A Scalable Path Protection Mechanism for Guaranteed Network Reliability Under Multiple Failures

Changcheng Huang, *Senior Member, IEEE*, Minzhe Li, and Anand Srinivasan, *Member, IEEE*

***Abstract*—We propose two versions of Link Failure Probability (LFP) based backup resource sharing algorithms, namely LFP based First-Fit algorithm, and LFP based Best-Fit algorithm for Generalized Multi-Protocol Label Switching networks. Customers' availability requirements are met by adjusting the availability of the protection paths with different sharing options. Information required for calculating the availability of both the working, and protection paths can be collected along the specific working, and protection paths, thus avoiding the requirement for flooding. This makes our algorithms scalable for a large network. Our algorithms work consistently against both single, and multiple failures. Furthermore, we propose extensions for the existing signaling protocols to demonstrate that our proposed algorithms require minimum changes to the existing protocols. Simulation results show that our proposal performs better than the conventional Dedicated Path Protection schemes in terms of Call Acceptance Rate, and Total Bandwidth Consumption. Finally, by comparing simulation results to analytical results for a simplified network, we provide some insights into the correctness, and efficiency of our proposed algorithms.**

***Index Terms*—Availability, GMPLS, network, protection.**

ACRONYM[1]

BP	Backup Path
DPP	Dedicated Path Protection
IP	Internetworking Protocol
ILP	Integer Linear Programming
LFP	Link Failure Probability
LFP-BF	LFP-based Best-Fit
LFP-FF	LFP-based First-Fit
GMPLS	Generalized Multi-Protocol Label Switching
NSFNET	National Science Foundation Network
QoS	Quality of Service
RBB	Reserved Bandwidth Block
RSVP-TE	Resource ReSerVation Protocol-Traffic Engineering
SCI	Sharing with Complete routing Information
SPI	Sharing with Partial routing Information
SPP	Shared Path Protection
SPS	Separate Path Selection
SRLG	Shared Risk Link Groups
SRLG-SPP	SRLG based Shared Path Protection
WP	Working Path

NOTATION

$G = (V, E)$	is a directed network graph having a set V of nodes and a set E of links
$V = \{v_1, v_2, \cdots, v_N\}$	is a set of nodes in a network where N represents the total number of nodes
$E = \{e_1, e_2, \cdots e_M\}$	is a set of links in a network where M represents the total number of links
$C = \{c_1, c_2, \cdots, c_K\}$	is a set of connections in a network, where each connection is denoted by a 3-tuple
$c_i = (s_i, d_i, x_i); s_i \in V$	is the source node, $d_i \in V$ is the destination node, and x_i is the connection identifier
WP_c	is a set of links representing the working path of a connection $c \in C$
BP_c	is a set of links representing the backup path of a connection $c \in C$
$T_{i,j}$	is a set of connections that use the link (i,j) as their backup link: $T_{i,j} = \{c_k \mid (i,j) \in BP_{c_k}, c_k \in C\}$
$T^n_{i,j}$	is a subset of $T_{i,j}$ representing the set of connections that share the n^{th} Reserved Bandwidth Block (RBB) of link (i,j) with the allocated bandwidth denoted by $RB^n_{i,j}$.
$AB_{i,j}$	denotes the total available (spare) bandwidth on link $(i,j) \in E$
$SB_{i,j}$	denotes the total service bandwidth on link $(i,j) \in E$ which is being used by the working paths

Manuscript received March 26, 2006; revised July 30, 2006 and January 24, 2007; accepted January 29, 2007. Associate Editor: M. Zuo.

C. Huang is with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON K1S 5B6, Canada (e-mail: huang@sce.carleton.ca).

M. Li was with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON K1S 5B6, Canada.

A. Srinivasan is with Eion Wireless Inc., Ottawa, ON K1Y 2X5, Canada (e-mail: anand@eion.com).

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

Digital Object Identifier 10.1109/TR.2007.896739

[1]The singular and plural of an acronym are always spelled the same.

$RB_{i,j}$	denotes the total reserved bandwidth on link $(i,j) \in E$ which is being used by the backup paths
$RB_{i,j}^{k}$	denotes the reserved bandwidth allocated to the k^{th} Reserved Bandwidth Block (RBB) of link $(i,j) \in E$
$B_{i,j}$	denotes the capacity of link $(i,j) \in E$
B_c	denotes the bandwidth demand requested by a connection $c \in C$
Q_c	denotes the overall reliability demand requested by a connection $c \in C$
Q_c^{BP}	denotes the backup path reliability demand or the residual reliability demand
P_c^{Conn}	denotes the failure probability of a connection $c \in C$
$P_{i,j}^{L}$	denotes the failure probability of link $(i,j) \in E$
P_c^{Path}	denotes the path failure probability
P_c^{WP}	denotes the failure probability of a working path WP_c
P_c^{BP}	denotes the failure probability of a backup path BP_c
R_c^{Conn}	denotes the calculated reliability of a connection $c \in C$
$R_{i,j}^{L}$	denotes the reliability of link $(i,j) \in E$
R_c^{Path}	denotes the calculated reliability of a path
R_c^{WP}	denotes the reliability of a working path WP_c
R_c^{BP}	denotes the reliability of a backup path BP_c
H_c	denotes the number of hops along the backup path BP_c
β_c^{BP}	denotes the calculated backup path failure probability constraint for the backup path BP_c
β_c^{L}	denotes the backup link constraint for each link along the backup path BP_c
R	denotes the average call acceptance rate
r	denotes number of servers in a M/M/r/r queuing system
δ	denotes restoration overbuild
λ	denotes the call arrival rate for a source-destination pair
M	denotes the number of source-destination pairs
μ,	denotes the inverse of average call holding time

I. Introduction

With the rapid increase in the processing power of computers, and the abundant bandwidth of high-speed networks, new real-time multimedia applications, such as internet telephony, video conferencing, and virtual private networks, are becoming more popular everyday. These applications have traffic characteristics, and performance requirements that are significantly different from existing data-oriented applications. As networks expand with the increasing deployment of broadband & optical technologies, the consequence of a failure or multiple failures becomes more pronounced. Because service disruptions due to a network failure can cause customers significant loss of revenue, network availability is becoming an important element of QoS requirements [1].

Network availability is defined as the capability of a network to maintain & provide an acceptable level of performance during network failures [1]. In this paper, the parameter Q_c is being used to denote the availability requirement, which is simply defined as the required minimum probability that a connection c is operational at any given instant of time [2]. Failures may have many causes, e.g. link failure, node failure, software failure, etc. as mentioned in [3]. Upon the onset of these failures, a network recovers by sending traffic to another part of the network instead of the failed part of the network. The key objective of network failure recovery mechanisms is to minimize the disruptions to user traffic when failures occur. Because node failure can be treated as a multiple-link failure problem, and software failure is actually a kind of node failure [4], we therefore focus only on link failures in this paper.

There are mainly two types of failure recovery mechanisms: protection switching, and rerouting. Protection switching is a type of failure recovery method where, upon the arrivals of new connection requests, the backup paths are assigned while the working paths are established. The rerouting mechanism, on the other hand, dynamically selects an alternative path when a failure occurs. It is easy to see that protection switching is faster than rerouting, but less efficient in bandwidth usage. In this paper we focus on protection switching for its superior property of fast recovery. Our goal is to minimize its reserved backup resources while maintaining the required availability. Discussions on various failure recovery mechanisms can be found in [5]–[14].

The approaches for allocating backup resources can be further classified as link-oriented, and path-oriented. The link-oriented methods, such as Node Cover, Ring Cover, and P-cycle, focused on allocating enough resources for protecting each individual link [4], whereas a path-oriented method used the end-to-end detouring scheme [15]. The end-to-end path-oriented protection schemes achieve failure resilience by using a link/node disjoint pair of WP, and BP from a source node, to a destination node. DPP, and SPP are two examples of path protection schemes. DPP, also called 1+1 protection, allocates a dedicated link/node-disjoint backup path with the same bandwidth as the working path. The DPP scheme provides high reliability, and fast restoration speed, but is resource inefficient because it consumes at least twice the amount of the required network resources. On the other hand, resource usage efficiency

can be improved by using the SPP scheme, which allows a new backup path to share the resources allocated to some existing backup paths as long as their working paths are not subject to the same point of failure. The SPP scheme is more efficient in terms of spare capacity utilization, but may not be as reliable as the DPP scheme. There exist numerous studies focusing on mesh-based shared-path protection design [16]–[21]. In [19], an algorithm was developed to dynamically find a link-disjoint working-backup path pair for each connection request. The SPS approach was used to find the link-disjoint backup paths. Because the SPS approach cannot guarantee 100% success in path selection, in [1] a k-shortest path-ranking algorithm was applied to search for a better possibility of allocating the working-backup path pair.

In papers [17], [18], the scalability of optimizing bandwidth allocation for working, and protection paths was investigated under two dynamic routing schemes, called Sharing with Partial routing Information (SPI), and Sharing with Complete routing information (SCI). In the SCI scheme, at the time of routing, the source node has the information of 1) the aggregate bandwidth used in each link by the working paths, 2) the aggregate bandwidth used in each link by the backup paths, and 3) the available bandwidth of each link. With this information, optimal resource sharing of the backup paths is possible at a cost of frequent flooding of link state information, which is not quite scalable for a large network.

