Generalized Survivable Network

Kwok Shing Ho and Kwok Wai Cheung

Abstract—Two important requirements for future backbone networks are full survivability against link failures and dynamic bandwidth provisioning. We demonstrate how these two requirements can be met by introducing a new survivable network concept called the “Generalized Survivable Network” (GSN), which has the special property that it remains survivable no matter how traffic is provisioned dynamically, as long as the input and output constraints at the nodes are fixed.

A rigorous mathematical framework for designing the GSN is presented. In particular, we focus on the GSN capacity planning problem which finds the edge capacities for a given physical network topology with the I/O constraints at the nodes. We employ fixed single-path routing which leads to wide-sense non-blocking GSNs. We show how the initial, infeasible formal mixed integer linear programming formulation can be transformed into a more feasible problem using the duality transformation. A procedure for finding the realizable lower bound for the cost is also presented.

A two-phase approach is proposed for solving the GSN capacity planning problem. We have carried out numerical computations for ten networks with different topologies and found that the cost of a GSN is only a fraction (from 39% to 97%) more than the average cost of a static survivable network. The framework is applicable to survivable network planning for ASTN/ASON, VPN, and IP networks as well as bandwidth-on-demand resource allocation.

Index Terms—Network Design, Non-blocking Network, Survivable Network, ASTN, ASON, VPN, IP Network.

I. INTRODUCTION

With the proliferation of the IP networks and services, the backbone network architecture is evolving towards a two-layered model. The top layer carries different communication services using IP with extensions of MPLS. The lower layer is the high-bandwidth optical transport network based on ASON/ASTN. The segregation of the connection-oriented optical layer and the packet-based IP layer allows easier management and performance guarantee of resources by the operating carriers. Based on this trend, two basic requirements for the future core networks are: 1) dynamic bandwidth provisioning and 2) full survivability against link failures. Dynamic bandwidth provisioning means that the network can be provisioned flexibly as long as the resources are available. This is a paradigm shift from the conventional static or semi-static network provisioning view because of the recent advancements in ASON/ASTN and GMPLS technologies that allow connections to be set up and torn down in a matter of seconds. Many carriers (e.g. AT&T [1]) have already begun deploying such networks. Full survivability against link failures means that a network remains intact up to a certain number of link failures (at least one) occurring anywhere in the network. Given the huge bandwidth capacity of any link, any non-restorable link failure results in tremendous traffic loss and also revenue loss. Therefore, full survivability against link failures has been regarded as an essential requirement for today’s telecommunication network. As an example, Deutsche Telekom has introduced a 1+1 route diversity for each of the links in its core mesh network to guarantee full survivability against link failures [2].

The flexibility of future optical network brings to network operators a powerful way of meeting dynamic service requirements instantly, but at the same time introduces problems in two aspects, namely, 1) the lack of guarantee of full survivability against link failures because of the dynamic changes in traffic demands, and 2) the difficulty in the optimization of network resources in the planning and operation of the network especially during routing and resource allocation.

For the first problem, there were a few attempts to incorporate the dynamic traffic demand requirement into the survivable network design (see e.g. [3,4]). These studies use a statistical approach. Session arrivals and departures are assumed to be stochastic processes. Based on this, a set of dynamic traffic patterns is formed for the planning process. Because of the probabilistic description, full survivability cannot be guaranteed in these approaches. This is considered unsatisfactory because full survivability is an essential requirement and need to be guaranteed with certainty as we discussed above.

For the second problem, a blocking probability is usually used to describe the situations where a routing path cannot be found for a connection request even when both the source and destination have the available capacity resources. For example, the “Protected Working Capacity Envelope” (PWCE) concept was proposed as a solution for dynamic survivable service provisioning [5]. In the framework of PWCE, network operators first specify the limit of capacity on each link based on a single traffic matrix or capital budget. Then optimization (without routing considerations) is performed to partition the total capacity into the working and spare capacity in such a way that the set of working capacities defines a protected operational envelope that can be adopted by as many potential...
demand patterns as possible. After the optimization, the network operators can choose a routing scheme for the resulting network (e.g. the shortest path routing). If a connection request cannot be satisfied with this specific routing scheme, the connection is blocked. There will be situations where a routing path cannot be found for a connection request even when both the source and destination have the available capacity resources.

Therefore, a natural question faced by a network planner is how we can plan a network that guarantees: 1) a routing path can always be found for a connection request when resources at both the source and destination are available, and 2) full survivability against link failures when traffic demand changes from time to time.

In this paper, a new network concept called “Generalized Survivable Network” (GSN) is proposed as a solution for the above question. GSN has the special property that it remains fully survivable against link failures no matter how traffic is provisioned dynamically, as long as the input and output constraints at the nodes are fixed. The term “Generalized” means the generalization to the set of all possible traffic demands under the deterministic constraints of the network – namely, the maximum input or output traffic flow that a node can handle.

The conventional approach for designing a survivable network is to start with one static traffic matrix and compute the required working and spare capacities for the links. We call the resulting network planned with this approach the static survivable network (SSN) in this paper. Different from the conventional paradigm, GSN considers all possible traffic patterns under the input and output constraints at the nodes. Thus GSN requires higher link capacities than the SSN. One key issue that concerns the planning and deployment of GSN in practice is how GSN compares with the traditional SSN in terms of capacity requirement. This is the extra cost for dynamic demand provisioning. We evaluate this issue by performing some numerical experiments and show that it is possible to obtain cost-effective GSN solutions satisfying both the dynamic demand provisioning and full survivability in the physical layer. A GSN generally costs only a small fraction more than an average SSN. Even though the motivation for studying GSN concerns the core optical network, the concept of GSN is general and can be applied to other kinds of networks like VPN and IP networks.

The rest of the paper is organized as follows. In Section II, the GSN concept will be presented. Section III describes the mathematical formulation we used to estimate the required GSN capacity. Two Mixed Integer Linear Programming (MILP) formulations will be presented. The first is a straightforward MILP formulation which turns out to be impossible to solve because of the infinite number of constraints. We show how to transform the straightforward formulation using the duality transformation into a finite MILP problem. Since the problem is NP-hard, we split the problem into two parts and optimize each part separately using the CPLEX solver. Based on the MILP formulation, we show how to derive a lower bound for the cost by relaxing the integer constraints and solving a linear program in polynomial time. Some experimental results will be presented in Section IV to show that GSN can be cost-effective to deploy practically together with some discussions. In Section V, we will draw the conclusions.

II. THE CONCEPT OF GENERALIZED SURVIVABLE NETWORKS

A simple way to understand the Generalized Survivable Network is that it incorporates the non-blocking network property into a survivable network. Non-blocking network was first studied in a classical paper by Clos [6] for electronic circuit switching networks. In a non-blocking network, given that the source node has the available outgoing capacity and the destination node has the available incoming capacity, a path is guaranteed to be found. The algorithms for finding the routing paths can further be classified into strictly non-blocking, wide-sense non-blocking and rearrangeably non-blocking algorithms in the literature [7]. Generalizations of the non-blocking network to wavelength-selective switching and wavelength switching were given in Refs. [8], [9].

