Multi-Flow Optimization Model for Design of a Shared Backup Path Protected Network

Brody Todd and John Doucette
Dept. of Mechanical Engineering, University of Alberta, Edmonton, AB, Canada T6G 2G8
and TRLabs, 7th Floor ECERF, 9107-116 Street NW, Edmonton, AB, Canada T6G 2V4
*corresponding author

Abstract: Designing optimal shared backup path protected networks is a difficult and time-consuming task, and considerable research has been done to develop near optimal heuristics and algorithms that will solve the SBPP model without extensive computing power, but by definition, such methods are sub-optimal. This paper introduces a slight modification to the SBPP problem that allows it to be optimally solved using conventional ILP techniques. By allowing working and backup paths to follow multiple routes, the new SBPP model eliminates the numerous 1/0 variables in the conventional model. The fundamental characteristics of SBPP remain intact, with the problem altering only slightly but it allows ILP solvers to find an optimal solution in a time measured in seconds to minutes compared to the days or longer needed for conventional models.

Keywords: network restoration and protection, shared backup path protection, network design, network optimization, network reliability, network availability

I. INTRODUCTION

As our dependence on communication networks continues to grow, their ability to continue operating despite equipment failure becomes more and more important. Already, government, business, and consumers rely on and expect high availability in their communication system [1]. As increasingly critical applications are developed in many industries (e.g., telemedicine), availability becomes an even more paramount consideration. While effort has been made in reducing their occurrences, disruptions caused by fibre cuts and other failures continue to occur at high rates [2]-[3], from causes that range from accidents, forces of nature, and even intentional activities. One of the goals addressed in the telecommunication network design literature is to optimize a network’s capacity configuration so as to provide restoration to traffic carried by a failed link, and many network design models have been developed to accomplish that.

While there are many adaptations and variations of each network survivability design approach, they primarily fall under three categories: survivable rings, end-to-end path protection, and localized span restoration. Each method comes with some unique advantages and disadvantages as measured in recovery time, cost, complexity, dual-failure restorability, availability, etc. [4]-[9]. The work in this paper focuses on shared backup path protection (SBPP), an end-to-end path-based protection method, and develops a novel approach to designing SBPP type networks [10]-[13]. Strictly speaking, this new model is based on SBPP, however, the results of the application of the model are distinct enough that it could be a separate survivability approach as well.

II. BACKGROUND

There has been a significant amount of research done in different areas of SBPP, as it is a popular network restoration scheme [14]. SBPP can be thought of as a shared form of 1+1 automatic protection switching (1+1 APS) [4], [9]. In a 1+1 APS network each working path has its own dedicated preconfigured protection path. In some configurations data is sent simultaneously across both paths, such that a tail end transfer is all that is needed to restore communication when a link fails on the working path, virtually eliminating data loss. This method of network protection, though providing good availability and quick restoration times, is highly inefficient, requiring at least double the total capacity needed for the working paths. SBPP takes the basic idea of 1+1 APS protection, and relaxes the requirements of dedicated backup capacity, allowing backup paths to share link capacity with other backup paths so long as their corresponding working paths are disjoint. SBPP is also alternatively known as shared path protection (SPP) or failure independent path protection (FIPP).

The conventional SBPP design problem is often approached as an ILP model [12], [15], although because of the extreme difficulty in solving the SBPP ILP model, algorithmic design approaches have also been developed [16]-[19]. The work herein develops a new ILP optimization model for solving the SBPP design problem that is significantly faster to solve than the conventional ILP models.

A. Conventional SBPP ILP Design Model

Here, we will define the SBPP ILP design model from [6], [12], [15] as the traditional one, while noting, however, that other variations do exist. This model follows the same general arc-path approach first developed in [7], where the primary decisions are an assignment of flows onto a set of eligible backup paths. In this model $D$ represents the set of all demands and $d$ is the integer magnitude of demand $r$, while $S$
is the set of all spans in the network, $s_i$ is the spare capacity allocated to a span $i$, and $w_i$ is the working capacity on that span. Predefined sets of eligible routes, $R_j$, for each demand $r_j$ are available for placing either working or backup paths. A subset of these routes, $R'_r$, is the set of eligible routes that cross span $j$, and a subset of spans $S'_j$ is the set of spans that route $q$ crosses. The parameter $c_{ij}$ is the “cost” of each unit of working or spare capacity on span $i$, and in the present work is proportional to the length of the span in the topology drawn. The decision variables used in the model are $x^r_i \in \{0,1\}$, which represents whether or not route $b$ is used as a backup route for demand $r$, $y^r_i \in \{0,1\}$, which represents whether or not route $p$ is used as a working route for demand $r$, and $z^{r,b}_i \in \{0,1\}$, which represents the multiplication of $x^r$ and $y^r$. Note that the latter is not a true multiplication of two other variables, as that would introduce a non-linearity in the model. Rather, as described more fully below, an interaction of several constraint equations effectively sets the value of $z^{r,b}$ to a value equivalent to that multiplication. To be clear, the constraints are as follows:

