Failure-Independent Path-Protecting p-Cycles: Efficient and Simple Fully Preconnected Optical-Path Protection

Adil Kodian, *Student Member, IEEE*, and Wayne D. Grover, *Fellow, IEEE*

***Abstract*—We propose a new technique for optical-network protection called failure-independent path-protecting (FIPP) p-cycles. The method is based on an extension of p-cycle concepts to retain the property of full preconnection of protection paths, while adding the property of end-to-end failure-independent path-protection switching against either span or node failures. An issue with applying the popular method of shared-backup path protection (SBPP) to an optical network is that spare channels for the backup path must be cross connected on the fly upon failure. It takes time and signaling to make the required cross connections, but more importantly, until all connections are made, it is not actually known if the backup optical path will have adequate transmission integrity. Thus, speed and optical-path integrity are important reasons to try to have backup paths fully preconnected before failure. With fully preconnected protection, not only can very fast restoration be attained, but the optical-path engineering can also be assured prior to failure. Regular p-cycles are fully preconnected, but are not end-to-end path-protecting structures. SBPP is capacity efficient and failure independent—failures only need to be detected at the end nodes and the end nodes activate and switch over to one predefined backup route for each working path—but the backup paths are not preconnected. FIPP p-cycles support the same failure-independent end-node-activated switching of SBPP, but with the fully preconnected protection-path property of p-cycles. As a fully preconnected and path-oriented scheme, FIPP p-cycles are, therefore, potentially more attractive for optical networks than SBPP. Results confirm that FIPP p-cycle network designs will exhibit capacity efficiency that is characteristic of path-oriented schemes and may be as capacity efficient as SBPP, but more conclusive comparisons on larger scale networks await further study.**

***Index Terms*—Failure-independent path-protecting (FIPP), p-cycles, path protection, preconnection, shared-backup path protection (SBPP), survivable networks.**

I. Introduction

RECENT years have seen considerable research related to the p-cycle concept since its first proposal as a means for fully preconnected and, hence, fast, but also capacity-efficient means of network protection in 1998 [1]. The main features of the p-cycle concept are that it combines the properties of ringlike speed with the capacity efficiency of a span-restorable mesh network. The practical importance of the p-cycle property of providing fully preconnected protection paths in optical networks was given recent emphasis in work motivated towards achieving this property with linear "pre-cross-connected trails" (PXTs) as protection structures [2]. In general, there will be a significant speed advantage if the protection structures are fully pre-cross-connected, compared to a network where protection paths are cross connected in real time on an on-demand basis. This is the main source of the speed that rings have always enjoyed, and it was one of the main motivations for the original development of p-cycles. But in an optical network, especially a transparent or translucent optical network, preconnection can be even more important than being simply an issue of restoration speed, because when optical-protection paths are preconnected, they can be preengineered and tested, and be in a known working condition prior to their use.

Without this prefailure assurance of transmission integrity, it is not merely an issue of speed, but a basic issue of whether protection paths will work or not in the first place. With today's state-of-the-art technology in switched optical networking, it is not realistic to expect that an on-the-fly concatenation of arbitrarily selected spare wavelength channels cross connected between different spans at the optical level will result in an end-to-end path with under 10^{-12} bit error rate (BER). Polarization, dispersion, power levels, amplifier-gain transients, intermodulation, and several other noise and nonlinear impairment processes must all be carefully engineered for a dense wavelength division multiplexing (DWDM) carrier path to achieve objectives for transmission integrity. Freeman [3] details nearly 20 different impairments that have to be mastered simply in designing a point-to-point 10-Gb/s (or above) optical link in a DWDM environment. We may hope eventually to standardize optical-wavelength channel design enough, and develop adaptive power-level schemes and so on, to the point where arbitrary [synchronous optical network (SONET)-like] interconnection of wavelength channels into end-to-end paths is possible on demand, but this will remain difficult or expensive for some time. Without relying on opaque (electronic core) cross connects to, in effect, cross connect the payloads, not the optical channels themselves, it remains a difficult problem to arbitrarily connect several optical channels directly with assurances of immediate end-to-end transmission quality at 10 to 40 Gb/s in a DWDM environment. This is often overlooked in the literature on dynamically switched DWDM optical networks, but we are told by industry colleagues that it is one of the biggest practical stumbling blocks for network operators that would like to consider dynamic mesh restoration or shared-backup path protection using transparent optical cross connects.