In the original definition which was applied to circuit switching systems, all circuits have the same bandwidth allocation. These fixed bandwidth circuits are routed from the ingress node to the egress node through some intermediate switching stages and the non-blocking property imposed certain well-known constraints on the number of intermediate stages required (see Fig. 1 for an illustration). Here we generalize the definition of non-blocking network by relaxing the fixed bandwidth allocation requirement for each circuit as follows. A network is said to be non-blocking if an egress node with an allowable outbound traffic capacity of \( \tau \) can be connected to any ingress node with an allowable inbound traffic capacity \( \geq \tau \) and vice versa. In this generalized definition, the constraints on the number of intermediate stages become the constraints on the edge (or link) capacities, assuming the traffic can be switched in an arbitrary manner at each network node or interconnection.

In a real network, the capacity constraints at the input and output interfaces of each network node (e.g. the bandwidth of the network interface cards) limit the traffic flowing into or out of a network. These are real constraints that are always present independent of whether we consider them in the planning and design of a network. Therefore, no matter how the traffic demand is changed dynamically, the traffic demands from a certain node to all the other nodes must satisfy the input/output (I/O) constraints. As shown in Fig. 2, if the input capacity of node 1 is limited to 1 unit (arrow into node 1), then the total demands from node 1 to node 2 (f2), node 3 (f3), and node 4 (f1) cannot exceed 1 unit. That is, \( f1 + f2 + f3 \leq 1 \). More generally, for any real network, the following deterministic constraints must hold:

\[
\sum_{d \in N} r_{d} \leq O_d, \quad \forall d \in N \tag{1}
\]

\[
\sum_{d,s \in N} r_{d,s} \leq I_s, \quad \forall s \in N \tag{2}
\]
where \( I_s \) is the maximum input capacity of node \( s \) and \( O_d \) is the maximum output capacity of node \( d \). \( r_{sd} \) represents the demand from node \( s \) to node \( d \). \( N \) is the set of network nodes.

If we plan a network in such a way that there is sufficient capacity for every valid traffic pattern that is consistent with the I/O constraints (1) and (2), then we can achieve a non-blocking network which guarantees a routing path can always be found for a connection request when resources at both the source and destination are available.

If we further design a non-blocking network by taking survivability into consideration, then the resulting network can guarantee the full survivability under the dynamic change of traffic. We refer to such network the “Generalized Survivable Network”, where the term “generalized” means the generalization to the set of all allowable traffic demands under the deterministic constraints of the network – namely, the maximum input or output traffic flow that a node can handle.

Interestingly, the non-blocking network concept has been exploited in the design of ATM networks [10] and VPNs called the hose model [11]. The GSN planning methodology is different from these works and is initiated with the concern for full survivability against link failures under dynamic traffic changes. Therefore, the GSN planning methodology incorporates the non-blocking network concept into the survivable network model whereas the other works consider the design of non-blocking networks only without considering the survivability aspect.

The four-node network shown in Fig. 2 is a GSN. In the figure, the number on the arrows going into / out of each network node represents the node’s I/O bandwidth capacity. The number without/with bracket on each network link represents the working/spare capacity of that link. This network can satisfy any possible demand matrix satisfying the I/O constraint with single-path routing while it is single-fault tolerant under link restoration and single-path rerouting. Six possible demand matrices that it can satisfy are shown in Fig. 3. Note that the demands need not be integers. The routing and rerouting information are shown in Table I. It should also be noted that different restoration schemes can be deployed in GSN even though we use link restoration mechanism in this example.

With this new GSN networking concept proposed, we create a new design methodology for the future network that can result in a network that fulfills the requirement of both dynamic bandwidth provisioning and full survivability. In addition to guaranteeing these two most important properties, the GSN design methodology is a suitable design methodology for the future networks because of the following additional advantages:

1) Ease of specification and characterization - for the GSN design methodology, only the input and output capacity bounds of each network node need to be specified but there is no need to specify the destinations, the exact demand elements, and the holding time or other statistical attributes of the traffic. These input and output capacity bounds are easily measured and can be easily predicted with the gravity model [12].

2) Deterministic - for the future networks, there are too many different types of traffic that may lead to the statistical variability in the individual source-destination traffic. If the design methodology is based on the prediction of future demands, there will be serious questions about whether the statistical model and sampling are accurate or appropriate. The GSN design methodology is based on the input and output capacity bounds which will not change once they are specified and therefore the GSN design methodology is deterministic.

III. A MATHEMATICAL FRAMEWORK FOR GSN

In this section, we present a rigorous mathematical framework using the MILP formulations to evaluate the capacity requirement for a GSN. Two MILP formulations will be given. The first is a straight-forward MILP formulation which turns out to be impossible to solve because there are an infinite number of constraints. By applying the duality theorem, we show how the straight-forward but infinite MILP formulation can be transformed into a finite MILP problem. Since the problem is NP-hard, we split the problem into two parts so that each part can be optimized separately using the commercial solver. A realizable lower bound for the cost is obtained by relaxing the integer constraints in this formulation to result in a linear problem which can be easily solved. Some numerical results are presented in Section IV based on the two-phase approach for comparison with the lower bound cost.

A. Notations, Definitions and Preliminaries

We let an undirected graph \( G(N,E) \) represent a network, where \( N \) represents the set of nodes equipped with some I/O devices or equipment, and \( E \) represents the set of edges. Each edge in \( E \) has two extremities of nodes \( i \in N \) and \( j \in N \) and will be denoted by \( \{i,j\} \). For each edge \( \{i,j\} \in E \), \( c_{ij} \) represents the cost of per-unit capacity reservation. For simplicity, all edges are assumed to have the same cost metric \( c_{ij}=1 \) and the same length in this paper. The capacity of each edge represents the maximum traffic that can flow through the edge from both directions simultaneously. For each node \( n \in N \), the input and output capacity bounds \( I_n \) and \( O_n \) are given. \( I_n \) and \( O_n \) are assumed to be non-negative integers.

We consider the traffic demands between all possible source-destination (sd) pairs. The traffic demands among all sd pairs are represented by the traffic matrix \( R \) in which each element \( r_{sd} \) represents a specific traffic demand from node \( s \in N \) to node \( d \in N \). We assume \( r_{sd}=0 \) for all \( n \in N \). A traffic matrix \( R \) is said to be allowable if all its elements simultaneously satisfy I/O constraints (1) and (2). The set of allowable traffic matrices \( R \) will be denoted by \( \mathcal{R} \). It should be noted that if the value of each element \( r_{sd} \) in \( R \) is a real number, the cardinality of \( \mathcal{R} \) will be infinite. Even if the value of each element \( r_{sd} \) in \( R \) is restricted to be an integer, the cardinality of \( \mathcal{R} \) is at least of \( \Omega(|N|!) \), which is an extremely large number.

