Dynamic Sharing Mechanism for Guaranteed Availability in MPLS Based Networks

Andrés J. González and Bjarne E. Helvik
Centre for Quantifiable Quality of Service in Communication Systems*
Norwegian University of Science and Technology, O.S. Bragstads plass 2E, N-7491. Trondheim, Norway
Email: {andresgm,bjarne}@q2s.ntnu.no
Phone: (+47) 735 92783 , 735 92667 Fax: (+47) 73 59 27 90

Abstract—**This paper proposes an algorithm for allocating connections with bandwidth and availability requirements in a telecommunication network where the connections may share resources in their backup paths. The allocation is dynamic in the sense that allocation requests have random arrival times, source and destinations, and allocated connections have a random duration. The algorithm has been designed for core backbone networks with continuous bandwidth distribution, like for example MPLS networks. An efficient bandwidth utilization was obtained by an intelligent sharing mechanism that takes into account the properties of networks with continuous bandwidth allocation. The problem may be formulated as an NIP (Nonlinear Integer Programming) problem. However, due to the well known complexity and scalability limitations in solving this kind of problems, the solution is based on heuristic procedures. A performance comparison with previously published algorithms is carried out for some ”reference networks”, demonstrating a substantially better resource usage.**

I. INTRODUCTION

Offer quality of service (QoS) to the connections that use a network is a matter of major impact. Availability is a significant element on providing a good QoS. It may be clearly defined in a Service Level Agreement as a parameter guaranteed by the network operator. Violation of the agreed value may have large consequences and for this reason is very relevant to develop techniques that help to fulfill the specific availability. At the same time another important concern is the efficient utilization of the network resources.

This paper proposes a mechanism that allocates connections in a network that may be affected by multiple failures, fulfilling specific availability requirements. Compared with previous related works, this mechanism reduces considerably the bandwidth reserved for backups. This is made through the use of a novel technique called *dynamic sharing*.

Protection is one of the most common strategies used to meet dependability requirements with a high probability. In this approach the network resources that provide connection between two points are planned when the connection arrives and before any failure affect it. For this, the use of predefined backups is a common policy. Depending on the characteristic of the network elements and on the implemented backup scheme, different dependability degrees may be obtained.

The backup scheme may be dedicated where the bandwidth of the working and the backup path is reserved exclusively for one connection. Works such as [7] and [1] have proposed relevant ideas for this. On the other hand a scheme known as Shared Backup Path Protection (SBPP) may be implemented, where the network resources may be used more efficiently. Under this scheme, a connection may share bandwidth in its backup path with other connections. For this reason the offered availability will depend not only on the network resources assigned to one connection but also on the behavior of the others. There are some approaches that try to model this problem assuming that maximum two failures may occur in the network [6]. In [2] the use of SBPP is studied including partial restoration. A novel concept that reduces the connection blocking probability in networks that use SBPP is proposed in [8], where the idea is to implement connection priorities that are proportional to the time that a connection has been allocated in the network, in this way, the availability of an existing connection will be unaffected by the establishment of new connections.

Finally it is important to highlight that when SBPP is used, the concept of Shared Risk Group (SRG) has to be considered in order to have robustness under single link failures. That means that the connections affected by one failure can not share any backup resource [9].

The problem of allocating connections in a network fulfilling bandwidth and availability requirements has been studied through the use of Integer Linear Programming (ILP). Formal optimization techniques have been successfully applied for the case of connections without protection and connection with dedicated path protection [11]. Anyhow for the case of shared path protection the problem can not be efficiently solved due to the non linear constraints in the shared availability inequalities [5]. Therefore in further SBPP works the use of heuristics algorithms is common. An interesting proposal is found in [4] where by solving an ILP, a working and a shared backup path are obtained. However, this solution uses Shared Risk Group (SRG) as the only availability constraint and therefore it can not be used to fulfill specific availability values. That means that through the use of ILP may be obtained efficiently a working path that is protected by a disjoint backup path but

---

*“Centre for Quantifiable Quality of Service in Communication Systems, Centre of Excellence” appointed by The Research Council of Norway, funded by the Research Council, NTNU, UNINETT and Telenor. http://www.q2s.ntnu.no

without achieving specific availability targets.