Manuscript received December 20, 2004; revised April 25, 2005.

The authors are with the Department of Electrical and Computer Engineering, Network System Research, TR Labs, Edmonton, AB T6G 2V4, Canada.

Digital Object Identifier 10.1109/JLT.2005.855697

Therefore, full preconnection of protection paths is a very important property, because the paths can be engineered and tested before failure, and will then be known to work with certainty when accessed for restoration.

The main contribution of this paper is to propose the concept of failure-independent path-protecting (FIPP) p-cycles, as an extension of the basic p-cycle concept into a path-oriented scheme. The most important properties of FIPP p-cycles appear to be that they remain competitive or at least comparable in capacity efficiency with shared-backup path protection (SBPP), and like SBPP provide simple failure-independent end-node activation and control against either span or node failure, but they employ only fully preconnected protection paths. This is the key to both ringlike speed, minimal real-time signaling, and the assurance of optical signal quality on the protection path. To explain FIPP p-cycles, let us first review some relevant background.

II. Background and Literature

A. Path Protection, Path Restoration, and the "Mutual Capacity" Issue

In path-oriented protection or restoration in general, the objective is to provide (or to form on demand) a surviving backup path for each affected working path, in the event of any single span or node failure that disrupts the path. Path-oriented survivability schemes are, generally, more capacity efficient than their span-based counterparts and end-to-end rerouting gives customers control on activating the backup path(s) for their affected services. Although adaptations or extensions of span-protecting schemes have been developed to cope with node failure, as well as span failure (see [9]–[11], for example), path-based schemes have a more inherent ability to react against either a span or node failure. On the other hand, most path-oriented survivability schemes are slower and more complex than span-oriented schemes. Notably, in the sense that the word "protection," as it originated in automatic protection switching (APS) and ring applications, really implies the existence of a fully pre-cross-connected bidirectional backup path between end nodes, it seems to us that no shared-mesh path-oriented protection scheme has, strictly, yet been developed. By this, we mean a scheme that has the following properties:

1) is capable of being designed for single failure protection of 100% of the network demands, if desired;
2) has spare capacity requirements that are very much less than 100% redundancy—close to the theoretical minimum of spare capacity set by optimal multicommodity maximum-flow (MCMF) solutions for restoration;
3) provides strictly and simply pre-cross-connected bidirectional protection paths between end nodes;
4) requires only end-node failure detection (no immediate fault localization is needed); and
5) needs only the end nodes to do any protection switching.

Other factors aside when comparing schemes, it is also generally agreed that a preferred protection scheme will always be one that requires less network state information and state-update dissemination (for robustness) and fewer logical constructs to create, change, and monitor to sustain network-survivability goals (for op-ex reduction), especially as demand patterns evolve. In considering the background literature, first, of course, we recognize that diverse dedicated $1 + 1$ APS has a protection path that goes end to end and is fully pre-connected. But the cost of this is at least 100% redundancy. There is no "shared mesh" aspect of efficiencies from reusing protection capacities over multiple different failure scenarios. In contrast, SBPP [16] has very good capacity efficiency, and is end-to-end oriented, but it is not a protection scheme (in the sense above). It is really a preplanned restoration scheme, which has no aspect of backup-path pre-cross-connection. The routes of backup paths are decided in advance, but a path must be formed on demand by seizing and cross connecting spare channels on that route when needed. More precisely, we would say that SBPP is a failure-independent preplanned path-restoration (PR) scheme. However, SBPP does stand as the most closely related and relevant scheme against which FIPP p-cycles are compared, and we will be returning to consider it in further detail.