A network is called non-blocking if it can satisfy all the allowable traffic matrices \( R \) in \( \mathcal{R} \) with a specific routing model. The definitions of wide-sense non-blocking and rearrangeably
non-blocking in circuit switching [7] are generalized as follows. If there exists a routing algorithm that can meet any allowable traffic demand without having to rearrange any existing connections, the network is said to be wide-sense non-blocking. If there exists a routing algorithm that can meet any allowable traffic demand by rearranging some existing connections, the network is said to be rearrangeably non-blocking. A wide-sense non-blocking network is a rearrangeably non-blocking network but not vice versa. For practical reasons wide-sense non-blocking networks are more interesting because existing network connections need not be torn down, rerouted and reconnected whenever there is a change in the traffic demand. This simplifies the network management and operations and improves the restoration speed when there is a link failure.

In this paper, we assume that the routing model is fixed single-path routing, i.e. each traffic demand \( r_{sd} \) is routed along one single path which will not change with time no matter what the current traffic matrix \( R \) is. A formulation based on fixed single-path routing will result in a network with sufficient capacity such that any given working demand can be carried by the fixed single route from a source to a destination. Therefore, the resulting network planned out by the model in this paper is a wide-sense non-blocking network and it can meet any allowable traffic demand without having to rearrange any existing connections.

Any link failure is assumed to cut off the bandwidth capacities in both directions. A network is said to be survivable if it has sufficient diversity (multi-edge-connected) and spare capacities for rerouting a certain portion of traffic demands under a set of failure scenarios. The process of rerouting the broken traffic is called restoration. If the restoration portion is 100\%, the network is a fully survivable network. For a survivable network, there are two general restoration schemes: 1) link restoration and 2) path restoration. In link restoration, when an edge fails, only the working flow on this edge will be rerouted with alternative routes that connect the two ends of the failed edge. In path restoration, all traffic demands \( r_{sd} \) passing through the faulty edge will be rerouted with alternative routes. The number of alternative routes can be single or multiple. In case of link failures, some route rearrangements are required by the restoration process and these rearrangements are not considered as traffic rearrangements under the non-blocking network definition.

A “Generalized Survivable Network” (GSN) has two properties: 1) it can satisfy dynamic traffic demands which vary with time under the deterministic input/output capacity constraints at each node, and 2) it is fully survivable under link failures. A GSN is \( \alpha \)-survivable (\( \alpha \) is an integer) if the network is fully protected up to \( \alpha \)-link failures anywhere in the network.

In this paper, for simplicity, we only consider the capacity planning of 1-survivable GSN under link restoration with single or multiple rerouting paths. The problem is stated as follows:

Given a network topology that is at least two-edge connected with a set of nodes \( N \) and a set of edges \( E \), the I/O constraints \( I_o, O_o \) for each node \( n \in V \), how to compute the working and spare capacity of each edge \( (i,j) \in E \) in order to obtain a 1-survivable GSN with minimum cost? The resulting GSN should be able to carry any allowable traffic demand with fixed single-path routing while guarantee full survivability under single link failure using link restoration.

The following variables will be commonly used in all mathematical formulations in this paper.

- \( w_{ij} \) working capacity on edge \( (i,j) \);
- \( x_{ij}^{sd} \) portion of flow for the traffic demand \( r_{sd} \) routed on edge \( (i,j) \) from node \( i \) to node \( j \);
- \( s_{ij} \) spare capacity on edge \( (i,j) \);
- \( z_{ij}^{fg} \) restoration flow on edge \( (i,j) \) for rerouting the demand volume through edge \( (f,g) \) (that is equal to \( w_{fg} \)) from node \( i \) to node \( j \) when link \( (f,g) \) is cut;

B. Normal Form

The capacity planning problem of GSN can be written down in a straightforward manner as follows:

**GSN Capacity Planning Problem (GSNCPP):**

\[
\text{Min } \sum_{(i,j) \in E} (w_{ij} + s_{ij}) \tag{3}
\]

Subject to:

\[
\sum_{j:(i,j) \in E} (x_{ij}^{sd} - x_{ij}^{sd'}) = \begin{cases} 
1 & \text{if } i = s \text{ and } i \in N, \\
-1 & \text{if } i = d \text{ and } i \in N \text{ and } s \neq d, \\
0 & \text{if } i \neq s, d
\end{cases} \tag{4}
\]

\[
\sum_{r_{sd},x_{ij}^{sd},x_{ij}^{sd'}} r_{sd}(x_{ij}^{sd} + x_{ij}^{sd'}) \leq w_{ij} \quad \forall R \in \mathbb{R}, (i,j) \in E \tag{5}
\]

\[
\sum_{j:(i,j) \in E} (z_{ij}^{fg} - z_{ij}^{fg'}) = \begin{cases} 
w_{fg} & \text{if } i = f \text{ and } i \in N, \\
-w_{fg} & \text{if } i = g \text{ and } i \in N \text{ and } f \neq g, \\
0 & \text{if } i \neq f, g
\end{cases} \tag{6}
\]

\[
z_{ij}^{fg} + z_{ij}^{fg'} \leq s_{ij} \quad \forall (i,j) \in E, (f,g) \in E \tag{7}
\]

\[
z_{ij}^{fg} = z_{ij}^{fg'} = 0 \quad \forall (i,j) \in E \tag{8}
\]

\[
w_{ij}, s_{ij} \text{ are non-negative integers, } \forall (i,j) \in E \tag{9}
\]

\[
x_{ij}^{sd}, x_{ij}^{sd'} \text{ are 0 or 1, } \forall (i,j) \in E, s, d \in N, s \neq d \tag{10}
\]

The GSN capacity planning problem defined by equations (3)-(11) will be referred to as Problem GSNCPP. The objective function (3) minimizes the total capacity installed in the network. We assume that once the capacities are installed, there will not be any additional routing cost. The GSNCPP formulation (3)-(11) assumes single-path routing and multiple-path rerouting in the case of single-link failure using link restoration. The GSNCPP formulation for single-path routing and single-path rerouting using link restoration is shown in Appendix A to avoid confusion. Similar formulations can be performed for path restoration but they are not considered here.
Constraints (4) are the classical flow balance constraints that ensure the routing between each $sd$ pair [13]. Constraints (5) ensure that enough working capacity is installed on each edge $[i, j]$ so that the total edge flow is satisfied for each allowable traffic matrix $R \in \mathbb{R}$. Constraints (6) are the flow balance constraints that ensure the rerouting of the broken working flow passing through the failed edge $[f, g]$ through multiple paths. Constraints (7) ensure that enough spare capacity is assigned on each edge $[i, j]$ so that the total rerouted flow is satisfied for any affected working flow upon any single link failure. Constraints (8) ensure the rerouted flow variables to be zero on the failed edge. Constraints (9) restrict the rerouted flow variables to be non-negative. Constraints (10) restrict the working and spare capacity to be non-negative integers. Constraints (11) restrict the flow variables to be either 0 or 1 so that single-path routing is ensured. Once the capacity allocation for the links is solved, some additional post processing is required to find the routing and rerouting paths but the problem can be solved explicitly in polynomial time [13].

