An Ultra-fast Shared Path Protection Scheme -Distributed Partial Information Management, Part II*

Dahai Xu Chunming Qiao Yizhi Xiong
Department of Computer Science and Engineering
State University of New York at Buffalo
{dahaixu, qiao, yxiong}@cse.buffalo.edu

Abstract—

This paper describes a novel, ultra-fast heuristic algorithm to address an NP-hard optimization problem. One of its significances is that, for the first time, the paper shows that a heuristic algorithm can also have better overall performance than its time-consuming, Integer Linear Programming (ILP) based counterparts in the on-line case, which is non-intuitive.

The proposed heuristic algorithm is useful for developing effective shared path (mesh) protection schemes that establish survivable connections in modern networks. The advantage of our heuristic algorithm over existing algorithms for finding a pair of link (or node) disjoint paths, Active Path (AP) and Backup Path (BP) comes from the following salient feature. It uses a so-called *Potential Backup Cost (PBC)* function when selecting an AP in the first phase, in order to take into consideration the backup bandwidth needed by the corresponding BP yet to be chosen in the second phase. The PBC function is derived mathematically based on rigorous statistical analysis of experimental data. While the use of PBC only requires partial aggregate information on existing connections and distributed control, it can also be applied even more effectively when complete information is available.

I. Introduction

Many emerging applications including nation-wide collaborative science and engineering projects require that reliable, high-bandwidth connections be dynamically set up and released between large computing resources (e.g., storage with terabytes to petabytes of data, clustered supercomputers and visualization displays). To meet the requirements of these emerging applications in an economical way, a network must be able to quickly provision bandwidth-guaranteed survivable connections (i.e., connections with sufficient protection against possible failures of network components such as links or nodes). Driven by service providers' desire to generate more revenues from existing infrastructure, much frequent connection set-ups and releases are expected in the future. This, in conjunction with the large (and increasing) size of the Internet, necessitates a scalable solution to the above on-line provisioning problem. Being able to fast provision survivable connections with guaranteed QoS (such as bandwidth) is also a *key feature* of the envisioned next generation networks [1–5]. In this work, we will introduce such a scalable scheme that can achieve optimal bandwidth efficiency (and revenue) at an ultra-fast request processing speed under distributed control.

*This project is supported in part by NSF under the contract ANIR 0208331

In a high-speed network, a link (e.g., a fiber) can carry up to a few terabits per second. Such a link may fail due to human mistakes (e.g., mis-configuration), software bugs, hardware defects (or simply the aging of some materials), natural disasters (e.g., flooding or earthquakes), or even perpetrators (e.g., terrorists or hackers). As our national security, economy and even day-to-day life rely more and more on computer and telecommunication networks, avoiding disruptions to information exchange due to unexpected failures becomes increasingly important.

A. Problem Description

Shared path protection is an effective method used in modern survivable networks to protect a connection from the failure of any single link or a single node (other than the ingress or egress node of the connection). With path protection, a link (or node)-disjoint pair of paths from an ingress node to an egress node, called Active Path **(AP)** and Backup Path **(BP)**, respectively, is used to satisfy each connection. Assume that a connection requires w units of bandwidth. The amount of bandwidth needed on each link along the AP to establish the connection, hereafter referred to as Active Bandwidth **(ABW)**, is w units. When there is no sharing of bandwidth of any kind (the case with bandwidth sharing will be discussed in the next paragraph), the amount of bandwidth to be reserved on each link along the BP, hereafter referred to as Backup Bandwidth **(BBW)**, is also w units. This BP will be used to (re-)establish the connection to re-route the information to be carried after the AP breaks due to a link failure along the AP (and before the AP can be restored). Such a path protection scheme will be referred to as the No Sharing **(NS)** scheme.

The concept of *shared path protection* is illustrated using the following example. Assume that two connections, requiring w_1 and w_2 units of bandwidth, respectively, are established using two link (or node) disjoint APs. Note that under such an assumption, the two APs cannot break at the same time *provided that* at most one link (or node) in the network can fail at any given time(or more precisely, no additional failures may occur before an existing failure is repaired, which is a fairly practical and reasonable assumption). Accordingly, their corresponding

Proceedings of the 10 th IEEE International Conference on Network Protocols (ICNP'02)
1092-1648/02 $17.00 © 2002 IEEE

BPs need not be used to (re)-establish the two connections, respectively, at the same time either. Hence, if the two corresponding BPs use the same link e, they can share the BBW without affecting the survivability of either connection. More specifically, with shared path protection, the total BBW that needs be allocated on link e (for the two BPs) is $\max\{w_1, w_2\}$ (instead of $w_1 + w_2$). While approaches other than shared path protection may be adopted to achieve a good degree of survivability, this work focuses on shared path protection for its quick restoration speed (without requiring fast fault localization) and high bandwidth efficiency.

In shared path protection, the problem of minimizing the Total Bandwidth **(TBW)** (which is the sum of ABW and BBW) needed to satisfy a given set of requests for establishing connections is NP-hard. Intuitively, this is because in order for two BPs to share their BBW on a common link, their corresponding AP's must be link (or node) disjoint. Thus, minimization of TBW requires joint-optimization of both the AP and BP selection. Many *off-line* schemes with applications to SONET, ATM and WDM networks case have been proposed (see for example the work in [6–8] and the references contained therein).

In the on-line case to be studied, not all requests arrive at the same time, and a decision as to how to satisfy a request for connection establishment (if possible at all) has to be made without knowing which requests will arrive in the future, and, for the sake of guaranteed QoS, *without* being able to rearrange the way existing connections are established. In such an case, a sensible approach is to try to allocate minimal TBW when satisfying *each* request for connection establishment, which is still an NP-hard problem due to possible BBW sharing among the new BP to be established and the existing BPs.

The above approach *may* lead to either the minimization of TBW needed to support a given number of requests, or the maximization of revenues for a given network capacity, or both (these two performance metrics will be described in more detail in Sec. VII). In fact, several schemes based on Integer Linear Programming **(ILP)** formulations have been proposed to optimize for *each* request in the on-line case. However, these schemes (as well as other schemes taking a similar "joint-optimization" approach) are bound to have a long request processing time, and thus impractical for on-line provisioning in a large network. On the other hand, simple and fast heuristics such as Active Path First **(APF)** [9], which selects an AP with the least amount of ABW first (without taking into consideration the BBW needed for the corresponding BP) and then selects a BP, each time using any well-known polynomial time shortest path algorithm, are known to yield less optimal results.