Another scheme that needs to be considered under the criteria above is "demandwise" shared protection [29]. The concept (in its most general form) is to split the total bundle of demand between any two end nodes over multiple mutually disjoint routes of the graph and also use those disjoint routes to support an end-to-end allocation of protection paths for each working route. Backup paths assigned to protect one working route may be co-routed with the other working routes, or separate. For instance, if three mutually disjoint routes exist between nodes A and B, and the total demand between A and B is 11 lightpaths, then the best solution is to assign $w_i = \{0, 6, 5\}$ working paths, respectively, and $s_i = \{6, 0, 0\}$, $i = 1, 2, 3$ spare paths. In this example, the pathwise logical redundancy would be $6/(11) = 54\%$. The true ratio of total capacity used above that required, simply, for the shortest path routing of demands (the "standard redundancy")[1] would be higher than this, however, because, in general, the disjoint paths are not the shortest paths. The overall-capacity efficiency is, thus, limited by the fact that sets of mutually disjoint routes tend necessarily to include routes that are much longer than the shortest route, and by the fact that the minimum edge cut between most node pairs in typical transport networks is only two or three. Between any two nodes with a min-cut of two edges, 100% is the best redundancy achievable. With a min-cut of three, it is, logically, 66%, but total capacity cost will not reflect this full benefit, because of the excess routing lengths of the three disjoint routes involved. Ultimately, these effects result in there being only 2.6% and 8.7% capacity-cost reduction relative to $1 + 1$ APS in the test-case results for protection of all demands in [29]. Therefore, while the "demandwise" shared-protection scheme

[1]Standard capacity redundancy is the ratio of total protection capacity-distance plus extra working capacity-distance over shortest paths to the working capacity needed only for a shortest path realization of the working paths alone. This avoids the degeneracy of measuring redundancy when working paths are themselves made longer than shortest paths. Otherwise, by increasing working path lengths, one can apparently reduce redundancy without reducing absolute spare capacity (see [13, pp. 47–49]).

would have the other desired properties, it does not have the characteristically high capacity efficiency that characterizes a "shared mesh" solution.[2]

Finally, in searching the literature for schemes that might combine all of the key properties above, we also found the approach of backup light trees, as developed in [30] and [31]. The basic idea involves adoption of a unidirectional tree-forming approach in the "incoming" direction on each node, as a destination node. Each individual tree is preconfigured out of a single channel of protection capacity and arranged so that some (not all) of the lightpaths incident on that node can enjoy a disjoint route from their origin to the respective destination node through the particular backup tree for that node in that direction. Multiple distinct unit-capacity backup trees are defined on each destination node until all demand flows in that direction to that node are made single-failure restorable. There are several issues when lined up for comparison to the criteria above. One is that the backup trees are preconfigured but not pre-cross-connected, in the hard and simple sense that preconnection is unusually meant, such as in rings, p-cycles, or $1 + 1$ APS. Rather, specific assumptions about incoming-signal mutual exclusion, valid signal recognition, and signal selection at merge nodes on the tree, and at the destination node, are required to support the use of the backup-tree concept on nodes where preconnection of degree > 2 is required in the trees. The arrangements for protection are also not bidirectional. Each direction of each working demand will wind up taking a different restored-status route over the network. In addition, capacity efficiency is, evidently, also not comparable to that of SBPP, or even span restoration (SR), as indicative of what is expected from mesh sharing of spare capacity: [31] reports that "results indicate the capacity savings due to (backup trees, instead of $1 + 1$ APS) are about 15% for networks with reasonably high degree (about 3.5) and large demand matrices suitable to the topology." In contrast, it is already well known to expect that under "shared mesh" architectures, such as SBPP, or even an SR on a degree-3.5 network, we can conservatively insist achieving something like 50% absolute standard redundancy or less (15% better than dedicated $1 + 1$ APS is still a very much higher value than this level and will often likely be still over 100%.)[3]

[2]It might be pointed out that in [29], larger savings, up to 28% relative to $1 + 1$ APS, are achieved if only two-thirds of the network demands are to be protected. But simple omission of any desired fraction of total demand from the protected demand is also an option in the present scheme. More generally if only two-thirds of each demand bundle is to be protected, we can expect a direct 33% reduction in total protection costs in this scheme.