In [20], an ILP formulation was presented for both the SCI, and SPI schemes. In [22], the author developed an analytical model for recovery in $M:N$ protection groups using both the recovery time, and the recovery failure probability. In [23], the author proposed a survivable routing scheme that can explore pool-based backup sharing to dynamically find the least-cost backup path for a given working path. Recently, the authors in [21] proposed a new scheme called SRLG-based Shared Path Protection (SRLG-SPP). In SRLG-SPP, unlike SPP, an SRLG constraint is imposed, which means if two working paths are using at least one common link, their backup paths should not share any link. But in [21], the authors only give a centralized implementation for this algorithm

Both the legacy DPP, and SPP schemes guarantee 100% availability only against single-failure events in the network [24]. When the single-failure assumption is not acceptable in the case where some failures affect the working path of a connection, and some other failures affect the backup path of the same connection at the same time; neither DPP, nor SPP can satisfy the availability requirements. In addition, the SPP scheme also fails if two or more failures affect at the same time some distinct working paths whose corresponding backup paths share the same protection resource [24]. Because the probability of multiple concurrent failures is high, especially in mobile wireless networks due to node mobility, several authors have studied the issues of protection against multiple failures [24]–[27]. In reality, some links which connect different nodes may be in the same conduit. Therefore, if the conduit is cut, many links fail at the same time. Fire or earthquakes can also damage a large number of nodes causing multiple node failures, and each node failure can be treated as multiple link failures. Moreover, multiple-link/node failures often happen in mobile wireless networks. Unlike fixed infrastructure networks where link failures are comparatively rare events, in mobile networks, the rate of link failures is directly related to node mobility. Therefore, in mobile wireless networks, the rate of link failures due to node mobility is the primary obstacle for routing [28], and restoration algorithms must be able to address these multi-failure situations.

In summary, the following challenges remain to be addressed: 1) design a distributed QoS path selection algorithm which does not require the global link state information for the sharing calculation of the backup resource, so that the communications overhead can be reduced for large scale networks; 2) design a failure recovery mechanism which not only guarantees the customer specific network availability requirement, but is also efficient in terms of network resource utilization; 3) design an efficient backup resource sharing algorithm which works consistently against both a single failure, and multiple failures; and 4) design an algorithm that require minimum modifications to the existing signaling protocols so that it can be easily integrated into the current Internet standards.

We propose two versions of LFP-based backup resource sharing algorithms, namely LFP-FF, and LFP-BF. Both the legacy DPP, and SPP algorithms can be considered as special cases of our proposal. We assume that working, and protection paths are optimally selected using certain routing algorithms which are based on certain criteria for optimization. K-shortest path is a good example of this kind of routing algorithm. Instead, we focus on improving the sharing efficiency of backup resources by taking into account the fact that different connections have different bandwidth requirements. Customers' requirements for availability are met by adjusting the availability of protection paths with different sharing options. Information required for calculating the availability of both the working, and protection paths can be collected along the specific working, and protection paths. No flooding is required. This makes our algorithm scalable for a large network. We propose signaling protocol extensions to distribute some of the link usage information (failure probability, and SRLG constraint) among network nodes. In this distributed signaling framework, the calculation of backup resource is done on each node along the backup path depending only on the local link state information, such that the network overhead is contained within an acceptable level.

The rest of this paper is organized as follows. In Section II, we introduce some definitions, notations, and the formulation of the constraints for our proposed algorithms. In Section III, we provide a detailed description of our proposed algorithms. Numerical examples are discussed in Section IV. We start with a simple network with some analytical results to gain some insights into our proposed algorithms. This is followed with a more complex network example. Section V provides conclusions for this paper.

II. Background Information

A. Definitions

Link Failure Probability $P^L_{i,j}$ is defined as the probability that link (i,j) is in a failure state over a period of time. It can be calculated as $\mathrm{MTTR}/(\mathrm{MTTF}+\mathrm{MTTR})$. The values of LFP

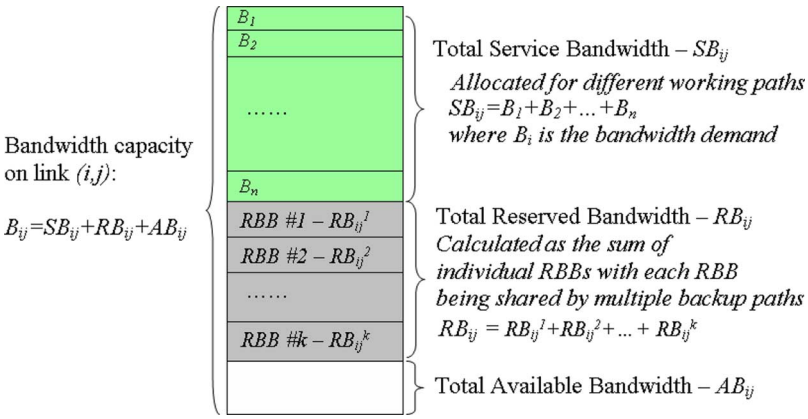

Fig. 1. Illustration of the service, and reserved bandwidth allocation.

are usually very small, e.g. less than 0.01%. In this paper, we assume that knowledge of these probabilities is readily available for the nodes directly connected by the link.

Path Failure Probability P_c^{Path} is defined as the probability of any single link or a combination of several links along the path c being in a failure state over a period of time. Because node failures can be considered as a special case of multiple link failures [4], our proposed algorithm can also be applied to node failure scenarios.

Residual Availability Q_c^{BP} is defined as the minimum availability requirement for backup paths so that the availability provided by both the working, and backup paths can meet the availability requirement for a connection. Assuming a working path is selected with a failure probability of P_c^{WP}, and the overall connection availability requirement is Q_c, the residual availability can be calculated as $Q_c^{BP} = (Q_c + P_c^{WP} - 1)/P_c^{WP}$.

A Connection is defined as a combination of a working path, and one or multiple backup paths. The implementation of our proposed algorithms requires a global connection ID assignment. A possible solution is to provide a globally unique connection ID using a 3-tuple consisting of the source node identifier s, the destination node identifier d, and the connection identifier c to identify each individual connection. An operational connection may refer to a single working path; or of both a working path, and one or multiple backup paths. In the case that a connection consists of one working path, and one backup path, the connection availability R_c^{Conn} is defined as the probability that either the working path, or the backup path, or both paths remain operational.

The total bandwidth on a link (i, j) can be dynamically divided into multiple parts called the Reserved Bandwidth Blocks (RBB), so that the resource in each of the RBB can be shared by multiple connections as their backup resource, as long as their availability constraints are satisfied (refer to Fig. 1 for illustration).

The allocated reserved bandwidth in the nth RBB of link (i, j) is denoted by $RB_{i,j}^n$. The total reserved bandwidth in link (i, j) is obtained from the sum of each individual reserved bandwidth as defined in (1), where N is the total number of RBB on link (i, j).

$$RB_{i,j} = \sum_{n=1}^{N} RB_{i,j}^n \tag{1}$$

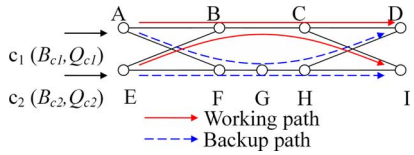

Fig. 2. Illustration of two connections within the same SRLG.

Note that all the RBB may have completely different sizes. One of the major contributions of this paper is to efficiently share bandwidth among connections with different bandwidth requirements. Two different RBB management algorithms are proposed in this study: the First Fit algorithm, and the Best Fit algorithm, which are explained in the next section.

The SRLG constraint requires that any two working paths sharing the same risk of failure cannot share the same protection resource. Paths that have a common link are said to be in the same SRLG. If two connections are in the same SRLG, and their corresponding backup paths also share some common links at the same time, they can not share the same RBB.

An example of the SRLG constrained connections is shown in Fig. 2, where two connection requests are being set up. The working paths A-B-C-D, and E-B-C-I share the link (B,C); therefore, connection c1, and connection c2 are in the same SRLG; and the SRLG constraint will prevent them from sharing the same RBB under the same links, e.g. link (F,G), and link (G,H).

During the signaling stage for the setup of a backup path, it is necessary for the source node to have the knowledge of the SRLG associated with each link of the corresponding working path. Fortunately, for the algorithms proposed in this paper, this information can be collected during the setup of the working path. Hence, no flooding or extra signaling messages are required. This is very different from the approach discussed in [21].

B. Some Assumptions

In this paper, it is assumed that the connection requests arrive sequentially, and randomly. Each connection request has the following parameters: a source node s, a destination node d, a bandwidth demand B_c, and an availability requirement Q_c. The working, and protection paths are optimally selected based on some routing algorithms, which again are based on some principles for optimization. K-shortest-path is a good example.

Issues related to routing are beyond the scope of this paper. This paper focuses on bandwidth sharing along backup paths instead of routing. It is also assumed in this paper that the LFP value of each link is known a priori, and is readily available to the nodes connected by the link, either through network measurement or other methods, and the LFP of different links are independent of each other. All the links are duplex, and the explicit routes for both the working, and backup paths are computed at the source node. The actual path setup can be done using the RSVP-TE[29] signaling mechanism in a MPLS network.

C. Calculations

Some equations related to our proposal are provided below. In this paper, we assume all failure events are independent. This independence can be achieved by selecting node disjoint paths, and avoiding SRLG as discussed in the following sections.

Lemma 1: The path failure probability can be calculated using (2).

$$P_c^{Path} = 1 - \prod_{(i,j)\in E,(i,j)\in Path_c} \left(1 - P_{i,j}^L\right) \tag{2}$$

Accordingly, the working path failure probability P_c^{WP}, and the backup path failure probability P_c^{BP} can be calculated using (3), and (4) respectively.

$$P_c^{WP} = 1 - \prod_{(i,j)\in E,(i,j)\in WP_c} \left(1 - P_{i,j}^L\right) \tag{3}$$

$$P_c^{BP} = 1 - \prod_{(i,j)\in E,(i,j)\in BP_c} \left(1 - P_{i,j}^L\right) \tag{4}$$

Lemma 2: If a connection consists of one working path (denoted by WP), and one backup path (denoted by BP), the connection availability constraint Q_c should be satisfied as (5) for the connection being operational.

$$R_c^{Conn} = 1 - P_c^{Conn} = 1 - P_c^{WP} \times P_c^{BP} \geq Q_c \tag{5}$$

where the connection availability R_c^{Conn} is defined as the probability that either the working path, or the backup path, or both the working path and the backup path remain operational.