We observe that even if one can allocate minimal TBW for *each* request, one *cannot guarantee* the best overall performance in terms of minimization of TBW needed to support *a given number* of requests that have arrived over time, or maximization of revenues for a given network capacity. This is why schemes based on ILP formulations for the on-line case may not yield the best overall performance (unlike in the off-line case). In fact, one of the pleasantly surprising results of this paper is that our heuristic algorithm proposed in this paper for determining an AP and a BP can perform *better* than schemes based on ILP formulations (with the same input), and at the same time, can process requests several orders of magnitude faster. To our best knowledge, this is also the first work that shows a heuristic algorithm performs better than ILP-based schemes (at least for the problem under consideration, which is a non-intuitive result.

B. Comparison with Related Work

In order to determine whether or not a new BP can share BBW with an existing BP on a given link, and consequently, exactly how much additional BBW on the link needs be allocated, a controller (either in centralized or decentralized control implementation) needs to maintain *complete per-flow* information as in the Sharing with Complete Information **(SCI)** scheme (which uses an ILP formulation) [10] or *complete aggregated* information as in the Survivable Routing **(SR)** scheme (which uses the APF heuristic) [9]. In both cases, the amount of information to be maintained by a controller is (more or less) (E^2), where E is the number of links in a network.

Several other schemes have been proposed which require only (E) partial (and aggregated) information (but overestimate the additional BBW needed by the new BP). They include the Sharing with Partial Information **(SPI)** scheme in [10] and our Distributed Partial Information Management **(DPIM)** scheme in [11], both of which use an ILP formulation, which is time-consuming and non-scalable (for example, to process one connection establishment request in an 80-node network, it takes about 10-15 minutes on a low-end workstation [10]).

In order to reduce the computational overhead associated with the ILP based approaches, a primal-dual based approach and an approach based on the Shortest Pair of Path **(SPP)** algorithm [12], were also proposed for SPI and DPIM, respectively. However, like all other heuristics, including a modified SPP algorithm where the costs of using a link on an AP and an BP can be different [13], these approaches do not perform as well as their ILP-based counterparts in terms of minimizing the TBW needed in the on-line case. While heuristics are desirable in terms of their scalability, many researchers take the resulting degraded performance for granted as a tradeoff, and few has questioned whether any heuristic can perform better than its ILP-based counterpart or not.

The proposed ultra-fast and optimal heuristic algorithm to determine an AP and a BP is applicable with either complete or partial information. The salient feature that distinguishes our path determination algorithm from the methods used by SCI and DPIM, and also offers many advantages over them, is the use of the so-called *potential backup cost (PBC) function* derived mathematically from rigorous statistical analysis

of experimental data, in conjunction with APF. Such a heuristic, named APF with *Potential Backup Cost* **(APF-PBC)**, decomposes the joint-optimization problem into two simpler (and tractable) phases: selecting an optimal AP, then an optimal BP (using any well-known shortest path algorithms) just as APF. However, unlike APF, it assigns a cost of $w + \beta_e(w)$ to each link before AP is selected (in the first phase), where w is the ABW required by the connection, and $\beta_e(w) \leq w$ represents the potential backup cost (i.e., BBW) to be incurred on the corresponding BP yet-to-be-chosen (in the second phase). In this way, the APF-PBC heuristic can take into consideration the inter-phase correlation (i.e., the impact of selecting an AP on the amount of BBW sharing) just as an approach based on an ILP formulation. In short, the proposed APF-PBC method combines the best of the APF heuristic and ILP-based approach while avoiding their shortcomings.

Hereafter, we will refer to the proposed two representative schemes that apply the proposed path determination algorithm APF-PBC, and use complete and partial information, respectively, as SCI-P and DPIM-P, respectively, where "P" stands for PBC. We will compare their performance with that of SCI and DPIM, as well as NS and SPI.

The rest of the paper is organized as follows. Section II provides background information including the notations to be used throughout the paper, and reviews prior solutions including SCI, SR, DPIM, and SPI. Section III describes the main idea and motivation behind the proposed APF-PBC, and the SCI-P and DPIM-P schemes that apply the APF-PBC heuristic algorithm when using complete and partial information, respectively. Sections IV through VI describe in detail the Potential Backup Cost **(PBC)** function, including its mathematical derivation, approximation and simplification, respectively, based on the analysis of experimental data. Section VII presents the performance evaluation model used, followed by numerical results of the performance comparison. Section VIII concludes the paper.

II. Background

In this section, we first present the notation to be used throughout the paper, and then briefly describe the basic ideas of existing schemes. In the following, we will limit our discussion to protection against a single link failure. Note that the problem of protection against the failure of a single node other than the ingress or egress node can be transformed to that of protection against a single link. For example, one can first treat a network as a directed graph, then split each node into two halves, one for all incoming links to the node and the other for all outgoing links, and finally interconnect the two halves with an "added" directed link. The failure of this added directed link thus becomes equivalent to the failure of the node.

A. Notation

To facilitate our presentation, the following notations will be used

: The set of directed links in a network (the number of the links in the set is E).

w: The amount of bandwidth requested by a survivable connection. To satisfy the request, the amount of additional ABW that needs to be reserved on each link along a new AP is w, but the amount of additional BBW that needs to be reserved on each link along a new BP could be less due to BBW sharing.

$_e$: Total (i.e., aggregated) ABW on link *e* dedicated to the set of connections whose APs traverse link *e*.

$_e$: Total BBW allocated on link *e* to the set of connections whose BPs traverse link *e*.

R_e: Residue bandwidth of link e. $R_e = C_e - \ _e - \ _e$, where C_e is the capacity of link e,

: Total amount of bandwidth required by the set of connections whose APs traverse link *a* and whose BPs traverse link *b*. It is a fraction of and .

$(e) = \{ \ _e | b \in \ \}$: *Profile* of ABW on a given link e. This is a *vector* consisting of a list (or set) of $_e$ values, one for each link b. It specifies the amount of ABW on link e that is protected by every link (e.g., b_1, b_2 $b \in$) in the network.