$$
\sum_{v \in L} y^r_v = 1 \quad \forall r \in D
$$

(2)

$$
\sum_{v \in L} y^r_v \cdot d' = w_i \quad \forall i \in S
$$

(3)

$$
\sum_{v \in L} x^r_v = 1 \quad \forall r \in D
$$

(4)

$$
y^r + x^r \leq 1 \quad \forall r \in D, \forall p \in R_r
$$

(5)

$$
y^r + x^r \leq z^{r,b} + 1 \quad \forall b \in R_r
$$

(6)

$$
\sum_{v \in L} x^r_v \geq y^r_v \quad \forall i \in S, \forall r \in D
$$

(7)

$$
\sum_{v \in L} z^{r,b}_v \cdot d' \leq s, \quad \forall i \in S, \forall j \in S | i \neq j
$$

(8)

The constraints in (2) select a single working route for each demand and (3) places sufficient working capacity along that route to accommodate them. (4) selects a single backup route for each demand and (5) ensures that a route is not assigned as both a working route and a backup route for a given demand. (6) is a constraint set that relates $x^r$, $y^r$, and $z^{r,b}$ together in a linear manner while assigning $z^{r,b}$ a value that is equivalent to $x^r \times y^r$. Because both $x^r$ and $y^r$ are 0/1 variables, (6) turns this multiplication into an iff relationship (see [15] for more information). (7) ensures disjointedness of a demand’s working and backup routes that makes (5) strictly redundant but we include both anyway. The use of both more directly limits the feasible solution space and generally improves runtimes. (8) assigns sufficient spare capacity to accommodate the largest set of concurrent backup routes for any span failure (the forcer set). To make this a spare capacity assignment (SCA) problem, a pre-processing step can assign values to the $y^r$ variables (say according to shortest path routing of working routes), which then become input parameters.

This traditional model, however, is quite difficult and time-consuming to solve for any medium to large networks, even with the SCA variant when provided with a very small eligible routes set (say, 5 per demand). In fact, some mid-sized networks we tested took days or weeks to solve with the SCA variant, and the JCA variant wouldn’t solve at all for any but the very smallest of our test networks, even when eligible working routes were limited to only 2 per demand. While those observations substantiate the need for algorithmic or improved ILP approaches, we thought it prudent (and fair) to provide that conventional model with all improvements possible so that the reference data to which we compare our new design model are the best attainable.

### B. Improved Conventional SBPP ILP Design Model

What follows is a variation of the conventional model above, which has been modified as much as possible while still respecting the spirit of the original to such an extent that it can still be considered a benchmark model. We make these changes so that we can provide the benchmark to which we compare our new model with all reasonable improvements that the benchmark can include. The main difference between the original model above and the improved model here is an explicit representation of separate eligible working and backup route sets. Here, we define $P'$ as the set of primary routes, and $B'$ is the set of backup routes available for demand $r$. $P'$ and $B'$ are the sets of working/restoration routes that cross span $j$.

Finally, we introduce two new binary variables, $\alpha'_r \in \{0,1\}$ and $\rho'_r \in \{0,1\}$, which indicate whether routes $p$ or $b$ were assigned as a working or backup route for demand $r$ (it equals 1 if it is assigned and 0 otherwise), these are similar to the $x^r$ and $y^r$ variables in the conventional model. The objective function of the improved model, which we call SBPP-imp, remains the same as the one from the original:

$$
\text{Minimize } \sum_{v \in L} c_j \cdot (s_j + w_j) \quad \forall j \in S
$$

(9)

The constraints are as follows:
SBPP networks for dual link failures. The algorithm looks at the tradeoffs between the resource utilization ratio and the protection switching time when provisioning backup capacity for dual link failure in a dynamic manner. The authors in [19] propose an algorithm for implementing SBPP type protection in an environment that has service level agreement constraints related to the reliability levels of each demand. An overview of various dynamic variations of SBPP is provided in [14].