Lemma 3: The Backup Path Failure Probability Requirement (or the inverse of the residual reliability requirement) is calculated by

$$\beta_c^{BP} = (1 - Q_c)/P_c^{WP} = 1 - Q_c^{BP} \tag{6}$$

Lemma 4: Given the number of hops H_c of a backup path, the Backup Link Constraint (assuming an Equal-Distribution policy [30] is being used) is calculated by

$$\beta_c^L = 1 - \left(1 - \beta_c^{BP}\right)^{1/H_c} = 1 - \left(Q_c^{BP}\right)^{1/H_c} \tag{7}$$

Note that our proposed algorithms in this paper do not depend on the Equal-Distribution policy. For the cases where residual availability is distributed unevenly among the links, a variation of (7) can be derived easily.

Theorem 1: Assuming $T_{i,j}^n$ is the set of connections which use link (i,j) as their backup link, and which are sharing the same RBB (RBB n) with allocated bandwidth capacity denoted by $RB_{i,j}^n$, the set of backup link constraints shown as in (8) should be satisfied for any connection $c_k \in T_{i,j}^n$, so that the overall availability requirement Q_c could be satisfied.

$$1 - \left(1 - P_{i,j}^L\right) * \prod_{\forall c_m \in T_{i,j}^n, m \neq k} \left(1 - P_{c_m}^{WP}\right) \leq \beta_{c_k}^L, \forall c_k \in T_{i,j}^n \tag{8}$$

Equation (8) can also be expressed as a matrix of K equations, as shown in (9), where K is the total number of connections in the set $T_{i,j}^n$.

$$\begin{cases} 1 - \left(1 - P_{i,j}^L\right) * \prod_{\forall c_m \in T_{i,j}^n, m \neq 1} \left(1 - P_{c_m}^{WP}\right) \leq \beta_{c_1}^L \\ 1 - \left(1 - P_{i,j}^L\right) * \prod_{\forall c_m \in T_{i,j}^n, m \neq 2} \left(1 - P_{c_m}^{WP}\right) \leq \beta_{c_2}^L \\ \cdots \\ 1 - \left(1 - P_{i,j}^L\right) * \prod_{\forall c_m \in T_{i,j}, m \neq K} \left(1 - P_{c_m}^{WP}\right) \leq \beta_{c_K}^L \end{cases} \tag{9}$$

Proof: For any connection $c_k \in T_{i,j}^n$, to meet the criteria that enough resource in the n^{th} RBB is available to recover from a failure of this working path, the following two conditions have to be satisfied.

Condition A_1: The link (i,j) is operational.

Condition A_2: None of the other connections except for c_k in set $T_{i,j}^n$ are occupying the resource of this RBB, i.e., the working paths of all other connections remain operational.

Satisfying the Backup Link Failure Constraint (β_c^L) means that the failure probability of either of the above two conditions should not exceed the constraint β_c^L, i.e.,

$$P(\overline{A_1} \cup \overline{A_2}) \leq \beta_c^L \tag{10}$$

because

$$\begin{aligned} P(\overline{A_1} \cup \overline{A_2}) &= P(\overline{A_1 \cap A_2}) \text{ (de Morgan's laws)} \\ &= 1 - P(A_1 \cap A_2) \\ &= 1 - P(A_1)P(A_2) \text{ (independence)} \\ &= 1 - [1 - P(\overline{A_1})] * [P(A_2)] \end{aligned} \tag{11}$$

$P(\overline{A_1})$ is the link failure probability, i.e. $P_{i,j}^L$, and $P(A_2)$ is the probability that the working paths of all other connections are operational. Thus

$$P(\overline{A_1}) = P_{i,j}^L \tag{12}$$

$$P(A_2) = \prod_{\forall c_m \in T_{i,j}^n, m \neq k} \left(1 - P_{c_m}^{WP}\right) \tag{13}$$

From (10), (11), (12), and (13), we have

$$1 - \left(1 - P_{i,j}^L\right) * \prod_{\forall c_m \in T_{i,j}^n, m \neq k} \left(1 - P_{c_m}^{WP}\right) \leq \beta_{c_k}^L, \forall c_k \in T_{i,j}^n$$

■

Theorem 2: For the n^{th} RBB in link (i,j), the following bandwidth constraint must be satisfied for the connection being operational.

$$RB_{i,j}^n + AB_{i,j} \geq B_c, \forall n \in \{1, 2, \ldots N\} \tag{14}$$

III. LFP-Based Backup Resource Allocation Algorithms

The LFP-based backup resource allocation algorithm is implemented as follows. When a restorable connection request c arrives with the following parameters (a source s, a destination d, a bandwidth demand B_c, and an availability requirement Q_c), the source node s first probes the working path WP_c to the destination node d. During the signaling process of the working path, the source node performs the data collection, and subsequently calculates the *Working Path Failure Probability* (P_c^{WP}) using (3), and the *SRLG constraint* using (8). If the calculated availability of the working path alone can not satisfy the availability requirement, i.e. $R_c^{WP} = 1 - P_c^{WP} < Q_c$, then the residual availability is calculated, and a backup path becomes mandatory for this connection request. There are two approaches for finding the backup paths which can satisfy the availability requirements. The first approach is that the source node would subsequently check the second node-disjoint path to the destination as the backup path. If the joint availability of the backup, and working paths still can not satisfy the availability requirement, a second backup path would be selected. This process should continue until the availability constraint is satisfied, or the connection request is rejected due to lack of resources. A second approach is that only one backup path that can meet the availability requirement is selected. If no single backup path exists in the network to satisfy the availability constraint, then the connection request is rejected. Both approaches can actually be considered as some kind of ordered exhaustive search algorithm. The first option has a higher chance of meeting the availability constraint earlier than the second option, and therefore is faster in average. However, the second option does not require splitting traffic among several backup paths, and therefore is easier to implement. Our proposed algorithms can work with both options. To simplify our description without loss of generality, it is assumed that one backup path is enough in the following discussions, and therefore both options have the same result.

The selected backup path BP_c is then probed by the source node s. The source node calculates the *Backup Path Failure Probability Constraint* (β_c^{BP}) using (6), and the *Backup Link Constraint* (β_c^L) using (7), based on the number of hops H_c of the backup path. The backup path bandwidth allocation is done link by link with signaling messages. Each link (i,j) along the backup path first checks if the existing RBB can be shared with this new backup path request. For the n^{th} RBB with bandwidth capacity $RB_{i,j}^n$, if sharing this RBB can still meet the set of *Backup Link Constraints* in (8) & (9), and the *Bandwidth Constraint* in (14) for all other connections in the set $T_{i,j}^n$, sharing this RBB is granted, and the bandwidth capacity allocated to this RBB is updated as $RB_{i,j}^n = \max(RB_{i,j}^n, B_c)$. If all the existing RBB are not sharable, then a new RBB is allocated for this connection request as long as the available bandwidth of this link is not exhausted.

A. Shared Block Management

The concept of RBB is used to manage the bandwidth sharing. If the current RBB is not available for sharing, a new RBB can be created dynamically, provided that extra bandwidth is available on the link according to either one of the three scenarios described below.

Scenario 1: Assume that no previous connection is using the link (i,j) as its backup link $(RB_{i,j} = 0)$, and the first request for a backup path arrives with the parameters Q_{c_1}, B_{c_1}, $P_{c_1}^{WP}$, and $\beta_{c_1}^L$, the first task is to check the following two constraints: 1) the bandwidth constraint such that $AB_{i,j} \geq B_{c_1}$, and 2) the availability constraint such that $P_{i,j}^L \leq \beta_{c_1}^L$. If both constraints are satisfied, B_{c_1} unit bandwidth can be allocated for the first RBB, i.e. RBB 1, as $RB_{i,j}^1 = B_{c_1}$. The total reserved bandwidth becomes $RB_{i,j} = RB_{i,j}^1 = B_{c_1}$. At the same time, the working path failure probability $P_{c_1}^{WP}$, and the backup link constraint $\beta_{c_1}^L$ are recorded into the local database.

Scenario 2: A second request for a backup path arrives at link (i,j), with the constraint parameters Q_{c_2}, B_{c_2}, $P_{c_2}^{WP}$, and $\beta_{c_2}^L$, then the following constraints should be checked: 1) the bandwidth constraint such that $AB_{i,j} + RB_{i,j}^1 \geq B_{c_2}$; 2) the availability constraints such that $1 - (1 - P_{i,j}^L) \times (1 - P_{c_1}^{WP}) \leq \beta_{c_2}^L$, and $1 - (1 - P_{i,j}^L) \times (1 - P_{c_2}^{WP}) \leq \beta_{c_1}^{L^2}$; and 3) the *SRLG constraint* such that if the working path of the first connection, and the working path of the second connection are in the same *SRLG* group, they cannot share the same RBB. If all three constraints are satisfied, then the resource in the first RBB can be shared by this new connection request. Consequently, the bandwidth allocated to the first RBB $(RB_{i,j}^1)$ is updated to $RB_{i,j}^1 = \max(RB_{i,j}^1, B_{c_2})$. In addition, $P_{c_2}^{WP}$, and $\beta_{c_2}^L$ need to be recorded into the local database for future sharing calculation. If RBB 1 cannot be shared by this new connection because its availability constraint or the *SRLG* constraint cannot be satisfied, a second RBB has to be allocated, which is denoted by RBB 2. Suppose $AB_{i,j} \geq B_{c_2}$, and $RB_{i,j}^2 = B_{c_2}$, then the total reserved bandwidth is the sum of the reserved bandwidth of the two individual bandwidth blocks $RB_{i,j} = RB_{i,j}^1 + RB_{i,j}^2$.