[3]An additional attribute of practical importance to operational manageability is the number of distinct logical protection structures that are required to exist in a fully protected network. With backup trees, every node has a set of unit capacity trees extending out to other nodes. There will be one distinct tree for some characteristic number of arriving lightpaths, say five or six per tree (this extent of sharing the backup structures is generously high). It seems reasonable, therefore, that each node may typically have several tens of individual backup trees incident on them. In comparison, we tend repeatedly to find in the results with FIPP p-cycles that entire networks are protected with significantly fewer distinct cycles than the number of nodes in the network. Demand volume and pattern affect the capacitation of these individual protection structures, but not the absolute number of different structures needed directly. In contrast, based on principles of how the backup trees are formed, the number of individually distinct backup trees cannot be less than a multiple of the number of nodes.

Failure independence is one of the practically most attractive and simplifying properties that both FIPP p-cycles and SBPP share and will also be discussed further. Protection trees or backup trees [6], [8] can be fully preconnected structures, but do not operate on an end-to-end basis as in [30] and [31]. They act as span-protection technologies in [6] and [8]. In the method of [8], there always remains a requirement for on-demand restoration path assembly to address failures on spans on the single backup tree itself. Reference [8] even mentions that because of rules prohibiting nodes from routing protection through one of its own child nodes, the scheme is also not always able to support 100% restorability. The earlier backup-tree scheme in [6] has no such limitation on restorability, and has good efficiency even with 100% fully preconnected restorability, but it does operate on a span-protection basis. Other schemes, such as oriented cycle covers, PR, and so on (see [13] for a comprehensive survey of survivability techniques), all omit at least one of the three main properties we now seek. To reiterate these properties are as follows:

1) capacity efficiencies well under 100% and characteristic of optimized SBPP (or path restorable) network designs;
2) path-oriented failure-independent end-node activation and control in response to either span or node failures;
3) completely preconnected end-to-end protection paths.

The most efficient scheme that is theoretically possible for survivability (without global rearrangement of unaffected paths) is failure-specific PR, as considered in [12]. Here, after the failure, a centrally controlled, or distributed adaptive process produces a coordinated set of restoration paths that restore each affected path on all affected end-node pairs in an efficient way. In this type of restoration, the composite set of restoration paths deployed considers the exact location of the failure and is allowed to reuse any working capacity on the failed paths up to and leading away from the failure location. This is referred to as "stub release." It makes PR inherently more efficient than SBPP, but also not "failure-independent" in its response, because failure localization is essential for stub release. Such failure-specific PR is, however, the most capacity-efficient form of restoration possible when the restoration path sets are computed by (or are equivalent to) a capacitated MCMF-type of optimized-routing model. However, to implement such a scheme assumes real-time dissemination of the failure information, and, therefore, is more complex than schemes for SR or for failure-independent-path end-node-controlled schemes such as SBPP. As such true PR tends often to be used only in theoretical studies in estimating the lower bounds of spare-capacity requirement for any possible restoration scheme.

