

Segment-based approaches to survivable translucent network design under various ultra-long-haul system reach capabilities [Invited]

Gangxiang Shen and Wayne D. Grover

TRLabs, 7th Floor, 9107-116 Street, N.W., Edmonton, Alberta, Canada T6G 2V4
{gshen,grover}@trlabs.ca

RECEIVED 8 SEPTEMBER 2003; REVISED 10 NOVEMBER 2003;
ACCEPTED 17 NOVEMBER 2003; PUBLISHED 22 DECEMBER 2003

One of the most practical architectural options for optical networking is a so-called translucent network based on a predominance of optically transparent switch nodes and a smaller number of strategically placed opaque (electronic core) switch nodes. In such a network it is technically easier to assume failure detection at the opaque nodes only and thus natural to consider viewing the transparent path segments between opaque nodes as the entities to be protected for network survivability, as opposed to single spans or entire end-to-end paths. We develop and test capacity-design models to compare this type of segment-based restoration scheme with conventional schemes. More important, however, a fast, nearly optimal, algorithm is proposed that can determine the placement of opaque nodes so that the fewest possible number is needed that ensures complete translucent reachability and single-failure survivability on the basis of the corresponding transparent path segments. Our data and methods also reveal the trade-off between the transparent reach obtainable by an ultra-long-haul (ULH) system and the corresponding number of opaque nodes required in the network (including survivability considerations), and thus we attain important insights to guide the relative allocation of research and development efforts on ULH systems as opposed to optical–electronic–optical cross-connect cost reduction. © 2003 Optical Society of America

OCIS codes: 060.4250, 060.4510.

1. Introduction

1.A. Background and Objective

A network is referred to as transparent if its nodes are all-optical cross connects (OXCs) without any electronic regeneration function. This requires that lightpaths in such a network can reach any destination with required signal-to-noise ratios (SNRs) because payload regeneration is not available en route. A transparent network in which each path must uniquely use the same wavelength on each fiber en route is called a wavelength-path (WP) network [1]. At the other extreme, if all nodes are equipped with full wavelength-conversion capability, the network is called a virtual WP (VWP) network [1]. In the latter case each lightpath can occupy different wavelengths on each fiber en route. Between these two extremes is perhaps the most practical option for many years to come—a partial wavelength-conversion (PWC) network [2–4]. A PWC network can be either based on a limited number of OXC nodes equipped with wavelength conversion on every signal path or realized by each node having a limited pool of wavelength converters.

A network in which regeneration (and implicitly also wavelength conversion) is available for all signals at every node is also a so-called opaque network. Here lightpath channels are optoelectronically detected at each node, digitally regenerated, and then reassigned to any available outgoing wavelength [5]. Transparent networks have the advantage of analog signal transparency, which implies complete bit-rate transparency and protocol transparency. However, complete optical transparency implies difficulties in network control and management because the signal is never accessed electronically for monitoring. Optical domain performance monitoring is possible but usually more difficult and expensive. In addition, even though advanced ultra-long-haul (ULH) systems [6] can now support an optical signal traveling a few thousand kilometers without electronic payload regeneration, the signal does eventually require electronic regeneration en route to retain digital SNRs [8]. A strictly transparent optical network therefore has some critical network diameter above which it cannot extend with a given optical transmission technology. On the other hand, an opaque network facilitates network monitoring, control, and management but requires optical–electronic (O–E) and E–O transponders and high-speed electronic switching at each node. The cost, space, power, and reliability implications of such complete O–E–O switching is a significant barrier preventing electronic-core optical switches from being installed in every node for the full volume of all signal flows through each node of the entire network.

Therefore, a practical approach is to strike a balance between transparency and opaqueness in an optical network in terms of “transparent islands” or, more generally, a “translucent” optical network [9]. In the former, a large-scale optical network is divided into several domains (i.e., islands) of optical transparency. In a domain, lightpaths can transparently reach any node without signal regeneration. But for communication between different domains, electronic switches are used at the domain boundaries. These switches act as 3R regenerators and wavelength converters while relaying the lightpaths crossing the domain boundaries.

However, a translucent optical network is more general than a network of transparent islands in that the regeneration capability is strategically distributed over the network as a whole. Rather than being dedicated to routing lightpaths only in and out of transparent islands, switches that have the electronic regeneration function can be shared by all paths of the network as a whole. One implementation of a translucent network is based on sparse placement of opaque switches. Here, one deploys a relatively small number of strategically chosen opaque (i.e., O–E–O) nodes at which wavelength conversion and regeneration is possible. All other nodes are lower-cost optically transparent OXCs [10]. Figure 1 illustrates an example of a translucent network, where nodes (1,3) are opaque nodes, whereas the rest of the nodes are transparent OXC nodes. Opaque nodes are able to regenerate optical signals and convert wavelengths electronically, while transparent OXC nodes have the optical switching function only. In a translucent network, there are two types of lightpaths: *translucent* lightpaths and transparent lightpaths. A lightpath is called translucent if there are some opaque nodes en route for signal regeneration and wavelength conversion, whereas a lightpath is called transparent if there are no opaque nodes en route. In Fig. 1 lightpath (2–1–4–3–7–6) is a translucent lightpath because nodes (1,3) are opaque nodes, and lightpath (2–5–8–9–6) is a transparent lightpath because all the nodes en route are transparent OXC nodes. We will call the lightpath segment between two neighboring opaque nodes a transparent segment. In Fig. 1 the translucent lightpath is made up of three successive transparent segments, which are segments (2–1), (1–4–3), and (3–7–6).

Another approach is to use translucent switches at all or some of the nodes in the network [9,11]. Each of these switches contains an optical switching core and an electronic matrix as well. Each core is generally smaller than it would be in a corresponding all-optical or all-electronic OXC because the electronic module is used only to provide regeneration

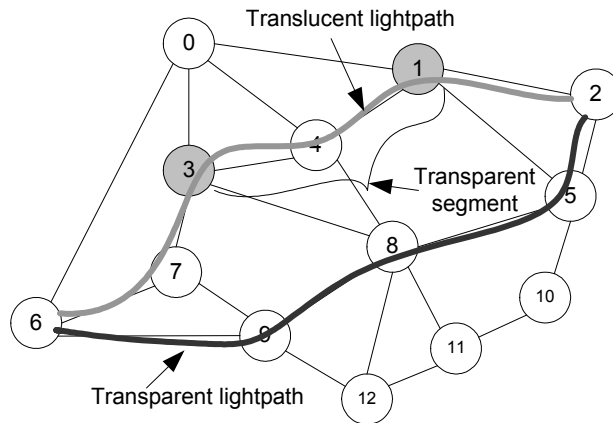


Fig. 1. Example of a translucent network, in which only two nodes (1,3) are opaque nodes.

and (implicitly) wavelength conversion to paths that need it. Each path through a translucent node can be either all-optically switched (via the optical module) or routed through the electronic core module, which regenerates its payload and assigns it any new wavelength desired (and available). The decision between optical or electronic switching is based on the analog noise properties of the optical path, i.e., whether regeneration is required before further transmission.

In this paper, we consider translucent networks based on the *sparse* placement of fully opaque switches such as in Fig. 1. Our focus is on the problem of placing the fewest number of opaque nodes so as to enable survivable routing between all node pairs. Related to this is the idea of planning protection based on the transparent segments between opaque nodes, rather than on conventional span- or path-oriented survivability schemes.

Let us now review some considerations about protection and restoration in optical transport network (OTN) design. Span-based and path-based restoration techniques are the basic techniques for survivable network design [12–15]. In a span-restorable (SR) network, the failure detection and restoration are carried out between the two terminating nodes of a failed span [12]. Because of the locality of response, SR networks are sometimes thought of as being fast but with relatively lower spare-capacity efficiency. In contrast, a path-restorable network is more efficient in spare capacity but often has a slower restoration speed [13,14]. Our interest here is in an intermediate and more general option—segment-based survivability. Segment-based survivability schemes carry out restoration of a failed path segment between the opaque nodes upstream and downstream of the actual span failure. A path segment can thus be a single span, a complete path, or any part of a complete path. There are at least three reasons to be interested in this approach for translucent networks:

1. Path segments will be generally shorter than complete paths, so restoration can be faster than schemes operating between the path end nodes because alarm propagation and signaling delays are fewer.
2. Rapid and low-cost failure detection is more easily based at the opaque nodes terminating a transparent segment than at the transparent nodes immediately adjacent to a failed optical span.
3. By considering the transparent segments inherently defined by a (sparse) set of opaque node placements—when placing such nodes, we have the prospect of be-

ing able to realize a least-cost survivable translucent network design, that is to say, a design using the fewest opaque nodes possible such that all node pairs can route demands between themselves, and in the event of any failure, a translucent or transparent alternate path also always exists.

Optical signal monitoring and failure detection are primary motivators for basing protection on segments, not on spans in a translucent network. Although techniques have been proposed for optical-domain monitoring at a transparent OXC node, they tend to be expensive and slow (scanning optical spectrum analyzers, for example, and scanned receivers). Electronic failure detection based on the O–E detection process and bit-error-rate (BER) measurements are much simpler, faster, and well developed. A dedicated per-channel transparent node failure detector is usually limited to detection of optical power loss, whereas an opaque node can look into the content on each channel and monitor detailed measure of channel health such as BER, synchronization, far-end failure status, and so on. Thus in a translucent network, it may be easier to use the opaque nodes to terminate transparent segments and detect and identify failures in those segments.