In our work the obtained constraints becomes non linear since specific availability values have to be fulfilled. For this reason the proposed solution uses heuristic techniques.

In MPLS networks, the link capacities are split for allocate multiple connections, and the bandwidth may be assigned in any proportion as long as the capacity limit is not exceeded. This is known as continuous bandwidth assignment. A recent work [3] deals with the problem of SBPP considering the properties of continuous bandwidth. This approach can be used to allocate connections in a network with guaranteed availability and additionally it scales very well. However, the sharing scheme used has some restrictions and the bandwidth utilization is therefore not completely efficient. Our proposal optimizes the amount of bandwidth reserved for backup paths through the use of dynamic sharing. This idea basically gives more flexibility on the selection of the connections that will share resources with an incoming connection.

This paper is organized as follows. In Section II the problem is defined. Section III explains the core idea of *dynamic sharing*. In section IV the optimization problem that minimize the backup bandwidth utilization is formulated. Section V shows the heuristic procedure that uses dynamic sharing to allocate connections, fulfilling bandwidth and availability requirements. Section VI shows some case studies that illustrate the performance of the proposed mechanism. Finally Section VII concludes the paper.

II. PROBLEM DEFINITION

The resilience of a network may be defined as the ability of a network to automatically react to failures through the use of alternative failure-free paths. Planning redundancies and make use of them to deal with failure situations is one of the keys to provide dependable services.

The mechanism developed in this paper is oriented to core backbone networks defined under the standard notation $G(V,E)$, where V represents a set of routers and E a set of links that interconnect those routers. The links l are characterized by a capacity B_l and a steady state availability ρ_l.

Another important elements for our mechanism are the connections that arrive and depart randomly and that have to be allocated and removed from the network according to the stipulated contract period. An arriving request C_n is characterized by a quadruple (s, d, b_n, a_n) where s and d are the source and destination routers respectively, b_n is the capacity requirement and a_n is the availability requirement . Through the use of routing algorithms a path in $G(V,E)$ to allocate C_n may be found. In MPLS networks this path is defined as an LSP (Label Switched Path). Once the path is known, the RSVP signaling protocol will perform the resource reservation in a continuous way. That means that the reserved resources fit exactly with the requested amount (this condition differentiate this work from the approaches where the link capacities are distributed in discrete amounts e.g. complete wavelengths). Section III explains how to take advantage of the continuous bandwidth allocation in order to optimize resource utilization.

In our work the protection mechanism will be implemented through the use backup paths that are used when the main connections fail. The working capacities are dedicated and the backup resources may be shared by several connections as long as the availability requirements are not violated.

Given that the bandwidth used on a link for working connections (B_l^W) is dedicated, the design of sharing strategies that optimize the bandwidth utilization is not possible. On the other hand, the bandwidth reserved for backup paths on an individual link (B_l^K) is shared. This property allows the implementation of techniques able to optimize the total backup capacity in the network through the selection of an appropriate sharing scheme.

The proposed mechanism tries to fit as best as possible the bandwidth reserved for a connection in the shared areas.

Finally it is important to mention that all the allocated connections have to fulfill the bandwidth constrain defined by:

$$B_l \geq B_l^W + B_l^K \ \forall l \in G(V,E) \tag{1}$$

A. Availability Calculations

When a connection request C_n arrives with requirements b_n and a_n to the network, first a working path W_n is found. This path is composed by several links l with respective values ρ_l. The availability of W_n is calculated using the following equation:

$$A_n^W = \prod_{\forall l \in W_n} \rho_l \tag{2}$$

Additionally the unavailability of the working path of C_n may be defined as $U_n^W = 1 - A_n^W$. If A_n^W is smaller than a_n it is a common policy to find a backup path K_n. The resources may be used more efficiently if K_n is shared with a group of other connections (G_n^s).

