks and Technologies (ANT 2013)

Enhanced Adaptive SLA-aware Algorithms for Provisioning Shared Mesh Optical Networks

Alireza Nafarieh*, Shyamala Sivakumarb, William Robertsona, William Phillipsa

aDalhousie University, Halifax, NS, Canada
bSaint Mary’s University, Halifax, NS, Canada

Abstract

The paper deploys an adaptive provisioning algorithm to the traffic with huge volume of high priority connection requests with long holding time by proposing a novel SLA-aware mechanism over optical shared mesh networks. The contribution presented in this paper follows three main characteristics: i) Proposing a new time-aware traffic engineering path constraint considering holding time of connections in addition to the availability, ii) Introducing a novel provisioning algorithm considering the proposed path attribute, and iii) Applying a high volume of high-priority dynamic traffic with long duration to the introduced mechanism in a new simulation environment to prove its effectiveness. The proposed mechanism benefits from dynamic service level agreement negotiation between a customer and service providers to buffer and further process the potentially blocked high priority connection requests. 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.

Keywords: SLA-aware provisioning algorithm; dynamic service level agreement negotiation; maximum path availability algorithm; time-aware maximum path availability

1. Introduction

The increasing demand of QoS-based traffic which carries a high volume of high-priority traffic requires new traffic engineering strategies, provisioning algorithms, and protection schemes to be developed. The new strategies typically take the SLA-aware algorithms into account to maintain a satisfactory level of QoS for the requested connection with regard to the parameters requested in the SLA. Connection availability is one of the most important QoS parameters specified in an SLA between...
customers and service providers over survivable WDM mesh networks. In addition to connection availability, connection holding time is another connection request characteristic which can play an important role in developing priority-aware algorithms for preserving high-priority requests [1]-[5].

As the first part of the contribution in this paper, a TE path constraint based on the combination of the connection availability, the connection holding-time, and the maximum path availability (MPA) [6] of the request is introduced. The proposed 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, it is assumed that the period during which a connection is valid, holding-time, is known a priori. As discussed in [7] and [8], 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 when the algorithm is serving dynamic traffic. The second part of the paper introduces a novel provisioning mechanism by which some of those high-priority connection requests which were blocked in other SLA-aware algorithms or protection schemes [9] are accommodated. The study focuses on a specific type of traffic which is of dynamic type and mainly high-priority class with long duration. The provisioning mechanism proposed in this paper employs three algorithms: i) An algorithm to calculate the maximum path availability discussed in [6] in detail, ii) An algorithm to find the matrix of time-aware maximum path availability of each connection, and iii) An algorithm by which the potentially blocked connections are buffered and served based on the connections’ holding times. The potentially blocked connections are those connections which are blocked by other existing mechanisms.

2. Related Work

As mentioned earlier, the standard shared mesh algorithm [10], [11] takes advantage of constraint-based shortest path algorithms for the path calculation process. The algorithm does not consider SLA parameters as the constraints in its path calculations. It is only considered in some performance evaluations as the original shared mesh path protection scheme. In [12], the authors have discussed an SLA-aware shared mesh protection algorithm in which the partial link-disjoint protection technique was presented and the link-availability and hop count were considered as the main constraints. The algorithm discussed in [12] has considered SLA parameters as important factors to guarantee customers’ requested reliability. The cost function definitions for both primary and backup paths’ calculations in [12] and [13] have enabled the algorithm to introduce a novel case of protection, partial link-disjoint protection, to increase the availability satisfaction rate, and to reduce the restoration time of shared mesh WDM networks. The authors in [12] and [13] have presented a lightpath provisioning mechanism by which primary paths are first calculated. Backup paths are calculated if the primary paths cannot satisfy the SLA requirements of connection requests. Since the path calculation process presented in [12] and [13] is an efficient way of utilizing network resources, it is adopted as the main path calculation scheme in this paper. Although the simulation results presented in [12] show that the mechanism has had an acceptable network performance, it has considered neither the connection priority nor the connection holding time as SLA requirements.