1.B. *Prior Research*

Translucent optical networks were first considered in comparison with opaque and transparent networks. Two operational strategies were later proposed for translucent lightpath establishment and evaluated in Refs. [16] and [17]. In Ref. [16] strategies for sparse translucent switch placement were also investigated in static and dynamic operations. Dynamic routing algorithms were also investigated in Ref. [19] for translucent networks with translucent nodes. Recently, a translucent network with sparsely placed opaque switching nodes was also studied in Ref. [10]. A simple but efficient sparse opaque switch-placement algorithm was also proposed and an almost optimal dynamic routing algorithm (the so-called two-dimensional Dijkstra algorithm) was developed specifically for translucent networks [10].

At the same time, generalized segment-based survivability approaches have also been under study recently. An approach for dynamically establishing segmented protection paths in WDM mesh networks was first proposed in Refs. [19] and [20], where an algorithm was developed for the segmented protection path selection, and performance in terms of average call acceptance ratio and wavelength use were evaluated. A protection-domain-based scheme was also proposed in Ref. [21]. It divided each working path into several overlapped protection domains, and each domain contained a working and protection path-segment pair, with the protection path-segment offering the protection function for the failures on the working path-segment in the domain. Another method, called subpath protection, was proposed in Ref. [22], which partitioned a network into several OSPF-like (open shortest path first) areas, and in each area, the shared-path protection approach was used for the failures occurring on the segment of a path crossing the area. Performance comparisons between path, subpath, and span restoration were also made in Ref. [23] on the basis of simulations where path-segment restoration (PSR) was carried out between an end node of the failed span and an end node of each affected working path. This is not generalized segment protection in the sense we consider. Also recently, an extension of the basic p -cycle concept in Ref. [24], so-called flow p -cycles, was developed to carry out segment-based protection in Ref. [25]. Because of the preconfigured characteristic of p -cycles, flow p -cycles have the merit of extremely fast restoration and a spare-capacity efficiency close to that of path restoration without stub release. Another recent study in Ref. [26] specifically considered survivable “transparent islands,” in which a subnet-partitioning algorithm was proposed and subnet-based survivable network designs were conducted for comparison with the design of the original unpartitioned network. More recently, some research in Ref.

[11] considered translucent survivable networks with sparsely placed translucent nodes, in which two heuristic algorithms were considered for the sparse placement of translucent nodes. Traffic-engineering schemes were also developed to design survivable translucent networks for future uncertain demands.

To our knowledge, however, no study has yet considered a survivable translucent network based on the idea of placing the fewest opaque switches, and taking advantage of the failure-detection capability of these nodes, to incorporate a segment-based approach to survivable network design. Most of the existing literature has focused on dynamic lightpath provisioning issues only in translucent networks or, where survivability has been considered, it is with end-to-end shared backup or short-leap protection schemes, with an interest in the blocking performance assuming *given* capacities. In contrast, we consider the minimum-cost capacity design for ensured segment-based protection with optimal solutions for the spare-capacity requirements and with methods that require the fewest possible number of opaque nodes. Most prior literature has been about simulation of blocking performance of different routing and wavelength assignment (RWA) and/or protection-provisioning schemes in networks in which node types (transparent or opaque) and capacities are already assumed. In contrast, this study approaches the translucent network primarily from an optimal design standpoint—taking ultra-long-haul (ULH) technology effects into account.

1.C. Outline and Contributions

We believe that this study makes several important contributions. First, we identify the *minimum* transparent reach (MTR) required for a translucent network to guarantee reachability between all node pairs by translucent lightpaths. In this analysis the number of opaque nodes in the network is not a constraint. There can be as many opaque nodes as required in the network. MTR is an important parameter for the ULH system selection because if the transparent reach (TR) of ULH systems is below this threshold, some node pairs cannot reach each other. We present a method to determine the required MTR that can guarantee full reachability between node pairs of a network, and we extend it to consider the same assurance under any single span failure to find the required minimum survivable transparent reach (MSTR). (Note that MTR and MSTR are properties of a network, whereas TR is a property of the optical transmission systems that may be used in that network.) A second contribution is an efficient heuristic to place opaque nodes when the transparent reach is larger than the required MSTR. The solution quality is almost optimal when we evaluate it on smaller-scale networks where we can compare it with the optimal solutions. But it runs very much faster than an optimal solution on large networks. Importantly, the node placement algorithm also takes failure situations into account. Another contribution is an integer linear programming (ILP) model for segment-based restoration design. We use it to study PSR and a segment-based variation on shared backup path protection (SBPP) called shared backup segment protection (SBSP). Finally, we also study the effects of ULH transparent reach on the number of opaque nodes needed in a translucent network, as well as the related spare-capacity requirements and relative restoration-speed measures in comparison with other known restoration schemes.

The rest of the paper is organized as follows. In Section 2 we present a method to determine the minimum transparent threshold for a given network. A simplified opaque node placement problem and an efficient heuristic for opaque node placement are proposed. A generalized ILP model for segment-based restoration is presented then in Section 3 and extended to model the method of SBSP. The segment-based methods are then employed to design translucent survivable networks considering two possible situations: (1) only opaque nodes are able to detect failures, and (2) all the nodes are able to detect failures. In Section 4 the performance of various translucent survivable designs is evaluated in comparison with

conventional span- and path-based restorable designs. Section 5 is a brief conclusion.

2. Identifying Minimum Survivable Transparent Reach and Placing Opaque Nodes

2.A. Identifying the Required Minimum Survivable Transparent Reach for a Translucent Network

To establish a translucent lightpath, we need to ensure that the ULH equipment reach is at least as large as the length of the *longest* transparent segment. Similarly, for a translucent network, to guarantee that all the nodes can reach each other by translucent lightpaths, the ULH equipment reach is required to be at least as large as the length of the longest transparent segment of the network. It is easy to understand that a network with more opaque nodes normally has a shorter longest transparent segment. Therefore, if we allow that there are as many opaque nodes as required, there should be a minimum length of the longest transparent segment for a network. We call such a minimum length the MTR of a translucent network. Only if the transmission reach exceeds the MTR can all node pairs possibly reach each other by a translucent lightpath.

The MTR can be initially thought of simply as the largest span distance of the network, since such a reach can always guarantee a successful establishment of a translucent lightpath between any node pair if the electronic regeneration is allowed at any node on demand. However, the largest span distance is found to be overestimated; there is a smaller reach that can guarantee full reachability of all the node pairs as well. For example, in the network as shown in Fig. 2 the longest span is span (0–6), whose length is 202. However, 202 is not the MTR of the network, because the network is still connected if the span is virtually removed, and as long as the network is connected, it is always possible to establish a translucent lightpath for each node pair given that there are sufficient opaque nodes in the network. We can similarly remove the next-longest spans from the remaining networks. After span (0–6), spans (1–2) and (4–5) can also be removed, while the remaining network still keeps connected. However, we cannot remove the next-longest span, i.e., span (2–5), because its removal will disconnect the network, where node 2 becomes isolated, unable to establish any lightpath with any other nodes. Accordingly, we find that the length of span (2–5), i.e., 121, is the MTR of the network. Note that this is much smaller than the length of the longest span, 202.

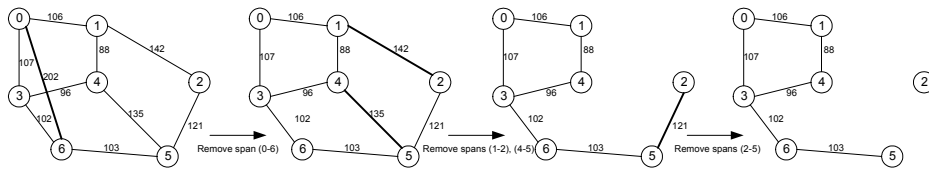


Fig. 2. Example to determine *minimum* transparent reach (MTR) for a network.

We can employ the above process to identify the MTR in a general case. The detailed steps are as follows:

MTR identification algorithm:

1. Order the spans in the network according to their physical lengths.
2. Virtually remove the longest span from the network and check the connectivity of the remaining network.

If the remaining network remains connected, rerun Step 2; otherwise, stop, and the physical length of the final removed span is the MTR.

However the above process does not yet take failure situations into account. To guarantee that there is a feasible lightpath existing between any node pair, even if the network is in any possible single-failure state, we extend the algorithm as follows:

MSTR identification algorithm:

1. Initialize MSTR, $MSTR := 0$.
2. For a network without any span failure, run the MTR identification algorithm above to get the required MTR_0 .
3. $MSTR := MTR_0$.
4. For (all single span-failure situations)
{Run the MTR identification algorithm to get the required minimum transparent reach MTR_f for the network with failed span removed.

If ($MTR_f > MSTR$)
MSTR := MTR_f .}
5. Return (with MSTR).

In the MSTR identification algorithm, we try to identify MTRs for the network when it is under various failure states, which include the state without any failure and all states of single span failure. We compare the resulting MTRs to find the largest one, which is the MSTR of the network as a whole. This is the distance at which the ULH system would be able to guarantee the full reachability of all nodes no matter whether the network is in a normal state or single-failure state.