$P \ _e = \max \ _e$: This is the maximum value over all the components in (e). It is also the *sufficient* amount of bandwidth that needs be reserved on any link in the network in order to protect against the failure of link e.

The following notations is useful only for the description of the proposed PBC function.

$\overline{P} \ _e$: This is the *average* value over all the components of a given ABW profile on link e.

$M = \max$: This is also equal to $\max_e P \ _e$.

Additional notations having only local significance (i.e., useful within a section) will be introduced when necessary. For a list of the acronyms used, see the end of the paper.

B. Prior Work

We now turn to previous solutions, which can be classified into two main categories, one using complete information (per flow or aggregated) and the other using partial (and aggregated) information.

1) Schemes Using Complete Information: We first review the SCI scheme in [10], where a controller maintains the set of connections whose APs traverse e and the set of connections whose BPs traverse e, in addition to $_e$ and R_e, for every link $e \in$. Based on such information, for every link a and link b (i.e., all possible combinations of a link pair) can be derived. Accordingly, if link b is to be used by the new BP, whose corresponding AP is already determined, the additional BBW (or backup cost) to be allocated on link b for the new BP, denoted by C , can be calculated as

$$C \ = \max\{ \max (\quad + w - \quad), 0\} \qquad (1)$$

(where , and should take their current value before any bandwidth is allocated to the new connection). In [10], an ILP formulation is described which can be used to determine a pair of link-disjoint paths with a minimal TBW for use as an AP and a BP, respectively.

The two schemes proposed in [9] also use complete information, but they differ from SCI mainly in that they use the APF heuristic instead of ILP formulation to determine an AP and a BP. More specifically, in these two schemes, each edge node maintains complete aggregate information, which is for every pair of links $a, b \in$. In the first scheme, called Survivable Routing **(SR)**, an AP with a minimal number of eligible links(i.e., links whose residual bandwidth R is larger than w) is found first using a shortest path algorithm. Then, the links used by the AP is removed (or assigned an positive infinite cost), and each remaining link b is assigned a cost equal to C given by Eq. 1. Thereafter, each link b with R C is removed, and a cheapest path found next is used as the BP. The second scheme, which is a variation of SR, is called Successive Survivable Routing **(SSR)**. The main difference between SR and SSR is that, in the latter, some existing BPs are allowed to change, not only in the way they are routed but also the amount of additional BBW reserved for them, after the matrix is updated as a result of setting up a new connection. Such changes may in turn trigger changes to other existing BPs until an equilibrium state is reached. Although SSR can achieve a better BBW sharing than SR, the iterative process involving changes to the existing BPs introduces a high signaling and control overhead, especially under distributed control. This is why in this paper, we will only consider schemes that *do not* require existing BPs to change.

There are several other schemes that use more or less the same order of magnitude of information as (E^2) with no better performance than SCI [6,14–16]. Later, we will use SCI as a representative scheme when comparing performance with that of the proposed SCI-P scheme.

2) Schemes Using Partial Information: We now review two schemes using partial information: one based on SPI [10], and the other based on DPIM [11].

In SPI, only the values of $_e$ and $_e$ (in addition to R_e) for every link e are maintained by each edge node. As a result, the amount of information to be maintained at each node is (E). However, the flip side is that, with such partial information, the additional BBW to be allocated on link b for the new BP, assuming its corresponding AP is already determined, can only be estimated as

$$C = \max\{\ \max\ (\quad + w - \quad), 0\} \qquad (2)$$

Since , Eq. 2 results in an over-estimation of the backup cost when compared to Eq. 1. Accordingly, when using an ILP formulation similar to that used in SCI, SPI obtains a less optimal pair of AP and BP than SCI.

In the DPIM scheme described in [11], every node (i.e., edge or core) maintains some ((E)) information for each local outgoing link e, which includes (e), $_e$, R_e, P $_e$. The last scalar is for convenience only as it can be derived from the first vector. In addition, an edge (ingress) node will also maintain three scalars, $_e$, R_e, and P $_e$ for each *remote* (i.e., non-local) link e. Note that, this still limits the amount of information to be maintained at each node to (E). How the local information is updated and remote/non-local information is exchanged, and how connections are established/released (as well as how bandwidth is allocated and deallocated along APs and BPs) through distributed control and signaling have been discussed in [11].

The DPIM scheme uses an ILP formulation similar to that used by SPI with several improvements. For example, in DPIM, the estimated backup cost can be obtained as follows:

$$C = \max\{\ \max\ \min\{(P \quad + w - \quad), w\}, 0\} \qquad (3)$$

Such an estimation is more accurate than that given by Eq. 2 since $\leq P \leq$. Due to this and other improvements, DPIM has been shown to perform much better than SPI. Later, we will compare the performance of DPIM (as well as SPI) with the proposed DPIM-P scheme.

III. Overview of Proposed APF-PBC and the Schemes based on APF-PBC

As discussed in the Sec. I, a major challenge in achieving efficient shared path protection in an on-line case is that, while a fixed amount of ABW (i.e., w units) is to be allocated on each link used by an AP, the amount of BBW to be allocated on each link along a BP depends on many factors including which links are used by the corresponding AP. This is why the APF heuristic (as well as other similar heuristics) is not ideal as it does not consider (nor cares about) the *potential* cost along the BP yet to be chosen when selecting the AP. In addition, the SPP algorithm such as the one in [12], which is suitable for the NS scheme, does not take possible BBW sharing into consideration either in that it essentially assumes that the cost of each link on a BP is also equal to w.

On the other hand, on-line "joint-optimization" schemes based on ILP formulations guarantee minimal allocation of TBW for each request by jointly optimize the selection of both AP and BP. However, they do not guarantee an optimal result for all requests that arrive over time. In addition, their computational complexity is too high to be scalable.