The spare capacity planning problem alone is known to be NP-hard [14,15]. GSNCPP is also NP-hard because the spare capacity planning problem is a special instance of the GSNCPP. If we restrict the GSNCPP problem to only one specific traffic demand that satisfies the I/O constraints, the problem reduces to the spare capacity planning problem which is NP-hard. What makes GSNCPP even harder is the exponentially increasing Constraints (5). The number of constraints in (5) depends on the total number of allowable traffic matrices. As discussed in Section III-A, the total number of allowable traffic matrices will be at least of order $O(|N|!)$ and can even be infinite and so are the number of constraints in (5). Therefore, we cannot construct the whole integer linear program explicitly. In order to find a lower complexity algorithm, we have to simplify the formulation first. This will be discussed next.

C. Dual Form

The GSNCPP Problem is large in size because of the at least exponentially increasing $(|N|!)$ or more) Constraints (5). In order to tighten the formulation and obtain a form suitable for optimization, we want to eliminate Constraints (5). First, if we fix the flow variables with feasible values for all $sd$ pairs in Constraints (5), it can be seen that the optimal working capacity of link $[i, j]$ is just equal to the maximum traffic through that link over all allowable traffic matrices $R$. The maximum traffic through the link $[i, j]$ with fixed flow variables can be calculated from the following linear program:

Maximum Traffic via link $[i, j]$ ($MT_{ij}$):
\[
\begin{align*}
\text{Max} & \sum_{s, d \in N, s \neq d} r_{sd} (x_{ij}^{sd} + x_{ji}^{sd}) \\
\text{Subject to:} & \sum_{s, d \in N, s \neq d} r_{sd} \leq O_d, \quad \forall d \in N \\
& \sum_{d \in N, s \neq d} r_{sd} \leq I, \quad \forall s \in N
\end{align*}
\]

(12)

(13)

(14)

The constraints of this linear program are actually the I/O constraints (1) and (2) and characterize all valid traffic matrices. The objective function maximizes the traffic through $[i, j]$.

It may appear that if we replace Constraints (5) of the Normal Form with the above linear program, we can represent all allowable traffic matrices implicitly and obtain a compact form. However, it is not so straightforward. If we simply eliminate Constraints (5) and insert the linear program into the Normal Form, the objective function (12) will become non-linear and thus the whole formulation becomes a non-linear program.

In order to avoid the non-linear constraints and use the linear program above to obtain a compact formulation, we can apply the duality transformation [16] to the linear program (12)-(15). The method is shown as follows:

Dual of $MT_{ij}$ ($DMT_{ij}$):
\[
\begin{align*}
\text{Min} & \sum_{s, d \in N, s \neq d} (I_d o_{ij}^d + O_d s_{ij}^d) \\
\text{Subject to:} & x_{ij}^{sd} + x_{ji}^{sd} \leq o_{ij}^d + s_{ij}^d, \quad \forall \{i, j\} \in E, s, d \in N, s \neq d \\
o_{ij}^d, s_{ij}^d \geq 0, \quad \forall \{i, j\} \in E, n \in N
\end{align*}
\]

(16)

(17)

(18)

By the strong duality theory of linear programs, the objective value of $MT_{ij}$ is equal to that of its dual. Since the objective of $DMT_{ij}$ is a linear function, we can obtain a compact formulation simply by putting the linear program of $DMT_{ij}$ into the Normal Form and eliminating Constraints (5). This Dual Form will not contain any nonlinear constraints and is given as follows:

GSN Capacity Planning Problem - Dual Form (GSNCPP-DF):
\[
\begin{align*}
\text{Min} & \sum_{[i, j] \in E} (w_{ij} + s_{ij}) \\
\text{Subject to:} & \sum_{[j, k] \in \delta \{i\}} (x_{ij}^{sd} - x_{kj}^{sd}) \begin{cases} 1 & \text{if } i = s, \\ -1 & \text{if } i = d, \\ 0 & \text{if } i \neq s, d \end{cases}, \quad \forall \{i, j\} \in E, s, d \in N, s \neq d \\
x_{ij}^{sd} + x_{ji}^{sd} \leq o_{ij}^d + s_{ij}^d, \quad \forall \{i, j\} \in E, s, d \in N, s \neq d \\
\sum_{s, d \in N, s \neq d} (I_d o_{ij}^d + O_d s_{ij}^d) \leq w_{ij}, \quad \forall \{i, j\} \in E \\
\sum_{[j, k] \in \delta \{i\}} (w_{ij} - w_{kj}) \begin{cases} w_{ij} & \text{if } i = f, \\ -w_{ij} & \text{if } i = g, \\ 0 & \text{if } i \neq f, g \end{cases}, \quad \forall \{i, j\} \in E, \{f, g\} \in E \\
z_{ij}^f + z_{ij}^g \leq s_{ij}, \quad \forall \{i, j\} \in E, \{f, g\} \in E \\
z_{ij}^f = z_{ij}^g = 0, \quad \forall \{i, j\} \in E \\
z_{ij}^f, z_{ij}^g \geq 0, \quad \forall \{i, j\} \in E, \{f, g\} \in E \\
s_{ij} \text{ are non-negative integers, } \forall \{i, j\} \in E
\end{align*}
\]

(19)

(20)

(21)

(22)

(23)

(24)

(25)

(26)

(27)
GSNCPP-DF problem. and can be constructed explicitly. We can now solve the
by any linear value, i.e., replacing Constraints (27), (28) and (29)
approach, the bound obtained in the cut plane approach may not
since routing constraints are not included in the cut plane
bound obtained by the cut plane approach shown in Ref. [17].
Since these two sub-problems are more manageable, we can
solve each sub-problem exactly for medium sized networks
with the standard BB algorithm in reasonable time.

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS
A set of numerical experiments was carried out to evaluate
the performance of the GSNCPP-DF two-phase approximation
algorithm. We consider ten networks of various network sizes
and topologies. Their topologies are shown in Fig. 4 to 13. The
details of the ten networks are shown in Table II. Networks 1
and topologies. Their topologies are shown in Fig. 4 to 13. The
details of the ten networks are shown in Table II. Networks 1
and 8 are drawn from Ref. [15]. Networks 2 and 7 are European
networks described in Ref. [18] and [20]. Network 5 is the
NFSNET. Other networks are drawn from Ref. [19]. We found
that the algorithm could easily handle GSN planning and
design up to a size of 26 nodes and 30 links using a desktop PC.
All formulations were coded in C with the CPLEX callable
library version 9.0 [21] and then solved exactly with the MILP
solver in CPLEX. All test runs were made on a PC of dual
XEON 3GHz with 1G DRAM. The platform was Windows XP.
We also attempted to find optimal solutions for the
GSNCPP-DF using the CPLEX solver directly for comparison
with our two-phase approximation results. Owing to the
complexity of the GSNCPP-DF algorithm, the solver could
only compute the smallest size GSN with ten nodes. The
two-phase approximation algorithm gives a result (42 units)
that is within 5% of the true optimal for this 10-node case (40
units) obtained by the CPLEX MILP solver. For networks with
more nodes, the solver could not produce a result within a
reasonable timeframe and so no further comparison was
possible.
Instead, we use the lower bounding procedure introduced in
Section III-D to gain some insight about the performance of
various optimization strategies. Table III shows the GSN
bounds that can be achieved by the joint optimization approach.
The bounds are obtained by relaxing the integer constraints in
(19)-(30) as discussed in Section III-D. The Redundancy of
a survivable network is the ratio of the required spare capacity

\[
\begin{align*}
\text{GSNCPP-DF – Working Capacity Part (GSNCPP-WC):} \\
\text{Min } \sum_{i,j \in E} w_{ij} \\
\text{Subject to: Constraints (20)-(22) and (28)-(30).}
\end{align*}
\]
\[
\begin{align*}
\text{GSNCPP-DF – Spare Capacity Part (GSNCPP-SC):} \\
\text{Min } \sum_{i,j \in E} x_{ij}^d \\
\text{Subject to: Constraints (23)-(27).}
\end{align*}
\]