When any link that belongs to the working path of C_n is affected by a failure, the found backup path will be used in order to keep the connection operational. Nevertheless the backup resources are available for connection C_n if the path itself is operational and additionally if: The backup resources are not being used by another G_n^s connection, or if the connection that is using the backup has lower priority than C_n and therefore can be preempted.

In the literature is well known that if the priority of C_n is assumed equal to zero, the total availability of a connection with sharing backup (A_n^S) may be defined by a lower bound as follows:

$$A_n^S \geq 1 - \left(1 - \prod_{\forall l \in W_n} \rho_l\right) \cdot \left(1 - \prod_{\forall l \in K_n} \rho_l \cdot \prod_{\forall C_x \in G_n^s} A_x^W\right) \tag{3}$$

If $A_n^S \geq a_n$ the C_n requirements are fulfilled.

During the allocation of C_n the offered availability to any connection previously established C_x must be kept over a_x.

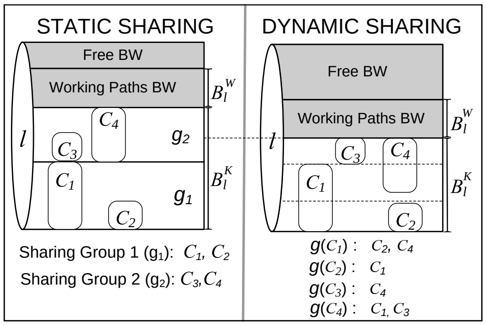

Fig. 1. Static and Dynamic Sharing Schemes

From (3) can be deduced that the availability of a connection x is affected only by the availability of the working path of the new arriving connection (A_n^W). Therefore it is necessary to include an additional admission policy that restricts the sharing of connection n with x depending on A_n^W if:

$$A_n^W \leq \frac{1 - \dfrac{1 - a_x}{[U_x^W]}}{(\prod\limits_{\forall l \in K_x} \rho_l) \cdot (\prod\limits_{\forall C_y \in G_x^s} A_y^W)} \tag{4}$$

Equations (3) and (4) will be used in this paper to verify the fulfillment of the availability requirements.

III. DYNAMIC SHARING

In MPLS networks the link capacities are split in many parts and the bandwidth may be assigned to several connections in any proportion as long as the capacity limit is not exceeded. This is a concept known as continuous bandwidth assignment.

Under this scheme, a link capacity B_l may be split in three parts. The first part is the capacity that has not been reserved by any connection and therefore is free to be used by any future request. The second part is the bandwidth used by working resources B_l^W. Finally there is a bandwidth reserved for backup paths B_l^K which may shared by several connections. Figure 1 shows this concept.

Previous works have defined bandwidth sharing mechanisms where the connections are grouped by a fixed scheme. If the availability requirements are not fulfilled for all of the connections inside a given group, new resources have to be used i.e. new groups are established. We define these traditional mechanisms as *static sharing*. An example of one of the latest static sharing approaches [3] is shown in Figure 1.

This paper proposes a new sharing strategy defined as *dynamic sharing* where the possible sharing group for an incoming connection C_n is not restricted by a fixed scheme. For this reason, any possible combination of the connections previously allocated on K_n may belong to the sharing group of C_n. Therefore the concept of a group containing a fixed number of connections sharing resources does not exist. Instead each connection has its own sharing group. Given this flexibility, the solution may be manipulated in order to optimize the resource utilization. For example, in the dynamic sharing example shown in Figure 1 C_1 shares the backup with C_2 and C_4 but it does not mean that C_2 has to share with C_4, which is the case in a static sharing group scheme. To the authors' knowledge, the problem of sharing resources in a complete dynamic way under bandwidth and availability constrains has not been previously considered.