The mechanisms and algorithms proposed in [14] and [9] are priority-aware mechanisms that take advantage of new path metrics proposed in these two papers to serve a higher percentage of high-priority requests. In [14], a static priority-aware pre-provisioning algorithm is proposed based on the SLA parameters negotiation for shared-mesh WDM networks. In [9], two priority-aware algorithms are introduced for survivable shared mesh WDM networks. Since the pre-provisioning algorithm presented in [14] benefits from static SLA parameters negotiation, the concept of dynamic dissemination of path availability information is extended in [9]. In addition, [14] and [9] focus on the priority-aware mechanisms which are designed for connections with a small percentage of high-priority requests and short durations. That is, the high-priority traffic with long connection holding times are not applied to and evaluated through such mechanisms. This paper extends the work presented in [14] and [9] by proposing
new path attributes and mechanisms to study the effect of variations on the number of high-priority requests and their durations on network performance. Considering the new type of traffic introduced in this paper, in addition to the priority level of the traffic, the duration of the connection requests plays an important role on how the connection requests should be served.

3. Novel TE Path Attribute: Time-aware Maximum Path Availability (TMPA)

To have a time-aware picture of the network resources from a service provider point of view, the information regarding the time characteristic of the connections needs to be propagated throughout the network. However, service providers will still need to disseminate the information about the maximum resource availability that they can offer (such as MPA discussed in [6] and [9]) to give customers an opportunity to choose the best provider in a multi-homed network. To facilitate this possibility, a novel time-aware TE path constraint is presented in this paper which can be disseminated over the entire network. The proposed path attribute gives service providers (or customers) the chance to know the future time for further processing of those connections for which enough resources are not available at the current time.

**Definition 1:** Established connection request matrix, ECM$_{(s,d)}$, is a matrix in which the first column of the matrix contains a list of established connections (C$_{j(s,d)}$), between a specific pair of source (s) and destination (d), path including Gold and Silver requests, and the second column is the associated release time (T$_{j(s,d)}$) of each connection C$_{j(s,d)}$. r is the number of established connections per s-d pair at the time of the calculation of T$_{(s,d)}$.

\[
\forall j \in \{1,2,\ldots,r\}, \quad \text{ECM}_{(s,d)} = \begin{bmatrix}
C_{1(s,d)} & T_{1(s,d)} \\
C_{2(s,d)} & T_{2(s,d)} \\
\vdots & \vdots \\
C_{r(s,d)} & T_{r(s,d)}
\end{bmatrix}
\]

**Definition 2:** Shared risk established connection matrix, SECM is a matrix in which the first column of the matrix contains the set of previously established connections affecting the links traveling between a pair of source and destination path, and the second column is the associated release time of each connection. k is the number of established connections affecting a path of the s-d pair. To determine what connections affect a path of a specific s-d pair, the SRLG of each link forming the s-d path are considered. L$_i$ is the $i^{th}$ link forming the path C$_i$ and G is the group number to which the SRLG belongs. The following algorithm defines the SECM matrix.

\[\exists[C_k \ T_k] \text{ in ECM AND } C_k \in G: \quad \text{SECM} = \{ECM[[C_k \ T_k]]\} \quad \text{SECM}_{(s,d)} = \begin{bmatrix}
C_{1(s,d)} & T_{1(s,d)} \\
C_{2(s,d)} & T_{2(s,d)} \\
\vdots & \vdots \\
C_{k(s,d)} & T_{k(s,d)}
\end{bmatrix}\]

**Definition 3:** The TMPA matrix, TMPA$_{\text{maxm}}$, is a matrix in which m is the number of nodes in the network. The method by which each element of this matrix is calculated is shown below. Each element of the TMPA$_{\text{maxm}}$ matrix, the TMPA$_{(s,d)}$, is a kx2 matrix by itself in which k is the number of established connections affecting the links traveling between a pair of source and destination. The TMPA$_{(s,d)}$ matrix contains MPA of all established pairs of connections affecting and s-d pair together with their associated offered times. In other words, T$_j$ shows the time at which the MPA value of MPA$_{(s,d)}$ can be offered to the connection C$_{j(s,d)}$. The matrix of TMPA$_{(s,d)}$ can be written as below in which p and q are node numbers among m nodes in the network. In the matrix SECM:
A general TMPA matrix element \( MPA_{(s,p,d,q)}^{C_j} \) will show the potential MPA value offered for the given pair of source and destination while the links forming the primary and backup paths of associated connections, \( C_j \), from the 1st to the \( j \)th row of the SECM matrix are released. During the period of \((T_{k-1} - T_k)\), the MPA that a service provider may offer with regard to the release of associated connections would be \( MPA_{(s,p,d,q)}^{C_k} \).