The working capacity and spare capacity planning problem of
GSN defined above will be referred to as Problem
GSNCPP-WC and GSNCPP-SC respectively. It should be
noted that \( w_{ij} \) for all links \( i,j \in E \) are variables that need to be
optimized in GSNCPP-WC during the first phase. They will be
the input parameters for optimizing the spare capacity in
GSNCPP-SC during the second phase.
Since these two sub-problems are more manageable, we can
solve each sub-problem exactly for medium sized networks
with the standard BB algorithm in reasonable time.

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS
A set of numerical experiments was carried out to evaluate
the performance of the GSNCPP-DF two-phase approximation
algorithm. We consider ten networks of various network sizes
and topologies. Their topologies are shown in Fig. 4 to 13. The
details of the ten networks are shown in Table II. Networks 1
and 8 are drawn from Ref. [15]. Networks 2 and 7 are European
networks described in Ref. [18] and [20]. Network 5 is the
NFSNET. Other networks are drawn from Ref. [19]. We found
that the algorithm could easily handle GSN planning and
design up to a size of 26 nodes and 30 links using a desktop PC.
All formulations were coded in C with the CPLEX callable
library version 9.0 [21] and then solved exactly with the MILP
solver in CPLEX. All test runs were made on a PC of dual
XEON 3GHz with 1G DRAM. The platform was Windows XP.
We also attempted to find optimal solutions for the
GSNCPP-DF using the CPLEX solver directly for comparison
with our two-phase approximation results. Owing to the
complexity of the GSNCPP-DF algorithm, the solver could
only compute the smallest size GSN with ten nodes. The
two-phase approximation algorithm gives a result (42 units)
that is within 5% of the true optimal for this 10-node case (40
units) obtained by the CPLEX MILP solver. For networks with
more nodes, the solver could not produce a result within a
reasonable timeframe and so no further comparison was
possible.

Instead, we use the lower bounding procedure introduced in
Section III-D to gain some insight about the performance of
various optimization strategies. Table III shows the GSN
bounds that can be achieved by the joint optimization approach.
The bounds are obtained by relaxing the integer constraints in
(19)-(30) as discussed in Section III-D. The Redundancy of
a survivable network is the ratio of the required spare capacity

\[
\begin{align*}
\text{GSNCPP-DF – Working Capacity Part (GSNCPP-WC):} \\
\text{Min } \sum_{i,j \in E} w_{ij} \\
\text{Subject to: Constraints (20)-(22) and (28)-(30).}
\end{align*}
\]
\[
\begin{align*}
\text{GSNCPP-DF – Spare Capacity Part (GSNCPP-SC):} \\
\text{Min } \sum_{i,j \in E} x_{ij}^d \\
\text{Subject to: Constraints (23)-(27).}
\end{align*}
\]
over the working capacity. It varies from 0.27 to 0.97. Table IV gives the GSN bounds that can be achieved by the two-phase algorithm (34) and (35). The Redundancy varies from 0.71 to 1.43. The results show that joint optimization may achieve a reduction up to 25% when compared with the two-phase optimization.

Another set of experiments was carried out to compare the cost of GSN with that of SSN for the ten test networks. The goal of this set of experiments is to assert whether a GSN can be deployed with a reasonable cost as compared to SSN. We adopt the following methodology. For each test network, we specify the I/O constraints of each node. We take twenty random samples among all allowable traffic matrices. Among these twenty random samples, ten samples are for sparse traffic matrices where all traffic originated from one node goes to a single destination (see e.g., Fig.3a - 3c), and the other ten are for dense traffic matrices where all traffic originated from one node goes to all the other nodes evenly except one (see e.g., Fig.3d - 3f). Whether a traffic matrix is dense or sparse, the I/O constraints (1) and (2) must be satisfied and the input and output capacities at every node are fully utilized. Then we plan a SSN under link restoration for each sampled traffic demand matrix, and take the average cost over these twenty sampled SSNs. We compare this average static cost with the corresponding cost of a GSN obtained by the two-phase approach. The cost for each sampled SSN was also obtained with the two-phase approach to reduce the side effect due to optimization and make a better comparison. The details are explained in Appendix B.

The traffic demands generated in these test cases are all asymmetric. All links are assumed to be identical with unity link cost per bandwidth. All nodes are assumed to be identical with both input and output capacity equal to unity. Other assumptions are the same as those given in Section III-A.

The comparisons of the required capacity between GSN and sampled SSN for all test networks are given in Tables V - IX. We have only shown the case of multiple-path rerouting because our aim of this experiment is to assert whether GSN can be deployed reasonably when compared with the static cases. The conclusion drawn from single-path rerouting is the same as that obtained from multiple-path rerouting. Of course, the total capacities for both SSN and GSN with single-path rerouting are higher than those with multiple-path rerouting. Therefore, the results for single-path rerouting are not shown to avoid redundancy.

Tables V and VI show the capacity required for the ten sampled sparse SSN and dense SSN respectively. Sparse SSN refers to the SSN planned with sparse traffic matrix while dense SSN refers to the SSN planned with dense traffic matrix. The Redundancy varies on average from 0.61 to 1.21 for the sparse and from 0.34 to 1.07 for the dense SSN. Table VII shows the two-phase optimized GSN cost. The Redundancy varies from 0.73 to 1.43.

Tables VIII and IX compare the GSN cost with the SSN cost. Table VIII compares the total capacity (sum of working and spare capacity) of the GSN with that of the SSN. The Dynamic Cost Ratio is the ratio of GSN cost to the average cost of the sampled SSN for each topology. From Table VIII, we observe that the cost of a GSN is about 32% to 84% more than the average cost of sparse SSN and is about 47% to 127% more than the average cost of dense SSN. When compared with the average cost of all twenty sampled SSN, GSN costs about 39% to 97% more but is within a factor of 2.

We believe the result is very encouraging. The cost of GSN is always higher than that of SSN because GSN needs to guarantee full-survivability and non-blocking performance for all allowable traffic matrices. The fact that the dynamic cost ratio is within a factor of 2 means that GSN can guarantee the full-survivability and non-blocking performance for at least $O(N!)$ allowable traffic demand matrices by paying no more than twice the cost of an average SSN which cannot give such guarantees. As mentioned in Section I, Deutsche Telekom has introduced a 1+1 route diversity for each of the links in its core mesh network (i.e. a total of 100% more capacity) to guarantee full survivability against link failures. Also, even though the ring network causes at least 100% redundancy, network operators still deploy ring networks for its simplicity and fast restoration speed. Here, it is shown that GSN can be deployed by paying no more than twice the cost of a SSN by providing full-survivability and non-blocking performance guarantee.