Finally, prior work in [20] also developed a new SBPP design model (they refer to SBPP as “semi-dedicated restoration”). That approach considered a set of cycles, where one side of a cycle is selected as the working route and the other is selected as a backup route. Among the key benefits of the model is that it allows other survivability mechanisms to be considered simultaneously in the same network, and also that it allows for easy control of route diversity (the use of cycles to enumerate working and backup routes automatically forces their disjointedness). One drawback with that approach, however, is that it still includes a large number of binary variables (two per eligible cycle per demand relation – the same order of magnitude as the number of binary variables in the benchmark model), which makes obtaining solutions difficult in large networks. Furthermore, that model requires a great deal of pre-processing to enumerate eligible cycles and separate the two “sides” of each cycle for matching to working and backup routes. While further work has been done on this issue [21], the run time issues of SBPP have not been addressed.

### D. Goals and Motivation

The work in [16] has already shown the SBPP design model to be NP-complete, and although other NP-complete problems can sometimes be relatively easy to solve in practice, the SBPP problem is not. Our experience shows that ILP solvers have a difficult time solving models with a large number of binary variables, which the traditional SBPP design models have in abundance. And while near optimal algorithms exist for solving the SBPP design problem, they are by definition sub-optimal. Furthermore, even mid-sized networks are often unsolvable with the conventional design model.

The reader may suggest that we might be able to obtain near-optimal solutions by prematurely terminating runtimes (and taking the best solution found up to that point), relaxing integrality requirements, or providing the design model with a heavily restricted set of eligible backup routes. Although those techniques are often successful in reducing runtimes for network design models that consider other survivability mechanisms, they are not appropriate for the present model. For instance, the conventional SBPP model cannot be solved for many larger test case networks even when provided with a minimum eligible backup route set, and in some of those cases, a feasible solution (let alone an optimal one) isn’t returned after many days of runtime. And because most of the integer variables in the conventional model are binary, relaxing integrality and rounding them off to integer values will result in a solution that is meaningless (e.g. when a variable represents use or disuse of a route, a value of 0.5 has

\[
\sum_{\forall b \in B_p} \rho^r_\omega = 1 \quad \forall r \in D \tag{10}
\]

\[
\sum_{\forall p \in P} \alpha^r_p = 1 \quad \forall r \in D \tag{11}
\]

\[
\sum_{\forall \omega \in \omega} \sum_{\forall b \in B} z^r_\omega, b = w^r_i \quad \forall i \in S \tag{12}
\]

\[
\sum_{\forall \omega \in \omega} \sum_{\forall j \in S} \rho^r_\omega = 1 \quad \forall r \in D \tag{13}
\]

\[
\rho^r_\omega + \omega^r_\rho = z^r_{b,p} + 1 \quad \forall r \in D, p \in P^r, b \in B^r \tag{15}
\]

\[
\sum_{\forall \omega \in \omega} \sum_{\forall \omega' \in \omega} z^r_{b,p} = 1 \quad \forall r \in D \tag{16}
\]

The constraints in (10) and (11) ensure that each demand has only a single backup and working route. (12) allocates sufficient working capacity on each span to accommodate all the traffic routed across it and (13) places enough spare capacity to accommodate the largest set of concurrent backup routes for any span failure. (14) enforces disjointedness of the backup and restoration routes. (15) is used for the same purpose as (6). One other equation, (16), was added to help limit the solution space. While this improved model is significantly faster to solve than the original, it is still excessively time consuming (see Section IV for numerical data). Though not documented herein due to page count limits, validation was done (on our smaller test case networks that the original version of the model could solve in a few days) to ensure that this improved version of the model and the original do in fact produce equivalent designs. They do.

Finally, we will reiterate that this improved benchmark mode is a variation of the conventional model in Section II.A. We use this as our benchmark because we want to ensure that our model that follows in Section III is compared to a benchmark that benefits from every possible enhancement in runtime. This will also serve to circumvent a criticism that is often offered up in response to development of new design models.

### C. Other Approaches

Due to the complexity of solving the SBPP ILP design models, there have been a number of algorithmic approaches proposed to design SBPP networks. Work in [16], for instance, develops a fast heuristic and an algorithm to produce near optimal network designs. The approach uses a preliminary heuristic to find an initial feasible solution, and then iteratively refines it using an optimization algorithm. [17] looked at the performance of the SBPP model under multiple network failures, and proposed an algorithmic approach to design a network to withstand dual link failures. This paper proposed a backup scheme that used multiple routes (f+1 disjoint routes for f link failures), however, the motivation, solution procedure, and results were fundamentally different from the work herein. In [18], an algorithm is proposed for provisioning
no physical meaning and rounding often results in an infeasible solution or one that is vastly over-provisioned).