Scenario 3: In general, if a connection request for backup path c_k arrives with the parameters Q_{c_k}, B_{c_k}, $P_{c_k}^{WP}$, and $\beta_{c_k}^L$, the existing RBB are searched similarly to scenario 2. If the current RBB can not be shared, the subsequent RBB needs to be checked. If there is no available RBB that can be shared, a new RBB is created as described in scenario 2.

B. LFP First-Fit vs. LFP Best-Fit

In the First-Fit backup resource allocation scheme, when a new connection request arrives, each node along the backup path checks the existing RBB in the link *sequentially*. Once the first suitable RBB satisfying all the constraints is found, sharing is granted, and the bandwidth of this RBB is updated to be the maximum value of all requests. If all the existing RBB can not be shared with this new request, a new RBB is allocated. The flow process of this algorithm is illustrated in Fig. 3.

A limitation of the LFP-FF algorithm is the fact that it may not check all existing RBB. In the case that an RBB is being shared by connections with very different bandwidth requirements, the resource utilization may not be efficient. An improvement over an LFP-FF algorithm is proposed, and called LFP Best-Fit (LFP-BF), as shown in Fig. 4.

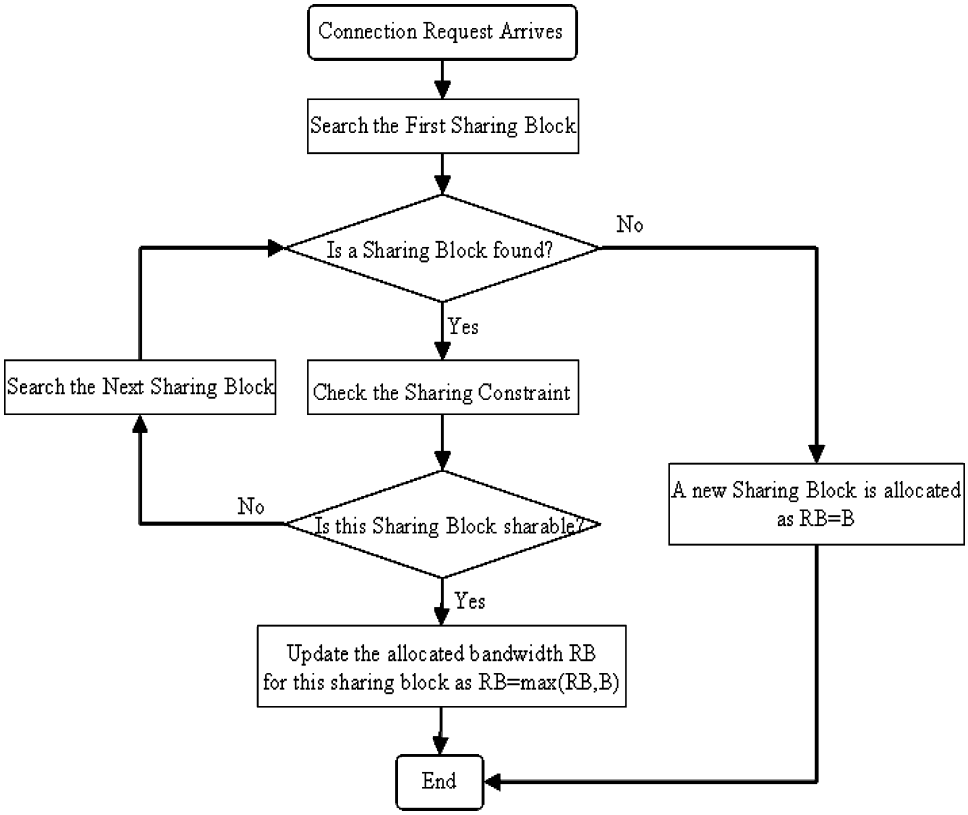

Fig. 3. Flow chart of the first-fit backup resource allocation.

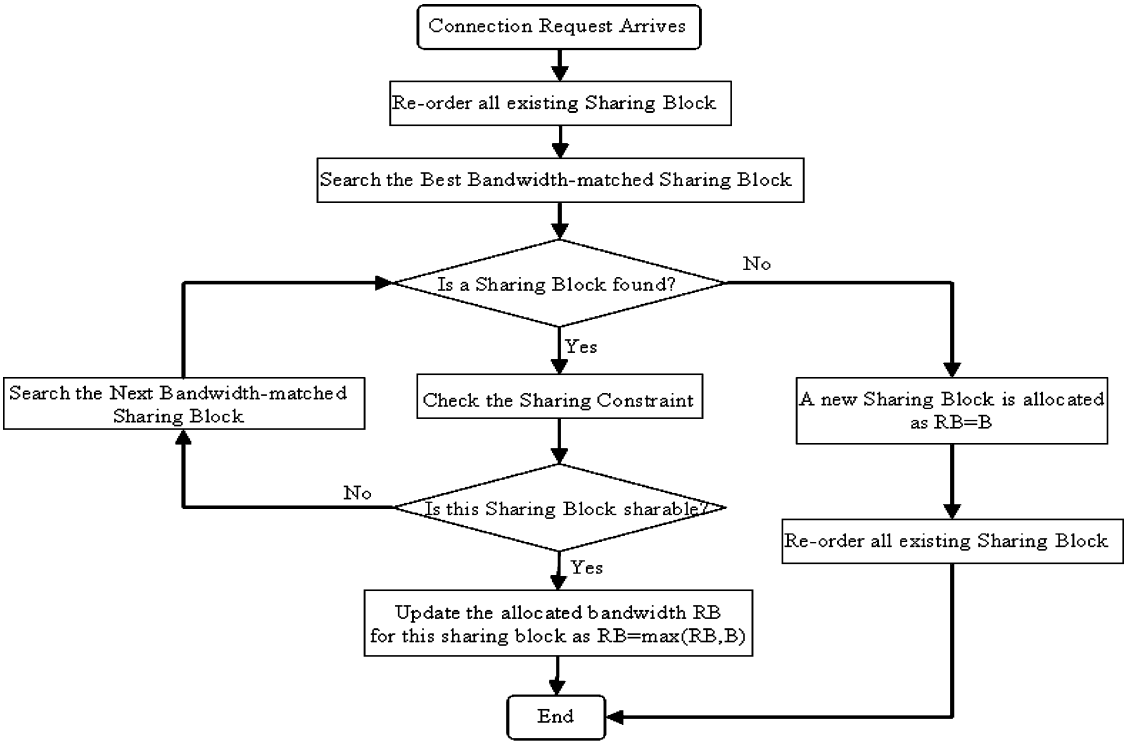

Fig. 4. Flow chart of the best-fit back-up resource allocation.

In the LFP-BF algorithm, all RBB are reordered by the values of the bandwidth reserved each time when a new connection is set up successfully. When a new connection request arrives, each node along the backup path first finds the RBB that matches its bandwidth requirement most closely, and then checks all the constraints. If the current RBB is not sharable, the node then

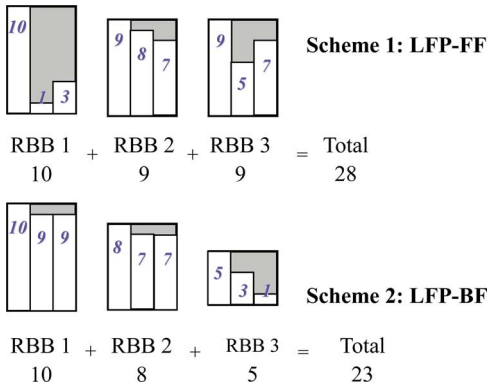

Fig. 5. Improvement of the backup bandwidth block management: LFP-BF vs. LFP-FF.

checks for a second best-matching RBB. This process will continue until all constraints are satisfied. The main idea behind the Best-Fit approach is to group the connections with similar bandwidth requirements into the same sharing block, as illustrated in Fig. 5. The overall bandwidth reserved can therefore be reduced.

C. Extensions to RSVP-TE

As mentioned above, the contributions of this paper are focused on the management of working, and protection paths instead of route selection. Therefore, we assume some kind of link state routing protocol is deployed, such as OSPF (Open Shortest Path First). The reason we need a link state routing protocol is that it can support the so-called K-shortest-path algorithm. A source node running a link state routing protocol maintains a global link state database. It first selects a shortest path based on Dijkstra's algorithm. Then the nodes that appear in the shortest path, except the source and destination nodes, are deleted from the database. The source node then selects another shortest path by rerunning the Dijkstra's algorithm again. This process continues until K shortest paths are selected, or all paths are exhausted. The K shortest paths selected in such a way are guaranteed to be node-disjoint. In the following parts, we assume K shortest paths for each source-destination pair are readily available at a source node.

In this paper, it is also assumed that the *RSVP Traffic Engineering* (*RSVP-TE*) extensions [29] are deployed for the setup, and tear-down of LSP. Extensions to support our proposed algorithms are discussed below. The whole signaling process is illustrated in Fig. 6. The LSP creation involves an initial signaling message from the source node to the destination node to set up the working path. An acknowledgement is returned from the destination node to the source node in order to complete the working path establishment. The acknowledgement message collects the information of the *SRLG* array, and the failure probability of each link along the working path, then brings the information back to the source node. The SRLG array is simply an array consisting of all connection ID whose working paths are in the same SRLG with the current working path. After the backup path is selected, the source node then issues another signaling message to the destination node over the backup path to reserve the backup resource. It carries the backup link failure probability constraint to each node along the backup path. If any node along the backup path can not satisfy all the constraints, the connection request is rejected immediately. Otherwise an acknowledge message will be sent back from the destination node to the source node over the backup path.