The TMPA matrix of all source and destination pairs for a network of \( m \) nodes is shown as below and is calculated in the TMPA module.

### 4. Enhanced Adaptive Provisioning SLA-aware (EAPSA) Mechanism

#### 4.1. EAPSA mechanism Structure

The elaborated diagram on how the EAPSA algorithm works is shown in Figure 1. 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 \( C_n \) \((s, d, A_r, p, T_{arrival})\) with the requested parameters: source, \( s \), destination, \( d \), availability, \( A_r \), the priority level, \( p \), and the arrival time, \( T_{arrival} \), respectively. The EAPSA algorithm uses the weighted maximum path availability (WMPA) concept presented in the DMPA algorithm in [9] rather than the MPA only. The request is first applied to the provisioning module of the DMPA algorithm. If the provisioning module of the DMPA algorithm can satisfy the on-going connection SLA requirement, the request is served and not sent to other modules for further processing. However, if the provisioning module of the DMPA algorithm facing a high-priority request has no choice other than blocking the request, the EAPSA algorithm sends the request to the TMPA and Buffering High priority Connection (BHC) modules for further processing. Algorithm 1 introduces pseudo code, and Figure 1 shows the block diagram of the EAPSA algorithm.

#### Algorithm 1 EAPSA algorithm

Input: CRM matrix  
Output: Optimal P-B pair of paths  
1. \( n \leftarrow 1 \)  
2. WHILE \( n \) is smaller than the number of the rows in CRM matrix \( \text{AND the connection type is Establish} \{n \leq \text{size (CRMrows)} \} \) \( \text{AND CRM} (n,8) = \text{Establish} \} \) DO Steps 3-10  
   3. Serve \( n \)th connection request  
   4. Call MAP algorithm [6] \{to calculate MPA\}  
   5. IF \( MPA_{(s,d)} = 0 \)  
      Block the connection \( C_n \) AND Go to Step 10  
   6. IF \( A_{Cn} \geq MPA_{(s,d)} \)  
      IF \( \text{WMPA}_{(s,d)} \geq 1 \)  
      \( A_{Cn} \leftarrow \text{MPA}_{(s,d)} \) AND Go to Step 7  
      ELSEIF \( p=\text{Gold} \) AND \( A_{Cn} > \text{MPA}_{(s,d)} \)  
      Call Algorithm 2 AND IF \( \text{TMPA}_{(s,d)} = 0 \)  
      Block the connection AND Go to Step 10  
      ELSE Call Algorithm 3 \{to buffer connections\}  
      ELSE Block the connection AND Go to Step 10  
   7. Call Routing module [9]  
   8. Call Wavelength assignment and graph update module [9]  
   9. RETURN Optimal P-B pair of paths
4.2. The TMPA Module

The Algorithm 2 shows how the TMPA matrix is calculated in the TMPA module. The TMPA matrix contains the MPA values offered to all pairs of source and destination and their associated release times. The Algorithm 2 calculates the \( \text{TMPA}_{m \times m} \) matrix for a network topology of \( m \) nodes. \( \text{MPA}(s,d) \) calculated in [6] is the maximum offered path availability for a certain source-destination pair in the \( n^\text{th} \) connection request, and has been discussed in detail in [9]. It is assumed that there is an automatic mechanism for SLA parameter 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 detail in [6]. If the \( \text{MPA}(s,d) \) is calculated after the release time of the connections associated with a specific pair of source and destination, the TMPA values can be propagated through the network. This is done in the same way that MPA information is disseminated over intra/inter-domain of networks involving multiple autonomous systems. TMPA information can be, and is, assumed to be propagated over the control plane of the network. To avoid buffering a request for an extremely long time, the TMPA algorithm is assumed to be applied to each connection request just once. If an already buffered request cannot be accommodated, it is not sent to the TMPA module for further processing and is simply blocked.

\[
\begin{align*}
1. & \; k \leftarrow 1, u \leftarrow 1, v \leftarrow 1 \\
10. & \; n \leftarrow n + 1 \\
11. & \; \text{END}
\end{align*}
\]

**Algorithm 2** TMPA algorithm

Input: CRM, \( C_0(s, d, A_{0I}, p, T_{0\text{arrival}}, T_{0\text{holding}}) \), \( m \) as the number of nodes in the network