From a theoretical standpoint, the central issue that makes any approach to PR (or protection) more complex than for SR is the coupling of multicommodity routing decisions under the finite spare capacities available on spans. This leads to so-called "mutual capacity" constraints, under which the detailed routing decisions for each restoration path for each affected end-node pair are all coupled, under a capacity constraint on each network span. This theoretically complicating central issue in path-oriented survivability (with 100% restorability

guarantees) is discussed at length in [12]–[14]. For present purposes, we can summarize the issue with an excerpt from [13]:

> "Consider a number of simultaneously affected origin-destination (O-D) pairs and imagine for argument's sake that they each took a turn in sequence to obtain restoration paths. One O-D pair may have 20 equivalently good route choices that will restore it, but for another O-D pair there may be only one particular route on which it can obtain restoration paths. What if the first O-D pair chooses a route that uses up the spare capacity on the only possible route for the second O-D pair? More generally, how can the exact route choices made by each of a multitude of O-D pairs be coordinated so that the spare capacity that is crucial for restoration of one O-D pair is not blindly used up by another, which may have had different routing options in any case. This is called the mutual capacity problem in reference to the central quandary of this situation: to which pairs should we allocate the spare capacity of a span among the many that try to seize it?" ([13, p. 383])

The relevance here is that all path-oriented schemes must have some way of directly or indirectly addressing the central issue of mutual-capacity coordination. However, one place where the issue is ignored is under recent proposals for generalized multiprotocol label switching (GMPLS)-based independent end-node "mass redial" as a restoration mechanism [14]. The price for ignoring the mutual-capacity issue is that any such restoration scheme is then inherently only a "best efforts" approach. There can be no assured outcomes or protection guarantees given by design. For any assured outcome from a PR process, the mutual-capacity issue must be addressed in one way or another. It is therefore important to the present work to understand how schemes other than GMPLS mass redial [14] do address the issue. SBPP addresses the mutual-capacity issue indirectly and elegantly in a prefailure way by permitting only working paths that are mutually disjoint [more specifically, have no shared-risk link groups (SRLG) in common] to share spare channels on other spans that are in their respective backup routes. Failure-specific PR addresses the problem more directly, but with considerable added complexity, through the solution of a capacity-optimization problem during the design phase that involves MCMF-type routing subproblems within its overall structure and the use of online MCMF routing decisions in real time or an equivalent distributed adaptive-restoration algorithm that senses conflicts in capacity allocation among paths for different end-node pairs being simultaneously restored [15]. In both cases, however, the mutual-capacity issue leads to considerable complexities for path-oriented survivability schemes, either in the advance planning (SBPP) or in the real-time phases (MCMF-like dynamic restoration). With the aim of being path-oriented, FIPP p-cycles will ultimately also have to deal with the mutual-capacity issue. One prior attempt to extend p-cycles to a path-oriented form [7] lead to segment-protecting p-cycles (reviewed later) that ultimately coped with the issue by being failure specific in response and adopting an MCMF-type resolution of the mutual-capacity allocation problem during design time. In contrast, the key to FIPP p-cycles will be to use an SBPP-like disjointness restriction on the working paths that can share spare channels in their protection paths. But unlike SBPP, we will retain the fully preconnected path-protected aspect of p-cycles.

B. Shared Backup Path Protection (SBPP)

As mentioned, SBPP is a preplanned PR scheme primarily developed by the Internet Engineering Task Force (IETF) [16] for use under Internet-style signaling protocols for protection of lightpaths in optical networks. SBPP is, however, logically identical to the asynchronous transfer mode (ATM) backup virtual-path (VP) scheme that preceded it [17]. Under SBPP, one backup route is predefined for each working path and no matter what fails on the working path, restoration is via a path assembled on demand over this one predetermined backup route. Conceptually, SBPP is like $1 + 1$ APS where two fully disjoint routes, a working and a backup, are established for each signal; but for efficiency in the use of spare capacity, we can share spare channels over the backup routes for different working paths. For this reason SBPP is also sometimes described as 1:1 APS with "backup multiplexing." The working paths are usually called "primary" paths. Usually, one or more backup routes is possible between the same end nodes of the primary path, but only one is chosen for the final design. To be eligible as a backup route, a route has to have no nodes or spans in common with the route of the primary path itself and no spans or nodes in common with any other primary path whose backup route has any spans in common with the route being considered. Together, these considerations ensure that when a primary path fails (under any single failure scenario) no span or node along its backup route is simultaneously affected. This means it will be possible to assemble a backup path along that route if sufficient spare channels have been preplanned. Fig. 1 illustrates the concept of spare-capacity sharing on predefined backup routes under SBPP.