2.B. Placing Opaque Nodes: the Optimal Problem and Heuristic Algorithm

Given a transparent reach of ULH equipment that is larger than the required MSTR of a network, we still need sufficient opaque nodes to guarantee reachability between all the node pairs. The opaque node placement problem is apparently an NP-hard problem because of its strongly combinatorial nature. For the purpose of research, one way to find the minimum number of required opaque nodes, and the optimal placements, is to enumerate all possible node combinations starting from a single opaque node and adding an additional opaque node at each step until full reachability is achieved; that is, first place one opaque node at one of N possible node positions and for each position, check whether the network has full reachability when the network is in the normal or single-failure state. When the network does have full reachability, the minimum number of opaque nodes optimal opaque node positions are found; otherwise, we add one more opaque node to the network and try again as in the previous step. Generally, if at least $K - 1$ opaque nodes are so far required, then for the next step we need to try a total of $C_{N,K}$ node placement combinations. If any one of these combinations provides full reachability between all node pairs under all single-failure situations, then the minimum number of required opaque nodes is K and an optimal opaque node placement combination is found. This process would be continued until the lowest K is found such that one or more of its combinations makes node pairs fully reachable (and survivable). Of course, when K is equal to N , the network is an opaque network. Thus the process will always stop at or before $K = N$. The above search process is optimal by

exhaustion, but blind. Among all the combinations enumerated, there are many in which the K nodes being considered are not even fully reachable among themselves. They can be excluded immediately with a corresponding test of transparent reachability between these nodes. For efficiency, we perform such exclusion in what follows. Note also that there can be more than one optimal combination for opaque node placement in a translucent network. The exhaustive search can easily identify all such equivalent optimal combinations at the critical value of K where the transparent threshold is just satisfied.

Of course, such an exhaustive combinatorial search is by its nature not suited for large-scale use—a heuristic is essential. Our interest in the purely exhaustive search is only as a way to obtain optimal solutions for research. These are necessary to assess performance of the following heuristic. The heuristic we describe next, for the same problem, is named the hub node first (HNF) algorithm. The steps for placing opaque nodes not considering span-failure situations are described first as follows:

Hub node first (HNF) algorithm:

1. With a given network and a transparent transmission reach capability, we construct an initial logical mesh graph, in which all the nodes are the same as those of the original network, and there is an edge between every node pair if there is a path existing between them whose distance does not exceed the transparent reach of available equipment.
2. If the logical mesh graph is fully connected, then stop. No opaque nodes are needed. Otherwise, go to the next step.
3. From the current non-opaque node list, select the node with the highest nodal degree in the logical mesh graph and deploy an opaque switch at the node. If there are multiple nodes with the same nodal degree, select the one with more paths transiting it; if such a measure is also the same for more than one node, select the one with highest nodal degree in the *original* network (not the logical mesh graph of step 2); if still the same for multiple nodes, use a random strategy to select one of them.
4. Update the nodal degrees for all the neighboring nodes of the opaque node placed at step 3. Add a direct edge between any two nodes in the above neighboring node list reflecting that the two nodes can at least reach each other with the opaque node as a relay.
5. Check whether the logical mesh graph becomes fully connected, that is, each node is connected to all the other $N - 1$ nodes in the network. If fully connected, then a feasible minimum node solution is found and the algorithm is terminated; otherwise, go to step 3.

Figure 3 gives an example to illustrate the HNF algorithm. Assume, for example, a transparent reach that is equal to one hop and a simple network as shown in Fig. 3(a). We can construct an initial logical mesh graph that is the same as the original network as shown in Fig. 3(a). From the node degrees in the logical graph and other measures, we see that nodes 1 and 2 have the same priority (for step 3 of HNF above). We therefore randomly select one of them, e.g., node 1, to place an opaque switch and update the nodal degree for each node in the logical graph to connect any two nodes in the neighboring node list of node 1. The new edges are shown by dotted lines in Fig. 3(b). The resultant logical graph shows that it is not fully connected yet, so we need to add more opaque nodes. Node 2 has a higher priority than node 4, because the former has a higher nodal degree in the original network as shown in Fig. 3(a), although nodes 2 and 4 both have the same nodal degree

in the logical graph, so we select node 2 to place an opaque switch and update the nodal degree for each node in a similar way. Now we find that the logical mesh graph becomes fully connected, so the algorithm is terminated with two opaque nodes placed at nodes 1 and 2.

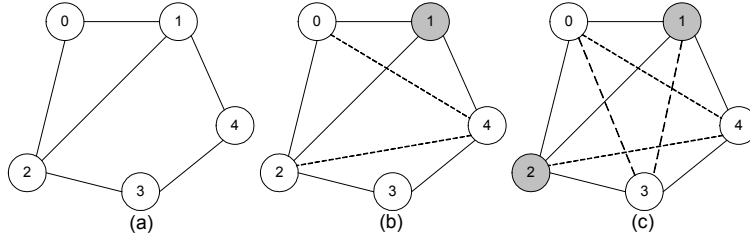


Fig. 3. Example to illustrate the HNF algorithm: two opaque nodes yield full translucent reachability.

Unlike the exhaustive combinatorial search, HNF finds only one final opaque node combination. Nonetheless, by construction, the nodes in the HNF solution are guaranteed to have the highest reachability to other nodes in the network as well as being placed so as to have the highest numbers of transiting primary paths and highest nodal degrees. That is why we call the algorithm hub node first (HNF). As pointed out, HNF is a general algorithm that does specifically not consider any span-failure situations. For a survivable network considering various span-failure situations, some extension to the HNF algorithm is as follows:

Survivable HNF (SHNF) algorithm:

1. For a nonfailure network, apply the HNF algorithm to identify a set of opaque nodes H .
2. For (each span failure situation)
 - {Remove the failed span from the nonfailure network and apply HNF algorithm above to the remaining network to identify an additional set of required opaque nodes F .
 - $H := H \cup F$.}
3. H is the final set of opaque nodes to deploy.

The proposed SHNF algorithm has high, but still polynomial computational, complexity. The reasoning for this is as follows: We need to consider a total of M network failure states. For each state, we first need to determine the initial logical mesh graph by examining whether the shortest path between each node pair exceeds or is within the transparent reach. There are a total of $N(N-1)/2$ node pairs, and for each of them the computation for the shortest path is of $O(N^2)$; therefore, for the logical mesh-graph creation, the required total computation is of $O(N^4)$. On the basis of the logical mesh graph, we then need to place opaque nodes. The worst case is to place N opaque nodes in the network, and for each opaque node placement, we need to find the node with highest nodal degree from at most N nodes and update the logical mesh graph to connect any two nodes in the prior logical mesh graph that are both connected to the newly placed opaque node. The above process needs a total computation of $O(N^3)$. Thus SHNF requires a computation of $O[M(N^4 + N^3)]$, simplified as $O(MN^4)$, for an N -node M -span network.

3. Generalized Segment-Based Protection Method and Its Application to Translucent Survivable Network Design

3.A. Segment-Based Protection Basics

Let us now look at the application to transparent-segment-based protection. Figure 4 illustrates span- and path-based restoration techniques and the proposed segment-based technique. As shown in Fig. 4(a), there is a working path (2–1–4–3–7–6). Upon a span failure the span-based approach reroutes the affected traffic demands on the failed span to restoration paths, whose end nodes terminate at the failed span. For example, if span (3–4) fails, paths (3–0–4) and (3–7–9–8–4) may be triggered to carry the affected lightpaths. In general, several distinct routes may be used. The path-based approach realizes a restoration at the two end nodes of each affected working path. As shown in Fig. 4(a), if span (3–4) fails, the affected traffic on the working path will be restored by alternate paths (2–1–0–6) and (2–5–8–9–7–6); the restoration switching happens at two end nodes (2,6) of the path. Path-based restorable techniques can be further divided into path restoration [13] and shared backup path protection (SBPP) [27]. Path restoration is also called failure-dependent path restoration, because it allows multiple protection paths to restore an affected primary path, and the protection paths are allowed to contain surviving spans of the affected primary path. SBPP is conversely called failure-independent path restoration because it uses a single protection path to restore the affected primary path, but the protection path must be link or node disjoint from the primary path. SBPP can be considered to be a special case of the general path restoration because all the eligible protection paths of SBPP are a subset of the protection paths of path restoration. Compared with path restoration, SBPP provides the advantage of not needing to know the location of the failure, which simplifies the restoration process. However, SBPP normally has a slightly inferior spare-capacity redundancy and possibly much lower service path availability compared with path restoration because the latter is an adaptive response.

The segment-based approach differs from the above methods in some important ways. As defined, a segment can be a span, a complete path, or any part of the complete path. As shown in Fig. 4(b), given a primary path between node pair (0,5), the segment-based approach allows us to restore a span failure, e.g., span (1–2), on any segment as long as one end node of the segment is upstream of the failed span and the other in the downstream of the span. Between any two such nodes, there can be multiple restoration segment options. We represent such restoration-path segment groups in Fig. 4 with dotted curves, where each curve contains all the possible segments between the node pair.