We believe that it is very reasonable to deploy the GSN by a network operator for its merits.

Table IX compares the working capacity of a non-blocking network with the average one of a SSN (no survivability). The ratio shows the price to pay for the flexibility to change the traffic demand dynamically. We observe that the working capacity (the non-blocking part) of a GSN is only about 26% to 64% more than the average one of a SSN. This indicates that the non-blocking property itself does not lead to too much extra capacity requirements.

On the other hand, the spare capacity (the survivability part) of a GSN is about 55% to 171% (not shown) more than the average counterpart of a SSN. This redundancy related to the non-blocking property causes a GSN to be over-provisioned in many cases. This can be further confirmed by the fact that the GSNs have at least 73% redundancy and 6 out of 10 GSNs have redundancy over 100% (see the last rows of Table VII). This is also the reason why the dynamic cost ratio for the dense SSN is always higher than that for the sparse SSN. For dense SSN, the capacity allocation can be more evenly distributed due to the more evenly distributed traffic. Therefore, the redundancy of the dense SSN is always less than that of the sparse SSN (see the last rows of Tables V and VI). As a result, the total capacity required by the dense SSN is less than that required by the sparse SSN and the dynamic cost ratio for the dense SSN is always higher than that for the sparse SSN.

After some investigations, it seems that the most probable reason for the high redundancy is that the non-blocking part of a GSN usually takes on a tree-like form after the first-phase optimization. Thus, the working capacity assignment is concentrated on a few links and that increases the redundancy during the second-phase optimization. As indicated by the
capacity reduction of the GSN lower bound with joint optimization, we believe that this condition may be improved by a joint optimization of both the working and spare capacity during the planning of GSN. This suggestion will be left for further study.

V. CONCLUSIONS

We have studied how the two important requirements for future backbone networks, namely, full survivability against link failures and dynamic bandwidth provisioning, could be met by introducing a new network concept called “Generalized Survivable Network” (GSN) that combines the non-blocking network concept with full survivability as a solution.

We have developed a basic framework for the planning and design of GSN. We showed how the initial, infeasible formal mixed integer programming formulation could be transformed into a more feasible problem using the duality transformation of the linear program, and then we showed how the problem could be decomposed into two manageable sub-problems: 1) the dimensioning of a network that can accommodate all allowable traffic demands limited by I/O constraints, and 2) the spare capacity planning of a working network.

A better lower bounding procedure for the cost has been obtained by relaxing the integer constraints to result in a linear program. The bound is realizable and useful for benchmarking various optimization strategies and routing algorithms. Based on these bounds, we observe that the joint optimization may bring a capacity reduction up to 25% when compared with the two-phase optimization.

Experimental results showed that GSN could be deployed with a reasonable cost (within a factor of 2) compared with statistical samples of static fully-survivable network. The results clearly indicate that GSN is a practical and deployable methodology for guaranteeing full-survivability and meeting dynamic traffic demand requirements.

Of course, there is still a lot of work to be explored on the various properties of GSN, such as the different classes of dynamic routing algorithms (wide-sense non-blocking or rearrangeable non-blocking), and a joint optimization framework (we use a separate two phase approach in this paper) etc. But the framework presented here should provide a firm foundation for further study.

In addition to offering a practical and guaranteed characterization framework for ASTN / ASON survivable network planning, GSN can also be used for guaranteeing the survivability of VPN and IP networks where the bandwidths are leased from the carriers and should be optimized. The GSN allows easy traffic redeployment and load balancing for networks by ISPs while guarantees the full-survivability of the network during reconfiguration. This should be a big advantage.

In conclusion, we believe GSN is an interesting, important and useful area that deserves our attention and efforts for further explorations.

REFERENCES


APPENDIX A

GSN Capacity Planning Problem - Dual Form (GSNCPP-DF)

for single-path routing and single-path rerouting in the case of single link failure with link restoration:

\[
\text{Min} \sum_{(i,j) \in E} (w_e + s_j)
\]  
(36)
Subject to:

\[
\sum_{j,k \in E} (x_{ij}^d - x_{ji}^s) = \begin{cases} 
1 & \text{if } i = s \\
-1 & \text{if } i = d \\
0 & \text{if } i \neq s, d 
\end{cases} \forall i \in N; \quad (37)
\]

\[
x_{ij}^d + x_{ji}^s \leq \alpha_{ij}^d + \alpha_{ji}^s, \forall \{i, j\} \in E, s, d \in N, s \neq d; \quad (38)
\]

\[
\sum_{s \in N} (I_s \alpha_{ij}^s + O_s \alpha_{ij}^d) \leq w_{ij}, \forall \{i, j\} \in E; \quad (39)
\]

\[
\sum_{j,k \in E} (z_{ij}^f - z_{ji}^d) = \begin{cases} 
1 & \text{if } i = f \\
-1 & \text{if } i = g \\
0 & \text{if } i \neq f, g 
\end{cases} \forall \{i, j\} \in E; \quad (40)
\]

\[
w_{ij} + M(\frac{z_{ij}^f + z_{ji}^d}{2} - 1) \leq s_{ij}, \forall \{i, j\} \in E, \{f, g\} \in E; \quad (41)
\]

\[
s_{ij} = z_{ij}^f = z_{ij}^d = 0, \forall \{i, j\} \in E; \quad (42)
\]

\[
z_{ij}^f, z_{ij}^d \geq 0, \forall \{i, j\} \in E, \{f, g\} \in E; \quad (43)
\]

\[
s_{ij} \text{ are non-negative integers, } \forall \{i, j\} \in E; \quad (44)
\]

\[
w_{ij} \text{ are non-negative integers, } \forall \{i, j\} \in E; \quad (45)
\]

\[
x_{ij}^d, x_{ji}^s \text{ are 0 or } 1, \forall \{i, j\} \in E, s, d \in N, s \neq d; \quad (46)
\]

\[
\alpha_{ij}^s, \alpha_{ij}^d \geq 0, \forall \{i, j\} \in E, n \in N; \quad (47)
\]

where \( M \) is the upper bound of \( w_{ij} = \min \left\{ \sum_{s \in N} I_s, \sum_{s \in N} O_s \right\} \).