Fig. 1(a) shows a sample network context. Fig. 1(b) shows a set of four mutually disjoint working routes between node pairs A–C, E–G, L–G, L–H. Fig. 1(c) shows possible routes of the disjoint protection paths for the four primaries in (a). For example, demand A–C follows the working route A–B–C for which the corresponding protection route is along A–D–O–G–C. Since these working routes are all mutually disjoint, spare capacity can be shared on their backup routes. For example, the A–D–O–G–C and E–D–O–G share the D–O–G segment and, therefore, as shown in Fig. 1(d), we require only one channel of spare capacity on D–O–G per working unit on working routes A–B–C and E–P–G. The gray shaded areas in Fig. 1(d) indicate the three spans, DO, OG, and NO, where sharing is possible. Of these, OG achieves maximal sharing with four separate working routes sharing a single unit of spare capacity (per-unit working capacity) along the backup route. Note in Fig. 1 that it is individual spare channels that SBPP is organized to share, and that it is not possible to have these channels cross connected in advance of failure because it is not known which of the specific backup paths in Fig. 1(c) might be needed until failure actually occurs. Ultimately, it is because SBPP sharing is structured on a per-channel basis, over groups of mutually disjoint primaries, and SBPP requires cross connection in real

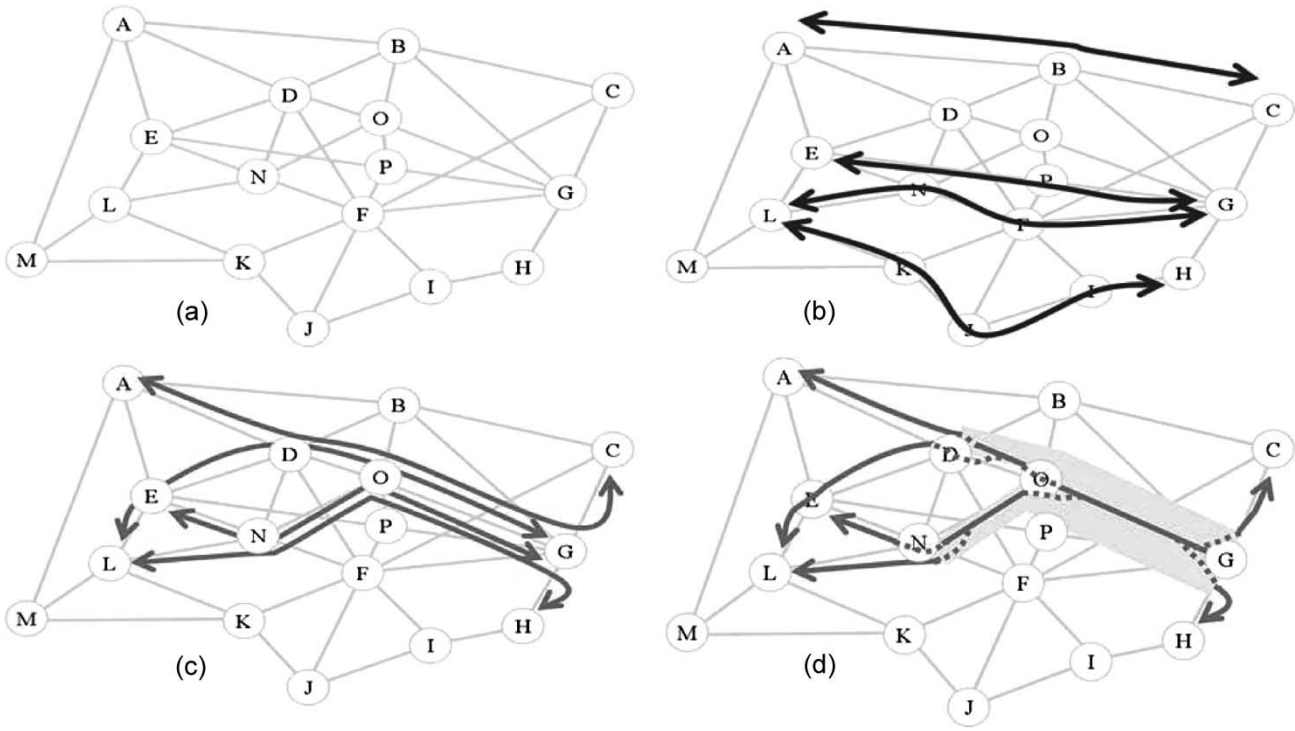

Fig. 1. SBPP principle—illustrating a set of four working routes that can share spare capacity.

time to form actual restoration paths. (A key point to follow will be that under FIPP p-cycles, such corresponding sets of primaries share entire preconnected structures, not individual channels.)

Optimization models for SBPP design are available in [18], and developed in more depth in [13]. More often, however, heuristic methods are used for SBPP network design such as, for example, [20] and [21]. The emphasis on heuristics for SBPP is partly because of the difficulty of solving the optimal SBPP design model even when the complete set of demands is given at once. And heuristics are also better suited to incremental survivable routing that is a strong practical orientation for the SBPP approach. "Incremental" in this context refers to the problem of routing new demands individually as they arrive and arranging the shared-capacity backup path for each arrival in the context of all other already-present demands and backup paths. Variations on this type of heuristic can either assume given capacities and try to perform survivable routing so as to minimize blocking, or, to treat each of a set of demands in sequence, attempting to route each one so that the least additional new capacity has to be added and, thereby, approximate an optimal solution for collective routing and minimum total spare-capacity placement (SCP).