We can implement the segment-based approach in two ways. We call these options *path-segment restoration* (PSR) and *shared backup segment protection* (SBSP). PSR can be regarded as a benchmark scheme of the segment-based approach. This method allows all distinct protection segments of a working path to be eligible to recover a failure, and all the protection segments are allowed to share the spare capacity in the network. PSR is similar in several aspects to conventional path restoration with stub release. Both are flow-based in the sense that restoration of each affected primary path is carried out in an independent fashion (i.e., they are not all bundled together). Multiple protection paths or segments can be used in the recovery effort. However, PSR is expected to be more flexible and have better spare-capacity efficiency than end-to-end path restoration without stub release, because PSR does not require strictly end-to-end replacement paths. When a segment is employed to restore a failure, it can reuse some parts of the stubs of an affected primary path. Here such stub reuse is similar to the stub release in conventional path restoration, but stub reuse is not strictly as efficient as stub release [15]. In terms of spare-capacity efficiency, PSR is therefore expected to have a redundancy between path restoration without, and with, stub release. In practice, PSR may also have a faster speed when failure propagation and

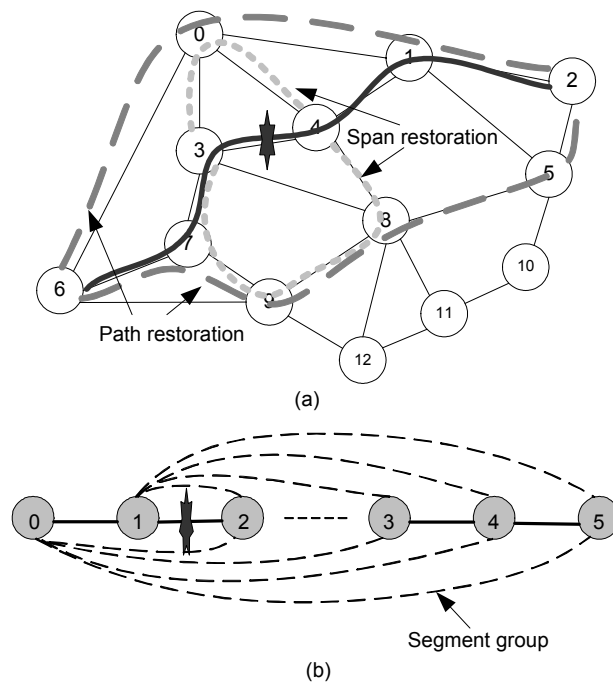


Fig. 4. (a) Examples of span- and path-based restoration, (b) example of segment-based restoration.

signaling times dominate, because segments are shorter than paths in general.

Just as PSR is related to path restoration, SBSP is the cousin of SBPP. It differs from PSR in that a single fully disjoint restoration-path segment is preplanned for each working-segment failure in SBSP. SBSP is expected to have a better spare-capacity efficiency and a faster restoration speed than SBPP for the same reasons that PSR will be slightly more efficient than end-to-end path restoration. However, a drawback of SBSP is that it is not a completely failure-independent scheme like end-to-end SBPP.

3.B. Integer Linear Programming Models

We now develop the integer linear programming (ILP) optimization models for both PSR and SBSP. Both are obtained from extension of conventional models for path restoration [13] and SBPP [27], respectively.

3.B.1. Model for Path-Segment Restoration

In the model of conventional path restoration, the basic protected entity is a path. Between a node pair exchanging a demand, there can be multiple working and preplanned protection routes. The ILP model is designed on a demand-pair basis to guarantee that all the failure-affected traffic demands, which may be made up of flows from multiple affected working paths, should be fully restored [13]. In PSR, the basic protection entity is a segment, which is any part of a full path. Therefore, the ILP model can be correspondingly designed on a segment basis to guarantee that all the traffic demands on an affected working path are fully restored. Bearing this in mind, we develop the model for PSR as follows:

Sets and parameters:

S is the set of spans on the network, indexed by i .

D is the set of demands between node pairs, indexed by r .

D_i denotes the affected demand set upon span failure i .

Q_i^r is the set of failure-affected working routes between node pair r upon span failure i .

$P_i^{r,q}$ is the set of protection segments for working route q between node pair r upon span failure i .

C_k represents the unit cost of capacity on span k .

d^r is the number of demand units between node pair r .

$g^{r,q}$ is the number of capacity units allocated on working route q of demand pair r .

$\zeta_i^{r,q}$ takes the value of one if working route q of demand pair r uses span i ; zero, otherwise.

$\delta_{i,j}^{r,q,p}$ takes the value of one if upon span failure i , protection segment p for working route q of demand pair r uses span j ; zero, otherwise.

Variables:

s_k is the number of spare-capacity units on span k .

$f_i^{r,q,p}$ is the number of capacity units allocated on protection segment p to protect working route q of demand pair r upon span failure i .

Objective: minimize the total spare-capacity cost.

PSR:

$$\min \left\{ \sum_{j \in S} C_j s_j \right\}.$$

Constraints:

$$\sum_{p \in P_i^{r,q}} f_i^{r,q,p} \geq g^{r,q} \quad \forall q \in Q_i^r; \quad \forall r \in D_i; \quad \forall i \in S, \quad (1)$$

$$s_k \geq \sum_{r \in D_i, q \in Q_i^r, p \in P_i^{r,q}} \delta_{i,k}^{r,q,p} f_i^{r,q,p} \quad \forall (i,k) \in S^2, \quad i \neq k. \quad (2)$$

Constraint (1) says that the lost demand on working route q between demand pair r upon span failure i must be fully restored by restoration flows assigned to the available protection segments. Constraint (2) says that the spare capacity on span k must be sufficient to meet the simultaneous demands of all restoration segments that use it to restore any single span failure. Note that if there is only a single working path between each node pair (e.g., a single shortest working route), the notation combination (r,q) , which represents working route q between node pair r , can be simplified as r alone representing the working flow between node pair r . This would be a common practical simplification of the general case above.

3.B.2. Model for Shared Backup Segment Protection

For this model, the additional notation to that of the PSR is as follows:

Parameters:

P_i^r is the set of eligible backup segments capable of restoring span failure i for the working path of node pair r . (Similar to SBPP, and unlike PSR, we assume that there is only a single restoration segment chosen for any working path between a node pair.)

$\delta_{i,j}^{r,p}$ takes the value of one if the eligible backup segment p for restoration of span failure i for demand pair r uses span j ; zero, otherwise.

Variables:

$x_i^{r,p}$ takes the value of one if backup segment p , eligible for restoring span failure i , for demand pair r is used; zero, otherwise.

Objective: minimize the total spare-capacity cost.

SBSP:

$$\min \left\{ \sum_{j \in S} C_j s_j \right\}.$$

Constraints:

$$\sum_{p \in P_i^r} x_i^{r,p} = 1 \quad \forall r \in D_i; \quad \forall i \in S, \quad (3)$$

$$s_k \geq \sum_{r \in D_i, p \in P_i^r} \left(\delta_{i,k}^{r,p} x_i^{r,p} d^r \right) \quad \forall (i,k) \in S^2, \quad i \neq k. \quad (4)$$

Constraint (3) says that there is only a single backup segment p actually selected to restore span failure i for demand pair r . Constraint (4) guarantees that there is sufficient spare capacity on each span to accommodate all the backup segments simultaneously crossing the span for the failure of any other span.

3.B.3. Applying the Segment-Based Approach to Translucent Survivable Design

Although we commented in the Introduction on how much easier failure detection is at the opaque nodes, for generality we do not rule out failure detection at the transparent nodes. Both failure-detection strategies can be used to *activate* the segment-based protection responses. If we deploy span-failure detection systems such as an optical power tap at each transparent OXC node, then we can detect span failures at any node. In this case we can apply the generalized segment-based approach to the translucent network design with only the constraint of the ULH transparent reach considered for the set of protection segments. We call this the span failure strategy. However, if we use just the opaque nodes to monitor optical channel integrity, then one more constraint is needed for the set of protection segments; i.e., all the segments must be started or terminated at opaque nodes or source or destination nodes of lightpaths. We call this the opaque node strategy.

Figure 5 illustrates the differences and how they further define which restoration-path segments through spare capacity are eligible for use in restoration. In Fig. 5(a), we assume that *all* nodes have span-failure detection ability. Upon a span failure [e.g., span (3–4)], the two end nodes (3,4) of the spans send alarm-indication signals (AIS) upstream and downstream to the source node 6 and destination node 2 of an affected working flow. All the nodes on the way, which include all the intermediate nodes (1,7), the two end nodes of the failed span (3,4), and source and destination nodes (6,2), will be notified on such a failure. Upon receiving an AIS, each of the nodes will check its preplanned restoration information to see whether it is responsible for a segment-based restoration for part of the affected working flow resulting from span failure (3–4). In this case preplans can be constructed according to the solution obtained from the optimization models, which (in its most general form) indicates which affected working flow resulting from which span failure should ask which node pair to restore how many units of affected traffic by which protection segment. If the node finds a match in its preplans, then it will activate a corresponding restoration-path segment. The signaling can be via a protocol such as RSVP-TE (Resource Reservation Protocol—traffic engineering) or CR-LDP (constraint-based routing—Label Distribution Protocol) to establish the restoration path. For example, after node 3 detects span failure (3–4), it triggers to establish a restoration path as shown with node 4 and assigns five units for the restoration flow. Similarly, after node 6 is notified with span failure (3–4), it calls

the signaling protocol to establish a two-unit restoration flow with node 2. Here the nodes that trigger the restoration can be either type, not necessarily opaque nodes or source or destination nodes of lightpaths.