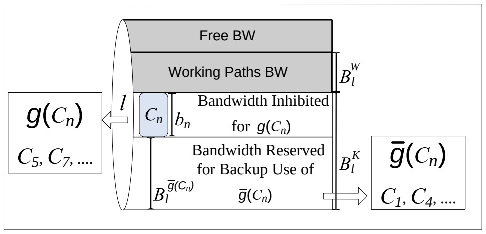

Fig. 2. Example that describes the use of backup capacity

A. Sharing Groups

The core idea of this paper is that each connection C_n has its own sharing group $g(C_n)$, independently of the sharing groups from the other connections.

On any link l that belongs to K_n, B_l^K may be used by a group of connections $G_l^{K_n}$.

The non sharing group of C_n ($\overline{g}(C_n)$) will be defined as $G_l^{K_n} - g(C_n)$ and represents the connections that use any link $l \in K_n$ as a backup path, but that can not share resources with C_n because of the violation of its availability constraint.

The selection of $g(C_n)$ is made in such a way that the availability requirements are kept for all the implied connections, c/o (3) and (4). Therefore if C_n is using the backup, a policy that inhibits all connections in $g(C_n)$ from utilizing B_l^K is introduced. Figure 2 shows an example.

On the other hand, given the case that C_n is using part of B_l^K, there will still be some remaining bandwidth available $B_l^{\overline{g}(C_n)}$ which is determined by $B_l^K - b_n$ (Figure 2). Given that C_n is not really sharing any capacity on l with $\overline{g}(C_n)$, a mechanism that assures that $B_l^{\overline{g}(C_n)}$ will be sufficient to accommodate the $\overline{g}(C_n)$ connections has to be implemented.

IV. MINIMIZING THE SHARED BANDWIDTH RESERVED FOR BACKUP CONNECTIONS

The objective as stated earlier is to minimize the total amount of bandwidth reserved for backup paths. This can be achieved if for each incoming connection C_n, B_l^K is minimized on all the links l that belongs to the backup path K_n by the appropriate selection of the C_n's sharing group $g(C_n)$. This can be expressed by the following equation.

$$min \sum_{l \in K_n} \theta_l \tag{5}$$

Where θ_l represents the increase in the bandwidth reserved for backup paths on link l.

The possible solutions to this problem are given by the feasible combination of connections in $G_l^{K_n}$ grouped in specific

$g(C_n)$ and $\overline{g}(C_n)$. For one specific combination is possible to find the values θ_l. The next equation shows how to make this.

$$\theta_l = \max\{\ 0\ ,\ b_n + B_l^{\overline{g}(C_n)} - B_l^K\ \} \tag{6}$$

In (6), B_l^K and b_n are known values. Obtaining $B_l^{\overline{g}(C_n)}$ is hard as it depends on the sharing and non sharing groups of the connections previously established. Assuming that $G_l^{K_n}$ contains a number of m connections, this sharing information may be captured by a *Sharing Matrix* $\mathbf{S}$ which is an m x m matrix defined as follows:

$$S_{i,j} = \begin{cases} 1 & \text{if } i{=}j \\ 1 & \text{if } C_i \text{ NOT share with } C_j \\ 0 & \text{if } C_i \text{ share with } C_j \end{cases} \tag{7}$$

In Figure 3 is observed an example of how to obtain θ_l and $B_l^{\overline{g}(C_n)}$ under an specific $g(C_n)$ and $\overline{g}(C_n)$.