In a network $G = (V, E)$, we assume the network diameter is R, where R is defined as the maximum number of links that must be traversed to send a message to any node. We also assume that an average of N^{WP} working paths go through a link in the network, an average of N^{BP} backup paths share the same sharing block, and the average total number of sharing blocks in a link is assumed to be K. Then an upper bound of the extra size S of the signaling message carrying such information is estimated by $S = N^{WP} * R$, which is linearly increased depending on the number of links in each path, and the number of working paths going through each link. At each node along the backup path, the computation complexity of the availability constraint is $O((N^{BP})^2 * K)$ depending on the number of backup paths sharing each link. If such information is to be flooded to each node in the network using a conventional routing protocol, the network overhead will be very significant in a large network. Assuming a reasonable network diameter, and an efficient encoding of the message, we do not anticipate any problem with the message size for our proposed algorithms because they do not need to flood information repeatedly. All required information is collected using signaling messages on an as-needed basis.

IV. Numerical Results

During this research, a discrete event-driven simulation model was built using C++. In the following parts, we present both analytical, and simulation results. Firstly, a simplified network model is introduced & analysed. By comparing the simulation results to the mathematical analysis for the simplified model, we provide some insights into the correctness, and efficiency of the proposed algorithms. Then these algorithms are evaluated in a more realistic network model, the NSFNET backbone model. Two performance metrics, Call Acceptance Rate, and Total Bandwidth Consumption, are investigated. Both sets of simulation results are compared with the performance of the DPP scheme. The reason that DPP is being used for the comparison instead of other SPP schemes is because the conventional SPP schemes have different goals from our proposed algorithm. As discussed in Section I, most of the SPP schemes are based on the algorithm for bandwidth efficiency under a single-failure assumption. Those algorithms will fail when multiple concurrent failures occur in the network. Therefore, the performances are not comparable with our proposed algorithms. Because the DPP algorithm can provide the same level of availability as our algorithms under the condition that the probability of a concurrent failure of the working and backup paths is low, we compare the performance of our proposal with the DPP scheme.

A. Description of a Simplified Network Model

For the best understanding of our LFP-based algorithms, and validating that the simulation model is built correctly, a simplified network model is analysed as shown in Fig. 7. Although this network model might not reflect a real system, it still gives

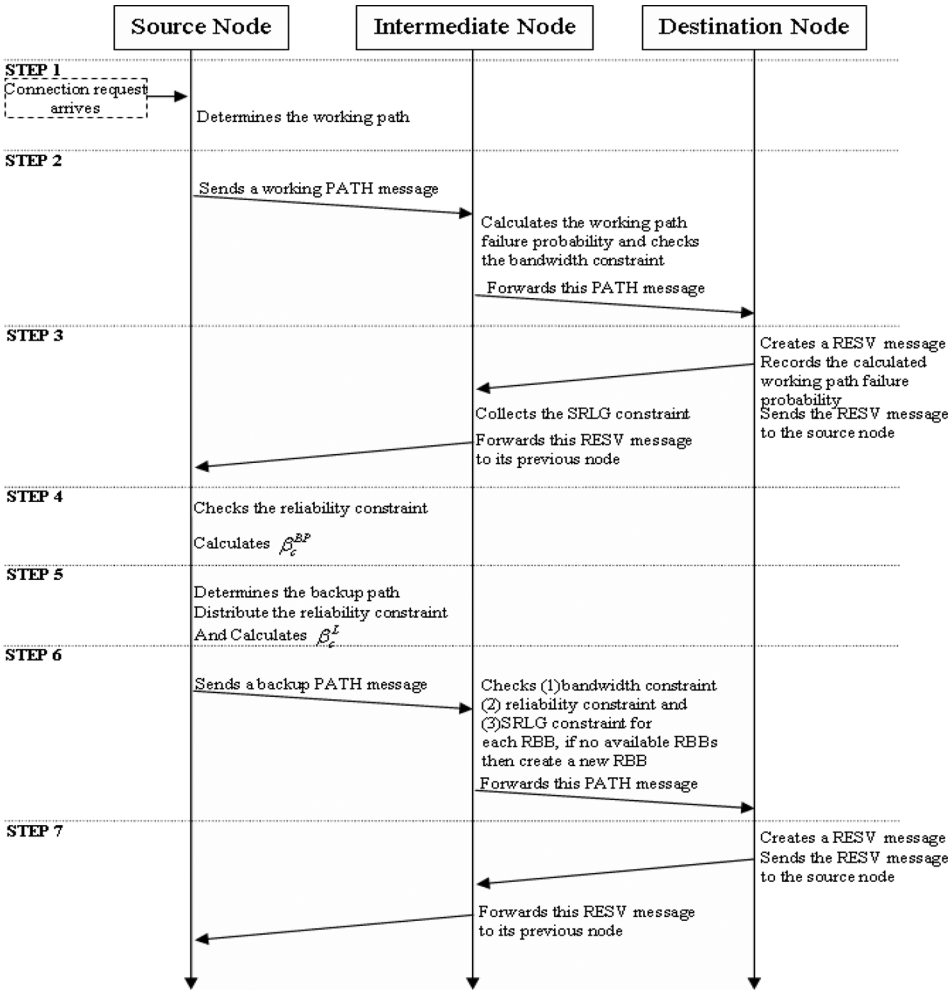

Fig. 6. The signaling process of LFP based algorithms.

us some insights about the impacts of different factors on the performance of the proposed algorithms. These factors include the correlated working paths (SRLG) that happen to share the same backup link, and the availability constraints on the calculation of backup resource sharing. For this set of experiments, the number of source-destination pairs M are fixed at $M = 40$. The link capacities along the working paths (e.g. links along the path 1-2, the path 3-4, and the path 5-6, etc.) are set to infinity. Because the backup paths of all connections go through the same link (m, n), the bandwidth capacity of link (m, n) is the bottleneck for setting up connections. If no available bandwidth in link (m, n) can be shared, a new connection request will be rejected. The working paths of the connections with the same source-destination pair go through the same set of links; therefore, they belong to the same SRLG. For example, both the working path of connection $c_1(WP_{c_1})$, and the working path of connection $c_2(WP_{c_2})$ belong to the same SRLG because they share the common links between the source node 1, and the destination node 2. Requests asking for restorable connections arrive at each source node (node 1, node 3, node 5, etc,) with the corresponding destination node (node 2, node 4, node 6, etc.) according to a Poisson process with an average rate of $\lambda = 1/50$. The average inter-arrival time of the requests is set to *50* hours. Furthermore, we assume the holding time of each connection request follows an exponential distribution with their mean values incremented from 10 to 2000 to simulate different loads. All connections can be restored with the same availability requirement $Q_c = 99.99\%$. All the working paths have the same failure probability $P_c^{WP} = 0.1\%$. The failure probability of the backup link (m, n) is set to $P_{m,n}^L = 0.1\%$, and the capacity of the link (m, n) is *50* bandwidth units. To make the description easier, we assume all other links of all the backup paths have infinite capacity, and 100% availability. The performance metric used in this simulation is the Call Acceptance Rate (the compliment of the Call Loss Rate), which is the ratio of the number of successful connections over the total number of connection requests, denoted by

$$R = \frac{Total\_Number\_of\_Successful\_Connections}{Total\_Number\_of\_Connections} \quad (15)$$

B. Mathematical Analysis for the Simplified Model

Two cases are analysed. In Case 1, the bandwidth demand is fixed to $B_c = 5$; and in Case 2, the bandwidth demand is a random variable governed by a discrete uniform distribution between 1, and 10, which is the same case as the NSFNET simulation scenario discussed later.

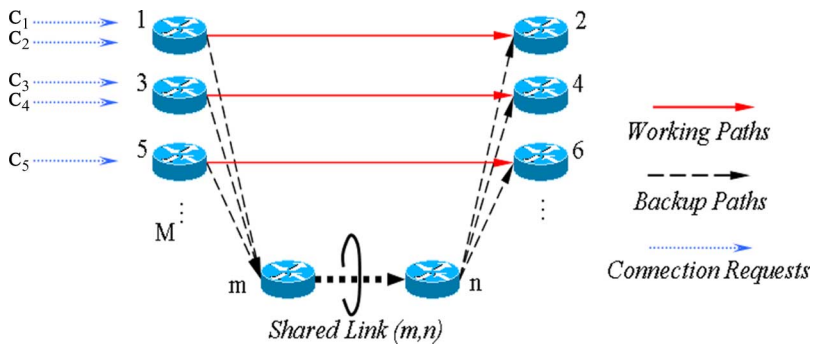

Fig. 7. A simplified network model.

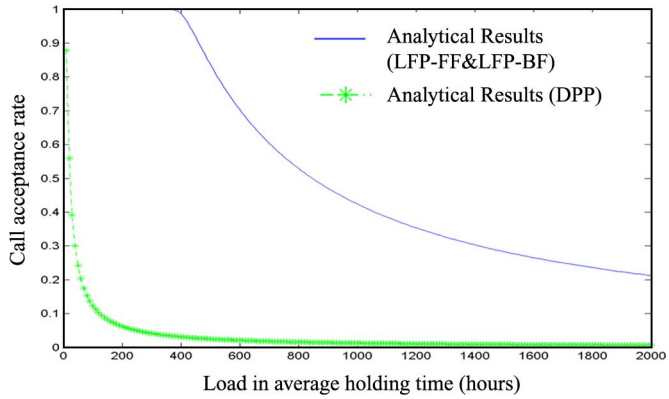

Fig. 8. Analytical results: call acceptance rate vs. network load in the simplified network model.