The main motivation for our work on APF-PBC is to overcome the disadvantages of the APF and ILP based schemes, while trying to combine the best of the two. More specifically, the path determination algorithms based on APF-PBC can run as fast as those based on APF. Yet, they can also take into consideration the BBW to be allocated when determining the AP as in ILP-based schemes. This is accomplished by assigning

an additional *potential cost* $\beta_e(w) \leq w$ to each link, based on which the AP is selected using a shortest-path algorithm. This $\beta_e(w)$ is derived mathematically based on the statistical analysis of experimental data (just as many theorems in quantum physics were), as to be described in the following sections. Note that, APF may be considered as a special case of APF-PBC where the value of $\beta_e(w)$ is always set to zero.

The main idea of a scheme based on APF-PBC, such as the proposed SCI-P and DPIM-P, is as follows. Given a network with some existing connections, any link e whose R_e w will be removed initially as such a link cannot be used by the new AP. Each remaining link a will be assigned a cost which is equal to $w + \beta\ (w)$. A cheapest path is then found for use as the AP (using any well-know shortest path algorithm). As $\beta\ (w)$ does not need to use more information than that used by DPIM, having more (e.g., complete) information will not change how the AP is selected. In other words, given the same starting point, both DPIM-P and SCI-P will select the same AP.

After the AP is selected, the links along it are then removed, but the other links removed initially should now be put back as they may have enough residual bandwidth for the corresponding BP yet to be chosen.

To select the corresponding BP, each link (other than those along the AP) will first be assigned a cost given by Eq. 1 if SCI-P is used, or Eq. 3 if DPIM-P is used. A cheapest path algorithm is then used to find the BP. Clearly, having complete information as in SCI-P leads to more accurate estimation of the additional BBW (or backup cost) for each link, and accordingly a better BP, than having partial information as in DPIM-P.

Note that APF-PBC can also be applied to modify the SPI scheme into what one may call SPI-P. Our results, though not shown in this paper, indicate that SPI-P can outperform SPI (which is based on ILP formulation), just as SCI-P and DPIM-P can outperform SCI and DPIM, respectively. In addition, our results also indicate that SCI-P and DPIM-P can outperform their counterparts that use naive heuristics such as APF and SPP (including SR).

IV. Potential Backup Cost - Derivation

In this section, we will describe how the PBC function $\beta\ (w)$ is derived based on the statistical analysis of experimental data. Even though the experimental data presented here is based on the traces collected from the simulation runs in which SCI is applied to the 15-node network shown in Fig. 1 assuming infinite link capacity (see Sec. VII for additional assumptions made in those experiments), we have found that the statistical characteristics of the data do not change significantly with topologies, demand patterns and shared path protection schemes used. More importantly, the derived PBC function works well in all the cases we have studied.

Note that a major challenge is that when trying to determine PBC function for link a, the ingress node responsible for path determination does not even know which link has been or will

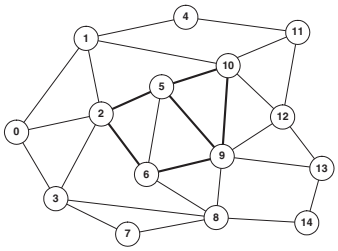

Fig. 1. A 15-node network

be used by BPs to protect against the failure of link a. If link b were to be used by a new BP (whose corresponding AP is going to use link a), and the amount of the BBW already allocated to link b (which is) were known to be x, the additional BBW needed on link b (or backup cost) is

$$C \ = \max\{s + w - x, 0\} \tag{4}$$

where $s =$, which should be no greater than x.

A. Step One

The first step towards "guessing" the potential backup cost is to assume that s is known for the time being (this assumption will be relaxed later), and proceed to calculate an "expected" backup cost using a *weighted average* over all possible values of x (i.e., of all $b \in$). More specifically, we will treat —, for an arbitrary link b, (where $M = \max$), as a random variable between 0 and 1 whose Cumulative Distribution Function **(CDF)** ( ), obtained experimentally, is illustrated in Fig. 2. As can be seen, (= —) can be approximated by a normal distribution function $N(\ ,\ )$ (shown in dashed curve), where is the mean value, and is its standard deviation. Let its corresponding probability density function be $f(\ )$.

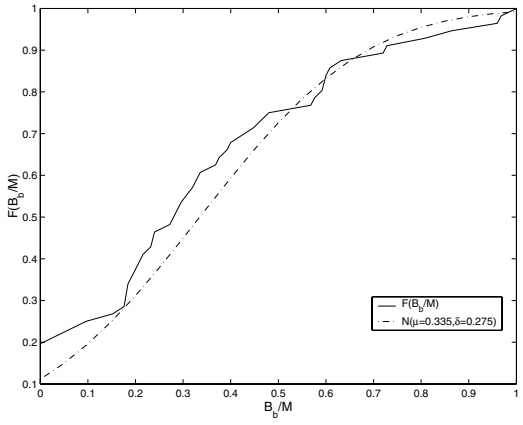

Fig. 2. Distribution of —

Based on Eq. 4 and the definition of a density function, the expected backup cost of using any link b by the BP to back up

an AP using link a, provided that the value of is equal to s, becomes

$$(w, s, M) = \ ^{(\frac{w}{} 1)}(s + w - x)\frac{(-)}{(\quad)}d(-) \quad (5)$$

where $\min(\frac{+w}{}, 1)$ is used as the upper bound for — due to the fact that $C = 0$ when $x \geq s + w$. In addition, $P(M \geq \geq s)$, which is equal to $(1) - (-)$, denotes the probability that a variable (representing the BBW on a link) is between s and M, and thus needs to be included since only those links whose $M \geq \geq s$ are valid.

Note that (w, s, M) is in fact independent of any particular link. The solid lines in Fig. 3 shows a typical graph for (w, s, M), where the horizontal and vertical axis are normalized to M and w, respectively. The value of (w, s, M) for a given w, s and M is obtained by using the adaptive Lobatto quadrature [17] method to evaluate the integral function in Eq. 5, and those values form the curves shown in Fig. 3. In Fig. 3, M is chosen to be 125 units, which is a typical value after a large number (e.g. 500) of connections have been established in each of the several experiments conducted.