To obtain $B_l^{\overline{g}(C_n)}$, we evaluate one by one the bandwidth contribution T_j that each connection C_j that belongs to $G_l^{K_n}$ may add to $B_l^{\overline{g}(C_n)}$. Nevertheless we only need to consider the connections that belong to $\overline{g}(C_n)$ given that they can not be inhibit by C_n. Therefore the following variable is defined.

$$I_j = \begin{cases} 1 & \text{if } C_j \in \overline{g}(C_n) \\ 0 & \text{Otherwise} \end{cases} \tag{8}$$

On each evaluation all the pervious calculations have to be kept, using a cumulative scheme that may be represented by the matrix $\mathbf{I}$ as follows:

$$\mathbf{I} = \begin{pmatrix} I_0 & 0 & 0 & \cdots & 0 \\ I_0 & I_1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \\ I_0 & I_1 & I_2 & \cdots & I_m \end{pmatrix} \tag{9}$$

The one by one evaluation may be summarized by an element matrix multiplication ($\bullet$) between $\mathbf{I}$ and $\mathbf{S}$. We define this new matrix as $\mathbf{A}$ which is given as:

$$\mathbf{A} = \mathbf{I} \bullet \mathbf{S} \tag{10}$$

$$\mathbf{A} = \begin{pmatrix} I_0 \cdot S_{1,1} & 0 & \cdots & 0 \\ I_0 \cdot S_{2,1} & I_1 \cdot S_{2,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ I_0 \cdot S_{m,1} & I_1 \cdot S_{m,2} & \cdots & I_m \cdot S_{m,n} \end{pmatrix} \tag{11}$$

Each row in matrix $\mathbf{A}$ contains a set of binary values that indicate which of the connections that belong to $G_l^{K_n}$ will contribute to the size of $B_l^{\overline{g}(C_n)}$ on each evaluation.

Finally it is important to highlight that when a contribution T_j is considered, this amount does not need to be considered again on future contributions T_h for all C_h that belong to $g(C_j)$. Therefore T_j may be modeled by the following equation:

$$T_j = \max\{\ 0\ ,\ \min(b_j - T_h)\ \}\ \forall\ C_h \in g(C_j) \tag{12}$$

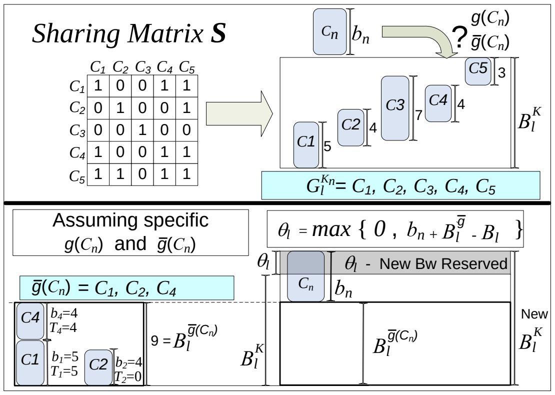

Fig. 3. Example that illustrates how to obtain θ_l

Where T_h is at the beginning equal to zero for all non yet evaluated C_h.

When all the m connections that belong to $G_l^{K_n}$ are considered, an iteration process is generated. This iteration may be captured by the following equation.

$$\mathbf{V_m} = \sum_{j=1}^{m} \mathbf{V_{j-1}} + \mathbf{A_j} \cdot I_j \cdot T_j \tag{13}$$

Where $\mathbf{A_j}$ is a 1 x m vector that represents the row j of the matrix $\mathbf{A}$. $\mathbf{V_m}$ is a vector of size m with initial value $\mathbf{V_0}$ defined as:

$$\mathbf{V_0} = [0, 0, 0, ...0] \tag{14}$$

At the end of the process, the maximum value of $\mathbf{V_m}$ contains $B_l^{\overline{g}(C_n)}$, therefore:

$$B_l^{\overline{g}(C_n)} = \max(\mathbf{V_m}) \tag{15}$$

The objective function is now fully defined by (5), (6) and (15). On the other hand the availability constrains are defined by (3) and bandwidth constrains are defined by (1). It can be observed that this formulation lies into the nonlinear integer programming. Given that the NIP solution of this problem is not scalable in a real world application the solution proposed is heuristic.