Firstly, we consider the case that the bandwidth demand by each connection request is fixed to $B_c = 5$ bandwidth units. Because connection requests arrive at each source node according to a Poisson process with an average rate of $\lambda = 1/50$, and the holding time is exponentially distributed with a mean value of $1/\mu$, this telecommunication system can be modeled as an $M/M/r/r$ queuing system. In an $M/M/r/r$ queuing system, the arrival process, and the service process follow Poisson distributions with mean arrival rate λ, and mean service rate μ, respectively. There are r servers, and no waiting queue in the system, so that the system capacity equals the number of parallel servers. Therefore, in such a system, an arriving customer finding all servers busy will not enter the system, and be considered lost. For the DPP scheme, the number of servers r is calculated as $r = B_{m,n}/B_c = 50/5 = 10$, where $B_{m,n}$ is the bandwidth capacity of link (m,n), and B_c is the bandwidth demand of each connection request c. For the LFP-FF, or LFP-BF algorithms, the maximum number of connections in the system (i.e. the number of servers) can be estimated as $r = C_{RBB} * K = 34 * 10 = 340$, where $C_{RBB} = 34$, calculated by (8), is the maximum number of connections an RBB in link (m,n) can accommodate, limited by the availability constraint.

The long run proportion of time that the system is full, or the probability that all servers are busy, can be calculated using the famous Erlang B Loss formula [31]. We calculate the mathematical results of the Call Acceptance Rate under different network loads, as shown in Fig. 8. The dotted curve is the mathematical result of the DPP scheme, and the solid curve is the result of LFP-FF or LFP-BF. Because the bandwidth demand is a fixed value, both LFP-FF, and LFP-BF schemes should achieve the same performance.

Now we consider the second case that the bandwidth demand by each connection request is uniformly distributed from 1 to 10 bandwidth units.

For the DPP algorithm, this system can be modeled as another kind of multi-server loss system called a Stochastic Knapsack System [32]. The classical stochastic knapsack problem involves a "knapsack" of capacity B resource units, and X classes of objects, with class-x objects having size b_x. Objects may be placed into the "knapsack" as long as the sum of their sizes does not exceed the knapsack capacity. This problem aims at placing the objects into the "knapsack" so as to maximize the long run number of objects admitted by the system. In our case, the capacity of link (m,n) is set at $B_{m,n} = 50$. The total number of classes is $X = 10$ with the request of each class-x denoted by $b_x = x$ $(x = 1, 2 \ldots, 10)$. Class-x objects arrive at the system according to a Poisson process with rate $\lambda' = \lambda * M/X$; where $\lambda = 1/50$ is the arrival rate of each connection request for each source-destination pair, and $M = 40$ is the total number of source-destination pairs. If an arriving class-x object is admitted into the system, it takes b_x resource units for a holding time that is exponentially distributed with mean $1/\mu$. In [32], an expression is given for calculating the class-x objects blocking probability P_x as in (16).

$$P_x = 1 - \frac{\sum_{n \in Sx} \prod_{j=1}^{X} \rho_j^{n_j}/n_j!}{\sum_{n \in S} \prod_{j=1}^{X} \rho_j^{n_j}/n_j!} \tag{16}$$

In (16), S is the set of all possible states in the system, and S_k is the subset of states in which the knapsack admits an arriving class-x object. A more efficient recursive algorithm is also presented in [32], which does not involve brute-force summation. Using this recursive algorithm, we draw the mathematical result of the DPP scheme shown as the solid curve in Fig. 9.

For LFP-FF, and LFP-BF algorithms, the system can still be approximately modeled as an $M/M/r/r$ queuing system, as discussed in Case 1, except that the server number r is random due to the fact that each block size is variable now. To simplify the process, we use the mean value of r as the number of servers. We start with the following Lemma.

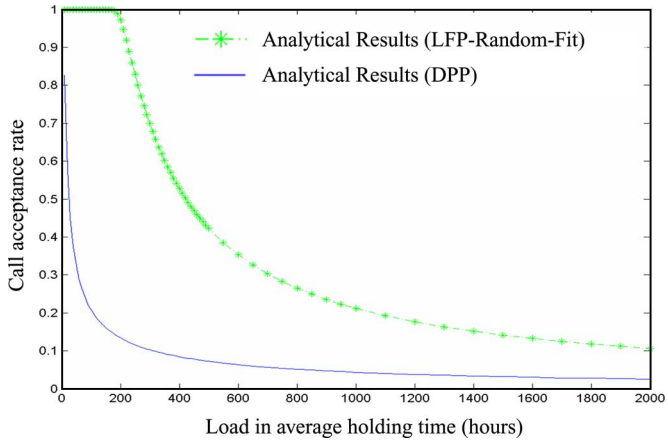

Fig. 9. Analytical results: call acceptance rate vs. network load in the simplified network model in case 2 (random bandwidth demand).

Lemma 5: Let $X_1, X_2, \cdots, X_N$ be a set of independent random variables having the distribution functions $F_{X_1}, F_{X_2}, \cdots, F_{X_N}$ respectively. Define $Y_N = \max(X_1, X_2, \cdots, X_N)$ to be the largest of those random variables. Then the probability distribution functions $F_{Y_N}(y)$ can be calculated by $F_{Y_N}(y) = F_{X_1}(y)F_{X_2}(y)\cdots F_{X_N}(y)$for all y.

From Lemma 5, assuming that each connection is randomly assigned to a RBB, the distribution function of RB^i_{mn} can be calculated by

$$F_{RB^i_{mn}}(y) = \prod_{c_j \in T^i_{mn}} \left[F_{B_{c_j}}(y)\right] = \left[F_{B_{c_j}}(y)\right]^{C_{RBB}}$$

Thus, the mean value of RB^i_{mn} is calculated by

$$\begin{aligned} E\left(RB^i_{mn}\right) &= \sum_{j=1}^{10} j * P\left(RB^i_{mn} = j\right) \\ &= \sum_{j=1}^{10} j * \left(F_{RB^i_{mn}}(j) - F_{RB^i_{mn}}(j-1)\right) \\ &= \sum_{j=1}^{10} j * \left(\left(\frac{j}{10}\right)^{C_{RBB}} - \left(\frac{j-1}{10}\right)^{C_{RBB}}\right) \\ &= 10 - \sum_{j=1}^{9} \left(\frac{j}{10}\right)^{C_{RBB}} \approx 10 \end{aligned}$$

Therefore, $r = C_{RBB} * E(RB^i_{mn}) = 34 * 50/10 = 170$ instead of 340. We therefore can calculate the blocking probability, or call acceptance rate for the LFP-based Random Fit scheme under Case 2. The analytical results are shown as the dotted curve in Fig. 9.

C. Simulation Results for the Simplified Model

Fig. 10 shows the simulation results for the simplified model under the case of fixed bandwidth demand.

The simulation results shown in Fig. 10 demonstrate that both the LFP-FF, and the LFP-BF schemes perform much better than the DPP scheme in terms of the Call Acceptance Rate in this simplified network model. There is no difference between the performance of LFP-BF, and LFP-FF for the given scenario because the bandwidth demands are a fixed value. With an increase in the network load, the Call Acceptance Rates of all three schemes decrease, but the DPP scheme diminishes more quickly than the other two schemes. Fig. 10 also shows that the simulation results are very much similar to the mathematical analysis. There is a minor difference between the simulation results, and the theoretical results of LFP-FF/LFP-BF under low network load. The reason for the difference is because, during the mathematical analysis, we neglect the correlation of the working paths (it was assumed that the working paths of all connections do not share any link). During the simulation, there exists the shared risk issue, which has to be considered; for example, two connections with the same source-destination pair will have the same shared risk, and thus can not share the same bandwidth block. The sharing contentions between the correlated working paths can block some potential connections, thus causing the capacity of the system to decrease. For the DPP scheme, the sharing feature is not implemented, and the contention between the correlated working paths does not exist; therefore the curve of the simulation results of DPP scheme matches the analysis results very well. Fig. 11 shows the simulation results for the simplified model under the case of random bandwidth demand.

The simulation results shown in Fig. 11 demonstrate that the performance of LFP-BF is better than LFP-FF for this scenario,

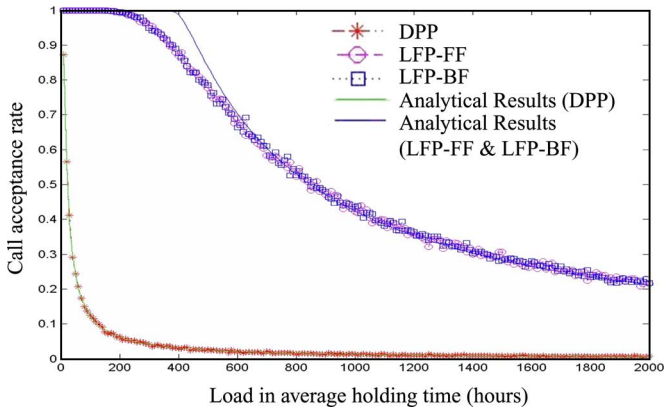

Fig. 10. Simulation results: call acceptance rate vs. network load in the simplified network model in case 1 (fixed bandwidth demand).

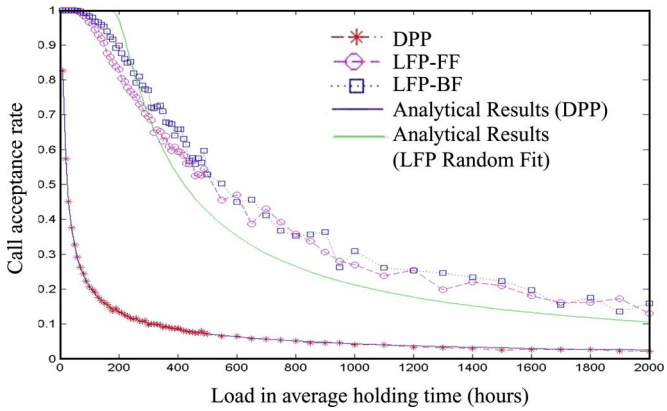

Fig. 11. Simulation results: call acceptance rate vs. network load in the simplified network model in case 2 (random bandwidth demand).