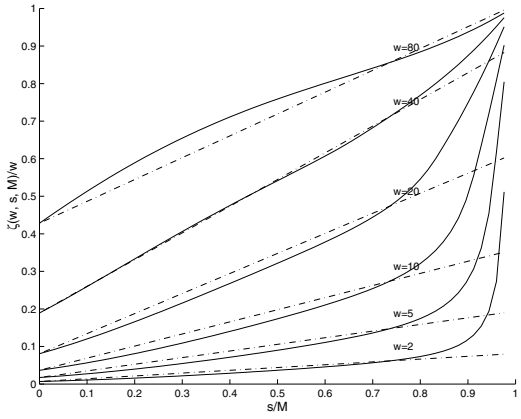

Fig. 3. Graph of ( )

B. Step Two

The second step towards guessing the potential backup cost is to relax the assumption that s is known by calculating a weighted average value of (w, s, M) over all possible values of s on link a. More specifically, let us treat —— as a random variable between 0 and 1 with a CDF $G(\ )$. Fig. 4 shows $G(= ——)$, obtained experimentally, for five randomly selected links in the 15-node network (specifically, links $0 \to 1, 2 \to 6, 8 \to 3, 9 \to 5, 9 \to 12$).

From Fig. 4, it seems that the CDF $G(\ )$ may be approximated by an exponential function (which matches our intuition/expectation), but such an approximation is not necessary to derive the potential cost function and hence will not be made.

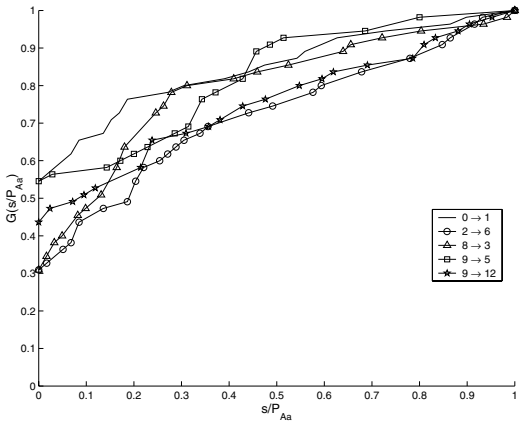

Fig. 4. Cumulative distribution function of ——

Let the corresponding density function be ( ) (whose actual distribution or its approximation is not important). Based on the above discussion, the potential backup cost is:

$$\beta\ (w) = \ ^{1} \ (w, s, M)\ (\frac{s}{P})d(\frac{s}{P}) \quad (6)$$

V. Potential Backup Cost - Approximation

In this section, we will simplify Eq. 6 with several reasonable approximations.

First, from the dashed lines in Fig. 3, it seems that each curve (or rather the points on each curve), which corresponds to a given value of w can be approximated (or fit) by a line of the form $= c_1 + c_2$, where $= \frac{(w \quad)}{w}$, $= —$, and the values of c_1 and c_2 depend on the given w and M (as to be discussed later). Although such an line-fitting approximation is good only up to the point where reaches about 0.75, it will not have much affect on the value of $\beta\ (w)$ given by Eq. 6 above. This is because as can be seen from Fig. 4, the probability that —— is larger than 0.75 is already very small. Moreover, since $P \leq M$, the probability that — is larger than 0.75 is even smaller, and in fact, almost negligible.

Furthermore, as can be seen from Fig. 3, we may fit the group of curves with lines as follows. First, the slopes of these fitting lines (shown in dashed lines), to be denoted by (w, M) (instead of c_1 to more clearly indicate their dependency on w and M), may be obtained as — at a reasonable value of (or —). Fig. 5 shows the values of (w, M) (obtained when $— = 0.75$) as a function of w normalized with M.

It is clear from Fig. 5 that (w, M) can be simply approximated by an exponential function $\eta(1 - e^{-\frac{w}{}})$ (shown in dashed curve), where $\eta = 1.2$ and $= 0.2$.

In addition, as can be seen from Fig. 3, $c_2 \approx 0.2$ if $w = 40$, and $c_2 \approx 0.1$ if $w = 20$ and so on. In other words, $c_2 \approx 0.05w$. But since in all these cases, $M = 125$, we may set $c_2 = \frac{w}{}$, where ≈ 0.6 (we use variable here to make the solution

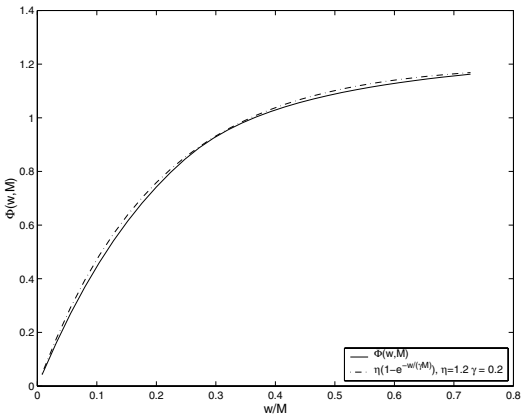

Fig. 5. Graph of ( )

more general so that we can further study its impact in the next subsection).

In short, we have the following approximation for (w, s, M):

$$(w, s, M) \approx \quad (w, M) \quad \frac{ws}{M} + \quad \frac{w^2}{M} \tag{7}$$

(where the right hand and left hand sides of Eq. 7, normalized with w, are plotted using the curves and lines, respectively, in Fig. 3).

We now plug in this approximation for (w, s, M) into Eq. 6. Since we have:

$$\int_s^1 (\frac{s}{P})d(\frac{s}{P}) = \int_s^\infty (\frac{s}{P})d(\frac{s}{P}) = \overline{P}$$

(this is because $\frac{}{} \leq 1$ and the expected value of s on link a, is, by definition, $\overline{P}$), and in addition

$$\int_1^1 (\frac{s}{P})d(\frac{s}{P}) = 1$$

,

we have the following approximation for $\beta\ (w)$:

$$\beta\ (w) \approx \frac{(w, M)\ \ w\ \ \overline{P}}{M} + \frac{w^2}{M} \tag{8}$$

VI. Potential Back Cost - Simplification

While one may use the potential cost function given in Eq. 8, it can be further simplified without affecting its usefulness or the performance of a scheme that adopts it.

First, we note that if Eq. 8 is used, each edge node needs to maintain $\overline{P}$ which is not needed in the DPIM. Every time an edge node satisfies a connection establishment (or release) request whose AP uses link a, it can simply increase (or decrease) $\overline{P}$ by $\frac{w}{}$, where is the number of links along the chosen BP, and multicast the updated value to all other edge nodes.