V. CONNECTION ALLOCATION MECHANISM USING DYNAMIC SHARING

In this section we explain the mechanism implemented to allocate an incoming connection C_n fulfilling its bandwidth need b_n and availability requirement a_n using dynamic sharing.

First, through the use of conventional routing algorithms like minimum cost path, a working path between the source and the destination is found. The availability of this path A_n^W is calculated using equation 2. When $A_n^W > a_n$ the connection may be provided without protection, i.e. the connection may be unprotected.

On the other hand, when $A_n^W < a_n$ a backup path is needed. There are two basic rules known in the literature for the backup

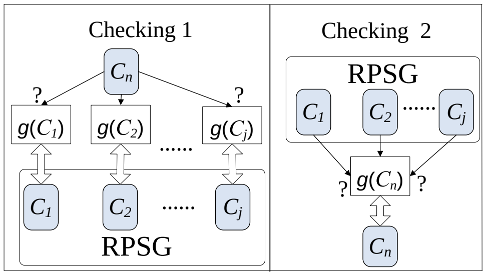

Fig. 4. Checking Procedures to find the sharing group of an incoming connection C_n

path selection. The first is that the working and the backup path have to be link disjoint. The second rule is known as one single link robustness under single link failure and basically states that given any link failure, the affected working connections can not share any backup resource [4].

Once those two rules are fulfilled, the selected backup path contains a number of links with connections previously allocated. Therefore in principle it would be possible for C_n share resources with all of them. This initial group of possible sharing connections will be called *potential sharing group* (PSG) and may be easily obtained using a link by link search through the backup path.

The next step is to guarantee that the availability offered to the PSG connections remains bigger than the respective availability requirement (Checking 1 Fig 4). Through the use of (4) we can easily verify if the availability of the working path of C_n is sufficient in order to share with each of the different connections that belong to PSG. The above procedure reduces the size of PSG obtaining a *Reduced Potential Shared Group* (RPSG). It is also important to highlight that the connections that do not allow sharing with C_n will belong to the *Non Sharing Group* ($\overline{g}(C_n)$). In [8] has been shown that the resources can be more efficiently used and recalculations are avoided if the availability of an existing connection is unaffected by the establishment of new connections, using a priorities scheme. Our mechanism is adaptable to this approach by including or removing the *Obtain RPSG* block (Figure 5).

Now the next step is to find which connections from RPSG should be included in $g(C_n)$ and which in $\overline{g}(C_n)$ as is illustrated in the *Checking 2* (Figure 4). To perform this step, all the connections from RPSG are evaluated one by one using (3). After being added into $g(C_n)$, the individual connections that generate a violation in the availability requirement a_n will be deleted from RPSG and added to $\overline{g}(C_n)$. This is represented in Figure 5 as *Update* $\overline{g}(C_n)$ step.

The next step is to find the RPSG connection combination that minimize the total bandwidth reserved for backup capacities. This combination is obtained, evaluating one by one the connections that currently belong to RPSG. On each iteration the connection under evaluation is temporarily added to $\overline{g}(C_n)$. Then C_n is temporarily allocated in the network under that conditions and using (5) an increase in the total bandwidth used for backup resources may be obtained. After performing those calculations the temporal changes are reversed and the next connection in RPSG is evaluated. When all the connections are analyzed is easy to see which one generates the minimum increase in backup bandwidth. This connection will be deleted from RPSG and added to $g(C_n)$ (New $g(C_n)$ Figure 5).