The goal of the present work, therefore, is to develop a new ILP based model that is quick to solve, and continues to adhere to the basic idea of SBPP, that being the idea that disjoint working routes may share spare capacity on their assigned backup routes. The model we describe herein will do just that. In fact, as we will show in the coming pages, the new model will provide more optimal solutions with significantly shorter runtimes than the conventional model, even when that conventional model is allowed to make use of improvements.

This new model uses concepts derived from SBPP, however, it is a new model, and produces designs that are distinct from designs that result from the SBPP-conv model. The following section outlines the new model and distinguishes it from the SBPP-conv model. The designs from this new model should be able to be implemented in a network that would use SBPP, as the concepts of spare capacity sharing and path-based protection are still intact.

III. NEW MULTI-FLOW SBPP ILP DESIGN MODEL

The approach we take here is to allow each demand to use more than a single working route and/or backup route. This alters the original SBPP problem, however, it does not significantly change the way protection is implemented in the network. Each path will have backup routes associated with it, and the spare capacity on these routes can be shared with other disjoint working routes. The effect on the ILP is that the binary variables are essentially converted into integer flow variables (requiring more than a simple relaxation of these variables) similar to those in other ILP design models [7]. \( \alpha' \), is now an integer variable that represents the number of working paths for demand \( r \) assigned to working route \( p \) and \( \rho' \) represents the integer number of backup paths for the working path \( p \) of demand \( r \) assigned to backup route \( b \). \( \zeta' \) encodes spans crossed by working routes. It is equal to 1 if working route \( p \) for demand \( r \) crosses span \( i \) and 0 otherwise. \( \varphi' \) is the equivalent parameter, encoding spans crossed by a backup route. Other notation is as used in (1) to (16).

While there is no explicit correlation between a specific working path and a backup path, the interaction of constraints in the model itself will effectively assign backup routes in a manner that is equivalent to that (see below). The objective function of this new ILP design model, which we call SBPP-MF, is the same as the one from the original models:

\[
\text{Minimize } \sum_{j \in D} c_j (s_j + w_j) \quad \forall j \in S
\]

The constraints are as follows:

\[
\sum_{p \in P} \varphi'_r = d_r \quad \forall r \in D
\]

Now rather than a constraint restricting the solution to having only a single working route per demand, we have new constraints in (18) that ensure that there are enough working paths for each demand to accommodate all of its demand. Likewise, a constraint assigning only a single backup route per demand is replaced by the equation in (19) assigning sufficient backup paths for each working route. (20) places sufficient working capacity on each span to accommodate all working routes crossing them, and (21) places enough spare capacity to accommodate the largest set of concurrent backup routes (forcer set) for any span failure. Finally (22) ensures that the demand assigned to a given working route has an associated disjoint backup flow. To convert this ILP model into an SCA type model, equations (18) and (20) can be removed (and \( \alpha' \), converted to an input parameter). Equivalently, the set of eligible working routes can simply be reduced to only a single working route per demand. We can also note that disjointedness of working routes from their associated backup routes is achieved by equation (22) but could also be done in the pre-processing step by temporarily removing the spans of a working route from the network topology before our depth-first-search route finder enumerates eligible backup routes that will protect it (SRLG considerations can also be included). Furthermore, the constraints in equation (21) consider all backup routes simultaneously applied to each span. For instance, if failure of span \( i \) affects three separate working routes that are all assigned backup routes that pass through span \( j \), then enough spare capacity will be placed on span \( j \) to accommodate all of them. If SRLGs exist, they can be considered by modifying the constraints in (21) to be valid for \( \forall i \in SRLG, \forall j \in \{ j \mid i \in j \} \) where \( SRLG \) is the set of all SRLGs and \( j \) is a set of SRLGs \( i \). Of course, the definition of \( \varphi' \) would have to be modified appropriately as well. So in this new model, route diversity (i.e., working and backup route disjointedness) is easily controlled and SRLG considerations are reasonably easy to implement.

The SBPP-MF model builds on the central idea of sharing backup capacity, however, it is not a mere relaxation of the 1/0 variables in the SBPP-conv and SBPP-imp models. The SBPP-MF model allocates backup capacity to any number of backup routes based on capacity allocated to individual working routes.