To avoid the need to maintain $\overline{P}$ at the edge nodes, one may replace $\overline{P}$ with P , where $= \frac{a}{a}$ is almost a constant as can be seen from its CDF shown in Figure 6, which is obtained experimentally. From the figure, it is clear that the CDF can be fit by a curve with a normal distribution function $N((\ , \ )$ (shown in dashed curve), whose mean is around 0.18 and whose standard deviation is only 0.043.

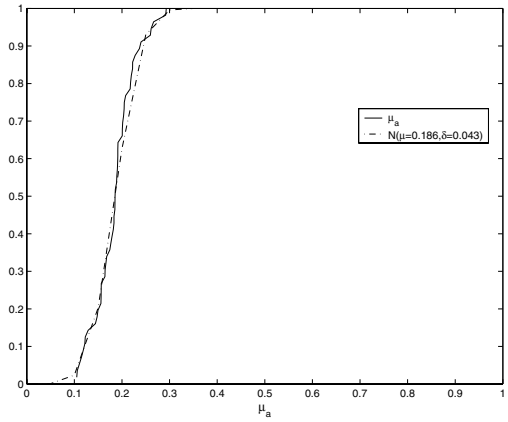

Fig. 6. Distribution of

To further simplify the PBC function, one may omit the second term in Eq. 8, that is, $\frac{w^2}{}$, as this term is not only small, but most importantly, independent of all links as well (as to be shown, this omission will not affect the usefulness of the PBC function).

Finally, we note that for any request with $w \quad 0$, the value of (w, M) is larger than 0 but no greater than 1.2 (see Fig. 5). In particular, it is independent of the link for which a PBC is being estimated.

The results shown in Fig 7, for example, indicate that any reasonable value of (w, M) between 0 and 1 may be used to estimate the PBC without any significant impact on the performance of the APF-PBC heuristic. In other words, we may replace (w, M) with a constant c to arrive at Eq. 9.

$$\beta\ (w) = c\ \ \frac{w\ \ P}{M} \tag{9}$$

As can be seen from Fig 7 which shows the bandwidth saving ratio of DPIM-P over NS in the 15-node network (detail in Sec. VII), the value of c does not matter much as long as $c \leq 1$. The figure also shows that with or without the second term in Eq. 8, (i.e., with $= 0.6$ or $= 0$), the performance of DPIM-P is pretty much the same.

Note that such a PBC function, though derived mathematically, is quite intuitive as the larger the w, the higher the PBC. In addition, for a given w, the larger the P , the more likely that a larger amount of additional BBW needs be allocated on BP (and hence a larger PBC).

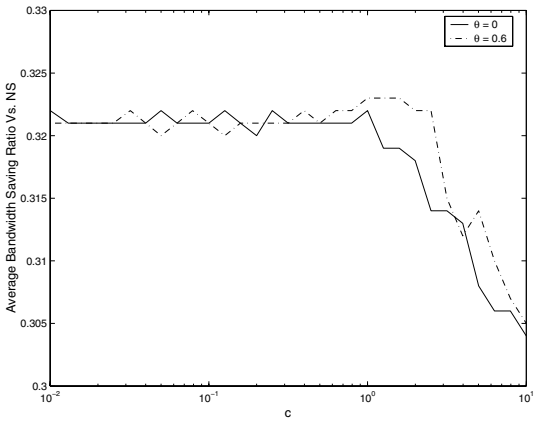

Fig. 7. The effect of constants and on the performance of APF-PBC

VII. Performance Evaluation

We are primarily interested in comparing the performance of the schemes based on APF-PBC, namely SCI-P, and DPIM-P, with their counterparts based on ILP, namely SCI and DPIM, Processing and signaling overheads are ignored in this quantitative comparison study. When simulating APF-PBC based schemes, c in Eq. 9 is set to 0.5. Note that many other potential functions derived from intuition may be used. However, we have tested some of them and have found that they do not perform as well as that given by Eq. 9.

In the rest of the section, we describe the network topology assumed, traffic types considered, and performance metrics used before presenting the results.

A. Network Topology

To facilitate a fair comparison between various schemes, we consider the topology shown in Fig. 1, which is the same as that used in [10] and has 15 nodes and 28 bi-directed edges (for a total of 56 links). The capacity of each link is assumed to be either infinite or limited as to be discussed in the next subsection. Another large network called USnet (with 46 nodes and 76 bi-directed edges [18]) is also considered, which consistent performance results have been obtained.

B. Traffic Types

We consider two types of traffic, one in which an established connection lasts forever (i.e., incremental traffic) as in [9, 10], and the other in which it may terminate after a certain duration (i.e., dynamic traffic) as in [11].

In both cases, the ingress and egress of a connection establishment request is evenly distributed among all nodes, and requests arrive in an on-line fashion. For the case with incremental traffic, the bandwidth required by the connections is uniformly distributed between 1 and 10 units as in [10]. Note that, any request arrival process may be assumed.

For the case with dynamic traffic, the bandwidth required varies from 1, 2, 3, 4, 6 and 12 units with probability being 20%, 10%, 30%, 10%, 10%, 20%, respectively. In addition, requests are assumed to arrive according to a Poisson process, and the connection duration has a Pareto distribution. This is just an attempt to model realistic traffic (which may be self-similar and whose bandwidth requirements range from OC-1 or 52Mbps to OC-12 or 622Mbps). Other possibilities, including uniformly distributed bandwidth requirements and exponentially distributed connection durations, have also been examined, and we have found that they have no significant impact on the performance of various schemes studied in this paper.

C. Performance Metrics

The following two performance metrics are used, one for each traffic type considered.

1) Bandwidth Saving (Ratio): To obtain this metric, it is assumed that the capacity of each link is infinite (and hence all requests will be satisfied), and the traffic is incremental. After an appreciable number of requests have been satisfied, TBW consumed (i.e., sum of ABW and BBW on APs and BPs, respectively) for each of the schemes is evaluated and consequently, bandwidth saving, in terms of TBW consumption ratio over the NS scheme, is determined as similarly done in [10]. Note that, for a given request, the BBW required need not be less than the ABW required in NS. Hence, even if an ideal scheme that achieves maximum BBW sharing is used, the bandwidth saving ratio will be upper-bounded by 50% (achievable only if no BBW is needed at all).