APPENDIX B

For each SSN, the spare capacity planning for the survivable part is the same as that of the GSN. Therefore we can use the same GSNCPP-SC algorithm for optimizing the spare capacity planning for each sampled SSN. However, we cannot use the GSNCPP-WC algorithm directly for optimizing the working capacity planning of each SSN. This is because there is only one traffic matrix for designing a SSN, so we have to replace the non-blocking network constraints (for calculating the maximum traffic flow among all allowable traffic matrices) by the constraints for calculating the maximum traffic flow with the given single traffic matrix. Therefore the following formulation has been used for optimizing the working capacity of designing SSN to replace GSNCPP-WC:

Static Working Capacity Planning (SWCP):

\[
\text{Min } \sum_{(i,j) \in E} w_{ij} \quad (48)
\]

Subject to: Constraints (20),(28),(29) and

\[
\sum_{s,d \in N, s \neq d} r_{sd} (x_{ij}^d + x_{ji}^s) \leq w_{ij}, \forall \{i, j\} \in E; \quad (49)
\]

Kwok-shing Ho received the M. Phil. degree in Information Engineering from the Chinese University of Hong Kong, Shatin, N.T. Hong Kong, China in 2002. Since 2002, he has been a Ph. D. candidate in Information Engineering from the Chinese University of Hong Kong, where he is active in survivable network planning research. His current research interests include the design and evaluation of survivable optical network. He received the best student paper award in APOC 2005.

Kwok-wai Cheung is currently Professor in the Department of Information Engineering, the Chinese University of Hong Kong. He received his B.S., M.S. and Ph.D. from the University of Hong Kong, Yale University and Caltech respectively. From 1987 to 1992, he worked as a member of Technical Staff at Bell Communications Research (now Telcordia Technologies), New Jersey, USA. He joined the Chinese University of Hong Kong in 1992. From 1996 to 2001, he served as the founding Director of the Centre for Innovation and Technology. The Centre has facilitated the spinning off of a number of technology start-ups. He has published over a hundred journal and conference papers, holds five U.S. patents, nine international patents, and over twenty others pending. He was awarded an Electronics Letters Premium in 1992.
Fig. 1. A symmetric Clos network.

Fig. 2. A four-node GSN.

Fig. 3. Six traffic matrices supported by the four-node GSN. Three matrices are sparse matrices (a, b, c) and the others (d, e, f) are dense matrices.

Fig. 4. Network 1.

Fig. 5. Network 2.

Fig. 6. Network 3.

Fig. 7. Network 4.

Fig. 8. Network 5.

Fig. 9. Network 6.
TABLE II
INFORMATION ABOUT THE TEN TEST NETWORKS.

<table>
<thead>
<tr>
<th>Network</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Nodes</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>17</td>
<td>18</td>
<td>20</td>
<td>23</td>
<td>26</td>
</tr>
<tr>
<td>Number of Edges</td>
<td>22</td>
<td>26</td>
<td>25</td>
<td>23</td>
<td>21</td>
<td>31</td>
<td>39</td>
<td>31</td>
<td>33</td>
<td>30</td>
</tr>
<tr>
<td>Number of OD pairs in R</td>
<td>90</td>
<td>110</td>
<td>132</td>
<td>156</td>
<td>182</td>
<td>272</td>
<td>306</td>
<td>380</td>
<td>506</td>
<td>650</td>
</tr>
<tr>
<td>Average Nodal Degree</td>
<td>4.40</td>
<td>4.73</td>
<td>4.17</td>
<td>3.54</td>
<td>3.00</td>
<td>3.65</td>
<td>4.33</td>
<td>3.10</td>
<td>2.87</td>
<td>2.31</td>
</tr>
</tbody>
</table>

TABLE III
GSN BOUND (JOINT OPTIMIZATION).

<table>
<thead>
<tr>
<th>Network</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total GSN Capacity</td>
<td>33.00</td>
<td>37.52</td>
<td>43.40</td>
<td>68.35</td>
<td>82.60</td>
<td>96.32</td>
<td>88.46</td>
<td>131.18</td>
<td>205.15</td>
<td>384.00</td>
</tr>
<tr>
<td>Redundancy</td>
<td>0.32</td>
<td>0.27</td>
<td>0.38</td>
<td>0.49</td>
<td>0.61</td>
<td>0.48</td>
<td>0.36</td>
<td>0.68</td>
<td>0.76</td>
<td>0.97</td>
</tr>
</tbody>
</table>

TABLE IV
GSN BOUND (SEPARATE OPTIMIZATION).

<table>
<thead>
<tr>
<th>Network</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total GSN Capacity</td>
<td>41.00</td>
<td>50.00</td>
<td>56.33</td>
<td>90.75</td>
<td>106.00</td>
<td>106.65</td>
<td>109.25</td>
<td>141.33</td>
<td>240.00</td>
<td>438.00</td>
</tr>
<tr>
<td>Redundancy</td>
<td>0.71</td>
<td>0.79</td>
<td>0.88</td>
<td>1.06</td>
<td>1.21</td>
<td>0.72</td>
<td>0.76</td>
<td>1.02</td>
<td>1.14</td>
<td>1.43</td>
</tr>
</tbody>
</table>
Table V: Capacity requirement for different sparse SSNs (fixed single-path routing, link restoration with multiple-path rerouting).

<table>
<thead>
<tr>
<th>Network</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample 1</td>
<td>25</td>
<td>26</td>
<td>33</td>
<td>48</td>
<td>62</td>
<td>67</td>
<td>63</td>
<td>116</td>
<td>140</td>
<td>200</td>
</tr>
<tr>
<td>Sample 2</td>
<td>29</td>
<td>34</td>
<td>37</td>
<td>61</td>
<td>61</td>
<td>66</td>
<td>68</td>
<td>101</td>
<td>138</td>
<td>228</td>
</tr>
<tr>
<td>Sample 3</td>
<td>24</td>
<td>33</td>
<td>34</td>
<td>49</td>
<td>62</td>
<td>66</td>
<td>67</td>
<td>109</td>
<td>150</td>
<td>280</td>
</tr>
<tr>
<td>Sample 4</td>
<td>30</td>
<td>29</td>
<td>35</td>
<td>41</td>
<td>60</td>
<td>64</td>
<td>66</td>
<td>100</td>
<td>148</td>
<td>246</td>
</tr>
<tr>
<td>Sample 5</td>
<td>26</td>
<td>29</td>
<td>34</td>
<td>61</td>
<td>58</td>
<td>59</td>
<td>49</td>
<td>97</td>
<td>157</td>
<td>238</td>
</tr>
<tr>
<td>Sample 6</td>
<td>26</td>
<td>33</td>
<td>32</td>
<td>47</td>
<td>57</td>
<td>74</td>
<td>72</td>
<td>116</td>
<td>164</td>
<td>252</td>
</tr>
<tr>
<td>Sample 7</td>
<td>27</td>
<td>32</td>
<td>36</td>
<td>45</td>
<td>57</td>
<td>60</td>
<td>69</td>
<td>105</td>
<td>142</td>
<td>278</td>
</tr>
<tr>
<td>Sample 8</td>
<td>25</td>
<td>30</td>
<td>40</td>
<td>51</td>
<td>58</td>
<td>64</td>
<td>54</td>
<td>126</td>
<td>152</td>
<td>220</td>
</tr>
<tr>
<td>Sample 9</td>
<td>25</td>
<td>28</td>
<td>39</td>
<td>59</td>
<td>49</td>
<td>71</td>
<td>50</td>
<td>106</td>
<td>163</td>
<td>246</td>
</tr>
<tr>
<td>Sample 10</td>
<td>25</td>
<td>28</td>
<td>39</td>
<td>50</td>
<td>52</td>
<td>71</td>
<td>60</td>
<td>98</td>
<td>182</td>
<td>246</td>
</tr>
<tr>
<td>Average Total Capacity</td>
<td>26.20</td>
<td>30.20</td>
<td>35.90</td>
<td>51.20</td>
<td>57.60</td>
<td>66.20</td>
<td>61.80</td>
<td>107.40</td>
<td>151.60</td>
<td>243.40</td>
</tr>
<tr>
<td>Average Redundancy</td>
<td>0.67</td>
<td>0.65</td>
<td>0.69</td>
<td>0.79</td>
<td>0.88</td>
<td>0.72</td>
<td>0.61</td>
<td>0.90</td>
<td>0.95</td>
<td>1.21</td>
</tr>
</tbody>
</table>

Table VI: Capacity requirement for different dense SSNs (fixed single-path routing, link restoration with multiple-path rerouting).