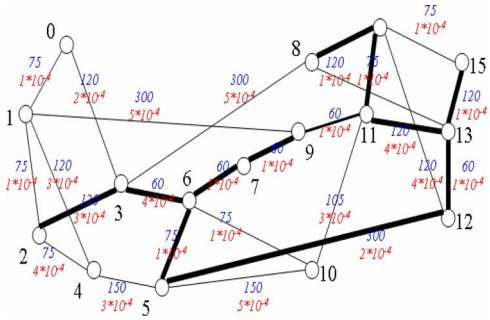

Fig. 12. A NSFNET simulation model.

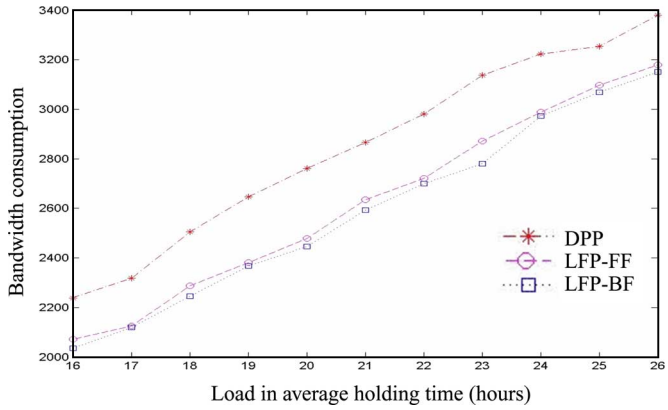

Fig. 13. NSFNET simulation result: total bandwidth consumption vs. network load.

but there is very little difference between the curves. Fig. 11 also shows that the simulation results of the DPP are similar to the analytical results.

Once again, the correctness of the simulation model is validated. Some configuration parameters are also used in the NSFNET simulation scenario discussed in the following paragraphs.

D. Simulation Results With NSFNET

To further investigate our proposed schemes, a simulation was performed using the *NSFNETt* backbone, as shown in Fig. 12.

Two metrics are investigated in the simulation to evaluate the performance of our backup resource allocation algorithms. The first metric is the Average Call Acceptance Rate (R),

$$R = \frac{Total\_Number\_of\_Successful\_Connections}{Total\_Number\_of\_Connections}$$

. The Average Call Acceptance Rate R is very important, as it measures the call throughput. The second metric is the Total Bandwidth Consumption $\sum_{(i,j)\in E}(SB_{ij}+RB_{ij})$, where SB_{ij} is the total service bandwidth used by the working paths in link (i,j), and RB_{ij} is the total reserved bandwidth used by the backup paths in link (i,j). $\sum_{(i,j)\in E}(SB_{ij}+RB_{ij})$ is the sum of the total consumed bandwidth over all the links in the network. The goal of this metric is to measure the capability of accommodating traffic for the three algorithms.

The topology used in this simulation is the backbone NSFNET (National Science Foundation Network). The example shown in Fig. 12 has 16 nodes, and 25 links. The capacity of each link is 120 (shown as thin lines), and 480 (shown as bold lines) bandwidth units. The links are bi-directional. Fig. 12 also illustrates the link weight (upper number), and the link failure probability (lower number). We assume traffic follows a uniform distribution, i.e. an arrival will choose one out of all possible source & destination pairs with equal probability. The connection requests arrive randomly at the same average rate for all source-destination pairs. Requests arrive at each source-destination pair according to a Poisson distribution with an average rate λ, and the holding times are exponentially distributed with a mean $1/\mu$. The simulations are run over a period of 10,000 hours. 30% of these requests are non-protected traffic, and 70% of these requests are protected traffic. All the protected traffic connections are setup by the above mentioned algorithms. Among the protected traffic connections, 30% are protected with the availability requirement of 99.95%, 30% with the availability of 99.97%, and the other 40% with the availability of 99.99%. The allocated bandwidth for each LSP is uniformly distributed between 1, and 10 bandwidth units.

In Fig. 13, the Total Bandwidth Consumption is plotted as a function of the Network Load for the NSFNET. LFP-BF has the least total consumed bandwidth, and the DPP scheme has the most total consumed bandwidth. The curve of the LFP-FF scheme lies between the LFP-BF, and the DPP scheme; and is very close to the LFP-BF. With the increase of the traffic load, the Total Bandwidth Consumptions for those three protection schemes also increase. The performances of the LFP-FF, and the LFP-BF are very close to each other. The reason is due to the fact that many factors may affect the performance of a LFP algorithm. The LFP-BF is only different from the LFP-FF on the way of selecting an RBB to be shared. Other factors such as the availability or the SRLG constraints remain the same for both algorithms. Therefore, in some cases, the LFP-BF performs much better than the LFP-FF if the bandwidth is the main criteria, such as in the Simplified Network Model; whereas in some other cases, the LFP-BF performs only slightly better than the LFP-FF. In this network model, the bandwidth sharing feature between the LFP-FF, and the LFP-BF is not particularly distinguishable.

Fig. 14 shows the Call Acceptance Rate under different network loads. The figure shows that both the LFP-FF, and the LFP-BF perform better than the DPP scheme in terms of the Call Acceptance Rate (or connection rejection rate). Under this particular network topology, there is little difference between LFP-FF, and LFP-BF. It should be noted that the results in Fig. 14 are similar to Fig. 11. This shows that the insights gained from the simplified network model should be still valid for more general networks like the NSFNET.

Another metric used in this simulation is the restoration overbuilds denoted by $\delta = \sum_{(i,j)\in E} RB_{ij} / \sum_{(i,j)\in E} SB_{ij}$, as mentioned by G. Li [33]. $\sum_{(i,j)\in E} SB_{ij}$ is the sum of service bandwidth over all links in the network, and $\sum_{(i,j)\in E} RB_{ij}$ is the sum of reserved bandwidth over all links in the network. The metric $\delta = \sum_{(i,j)\in E} RB_{ij} / \sum_{(i,j)\in E} SB_{ij}$ calculates the

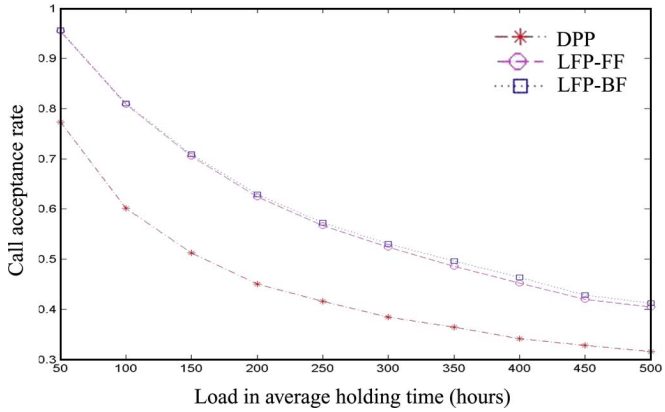

Fig. 14. NSFNET simulation results: call acceptance rate vs. network load.

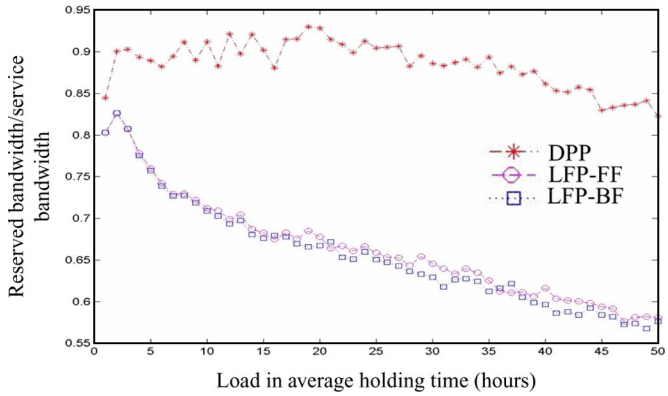

Fig. 15. NSFNET simulation results: ratio of total reserved bandwidth to total service bandwidth vs. network load.

extra bandwidth that is needed to meet the network restoration objective.

Fig. 15 shows that the LFP-FF scheme, and the LFP-BF scheme have better bandwidth usage than the DPP scheme under a heavy load. If the network load is not heavy, the difference in performance is very limited. LFP-BF performs better than LFP-FF, but there is not much difference between the two schemes because of this specific network topology.

V. Conclusion

In Section I, it was discussed that the legacy SPP algorithms can guarantee 100% availability only against single-failure events in a network. The DPP algorithms tried to solve this problem by allocating dedicated bandwidth to each backup path. However, there exist two immediate problems with this approach. Firstly, the bandwidth utilization is low because backup paths are typically idle most of the time. Secondly, the DPP schemes may also fail to satisfy customers' availability requirements if the probability that both the working path, and the backup path fail at the same time is high.

Our proposed solutions addressed multiple failure scenarios by directly taking customers' availability requirements into account. The probability of multiple failures was incorporated through (2) to (9), under the assumption that all failures are statistically independent. Under a single failure scenario, if the availability of the working path itself cannot satisfy the availability requirement, our proposed algorithms will be similar to the legacy SPP algorithms by setting up a sharable backup path. Therefore, all the legacy SPP algorithms are a special case of our proposed algorithms. On the other hand, if the backup path happens to use a new sharing block on each of the links it traverses due to the availability constraint of (8), the proposed algorithm will be similar to the DPP algorithms. Therefore, the DPP algorithms are also a special case of the proposed algorithm.

Because our proposed backup resource sharing algorithms are based on the analysis of the LFP value of each link along the working paths, and the backup paths; not only the availability requirement of the current connection can be satisfied, but the potential impacts of the current connection on the availability of other existing connections are also considered. Therefore, all the availability constraints are satisfied concurrently for all connections in the network.

During the selection of an RBB to be shared, we have taken variable bandwidth requirements of different connections into account. First Fit, and Best Fit are two algorithms proposed to maximize the efficiency of bandwidth sharing.