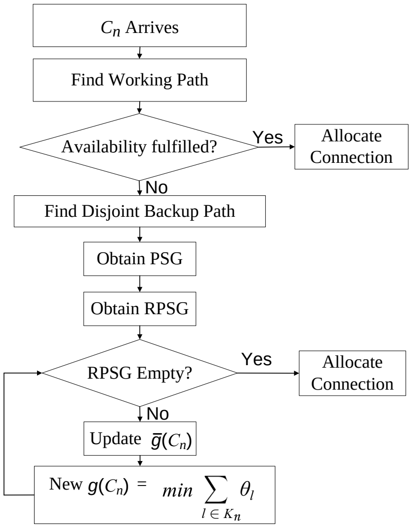

Fig. 5. Flow Chart of the Allocation Mechanism

After this step, there is a new scenario where RPSG has one element less and $g(C_n)$ one more. That means in one hand a change in A_n^S and therefore repeat the *Update* $\overline{g}(C_n)$ step is needed. On the other hand if RPSG is not empty there are still connections that need to be evaluated. Therefore the explained procedure will be performed until RPSG becomes empty. Then the algorithm stops and C_n can be allocated through the backup path with the found sharing schema. The complexity of the algorithm described in this section is $O(E.\delta^3)$ and can be deduced from the loop shown in figure 5 and equation 13 that is implied inside this process. δ represent the connections that are in RPSG which for the worst case may be equal to the number of allocated connections in G when C_n arrives.

VI. CASE STUDIES

In order to verify the performance of our mechanism, and compare the behavior of dynamic and static sharing schemes, some simulations were performed. The experiment setup was implemented in C++, using the topology COST266 network which is an European backbone with 37 nodes and 112 links (Figure 6) [10]. It was assumed that each link has $\rho_l = 0.99$ and a capacity of 100Gb/s which is big enough compared with the connection bandwidth request assumed (1-150Mb/s), in order to avoid bandwidth bottlenecks.

In the first experiment was assumed that requests with random source and destination nodes arrive and depart at any time. Nevertheless the *"number of allocated connections in the networks"* (load) is kept always constant for each point. Those allocated connections have bandwidth and availability requirement uniformly distributed with values among 1Mb/s -

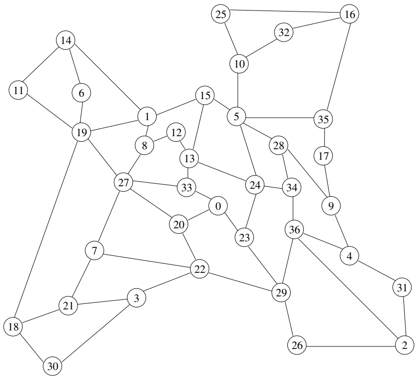

Fig. 6. The COST266 Network Topology

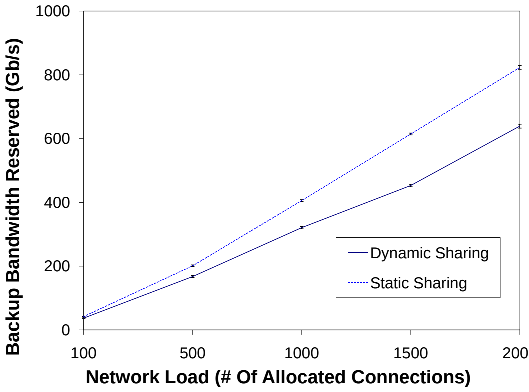

Fig. 7. Bandwidth Utilization Under Different Network Loads

150Mb/s and 0.991 - 0.999 respectively. Figure 7 shows the result of this experiment where there is a clear more efficient resources utilization for all the possible number of allocated connections. Nevertheless, for bigger network loads the saving becomes considerably better.

A second experiment studies the behavior of dynamic and static sharing schemes under different availability requirements. For each of the points shown in figure 8 the load is 1000 connections with the same availability requirement (a_n) but random source, destination and bandwidth (b_n). In this experiment is observed that for low availability requirements the utilization on both schemes is similar due to the connections may not need backup. The same tendency is observed for high availability requirements where the connections need dedicated path protection instead of shared. That means that for all the areas where sharing is really needed, the performance of our mechanism is better.

VII. CONCLUSIONS

This paper shows an improvement in the utilization of the network resources in a SBPP scheme, using the properties of continuous bandwidth assignment.