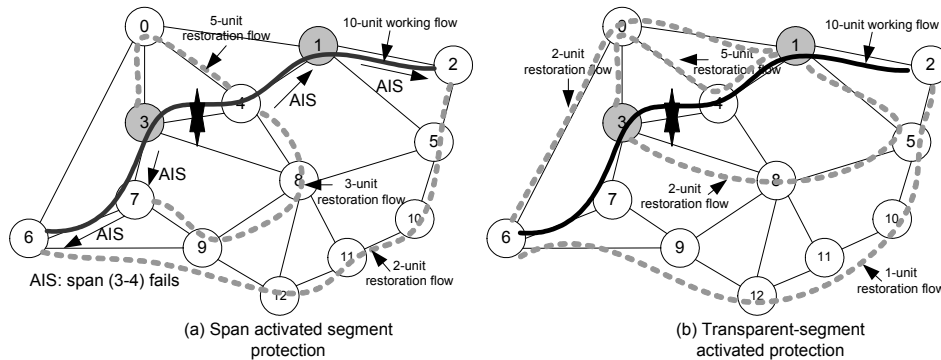


Fig. 5. Two failure-detection strategies and examples of their corresponding restoration processes: (a) each OXC node deployed with span-failure detection systems, (b) opaque nodes to function as channel-failure detectors

Figure 5(b) illustrates the strategy of using opaque nodes only to function as transparent-segment-failure detectors. Under this strategy, only opaque nodes can monitor and detect channel failures. When span (3-4) fails, without an AIS message the opaque nodes 3 and 1 and source and destination nodes 2 and 6 will detect the interruption of the working flow, so they will trigger the restoration processes. A similar set of preplans can be constructed at each opaque node with the information obtained from the optimization models, and signaling protocols such as RSVP-TE and CR-LDP can also be employed to establish restoration flows. There are five- and two-unit restoration flows, respectively, on the restoration segments (1-4-0-3) and (1-5-8-3) between the opaque nodes 1 and 3. There is a two-unit restoration flow on the restoration segment (1-0-6) between nodes 1 and 6 and a one-unit restoration flow between nodes 2 and 6. With segment-based activation there is no need for AIS notification messages because each opaque node and source and destination nodes of a lightpath detect the interruption of a faulty channel directly. And instead of allowing any protection segment between any node pair on the working route to act as in the *span* failure strategy, the opaque node strategy allows restoration-path segments to be deployed only when terminating nodes are opaque nodes or source or destination nodes of lightpaths. This restricts the set of eligible segments used for the optimization design but defines a simpler operational model.

4. Test Methods and Discussion of Results

4.A. Test Methods

We conducted experiments on two test networks: ARPA2 (in Fig. 6) and NSFNET (in Fig. 7). Each network is shown with the Euclidean distances of each span noted. We first identified the required minimum survivable transparent reach (MSTR) for each network. For various ULH transparent reaches that were greater than MSTR, the SHNF algorithm was then applied to identify a set of opaque node locations for the networks. Following that, we applied the PSR and the SBSP models to design translucent survivable networks, and their capacity efficiencies were evaluated. To start with we assumed fully all-optical wavelength-conversion capability at each transparent OXC node. For all the opaque nodes we assumed that they implicitly had the capability of electronic signal regeneration and wavelength conversion. Each node in the networks can be either a transparent OXC or an

opaque node. We used the Euclidean distance of each span (as drawn below) as a measure of its unit capacity cost, C_k . Traffic demands were generated following a uniform random distribution in the range $[1, \dots, 20]$ for each node pair. A single shortest-distance route that complies with a given transparent reach is preestablished for each demand pair, but the design models consider all eligible segments for the segment-based protection approaches. For reference comparison, we also did optimal designs for span restoration (SR) [12], path restoration (PR) without and with stub release [13], and SBPP [27] under the same experimental conditions. It should be noted that in the experiments, we found it was impossible to find a link-disjoint protection path for some node pairs in the ARPA2 network after the shortest working routes were selected. For example, in Fig. 6 after we select the shortest route (8–11–12–13–14–15–17) as the working route for node pair (8,17), it is impossible to find an end-to-end protection route that was link-disjoint from the working route. This is due to a “trap” subgraph in the network [28]. To avoid such situations, we may apply the shortest disjoint path pair algorithm in Ref. [29] to find the working and protection routes. Such a route pair is the most efficient for the 1 + 1 dedicated protection scheme where the spare capacity is not allowed to be shared. However, for the schemes with spare-capacity sharing, we prefer to select the shortest working route as the chief objective because under the spare-capacity sharing, working paths overwhelm protection paths in the capacity consumption. Therefore, to have a better network design efficiency, we searched the distinct route set of each node pair from the shortest to the longest to find the first as well as shortest working route, which had at least one link-disjoint protection route. However, for path restoration, path-segment restoration, and SBSP, we always selected the shortest eligible route as the working route for all the node pairs because none of these approaches have the same difficulty as SBPP in this regard. All the design problems were solved to a mipgap = 0.001 within several minutes with AMPL/CPLEX 7.1 on an Ultrasparc Sun Server at 450 MHz with 4 GB of RAM, except for SBPP and SBSP problems on NSFNET, which needed several hours to terminate at mipgap = 0.01.

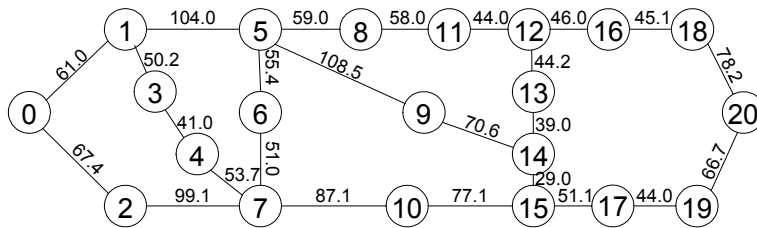


Fig. 6. 21-node ARPA2 network.

4.B. Results and Discussion

4.B.1. Comparison between Schemes without Transparent Reach Limitations

To first compare performance of the various survivable design schemes in a general architectural sense, we start by assuming that the transparent reach of the ULH system is *infinite* and the span failure strategy is employed to detect network failure. Table 1 shows the working- and spare-capacity costs and spare-capacity redundancy of various schemes that result. As expected, span restoration and path restoration with stub release provide the upper and lower bounds of redundancy among all these mesh-oriented schemes. The redundancy of PSR also falls nicely in the gap between path restoration with and without stub release, as expected. However, the redundancy improvement of path-segment restoration relative to path restoration without stub release is marginal. This suggests that although

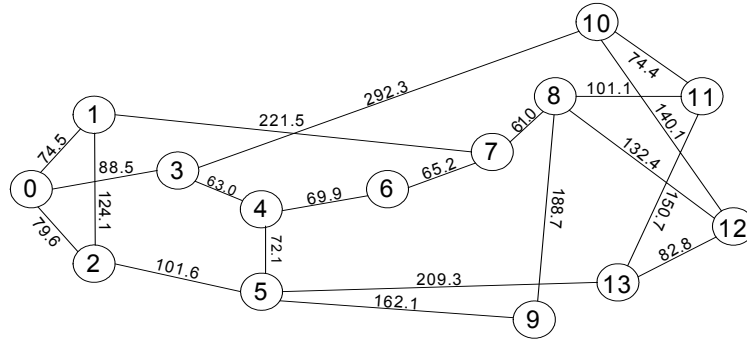


Fig. 7. 14-node NSFNET network.

PSR can reuse some part of the stubs of an affected working path, this is not as efficient as path restoration with true end-to-end stub release and optimal reuse of the stub capacities to establish restoration paths. In addition, when comparing the redundancies of SBPP and SBSP, we see that a similar difference exists between them with SBSP having a less than 1% spare-capacity improvement over SBPP. Note that in the results of ARPA2, we observe that SBPP “abnormally” shows lower redundancies than its peer, SBSP. This normally should not happen, because the previous scheme is just a special case of the latter (if the TR is infinite). The reason behind this is ascribed to the different working routes that had to be used by the different schemes. To satisfy the requirement of link-disjointedness between a working path and a protection path, SBPP has selected some longer routes, instead of the shortest routes as in path restoration and PSR, to function as working routes. Consequently, this results in an increase in the total working-capacity cost and hence a somewhat misleading decrease in the redundancy. Therefore, it actually is not abnormal for ARPA2 to have a redundancy of SBPP lower than that of PSR. (The “standard redundancy” as defined in Ref. [15], p. 50, takes any such differences in working capacity into account. If the standard redundancy is used, the rankings are again as expected.) For NSFNET, there is no such abnormality, because in all the survivability schemes all shortest routes have been used to establish the working routes.