By allowing multiple working and backup routes the SBPP-MF model has the added flexibility to reduce forcer sets, spread capacity throughout the network, and more fully take

\[
\sum_{i \in S} \alpha'_r = \omega'_r \quad \forall r \in D, \forall p \in P'
\]

\[
\sum_{i \in S} \zeta'_r \cdot \varphi'_r = w_i \quad \forall i \in S
\]

\[
\sum_{i \in S} \sum_{j : j \in S'} \zeta'_r \cdot \varphi'_r \cdot \varphi'_j \leq s_j \quad \forall i \in S, \forall j \in S \mid j \neq i
\]

\[
\sum_{i \in S} \zeta'_r \cdot \varphi'_r \cdot \rho'_{r,b} = 0 \quad \forall r \in D, \forall p \in P'
\]
advantage of spare capacity sharing when compared to the SBPP-conv model.

IV. EXPERIMENTAL METHODS AND COMPUTATIONAL CONSIDERATIONS

Our analyses made use of 163 test case networks of 15 nodes to 40 nodes and divided into 6 families, each headed by a master network of an average nodal degree of 4. Each member of a network’s family is obtained by applying a succession of span removals to create a series of networks of decreasing average nodal degree. Each network is identical to the next higher-degree network of the same family except that one span has been removed. The master networks are shown in Figure 1. The 15-node family consists of 15 members, the 20-node family consists of 20, the 25-node family has 26, the 30-node family has 29, the 35-node family has 34, and the 40-node family has 39 members. Each network used a full mesh of demands, where each node pair exchanged a uniform random integer number of working paths between 1 and 10. All test case networks in the same family used the same demand matrix.

Since preliminary investigations using the joint working and spare allocation (JCA) design models took considerable lengths of time (days and weeks for even the smallest networks) we decided to look solely at the SCA version of problem. Working paths for each demand were from a shortest path routing except where a routing infeasibility existed due to the existence of a trap topology [11]. In those cases, the next shortest path was chosen as the working route. The per-unit cost of capacity on each span is proportional to its length. Sets of 10 eligible backup routes, where available, were provided for each working route, such that those routes chosen were the 10 shortest disjoint from the working route. The test case networks, demand matrices, and eligible backup route sets are the same as those used in [15].

The ILP design models were implemented in AMPL and solved using CPLEX 9 on a Sun 1.6 GHz quad CPU machine with 16 GB of RAM. The MIP gap was set to 0.01, so solutions were guaranteed to be within 1% of optimal. Such a high MIP gap was due to the very long runtimes for the conventional SBPP model. Run times were recorded from the CPLEX log output, and were measured as the solver’s runtime only. Data pre-processing and any post-processing are exclusive of those runtimes, but there was no significant difference between the conventional model and the new SBPP-MF model in either of those. We should also note that these runtimes was based on approximately 50% utilization of the server and there were variances in the server’s utilization over the series of test cases we ran. However, as will be discussed below, the runtime differences between the two models is at least an order of magnitude (and sometimes two), which is much larger than any potential error introduced into the data from that source.

V. RESULTS AND DISCUSSION

The first analysis we did was a comparison of runtimes between the two models, and it became clear very quickly that there was a significant difference between them, as shown in Figure 2. Each data point in the figure corresponds to the average of all runtimes for test case networks of that average nodal degree (actually, a bin of average nodal degrees centred on the value indicated, since not all families had a member of exactly the same average nodal degree). These runtimes show an enormous advantage for the new SBPP-MF model, especially in highly connected networks. In the lowest connectivity test networks, the new model took on average only 1% of the runtime needed on average for the conventional model (approximately two orders of magnitude). The longest runtime for the new SBPP-MF model over all of...
the test case networks was only 220 seconds, while the longest for the conventional model was several days for one particular hard-to-solve network (though most networks completed in 5 hours or less). In short, the SBPP-MF model provided significantly improved run times over the conventional model, making this model much more practical to solve.