For our proposed schemes, both the LFP parameters, and SRLG arrays are collected by signaling messages during the connection setup stage. No repetitive flooding of global information is required. Information is distributed only on an as-needed basis. This has significantly reduced the overhead needed to support our proposed algorithm, and made our algorithms scalable to large networks. We have demonstrated that our proposed algorithms can be integrated into existing signaling protocols seamlessly with minor changes.

In summary, the proposed algorithms incorporate various features, such as multiple failures, SRLG issues, network scalability, resource efficiency, guaranteed availability, etc. Joint consideration of these issues helped achieve a better performance in terms of Call Acceptance Rate, and Total Bandwidth Consumption in a large network.

Simulation, and analytical results have demonstrated that our proposed algorithms outperform DPP in both bandwidth consumption, and call acceptance rate. LFP-BF performs slightly better than LFP-FF, and the gain is variable with different network & traffic scenarios.

This research can be used to guide many other future researches. One such research direction is to combine the proposed algorithm with a network resource optimization algorithm so that a new routing approach can be developed. Another area of future research is to apply our proposed algorithms to other protection schemes, such as the Local Repair, or the Segmented Repair. Future work will also need to focus on the improvement of the sharing blocks management algorithms to further improve the performance.

References

[1] R. Bhandari, *Survivable Networks—Algorithms for Diverse Routing*. Norwell, MA: Kluwer Academic Publishers, 1999.

[2] S. Jha and M. Hassan, *Engineering Internet QoS*. : Artech House, August 2002.

[3] A. Farrel and B. Miller, Surviving Failures in MPLS Networks February 2001, Technical report, Data Connection.

[4] P.-H. Ho and H. T. Mouftah, "Reconfiguration of spare capacity for MPLS-based recovery in the internet backbone networks," *IEEE/ACM Trans. Networking*, vol. 12, pp. 73–84, February 2004.
[5] S. Ayandeh, "Convergence of protection and restoration in telecommunication networks," *Photonic Network Communications*, vol. 4, pp. 237–250, July 2002.
[6] B. T. Doshi, S. Dravida, P. Harshavardhana, O. Hauser, and Y. Wang, "Optical network design and restoration," *Bell Labs Technical Journal*, pp. 58–84, January–March 1999.
[7] G. Mohan and C. S. R. Murthy, "Lightpath restoration in WDM optical networks," *Proceedings, IEEE Network*, vol. 14, pp. 24–32, Nov.–Dec. 2000.
[8] S. Ramamurthy and B. Mukherjee, "Survivable WDM mesh networks. Part I—Protection," in *Proceedings, IEEE INFOCOM '99*, March 1999, vol. 2, pp. 751–774.
[9] S. Ramamurthy and B. Mukherjee, "Survivable WDM mesh networks. Part II—Restoration," in *ICC'99*, June 1999, vol. 3, pp. 2023–2030.
[10] V. Sharma and F. Hellstrand, "Framework for multi-protocol label switching (MPLS)-based recovery," RFC-3469, Feb. 2003.
[11] P. H. Ho, J. Tapolcai, and H. T. Mouftah, "On optimal diverse routing for shared protection in mesh WDM networks," *IEEE Trans. Reliability*, vol. 53, no. 6, pp. 216–225, June 2004.
[12] T. J. Li and B. Wang, "On optimal p-cycle-based protection in WDM optical networks with sparse-partial wavelength conversion," *IEEE Trans. Reliability*, vol. 55, no. 3, pp. 496–506, Sep 2006.
[13] A. Haque and P. H. Ho, "A study on the design of survivable optical virtual private networks (O-VPN)," *IEEE Trans. Reliability*, vol. 55, no. 3, pp. 516–524, Sep 2006.
[14] P. H. Ho, J. Tapolcai, and H. T. Mouftah, "On achieving optimal survivable routing for shared protection in survivable next-generation Internet," *IEEE Trans. Reliability*, vol. 53, no. 2, pp. 216–225, Jun 2004.
[15] C. Huang, V. Sharma, K. Owens, and S. Makam, "Building reliable MPLS networks using a path protection mechanism," *IEEE Communications Magazine*, vol. 40, no. 3, pp. 156–162, March 2002.
[16] E. Bouillet, J. Labourdette, G. Ellinas, R. Ramamurthy, and S. Chaudhuri, "Stochastic approaches to compute shared mesh restored lightpaths in optical network architectures," in *Proceedings, IEEE INFOCOM 2002*, June 2002, vol. 2, pp. 801–807.
[17] M. Kodialam and T. V. Lakshman, "Dynamic routing of bandwidth guaranteed tunnels with restoration," in *Proceedings, IEEE INFOCOM 2000*, March 2000, vol. 2, pp. 902–911.
[18] M. Kodialam and T. V. Lakshman, "Dynamic routing of locally restoration bandwidth guaranteed tunnels using aggregated link usage information," in *Proceedings, IEEE INFOCOM 2001*, April 2001, vol. 1, pp. 376–385.
[19] G. Mohan and A. K. Somani, "Routing dependable connections with specified failure restoration guarantees in WDM networks," in *Proceedings, IEEE INFOCOM 2000*, March 2000, vol. 3, pp. 1761–1770.
[20] Y. Xiong *et al.*, "Achieving fast and bandwidth-efficient shared-path protection," *IEEE Journal of Lightwave Technology*, vol. 21, no. 2, pp. 365–371, February 2003.
[21] Y. Wang, Q. Zeng, and H. Zhao, "Dynamic survivability in WDM mesh networks under dynamic traffic," *Photonic Network Communications*, vol. 6, no. 1, pp. 5–24, July 2003.
[22] S. Lee, D. Griffith, and N.-O. Song, "An analytical approach to shared backup path provisioning in GMPLS networks," in *Proceedings, IEEE MILCOM 2002*, October 2002, vol. 1, pp. 75–80.
[23] S. Datta, S. Sengupta, S. Biswas, and S. Datta, "Efficient channel reservation for backup paths in optical mesh networks," in *Proceedings, IEEE GLOBECOM 2001*, November 2001, vol. 4, pp. 2104–2108.
[24] M. Tacca, A. Fumagalli, and F. Unghvary, "Double-fault shared path protection scheme with constrained connection downtime," in *Proceedings, DRCN 2003*, October 2003, pp. 181–188.
[25] H. Choi, S. Subramariam, and H.-A. Choi, "On double-link failure recovery in WDM optical networks," in *Proceedings, IEEE INFOCOM 2002*, June 2002, vol. 2, pp. 808–816.
[26] K. Wu, L. Valcarenghi, and A. Fumagalli, "Restoration schemes with differentiated reliability," in *IEEE ICC 2003*, May 2003, vol. 3, pp. 1968–1972.
[27] W. D. Grover and M. Clouqueur, "Availability analysis of span-restorable mesh networks," *IEEE Journal in Selected Areas in Communications*, May 2002.
[28] A. B. McDonald and T. Znati, "A mobility based framework for adaptive clustering in wireless ad-hoc networks," *IEEE Journal in Selected Areas in Communications (JSAC)*, vol. 17, no. 8, pp. 1466–1487, August 1999.
[29] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP tunnels," IETF RFC 3209, December 2001.
[30] P. Pongpaibool and H. S. Kim, "Guaranteed service level agreements across multiple ISP networks," in *Proceedings, DRCN 2003*, October 2003, pp. 325–332.
[31] C. H. Ng, *Queueing Modeling Fundamentals*. : John Wiley & Sons, Jan. 1997, p. 116.
[32] K. W. Ross, *Multiservice Loss Models for Broadband Telecommunication Networks*. : Springer, 1995.
[33] G. Li, D. Wang, C. Kalmanek, and R. Doverspike, "Efficient distributed path selection for shared restoration connections," in *Proceedings, IEEE INFOCOM 2003*, October 2003, vol. 11, pp. 761–771.

Changcheng Huang (M'00–SM'06) received his B.Eng. in 1985, and M.Eng. in 1988, both in Electronic Engineering from Tsinghua University, Beijing, China. He received a Ph.D. degree in Electrical Engineering from Carleton University, Ottawa, Canada in 1997. From 1996 to 1998, he worked for Nortel Networks, Ottawa, Canada where he was a systems engineering specialist. He was a systems engineer, and network architect in the Optical Networking Group of Tellabs, Illinois, USA during the period of 1998 to 2000. Since July 2000, he has been with the Department of Systems and Computer Engineering at Carleton University, Ottawa, Canada where he is currently an associate professor.

Dr. Huang won the Canada Foundation for Innovation (CFI) new opportunity award for building an optical network laboratory in 2001. He was an associate editor of IEEE COMMUNICATIONS LETTERS from 2004 to 2006. He is currently a senior member of IEEE.

Minzhe Li received his M.A.Sc. in Electrical Engineering from Carleton University, Ottawa, Canada in 2004.

Anand Srinivasan holds a Ph.D., and M.Sc. in computer science from the University of Victoria, British Columbia, Canada. In addition, he holds a Master's degree in Computing from JNU, New Delhi, and a Bachelor's degree from the University of Delhi, India.

Dr. Srinivasan has over 15 years experience in the system, and network design, and performance for large scale wired, wireless, and satellite networks. He is the principal architect for developing EION's Mobile Ad Hoc networking wireless technology for defense applications, and QoS & MPLS protocol suites for IP networks. He is at present the director of engineering at EION Wireless working towards development of Wimax based products. He contributed as a technical specialist in the areas of performance, and simulation of large-scale metro optical networks using innovative multi-constrained routing algorithms. He holds two patents for multi-constrained routing in optical networks. Prior to joining EION, Dr. Srinivasan worked in Nortel Networks on product performance. He has published over 25 papers in the areas of operating systems, distributed systems, fault-tolerance, and optimization.

Dr. Srinivasan has served in several research panels, and conference technical committees. He is a member of IEEE, and AIAA.