Output: \( \text{TMPA}(s,d) \) matrix
2. FOR all the values of $u \leq m$ AND $v \leq m$ DO Steps 3-17 {to build the matrix $\text{TMPA}_{mxm}$}

3. IF $u = v$
   \[ \text{TMPA}_{(s,u,d,v)} \leftarrow 0 \] AND go to step 17
4. Build the ECM matrix of associated $(s_u,d_v)$ pair (see Definition 1)
5. IF $r \leq 1$ THEN Block the request $C$ AND Go to step 17
6. Build SECM matrix (see Definition 2)
7. Sort the columns of SECM based on ascending order of the release times \{T_j\}
8. FOR all values of $k; k \leq t$ DO Steps 9-15 {to build $\text{TMPA}_{(u,d)}$ array}

   9. Save current wavelength, graph, and link availability matrices in new matrices
10. Release connections $C_{1(u,d)}$ to $C_{k(u,d)}$ from the SECM$_{(u,d)}$ matrix {associated primary and backup paths of the 1st to $k^{th}$ connections (rows) of the SECM$_{(u,d)}$ matrix}
11. Update new wavelength, graph, and link availability matrices
12. Call MPA algorithm [6] {to calculate the MPA associated to $k^{th}$ release: $\text{MPA}_k^{(u,d)}$}
13. Calculate the $k^{th}$ row of the TMPA matrix (see Definition 3)
14. RETURN the array $\text{TMPA}_{(u,d)}$
15. $k \leftarrow k+1$

16. END
17. $u \leftarrow u+1$, $v \leftarrow v+1$
18. END

4.3. The BHC Module

Algorithm 3 defines the steps involved in the BHC module. It is assumed that the $n^{th}$ connection request arriving at $t_n$ 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. The algorithm looks for the minimum amount of buffering 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 a better chance of being established. The circumstances under which the buffered request is established are discussed in Algorithm 3. If the requested availability of the Gold connection is higher than the MPA$^{(u,d)}$, the BHC module checks the TMPA matrix row by row to find which row has a smallest release time and can satisfy the connection requirements. When the BHC module finds the TMPA matrix’s row which meets the connection requirements, if it satisfies the threshold time boundaries applied by the control plane of the network, it modifies the request’s arrival time and inserts it in the CRM matrix as a connection request with a new arrival time and updates this matrix.

If it is assumed that the $k^{th}$ row of the TMPA matrix is selected as the best offered MPA value for a specific pair of source and destination, $T_k$ will be the earliest time after which the potentially blocked connection can be served. However, if the value of $T_k$ is too long, it buffers the connection for extremely long time and may not let the request be served. As a result, some constraints should be considered for the $T_k$ parameter, mainly lower and upper bounds. The release time of the first and the earliest connection should be considered as the lower bound. This makes the algorithm sure that the releasing process has started. The higher bound has been approximated using the following equations.

As shown in Figure 2, $N_{T_k}$ is the number of high-priority connections requested before $T_k$. To make sure that there are enough resources after $T_k$ for the $n^{th}$ connection, it is assumed that the number of newly established connections should be the same as the number of released ones. Although the equal number of established and released connections is not necessarily mapped into an equal number of utilized resources, the result of the average number of wavelengths per connections in NSFNet network [6], [9] shows that the assumption is a good estimate. The threshold buffering period of the queued connection is written as $T_{th}$. As observed in Figure 2, if the number of high-priority requests before $t_n$ is considered as $N_{HP}$, it can be assumed that on average every $t_n/N_{HP}$ unit of time a Gold connection is requested.
The higher and the lower bound of the buffering time are calculated as follows, where $t_0$ is the time at which the connection $C_0$ is being processed, $T_0$ is the time at which the connection $C_0$ is released, $T_{th}$ is the maximum amount of time during which the request will be queued, $N_{H_P}$ is the number of the established high-priority connections between either $s$-$d$ or $d$-$s$ pairs by $t_n$, $N_{sd}$ is the total number of the requests between either $s$-$d$ or $d$-$s$ pairs by $t_n$, $\xi$ is an integer, $\xi$ shows the percentage of high-priority requests, and $T_{sd(\text{avr})}$ is the average arrival period of connections after which a high-priority request between $s$-$d$ or $d$-$s$ pairs may show up.