<table>
<thead>
<tr>
<th>Network</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample 11</td>
<td>22</td>
<td>23</td>
<td>29</td>
<td>42</td>
<td>50</td>
<td>61</td>
<td>55</td>
<td>96</td>
<td>137</td>
<td>225</td>
</tr>
<tr>
<td>Sample 12</td>
<td>22</td>
<td>23</td>
<td>29</td>
<td>42</td>
<td>50</td>
<td>62</td>
<td>54</td>
<td>97</td>
<td>139</td>
<td>224</td>
</tr>
<tr>
<td>Sample 13</td>
<td>22</td>
<td>22</td>
<td>30</td>
<td>43</td>
<td>49</td>
<td>62</td>
<td>54</td>
<td>98</td>
<td>138</td>
<td>225</td>
</tr>
<tr>
<td>Sample 14</td>
<td>22</td>
<td>22</td>
<td>30</td>
<td>42</td>
<td>50</td>
<td>63</td>
<td>55</td>
<td>97</td>
<td>139</td>
<td>226</td>
</tr>
<tr>
<td>Sample 15</td>
<td>21</td>
<td>23</td>
<td>30</td>
<td>42</td>
<td>50</td>
<td>63</td>
<td>55</td>
<td>97</td>
<td>139</td>
<td>227</td>
</tr>
<tr>
<td>Sample 16</td>
<td>22</td>
<td>23</td>
<td>30</td>
<td>42</td>
<td>50</td>
<td>62</td>
<td>55</td>
<td>97</td>
<td>138</td>
<td>224</td>
</tr>
<tr>
<td>Sample 17</td>
<td>22</td>
<td>24</td>
<td>30</td>
<td>42</td>
<td>51</td>
<td>61</td>
<td>55</td>
<td>97</td>
<td>139</td>
<td>228</td>
</tr>
<tr>
<td>Sample 18</td>
<td>22</td>
<td>24</td>
<td>30</td>
<td>43</td>
<td>50</td>
<td>62</td>
<td>55</td>
<td>96</td>
<td>139</td>
<td>226</td>
</tr>
<tr>
<td>Sample 19</td>
<td>22</td>
<td>22</td>
<td>30</td>
<td>42</td>
<td>50</td>
<td>62</td>
<td>55</td>
<td>96</td>
<td>139</td>
<td>226</td>
</tr>
<tr>
<td>Sample 20</td>
<td>22</td>
<td>23</td>
<td>30</td>
<td>42</td>
<td>51</td>
<td>62</td>
<td>55</td>
<td>97</td>
<td>139</td>
<td>227</td>
</tr>
<tr>
<td>Average Total Capacity</td>
<td>21.90</td>
<td>22.90</td>
<td>29.80</td>
<td>42.20</td>
<td>50.10</td>
<td>62.00</td>
<td>54.80</td>
<td>96.80</td>
<td>138.60</td>
<td>225.80</td>
</tr>
<tr>
<td>Average Redundancy</td>
<td>0.37</td>
<td>0.34</td>
<td>0.49</td>
<td>0.56</td>
<td>0.67</td>
<td>0.55</td>
<td>0.41</td>
<td>0.76</td>
<td>0.83</td>
<td>1.07</td>
</tr>
</tbody>
</table>

Table VII: Capacity requirement for different GSNs (fixed single-path routing, link restoration with multiple-path rerouting).

<table>
<thead>
<tr>
<th>Network</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total GSN Capacity</td>
<td>42</td>
<td>52</td>
<td>63</td>
<td>91</td>
<td>106</td>
<td>107</td>
<td>113</td>
<td>142</td>
<td>240</td>
<td>438</td>
</tr>
<tr>
<td>Redundancy</td>
<td>0.75</td>
<td>0.86</td>
<td>1.10</td>
<td>1.07</td>
<td>1.21</td>
<td>0.73</td>
<td>0.82</td>
<td>1.03</td>
<td>1.14</td>
<td>1.43</td>
</tr>
</tbody>
</table>

Table VIII: Dynamic cost ratio of different GSNs (1-survivable).

<table>
<thead>
<tr>
<th>Network</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ratio of GSN to Sparse SSN</td>
<td>1.60</td>
<td>1.72</td>
<td>1.75</td>
<td>1.78</td>
<td>1.84</td>
<td>1.62</td>
<td>1.83</td>
<td>1.32</td>
<td>1.58</td>
<td>1.80</td>
</tr>
<tr>
<td>Ratio of GSN to Dense SSN</td>
<td>1.92</td>
<td>2.27</td>
<td>2.11</td>
<td>2.16</td>
<td>2.12</td>
<td>1.73</td>
<td>2.06</td>
<td>1.47</td>
<td>1.73</td>
<td>1.94</td>
</tr>
<tr>
<td>Ratio of GSN to All SSN</td>
<td>1.75</td>
<td>1.96</td>
<td>1.92</td>
<td>1.95</td>
<td>1.97</td>
<td>1.67</td>
<td>1.94</td>
<td>1.39</td>
<td>1.65</td>
<td>1.87</td>
</tr>
</tbody>
</table>

Table IX: Dynamic cost ratio of different GSNs (0-survivable).

<table>
<thead>
<tr>
<th>Network</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ratio of GSN to Sparse SSN</td>
<td>1.53</td>
<td>1.53</td>
<td>1.42</td>
<td>1.54</td>
<td>1.56</td>
<td>1.61</td>
<td>1.61</td>
<td>1.24</td>
<td>1.44</td>
<td>1.63</td>
</tr>
<tr>
<td>Ratio of GSN to Dense SSN</td>
<td>1.50</td>
<td>1.64</td>
<td>1.50</td>
<td>1.63</td>
<td>1.60</td>
<td>1.55</td>
<td>1.59</td>
<td>1.27</td>
<td>1.48</td>
<td>1.65</td>
</tr>
<tr>
<td>Ratio of GSN to All SSN</td>
<td>1.51</td>
<td>1.58</td>
<td>1.46</td>
<td>1.58</td>
<td>1.58</td>
<td>1.58</td>
<td>1.60</td>
<td>1.26</td>
<td>1.46</td>
<td>1.64</td>
</tr>
</tbody>
</table>