Table 1. Working- and Spare-Capacity Costs and Spare-Capacity Redundancy of Various Survivability Schemes

Network	ARPA2			NSFNET		
	Working	Spare	Redundancy	Working	Spare	Redundancy
SR	443873	514582	1.159	221574	199115	0.899
SBPP	446009	397974	0.892	221574	148189	0.669
SBSP	443873	400591	0.902	221574	146560	0.666
PR without stub release	443873	403303	0.909	221574	145390	0.656
PSR	443873	400582	0.902	221574	145253	0.656
PR with stub release	443873	374629	0.844	221574	121445	0.548

Restoration speed is another measure to evaluate a survivability scheme. To compare the restoration speeds of all the survivability schemes, we use the hop lengths of restoration paths as surrogates to approximate the relative speed of each scheme. This is valid simply for comparisons (not absolute speed predictions) because all the schemes fall into the category of spare-capacity-sharing schemes without preconfigured protection paths. The total

signaling time between the nodes to establish each protection path or segment (to reserve bandwidth, configure switch status, and eventually make the restoration path ready to take the traffic) is thus assumed to be proportional to the number of hops involved. From an optical-networking perspective the lengths of the restoration paths are also of interest in their own right because this relates to the difficulty of the optical path design requirements. Short rerouting is always more desirable if the accumulation of transmission impairments is significant.

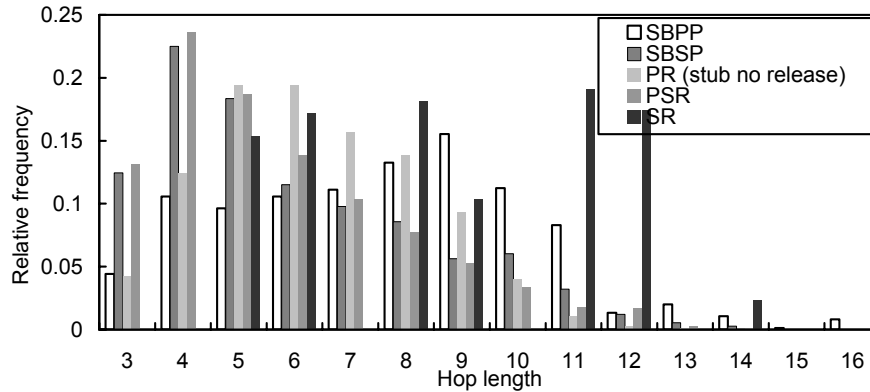


Fig. 8. Restoration-path or path-segment hop-length distribution of different restoration approaches (ARPA2 network).

For ARPA2, we display the distribution of hop lengths of all the restoration paths and/or path segments employed in each design and compute the mean of hop lengths and the 90th percentile level of restoration path or segment lengths. All these measures are over the set of routes used for restoration but are not weighted by the amount of demand that uses each route. In Fig. 8 we see that path-segment-based approaches have more segments at small hop lengths. In particular, both PSR, which has a mean of hop lengths at 5.72 hops and a 90th-percentile level at 9 hops, and SBSP, which has a mean of hop lengths at 5.96 hops and a 90th-percentile level at 10 hops, reach a percentage peak at hop length of 4, and with the increase of hop length, the percentage drops quickly with a tiny tail (a very few restoration segments over 10 hops). In comparison, the path-based approaches reach their peaks at some larger hop lengths. Path restoration without stub release, which has a mean of hop lengths at 6.38 hops and a 90th-percentile level at 9 hops, reaches its peak at hop length of 5 or 6, and SBPP, which has a mean of hop lengths at 7.67 hops and a 90th-percentile level at 11 hops, reaches its peak even at hop length of 9. These results show that in addition to slightly better spare-capacity efficiency, the segment-based approaches can generally restore a failure using shorter reroutes than the path-based approaches. Note that in this comparison, we do not consider the case of path restoration with stub release because an extra stub release requires some additional time. If we ignore it, it is unfair for the other approaches to make such a comparison purely based on the two-way “communication” delay.

An observation that was not expected is that span restoration, with a mean of hop lengths at 8.71 hops and a 90th-percentile level at 12 hops, needs longer restoration routes to restore failures than any other approach. This is attributed to the significant loop-back effect of span restoration in such a sparse topology.

Similar observations can be made for NSFNET from the hop-length distributions without demand weighted as shown in Fig. 9, where the means of hop lengths and the 90th percentile levels are at (4.45, 7), (3.69, 5), (3.59, 5), (3.33, 5), and (5.10, 6) hops, respec-

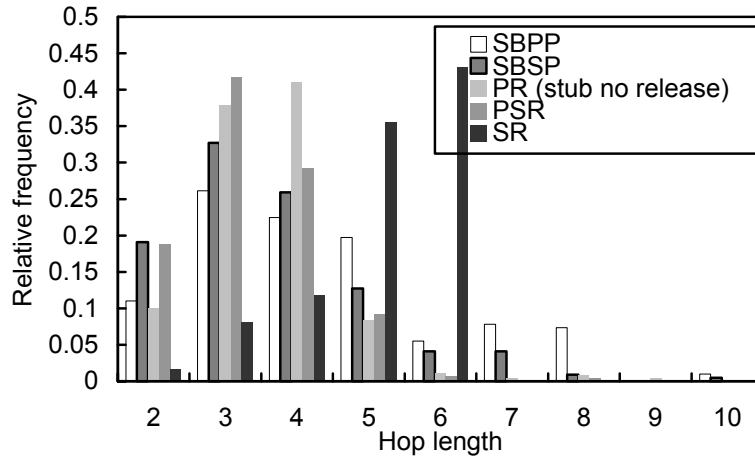


Fig. 9. Restoration-path or path-segment hop-length distribution of different restoration approaches (NSFNET network).

tively, for the schemes of SBPP, SBSP, PR, PSR, and SR. However, comparing the overall performance of ARPA2 and NSFNET, we observe that the segment-based approaches retain shorter paths when the network becomes sparser. The explanation for this is as follows: A sparser network normally has more long working and protection routes, which therefore means in the hop-length distribution charts of the path-based approaches that there are more protection paths gathering at the regions of large hop lengths. However, for the segment-based approaches that restore a failure on a segment basis, the restoration segments can be short even though their corresponding working paths are long. Therefore the segment-based approaches can have distributions with peaks at some small hop lengths, whereas the path-based approaches have distributions with peaks at some large hop lengths.

4.B.2. Opaque Node Placement: Optimal Versus Heuristic

Recall that the previous results all assume $TR = \text{infinity}$. If instead we have a limited transparent reach, we will need to determine the minimum required number of opaque nodes and where the opaque nodes will be placed in the network so as to guarantee full reachability between any node pair in a translucent network. Here we bring this aspect into the results of this study, making use of the SHNF algorithm and comparing it with optimal node sets found by the exhaustive search method. For the NSFNET and ARPA2 networks, we ran the exhaustive search processes to identify optimal opaque node locations for various ULH TR values. These are shown in Tables 2 and 3.

Table 2 gives the resultant opaque node locations obtained by the optimal searching and the SHNF heuristic for NSFNET. The time that the two methods consumed is also compared. For various TR from 189 to 550 [where 189 is the physical length of span (8–9)—the required MSTR to ensure full reachability in NSFNET], we see that the SHNF heuristic finds solutions in subseconds, whereas the exhaustive search takes much longer, especially with the increase of the number of required opaque nodes. Note that for $TR = 189$, because the exhaustive search algorithm could not find an optimal solution when terminated after it had run for one day, we cannot provide a known optimal placement of opaque nodes and the exact searching time for the comparison. Yet by comparing the node locations that can be found by the exhaustive search, we see that the heuristic is very close to the optimal solutions. When $TR = 550$ and 450, SHNF always finds an optimal placement, which falls in the set of optimal combinations identified by exhaustive search.

For situation TR = 350, although the SHNF algorithm needs one more additional opaque node, the solution is still efficient to see that there are two optimal placements whose nodes are all included in the SHNF solution. A similar observation can be made for the situation TR = 250. Table 3, which does not provide all the optimal placements in detail because of the extremely long computation times for the exhaustive searches, lists the opaque node placements by the SHNF heuristic only for ARPA2 under different transparent reaches, where the required MSTR that guarantees the full node-pair reachability is 109, i.e., span (5–9).

We observe that at small TRs the sensitivity to changes in TR is quite high in terms of how it changes the number of nodes required. For example, in NSFNET a TR increasing from TR = 189 to TR = 250 can save three opaque nodes, whereas from TR = 450 to TR = 550, only one node. Similarly, In ARPA2 a TR increasing from 109 to 200 can save six opaque nodes, whereas from TR = 500 to TR = 600, only one node. This is a saturating process. At a low TR many opaque nodes may be required, but some of these nodes may exist only to relay certain special paths (sometimes, a single path). With TR increased, these paths will eventually not need the opaque nodes. An increase in TR from a smaller initial value therefore has more chances to release more opaque nodes, which are specifically used to relay certain sets of special paths. When the TR is larger, the remaining opaque nodes are central and essential enough to be related to so many paths that it is more difficult to release them with a small increase in TR. This suggests that studies of this type can guide ULH technology development. Once the TR reaches a certain level of the network diameter (say, 400 for ARPA2), further development in ULH reach technology may not pay proportional returns in the network.