$$N_{Tk} - (n - 1) \leq k \rightarrow N_{Tk} \leq n + k - 1 \quad N_{Tk} = \frac{T_k}{T_{sd(\text{avr})}} \rightarrow T_k \leq (n + k - 1) \times T_{sd(\text{avr})} \rightarrow T_0 \leq T_k \leq (n + k - 1) \times T_{sd(\text{avr})}$$

$$T_{sd(\text{avr})} = \frac{k_n}{N_{H_P}} \quad T_{thmax} = \frac{v \times T_0}{\xi \times N_{sd}} \quad \text{Assuming } k < n, \quad v_{\text{min}} = \frac{n}{2} \rightarrow T_0 < T_{th} < 2T_0$$

4.4. The RWA and Update Modules

To have a fair comparison between the existing (SSPP [10] and [11], SLA-aware [12]) and the proposed (EAPSA) mechanisms, the RWA and wavelength update modules used in this paper are the same as discussed in detail in [9].

**Algorithm 3** BHC module

Input: ECM$_{(s,d)}$, Original request $\{C_j(s, d, A_r, p, T_{arrival})\}$, TMPA$_{(s,d)}(m \times m)$, $m$ as number of nodes in the network
Output: Modified Request $\{C_j(s, d, A_r, p, T'_{arrival})\}$

1. $\forall k \in \{1,\ldots,m\}$: Check TMPA matrix, Find the smallest $k^{th}$ row of TMPA matrix so: $A_{(s,d)} \leq MPA_{(s,d)}^k$
2. IF $T_k < T_{th}$
   $T'_{arrival} \leftarrow (T_k + \epsilon)$ AND Update CRM matrix with modified request $\{C_j(s, d, A_r, T'_{arrival})\}$
   ELSE Block the request $C_j$ AND Terminate the subroutine
3. $\forall$ MPA$_{(s,d)}^k \in$ TMPA$_{m \times m}$: IF $A_{(s,d)} > MPA_{(s,d)}^k$
   Block the request $C_{(s,d)}$ AND Terminate the subroutine

5. Performance Evaluation

To evaluate the network performance, dynamic traffic has been selected for performance analysis, and the availability satisfaction rate (ASR), the blocking probability (BP), and the resource overbuild (RO) of the EAPSA mechanism are compared with other existing algorithms. The RO computes the ratio of the sum of wavelength links for established backup paths to the sum of wavelength links for established primary paths. The performance of the EAPSA algorithm is compared with SLA-aware algorithm presented in [12]. The EAPSA algorithm has been evaluated based on the simulation environment discussed in the previous work [9]. The changes applied to the injected traffic are as follows: 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, the percentage of high-priority requests is considered variable values ranging from $\xi=10\%$ to $80\%$. The connection arrival process is a Poisson process with constant arrival rate of $\beta=40$ connections per unit of time. The holding time of the connections follows an exponential distribution with a mean value ranging from $\mu=50$ to $500$ units of time. Despite the simulation environments in [6], [9], and [14] in which $\mu$ was constant and $\beta$ changed, in this paper $\beta$ is considered constant and $\mu$ changes. This helps to show the effect of the time variation on the evaluation process. For the sake of simplicity and to better show the effect of the algorithm on serving high-priority connections, it is assumed that $T_{ih}$ takes its maximum value. To achieve a 95% confidence interval, $10^5$ connection requests are simulated in every experiment which may introduce a maximum error of $3\times10^{-3}$.

Figure 3(a) investigates how the proposed EAPSA mechanism behaves when the percentage of Gold requests ranges $10\%$ to $80\%$. As Figures 3(a) shows, the EAPSA algorithm has an average increase of $9.1\%$ in RO for $\xi$ ranging from $10\%$ to $80\%$ while the SLA-aware algorithm has an average growth of 17.2%. In addition, Figures 3(a) shows for longer connection durations, the EAPSA algorithm behaves better since the curves for different values of $\xi$ in the EAPSA algorithm converge to the lower RO. 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. Figures 3(b) shows that as $\xi$ increases, BP-gold of the EAPSA and SLA-aware algorithms decreases since both are priority-aware algorithms. However, the EAPSA mechanism better accommodates high-priority requests as $\xi$ increases from $10\%$ to $80\%$, and decreases blocking probability by $3.5\%$ on average while SLA-aware algorithm only drops it $1.6\%$ on average. That is, the other algorithm improves the blocking rate of high-priority requests less than the EAPSA when the number of such requests increases.