2) Total Earning (Ratio): The bandwidth saving measure may not mean much since in a practical case, all links have a finite capacity and thus not all requests can be satisfied.

Accordingly, in this set of experiments (simulation), we assume that each link has a finite capacity and dynamic traffic is considered. For example, in the Fig. 1 above, each dark (bold) link (consisting of two unidirectional links) is assumed to have a capacity of 192 units in each direction (to model an OC-192 link), and each of the other links has a capacity of 48 units in each direction (to model an OC-48 link). As a result, some requests will be rejected under a heavy traffic load.

The total number of rejected connection establishment requests (after an initial set of requests are satisfied) using each scheme has been used as a performance measure (e.g., in [10]). However, comparison between different schemes based on such a measure (or equivalently blocking probability) may not be fair as it does not differentiate one request from another [11].

This motivates us to use the total earning (or revenue) as a metric as in [11] based on a scheme-independent *Earning Rate* matrix whose entry at $(i,\ )$ represents earnings per bandwidth unit and time unit by a connection from ingress node i to egress node . The earnings from a connection from i to is thus the product of the earning rate, requested units of bandwidth, and the connection duration.

In this study, for lack of a better alternative, the earning rate is based on the cost of using the cheapest (or shortest) pair of AP and BP in the network from i to (assuming there were infinite capacity in the network), and hence is independent of the current load in the network [1]. An important and desirable consequence of using the assumed earning rate (along with the earnings from a connection) is that it tends to discourage an algorithm that tries to maximize earnings from choosing an unnecessarily expensive (or long) path to establish the connection. Because choosing an expensive/long path under such a model may prevent other (future) connections from being established and thus resulting in lost revenues.

We compare the total earnings of each scheme and in particular, the improvement ratio over the NS scheme.

D. Simulation Results

Fig. 8 shows the total bandwidth consumed after satisfying 200 connection establishment requests (or demands) in the 15-node network from 10 experiments (simulation runs). It can be seen from the figure that, given the same (E) partial information, DPIM-P consistently outperforms DPIM (which in turn consistently outperforms SPI). In addition, SCI-P also outperforms SCI slightly. The difference between SCI-P and DPIM-P may not be big enough to warrant the additional overhead involved in maintaining (E^2) complete information as required by SCI-P.

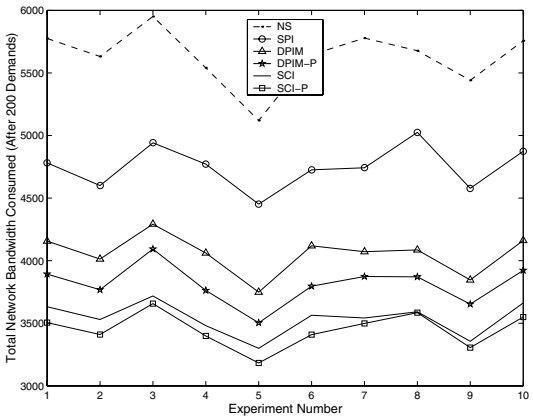

Fig. 8. Bandwidth consumed (from 10 experiments)

Table I shows the average bandwidth saving ratio (vs. NS) (over the 10 experiments) in the 15-node and 46-node networks. For each network, the first row is for schemes using ILP, and the second is for their corresponding schemes using APF-PBC.

In order to evaluate the performance of various schemes in networks with limited link capacity and dynamic traffic, another 10 experiments have been conducted in which each network is loaded with heavy (dynamic) traffic (under a light load, the differences between various schemes are insignificant).

[1] If the earning rate is load-dependent, it will become scheme-dependent also

TABLE I
AVERAGE BANDWIDTH SAVING RATIO

15-node network	
37.2%(SCI)	28.0%(DPIM)
38.7%(SCI-P)	32.1%(DPIM-P)
46-node network	
34.0%(SCI)	25.9%(DPIM)
34.7%(SCI-P)	28.8%(DPIM-P)

Fig. 9 shows the total earnings after processing 500 demands (not all of them are satisfied).

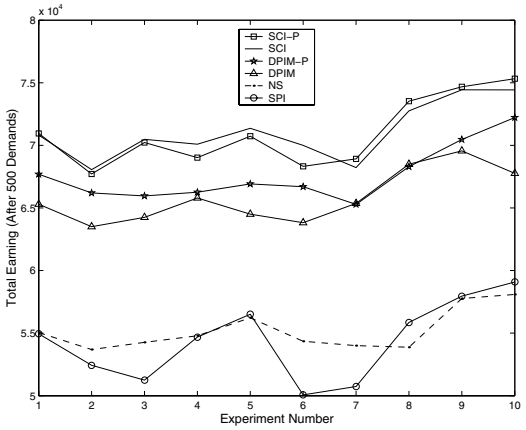

Fig. 9. Total earning after 500 demands (from 10 experiments)

An interesting observation is that while the SCI and DPIM based schemes perform well, SPI performed poorly and in fact, worse than NS in some experiments. This is because C given by Eq. 2 may be larger than w (which is needed by NS), that is, SPI may overly estimate the BBW needed, and allocate excessive BBW. This is especially problematic in a network with limited capacity as many connection requests that could have been satisfied even in NS will now be rejected.

Table II shows the average total earning ratio (vs. NS) for SCI and DPIM based schemes. These results also show that schemes based on APF-PBC outperform their counterparts based on ILP.

In short, our performance comparison study for both incremental and dynamic traffic patterns reviews that:

SCI-P > SCI > DPIM-P > DPIM >> SPI/NS

(where the sign " " means either "uses less TBW than" or "generates more earning than").

Note that, while the performance improvement of the APF-PBC based schemes over their ILP-based counterparts may not be significant, the fact that they do improve the performance and at the same time, reduce the computational time by several orders of magnitude makes them far more superior than their ILP based counterparts or any other existing heuristics for shared path protection.