Table 2. Opaque-Node Placements for the NSFNET Network.

Transparent Limits	Opaque Node Positions		Searching Time (s)	
	Optimal	SHNF	Optimal	SHNF
TR=189	—	(4,7,8,3,11,2,9,5,12,13)	>1 day	<1.0
TR=250	(1,13,11,7,5)...(1,13,8,7,5)...(2,13,8,5,4), (1,13,8,5,4)...(1,13,12,7,5)	(7,5,4,2,1,8,13)	2078.0	<1.0
TR=350	(0,9,8),(2,13,7)...(4,13,7)...(4,13,12)	(7,4,2,13)	48.0	<1.0
TR=450	(0,4),(1,4)...(4,8)...(8,13)	(4,8)	14.0	<1.0
TR=550	(4),(5),(6),(12)	(4)	2.0	<1.0

Note: Optical placement of TR = 189 cannot be obtained within one day.

Table 3. Opaque-Node Placements by the SHNF Algorithm for the ARPA2 Network

Transparent Reach	Opaque Node Positions (SHNF)
TR=109	(7,5,8,12,14,15,1,19,2,4,0,10,20,18)
TR=200	(14,7,5,1,12,19,15,16)
TR=300	(10,7,5,15,12)
TR=400	(7,15)
TR=500	(7,15)
TR=600	(7)

4.B.3. Comparison between the Two Failure Detection Strategies and Transparent Reach Effects

On the basis of the opaque node placements given in Tables 2 and 3, we ran the ILP capacity-design models for the various survivability techniques, which include PR without stub release, PSR, SBSP, and SR with finite transparent reach limits on any transparent

path or path segment. Now, for the working path, the whole set of k -shortest paths (KSPs) between a node pair was found first, and the one that was the shortest and complies with the transparent reach limit in each of its transparent segments was selected to function as the working path. Likewise, for the protection paths or segments, the whole set of such kinds of paths or segments was found first, and only those that comply with the transparent reach limit were kept as eligible candidates.

Table 4 reports the network redundancies of various survivability approaches for NSFNET at various TR values. For SBSP and PSR, we consider the two failure-detection strategies, which are distinguished by (s) and (o) attached after each abbreviation. SR and PSR with span-failure detection strategy [i.e., PSR(s)] upper and lower bound the spare-capacity redundancies pretty well for all the survivability schemes. Although there are some improvements, the redundancy difference between PR and PSR is found to be marginal. In addition, in comparing the two failure-detection strategies, we find that there is almost no redundancy difference (<1%) between them either for SBSP or PSR. This implies that although there is one more constraint on the protection-segment selection required by the opaque node strategy (i.e., only the segments whose terminating nodes are opaque node or source or destination nodes of lightpaths are allowed to recover a span failure), this does not strongly affect the network spare-capacity efficiency.

Table 4. Spare-Capacity Redundancies of Various Survivability Approaches (NSFNET)

Transparent reach	SR	PR	SBSP(o)	SBSP(s)	PSR(o)	PSR(s)
TR=189	1.903	1.470	1.459	1.459	1.459	1.459
TR=250	1.273	0.936	0.936	0.932	0.931	0.931
TR=350	1.198	—	—	—	0.921	0.916
TR=450	1.251	0.810	—	—	0.792	0.792
TR=550	1.014	0.681	0.694	0.686	0.681	0.681
TR=infi.	0.899	0.656	0.666	0.666	0.656	0.656
Opaque network	0.899	0.656	0.666	0.666	0.656	0.656

Note: We do not provide values in the fields with symbol — because we cannot find eligible protection paths or segments for some node pairs.

To identify the effects of different TR capabilities, we evaluated the redundancies for the network under various transparent reaches ranging from TR = 189 to TR = infinity. We see that there is a bigger redundancy improvement gained by each step of TR increasing when TR is smaller. For example, from TR = 189 to TR = 250 with TR increased only 60 units, there are larger redundancy improvements ranged between 63% (SR) and 52% (the segment-based schemes), whereas from TR = 250 to TR = 550 with TR increased 300 units, the redundancy improvements are much smaller, at only ~25% for all the approaches. Therefore, similar to the number of required opaque nodes, it is also a saturating process for the TR increase to improve the network redundancy; after TR exceeds a certain threshold, the redundancy improvement will be marginal. For NSFNET such a threshold falls in the range around TR = 250. In addition, besides giving the data of TR = infinity, Table 4 also provides the redundancies of the opaque network, which makes a logical or correctness check on the results. They should wind up being equivalent and showing the same values.

Apart from spare capacity redundancy, we also compare *average* protection path or segment lengths (APPL or APSL) in hops for various survivability approaches. Like the hop-length distribution of protection routes, the average length also approximately reflects the restoration speed of a survivability scheme. Typically, the average protection path or

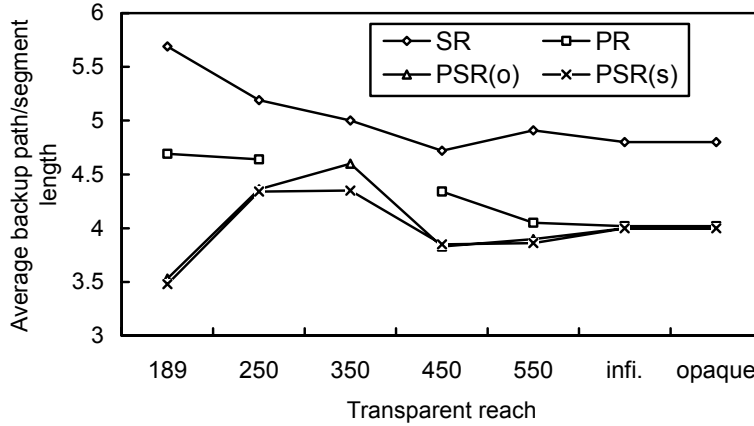


Fig. 10. Average backup path- or segment-length changes with different transparent reaches (NSFNET).

segment length of PSR is defined as follows:

$$APSL = \frac{\sum_{i \in S, r \in D_i, q \in Q_i^r, p \in P_i^{r,q}} f_i^{r,q,p} \left(\sum_{j \in S} \delta_{i,k}^{r,q,p} \right)}{\sum_{i \in S, r \in D_i, q \in Q_i^r} g_i^{r,q} \zeta_i^{r,q}}, \quad (5)$$

where $\sum_{j \in S} \delta_{i,k}^{r,q,p}$ is segment length in hops of flow $f_i^{r,q,p}$, so the numerator is the total segment length-weighted demand units restored in all the span-failure situations, and the denominator is the total affected working demand units of all the span-failure situations. Therefore, the ratio between them can be defined as the average protection segment length. For the rest of the survivability schemes, similar definitions can be made as well.

Figure 10 shows the average protection path or segment lengths of SR, PR, and PSR under various TR ranging from TR = 189 to TR = infi. It is again found that SR needs a longer average protection path. PR has a medium average protection path length, which is much larger than that of the PSR schemes when TR is small, and merge to the latter when TR becomes larger. As the best one, the two PSR schemes have almost the same average protection segment lengths, although in one situation (i.e., TR = 350) PSR(o) has a shorter average length.

From the above results both of redundancy and average protection path or segment length, we see that both *span*-failure strategy and opaque node strategy have a close performance, although the latter is probably more practical, as it uses the implicit channel-monitoring capability of an opaque node to monitor network failure. Therefore, we can conclude that the translucent segment-based restoration approach and the opaque node-based failure-detection strategy provide an economic viable way to design a translucent survivable network.

Experiments at limited TRs were also done for the ARPA2 network. These results are reported in Table 5 and Fig. 11. Similar observations can be made for the ARPA2 network except for two aspects. First, it seems that the saturating process is not so obvious for the SR scheme, although the rest of the schemes all show obvious saturating processes (in terms of the benefits of an even higher TR.) Second, the shortness of restoration-path segments under PSR is more obvious in the ARPA2 network. It is found that PSR always has an average protection segment one hop shorter than that of PR under all the experimental

situations. This can also be ascribed to the sparser connectivity of the ARPA2 network.

Table 5. Spare-Capacity Redundancies of Various Survivability Approaches (ARPA2)

Transparent Reach	SR	PR	SBSP(o)	SBSP(s)	PSR(o)	PSR(s)
TR=109	1.207	1.056	1.052	1.052	1.052	1.052
TR=200	1.194	0.916	0.910	0.910	0.910	0.910
TR=300	1.251	0.923	0.923	0.923	0.923	0.923
TR=400	1.269	0.968	—	—	0.945	0.945
TR=500	1.214	0.919	—	0.914	0.919	0.914
TR=600	1.184	0.909	—	0.903	0.909	0.903
TR=infi.	1.159	0.909	0.903	0.903	0.909	0.903
Opaque network	1.159	0.909	0.903	0.903	0.903	0.903

Note: We do not provide values in the fields with symbol — because some working paths of the situations do not have link-disjoint backup segments.