One aspect of the findings in the figure that needs explaining is the levelling out (and in some cases even slight drops) of run times as our test case networks exceeded an average nodal degree of approximately 3.2. In the SBPP-MF curve, this is caused by a change in the characteristics of the available backup paths at this point. At approximately that degree of connectivity, there appears to be an almost step change in the number of disjoint backup paths that are close to the same length, making them easy choices for the solver to make and subsequently making the problem relatively easy to solve. In the benchmark model curve, the solver would not return optimal solutions after 8 days of runtime for many of the larger network families with high connectivity. Those particular test cases were excluded from the data in that curve, reducing the reported runtimes. If we’d had the patience to wait the solver out until those particularly large test cases solved, then the difference relative to the new model would have been even greater. Note that for the largest test case network (40 nodes and 80 spans, the new SBPP-MF model was able to solve it in only 285 seconds).

those capacity reductions were in fact significantly greater than the MIP gap setting. In fact, as can be seen in Figure 3 through Figure 8, those capacity reductions were as large as 6% in some test cases, and averaged 1.7% over all test networks. In those figures, the blue diamond markers correspond to the percentage reduction in capacity requirements. The solid curve of the same colour is the 2nd order polynomial best fit trend line for that data, with coefficients of determination, $R^2$, shown ($R^2$ of 1 implies a perfect fit).

**Figure 2 – Average solution runtimes for the two SBPP ILP design models.**

In addition to the significantly reduced runtimes, we also expected that the new SBPP-MF design model may provide improvements in capacity requirements over the conventional model. Since any design produced by the conventional model can be exactly duplicated by the new model (by having $\rho^* = d$, for one backup route and $\rho^* = 0$ for all others), then the new model will require the same capacity as the conventional model in the worst case. However, it is conceivable that in some cases, allowing multiple lower capacity backup routes will reduce the so-called forcer set of backup paths on some spans (in constraint equation (21)) so that the spare capacity required on them is also reduced. While minor differences within the 1% MIP gap setting of the solver wouldn’t necessarily imply any improvement, we found that in most test cases (particular those with higher connectivity),

**Figure 3 – 15n30s1 network connectivity and scaled average multi-flow vs. capacity reduction in SBPP-MF.**

**Figure 4 – 20n40s1 network connectivity and scaled average multi-flow vs. capacity reduction in SBPP-MF.**

**Figure 5 – 25n50s1 network connectivity and scaled average multi-flow vs. capacity reduction in SBPP-MF.**
The other set of data shown in Figure 3 through Figure 8 (the red squares and their 2nd order polynomial trend lines, using the upper x axis) is the capacity reduction plotted versus a multi-flow factor, where that multi-flow factor represents the total number of paths used divided by the number of demands (if a network has 4 demands, and one of them uses two backup routes, the multi-flow factor would be 1.25). The number of demands is the number of unique node pairs that have a communication demand between them, regardless of the number of demand units required to fulfill this requirement. It is clear that there is a correlation between the multi-flow factor and the capacity reductions achieved with the new model; when a network is able to better exploit the new model’s ability to split backup paths onto different routes, capacity reductions improve. The reason is as alluded to above. The spare capacity required on a span is defined by the largest concurrent set of backup paths crossing the span. Splitting backup paths onto multiple routes can potentially reduce that largest concurrent set of backup paths crossing a given span, reducing the spare capacity requirement on it. This process, however, has limits since splitting the backup paths on multiple routes often requires that some of the backup paths follow longer routes. Further analysis showed that if backup paths for a specific demand are split onto multiple routes, it is rarely on more than three routes, and in almost all cases, only onto two. In fact, 85.8% of all the demands in all test cases were routed on only a single route, 12.6% were routed on two, 1.4% on three, and only 0.1% on more than three.

VI. CONCLUDING DISCUSSION

The conventional SBPP ILP design model, even in a vastly improved version with only a small number of eligible backup routes is difficult and time consuming to solve. This is due to the very structure of the model, and the large number of binary variables it contains. Here we developed a new SBPP based ILP design model that has been shown to produce SBPP network designs in only a fraction of the time required by the conventional model with a slight change in the problem definition. At the same time, the new model also takes advantage of splitting backup paths onto multiple routes, reducing capacity requirements relative to the conventional model where only a single backup route is allowed for each demand.

One argument against using the new SBPP-MF model might be that some customer may, for some reason, require all of their traffic to follow a single route. In such cases, the ILP model could be modified to accommodate that requirement by re-introducing the original binary variable and corresponding constraints below in (23) and (24), where $\rho_b^r$ is the binary variable indicating usage of a backup route and $D'$ is the set of demands requiring only a single backup route. While this will have the effect of increasing the runtime of the model, practice has shown that if we use only a very limited number of binary variables, they will not have a major impact on runtimes.

$$\sum_{\forall b \in B'} \rho_b^r = 1 \quad \forall r \in D'$$  \hspace{1cm} (23)

$$\rho_b^r \times d_r = \rho_b^r \quad \forall r \in D', \forall b \in B'$$  \hspace{1cm} (24)

VII. REFERENCES