![Figure 3](image1.png)

Likewise in Figures 3(c), the EAPSA algorithm better accommodates high-priority requests as it increases the ASR of Gold requests with long connection duration, ASR-gold, by almost $4\%$ when the percentage of high-priority requests, $\xi$, varies from $10\%$ to $80\%$. However, the SLA-aware algorithm cannot accommodate Gold requests with long duration connections, when $\xi$ varies from $10\%$ to $80\%$. The figure also shows ASR-gold performance degrades when the connections duration becomes longer. The effect of changes in the number of high-priority requests and holding time on standard shared mesh protection scheme is shown in Figures 4. As it is expected and verified through the graphs in Figures 4 the SSPP protection scheme has no understanding of the class of traffic it receives and treats all the requests the same. This is the reason why the blocking rate, resource overbuilt, and availability satisfaction rate
for different values of $\xi$ is almost the same. In addition to the number of Gold requests, Figures 4 shows that the SSPP highly degrades network performance of the requests of long durations with increasing blocking rate and resource overbuild, and decreasing availability satisfaction ratio.

6. Conclusion

In this paper, an enhanced adaptive SLA-aware algorithm has been introduced for provisioning shared mesh survivable WDM networks. The proposed mechanism consists of two parts, a novel 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 EAPSA mechanism has been developed to overcome the shortcomings of other existing algorithms to better serve a large number of high-priority connection requests with fairly long durations. To achieve this goal, a novel traffic engineering path constraint, TMPA, has been introduced. The TMPA path constraint benefits from two important SLA connection parameters, requested availability and holding time. The TMPA metric helps the EAPSA algorithm to buffer the potentially blocked high-priority requests and to serve them in a future time rather than blocking them. The mechanism has also considered a means of controlling the buffering duration. The simulation results show the network performance improvement of the proposed EAPSA algorithm when the number of high-priority requests increases. They also show how the EAPSA mechanism better accommodates high-priority connections with dramatically long holding time.

References

[1] M. Tornatore, C. Ou, J. Zhang, A. Pattavina, and B. Mukherjee, "Efficient shared-path protection exploiting the knowledge of connection-holding time," in Optical networks design and modeling, 2005, pp. 65-72.
[2] M. Tornatore, D. Lucerna, L. Song, B. Mukherjee, and A. Pattavina, "Dynamic SLA Redefinition for Shared-Path-Protected Connections with Known Duration," in National Fiber Optics Engineers, 2008.
[3] L. Song, J. Zhang, and B. Mukherjee, "Dynamic provisioning with reliability guarantee and resource optimization for differentiated services in WDM mesh networks," in Optical fiber communication conference, vol. 3, 2005.
[4] 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," in IEEE ICC, 2010, pp. 1-5.
[5] X. Wei, L. Guo, X. Wang, Q. Song, and L. Li, "Availability guarantee in survivable WDM mesh networks: A time perspective," Journal of information science, vol. 178, no. 11, pp. 2406-2415, 2008.
[6] 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 ICINC, 2010.
[7] 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, pp. 3138-3146, 2005.
[8] 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 GLOBECOM, 2006, pp. 1-5.
[9] A. Nafarieh, S. Sivakumar, W. Phillips, and B. Robertson, "Dynamically provisioned priority-aware algorithms in shared mesh optical networks," in Int. ICST Conf. on Heterogeneous Networking for Quality, Reliability, Security and Robustness, QShine, 2010.
[10] J. Lang, B. Rajagopalan, and D. Papadimitriou, "GMPLS recovery functional specification," IETF RFC 4426, 2006.
[11] E. Mannie and D. Papadimitriou, "Recovery (Protection and Restoration) Terminology for GMPLS," IETF RFC 4427, 2006.
[12] 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, pp. 429-444, 2007.
[13] R. Lin and L. Li, "A New Network Availability Algorithm for WDM Optical Networks," in 5th International Conference on Computer and Information Technology, 2005, pp. 480 - 484.
[14] A. Nafarieh, W. Phillips, B. Robertson, and S.C. Sivakumar, "Statically Pre-provisioned Priority-aware Algorithm for Shared-Mesh Optical Networks," in International Conference on Transparent Optical Networks, ICTON, 2010, pp. 1-4.