TABLE II
AVERAGE TOTAL EARNING RATIO

15-node network	
28.7%(SCI)	19.3%(DPIM)
29.6%(SCI-P)	23.1%(DPIM-P)
46-node network	
45.6%(SCI)	32.0%(DPIM)
45.7%(SCI-P)	35.8%(DPIM-P)

VIII. CONCLUSION

In this paper, we have proposed a novel heuristic called *APF-PBC* to determine a pair of active and backup paths for each on-line request to establish a connection using shared path protection. Its basic idea is to attach a potential backup cost (PBC) to each link so that one may select an active path first (APF), while still taking into consideration the impact of bandwidth sharing along a yet-to-be-chosen backup path.

We have mathematically derived a potential backup cost (PBC) function based on the statistical analysis of experimental data, which can be used to minimize the total bandwidth or maximize the total earnings. Although the APF-PBC heuristic can be used with only partial information under distributed control, it can also be applied when complete information is available and/or under centralized control. In fact, its basic idea may also be extended to other joint optimization problems.

One of the interesting results obtained from this study is that the proposed APF-PBC, of which APF is a special case, can not only run much faster than ILP based schemes, but also *outperform* them. Our performance evaluation results have revealed that this is true in all the on-line cases we have considered in this paper, as long as APF-PBC and ILP have the same information (either complete or partial), traffic load (either incremental or dynamic) and constraints (such as no rearrangements of existing connections). We have explained the reason for this ground-breaking and pleasantly surprising result, which should encourage people to look for more practical and efficient heuristics when tackling similar on-line optimization problems instead of simply resorting to ILP. Finally, we will describe how to extend the proposed APF-PBC heuristic to the case where a pair of paths that are Shared Risk Link Group **(SRLG)** [19, 20] disjoint need to be established in a future paper.

ACRONYMS

ABW	Active Bandwidth
AP	Active Path
APF	Active Path First
APF-PBC	APF with *Potential Backup Cost*
BBW	Backup Bandwidth
BP	Backup Path
CDF	Cumulative Distribution Function
DPIM	Distributed Partial Information Management
DPIM-P	DPIM with APF-PBC
ILP	Integer Linear Programming
NS	No Sharing
PBC	Potential Backup Cost
SCI	Sharing with Complete Information
SCI-P	SCI with APF-PBC
SPI	Sharing with Partial Information
SPP	Shortest Pair of Path
SR	Survivable Routing
TBW	Total Bandwidth

REFERENCES

[1] C.Assi, A. Shami, M. A. Ali, and et al., "Optical networking and real-time provisioning: An integrated vision for the next generation internet," in *IEEE Network, Vol. 15, No. 4*, Jul.-Aug. 2001, pp. 36–45.

[2] T.M. Chen and T.H. Oh, "Reliable services in MPLS," in *IEEE Communications Magazine*, Dec. 1999, pp. 58–62.

[3] A. Benerjee, J. Drake, J. Lang, and B. Turner et al., "Generalized multiprotocol label switching: An overview of signaling enhancements and recovery techniques," in *IEEE Communications Magazine, Vol. 39, No. 7*, Jul. 2001, pp. 144–151.

[4] D.O.Awduche, L. Berger, and et al, "RSVP-TE: Extensions to RSVP for LSP tunnels," in *Draft-ietf-mpls-rsvp-lsp-tunnel-07*, Aug. 2000.

[5] Der-Hwa Gan, Ping Pan, and et al., "A method for MPLS LSP fast-reroute using RSVP detours," in *Draft-gan-fast-reroute-00*, Apr. 2001.

[6] B. Doshi and et al., "Optical network design and restoration," *Bell Labs Technical Journal*, pp. 58–84, Jan.-Mar. 1999.

[7] Yijun Xiong and Lorne G. Mason, "Restoration strategies and spare capacity requirements in self-healing ATM networks," in *IEEE/ACM Trans. on Networking, Vol. 7, No. 1*, 1999, pp. 98–110.

[8] Ramu Ramamurthy and et al., "Capacity performance of dynamic provisioning in optical networks," *Journal of Lightwave Technology*, vol. 19, no. 1, pp. 40–48, 2001.

[9] Yu Liu, D. Tipper, and P. Siripongwutikorn, "Approximating optimal spare capacity allocation by successive survivable routing," in *INFOCOM'01*, 2001, pp. 699–708.

[10] Murali Kodialam and T V. Lakshman, "Dynamic routing of bandwidth guaranteed tunnels with restoration," in *INFOCOM'00*, 2000, pp. 902–911.

[11] Chunming Qiao and Dahai Xu, "Distributed partial information management (DPIM) schemes for survivable networks - part I," in *INFOCOM'02*, Jun. 2002, pp. 302–311.

[12] J.W. Suurballe and R.E. Tarjan, "A quick method for finding shortest pairs of disjoint paths," *Networks*, vol. 14, pp. 325–336, 1984.

[13] C. Li, S. T. McCormick, and D. Simchi-Levi, "Finding disjoint paths with different path costs: Complexity and algorithms," in *Networks, Vol. 22.*, 1992, pp. 653–667.

[14] C. Dovrolis and P. Ramanathan, "Resource aggregation for fault tolerance in integrated service networks," in *ACM Computer Communication Review, Vol. 28, No. 2*, 1998, pp. 39–53.

[15] Ramu Ramamurthy, Sudipta Sengupta, and Sid Chaudhuri, "Comparison of centralized and distributed provisioning of lightpaths in optical networks," in *OFC'01*, 2001, pp. MH4–1.

[16] Ching-Fong Su and Xun Su, "An online distributed protection algorithm in WDM networks," in *ICC'01*, 2001.

[17] W. Gander and W. Gautschi, "Adaptive quadrature - revisited," in *BIT, Vol. 40, This document is also available at http://www.inf.ethz.ch/personal/gander*, 2000, pp. 84–101.

[18] S. Baroni, P. Bayvel, and R.J.Gibbens, "On the number of wavelength in arbitrarily-connected wavelength-routed optical networks," in *University of Cambridge, Statistical Laboratory Research Report 1998-7, http://www.statslab.cam.ac.uk/reports/1998/1998-7.pdf*, 1998.

[19] J. Luciani et al., "IP over optical networks a framework," in *Internet draft, work in progress*, Mar. 2001.

[20] D. Papadimitriou et al., "Inference of shared risk link groups," in *Internet draft, work in progress*, Nov. 2001.