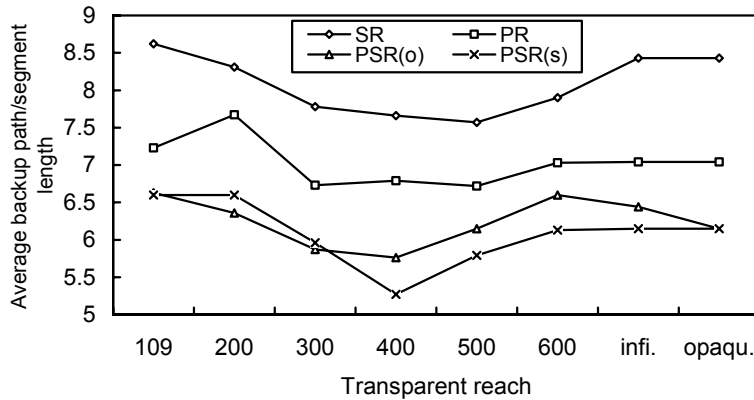


Fig. 11. Average backup path- or segment-length changes with different transparent reaches (ARPA2).

5. Conclusion

In this paper we have proposed a generalized segment-based approach to designing and operating survivable translucent networks and developed the related ILP design models from extension of the conventional models for path restoration and SBPP. The approach has been applied to translucent survivable network design. An opaque node-based failure-detection strategy was compared with a strategy based on span-failure detection at each OXC node. The main finding was that the strategy of detecting failures only at opaque nodes and basing restoration on segments between these nodes has almost the same efficiency as when all nodes have failure-detection ability. In addition, performance comparison between various survivability schemes showed that the segment-based approach is a promising technique to achieve a good spare-capacity efficiency and fast restoration in an architecture that is naturally suited to translucent networks based on sparse opaque-switch placements. Pulling all consideration from this research together, we can offer a prospective overall strategy for a survivable translucent optical network design:

1. Given a TR for the ULH technology, use SHNF to find the opaque node locations.

2. For practicality use the simpler opaque node strategy for failure detection between those nodes.
3. Use PSR design for the protection planning for minimum-capacity restoration, high restoration speed, and shortness of restoration segments.

Moreover, the design methods here can be further employed for a complete study of total network cost, which includes node cost and span-capacity cost, as it depends upon the TR of ULH. Overall, the combination of using SHNF to find optimal opaque node sets as function of TR, plus the use of PSR to design required protection capacity, can lead to a near-minimum total cost design method if a search is implemented step by step over the full range of TR.

References and Links

- [1] I. Chlamtac, A. Ganz, and G. Karmi, "Lightpath communication: an approach to high bandwidth optical WAN's," *IEEE Trans. Commun.* **40**, 1171–1182 (1992).
- [2] K. Lee and V. O. K. Li, "A wavelength-convertible optical network," *J. Lightwave Technol.* **11**, 962–970 (1993).
- [3] S. Subramaniam, M. Azizoglu, and A. K. Somani, "All-optical networks with sparse wavelength conversion," *IEEE/ACM Trans. Netw.* **4**, 544–557 (1996).
- [4] J. Yates, J. Lacey, D. Everitt, and M. Summerfield, "Limited-range wavelength translation in all-optical networks," in *Proceedings of INFOCOM* (IEEE, New York, 1996), pp. 954–961.
- [5] E. Iannone, "Wavelength conversion and routing strategies in opaque optical networks," in *Proceedings of the 11th Annual Meeting of the IEEE Lasers and Electro-Optics Society (LEOS'98)* (IEEE, New York, 1998), Vol. 1, pp. 354–355.
- [6] J. Nagel, "The role of ultra-long transmission systems in nationwide networks," in *Proceedings of the 14th Annual Meeting of the IEEE Lasers and Electro-Optics Society (LEOS'01)* (IEEE, New York, 2001), Vol. 1, pp. 352–353.
- [7] D. Chen, S. Wheeler, D. Nguyen, B. Davis, M. Glavanovic, J. Khaydarov, I. Koruga, S. Hegarty, F. Cokic, and F. Zhu "40 channels 4000 km DWDM ULH transmission field trial without Raman amplification and regeneration," in *Optical Fiber Communication Conference (OFC 2002)*, Vol. 70 of OSA Trends in Optics and Photonics Series (Optical Society of America, Washington, D.C., 2002), pp. FC10-1–FC10-3.
- [8] B. Ramamurthy, D. Datta, H. Feng, J. P. Heritage, and B. Mukherjee, "Impact of transmission impairments on the teletraffic performance of wavelength-routed optical networks," *IEEE J. Lightwave Technol.* **17**, 1713–1723 (1999).
- [9] B. Ramamurthy, D. Datta, H. Feng, J. P. Heritage, and B. Mukherjee, "Transparent vs. opaque vs. translucent wavelength-routed optical networks," in *Optical Fiber Communication Conference (OFC 1999)* (Optical Society of America, Washington, D.C., 1999), Vol. 1, pp. 59–61.
- [10] G. Shen, W. D. Grover, T. H. Cheng, and S. K. Bose, "Sparse placement of electronic switching nodes for low blocking in translucent optical networks," *J. Opt. Netw.* **1**, 424–441 (2002), <http://www.osa-jon.org/abstract.cfm?URI=JON-1-12-424>.
- [11] E. Yetginer and E. Karasan, "Regenerator placement and traffic engineering with restoration in GMPLS networks," *Photonic Netw. Commun.* (to be published).
- [12] M. Herzberg and S. Bye, "An optimal spare-capacity assignment model for survivable network with hop limits," in *Proceedings of GLOBECOM* (IEEE, New York, 1994), pp. 1601–1607.
- [13] R. R. Iraschko, M. H. MacGregor, and W. D. Grover, "Optimal capacity placement for path restoration in STM or ATM mesh-survivable networks," *IEEE/ACM Trans. Netw.* **6**, 325–336 (1998).
- [14] R. R. Iraschko and W. D. Grover, "A highly efficient path-restoration protocol for management of optical network transport integrity," *IEEE J. Sel. Areas Commun.* **18**, 779–793 (2000).
- [15] W. D. Grover, *Mesh-Based Survivable Networks, Options and Strategies for Optical, MPLS, SONET, and ATM Networking* (Prentice Hall, Englewood Cliffs, N.J., 2003).

- [16] X. Yang and B. Ramamurthy, "Sparse regeneration in a translucent WDM optical network," in *Proceedings of the SPIE Asia-Pacific Optical and Wireless Communications Conference* (SPIE, Bellingham, Wash., 2001).
- [17] B. Ramamurthy, S. Yaragorla, and X. Yang, "Translucent optical WDM networks for the next-generation backbone networks," in *Proceedings of GLOBECOM* (IEEE, New York, 2001).
- [18] X. Yang and B. Ramamurthy, "Dynamic routing in translucent WDM optical networks," in *Proceedings of ICC* (IEEE, New York, 2002).
- [19] C. Saradhi and C. Murthy, "Dynamic establishment of segmented protection paths in single and multi-fiber WDM mesh networks," in *Proceedings of OPTICOMM* (SPIE, Bellingham, Wash., 2002), pp. 211–222.
- [20] K. Gummadi, M. Pradeep, and C. Murthy, "An efficient primary-segmented backup scheme for dependent real-time communication in multihop networks," *IEEE/ACM Trans. Netw.* **11**, 81–94 (2003).
- [21] P. Ho and T. M. Hussein, "A framework for service-guaranteed shared protection in WDM mesh networks," *IEEE Commun. Mag.* (February 2002), pp. 97–103.
- [22] C. Ou, H. Zang, and B. Mukherjee, "Sub-path protection for scalability and fast recovery in WDM mesh networks," in *Optical Fiber Communication Conference (OFC 2002)*, in Vol. 70 of OSA Trends in Optics and Photonics Series (Optical Society of America, Washington, D.C., 2002).
- [23] J. Wang, L. Sahasrabudde, and B. Mukherjee, "Path vs. subpath vs. link restoration for fault management in IP-over-WDM networks: performance comparison using GMPLS control signaling," *IEEE Commun. Mag.* (November 2002), pp. 80–87.
- [24] W. D. Grover and D. Stamatelakis, "Cycle-oriented distributed pre-configuration: ring-like speed with mesh-like capacity for self-planning network restoration," in *Proceedings of ICC* (IEEE, New York, 1998), pp. 537–543.
- [25] W. D. Grover and G. Shen, "Extending the p -cycle concept to path-segment protection," in *Proceedings of ICC* (IEEE, New York, 2003), pp. 1314–1319.
- [26] E. Karasan and M. Arisoylu, "Subnetwork partitioning and section restoration in translucent optical networks," in *Proceedings of OPTICOMM* (SPIE, Bellingham, Wash., 2003), pp. 114–125.
- [27] W. D. Grover and J. Doucette, "Design of a meta-mesh of chain sub-networks: enhancing the attractiveness of mesh-restorable WDM networking on low connectivity graphs," *IEEE JSAC Special Issue on WDM-based Network Architectures*, **20**, 47–61 (2002).
- [28] D. A. Dunn, W. D. Grover, and M. H. MacGregor, "A comparison of k -shortest paths and maximum flow methods for network facility restoration," *IEEE J. Sel. Areas Commun.* **12**, 88–99 (1994).
- [29] J. W. Surrballe and R. E. Tarjan, "A quick method for finding shortest pairs of disjoint paths," *IEEE Netw.* **14**, 325–336 (1984).