Considerable bandwidth amounts are saved through the use of our proposed mechanism. This advantage is valid for a wide range of connection availability requirements and network loads. Nevertheless the complexity of our mechanism is approximately one order of magnitude bigger than the complexity obtained by the use of static sharing.

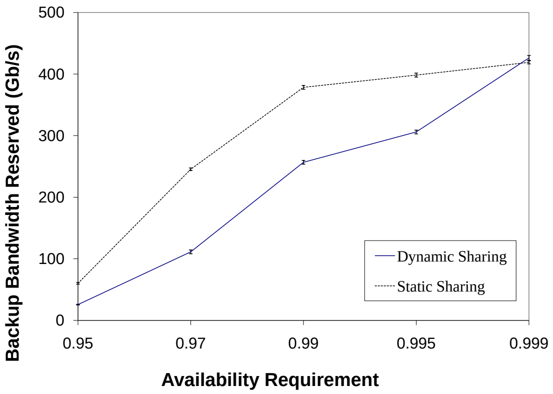

Fig. 8. Bandwidth Utilization Under Different Availability Requirements

Finally, if the availability requirements are considerably high compared with the average link availability, both schemes will behave very similar given that under these conditions they perform as a dedicated path protection scheme.

REFERENCES

[1] A. Fumagalli and M. Tacca. Differentiated reliability (dir) in wavelength division multiplexing rings. *TRANSACTIONS ON NETWORKING.*, 14(1):159–168, Feb. 2006.

[2] Pin-Han Ho, J. Tapolcai, and A. Haque. Spare capacity reprovisioning for shared backup path protection in dynamic generalized multi-protocol label switched networks. *TRANSACTIONS ON RELIABILITY*, 57(4):551–563, Dec. 2008.

[3] Changcheng Huang, Minzhe Li, and A. Srinivasan. A scalable path protection mechanism for guaranteed network reliability under multiple failures. *TRANSACTIONS ON RELIABILITY*, 56(2):254–267, June 2007.

[4] M. Kodialam and T. V. Lakshman. Dynamic routing of bandwidth guaranteed tunnels with restoration. In *Proc. IEEE Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM 2000*, volume 2, pages 902–911, 26–30 March 2000.

[5] Hongbin Luo, Lemin Li, and Hongfang Yu. Routing connections with differentiated reliability requirements in wdm mesh networks. *TRANSACTIONS ON NETWORKING.*, 17(1):253–266, Feb. 2009.

[6] D. A. A. Mello, D. A. Schupke, and H. Waldman. A matrix-based analytical approach to connection unavailability estimation in shared backup path protection. 9(9):844–846, Sep 2005.

[7] A. Mykkeltveit and B.E. Helvik. Comparison of schemes for provision of differentiated availability-guaranteed services using dedicated protection. In *Proceedings of the Seventh International Conference on Networking (ICN)*, April 2008.

[8] A. Mykkeltveit and B.E. Helvik. On provision of availability guarantees using shared protection. In *Proceedings of the 12th Conference on Optical Network Design and Modelling (ONDM)*, March 2008.

[9] R. Ramamurthy, Z. Bogdanowicz, S. Samieian, D. Saha, B. Rajagopalan, S. Sengupta, S. Chaudhuri, and K. Bala. Capacity performance of dynamic provisioning in optical networks. *JOURNAL OF LIGHTWAVE TECHNOLOGY*, 19(1):40–48, Jan 2001.

[10] SNDlib. Library of test instances for Survivable fixed telecommunication Network Design. [Online]. Available: http://sndlib.zib.de/.

[11] Jing Zhang, Keyao Zhu, Hui Zang, N. S. Matloff, and B. Mukherjee. Availability-aware provisioning strategies for differentiated protection services in wavelength-convertible wdm mesh networks. *TRANSACTIONS ON NETWORKING.*, 15(5):1177–1190, Oct. 2007.