The key ideas for routing under SBPP are that one tries to route the working path over the shortest or least-cost path over the graph while also considering the possible backup routes (disjoint from the chosen working path) and their potential for sharing of spare capacity. On an incremental-arrival basis, the optimal incremental-SBPP setup is one where the sum of the capacity used for the working path plus new spare channels that had to be provided to allow for the required backup path is minimal. It is because the backup route is fully disjoint from the route of the corresponding primary path that the switching action is "failure independent." When anything goes wrong on a primary path, its end nodes switch over to the predefined backup route. It does not matter what failure occurred or where it occurred along the path, or whether it was a span or node failure. The existence of a failure is always immediately observable at the end nodes and the activation and switch over to a path on the one preplanned backup route always follows. Under this scheme, it is also possible for the users to know ahead of time exactly where their service will be rerouted in a failure. An especially important advantage of this is that fault location is not necessary in real time to determine the restoration response. Fault detection still happens in real time, at the end nodes, but the action to take does not then depend on the localization of the actual failure. [4] This property, that FIPP p-cycles inherit, is called "failure independence" and its main advantage is in transparent or translucent optical networks, where fault location is slow or difficult. On the other hand, SBPP has some drawbacks. One of these is the extensive database dependencies, discussed in [22] and [23], arising

[4]Fault localization refers to the determination of which span or node actually was the cause of the path failure. This information is needed as part of a failure-specific response, such as for path restoration [12] or flow p-cycles [7]. In a transparent optical network, this information is, however, not necessarily available quickly because the payload signal is not accessed electronically at every intermediate node to inspect its overhead bytes. In addition, transparent optical cross connects typically do not generate "keep alive" optical-output signals, so that when an outright loss of light occurs, this alarm goes off at all downstream nodes. A centralized interrogation of alarm data may then be needed to find the node nearest to the actual physical source of the failure. Failure independence of the protection reaction is, therefore, desirable, as it requires only end-node failure detection, not failure localization, to be able to act.

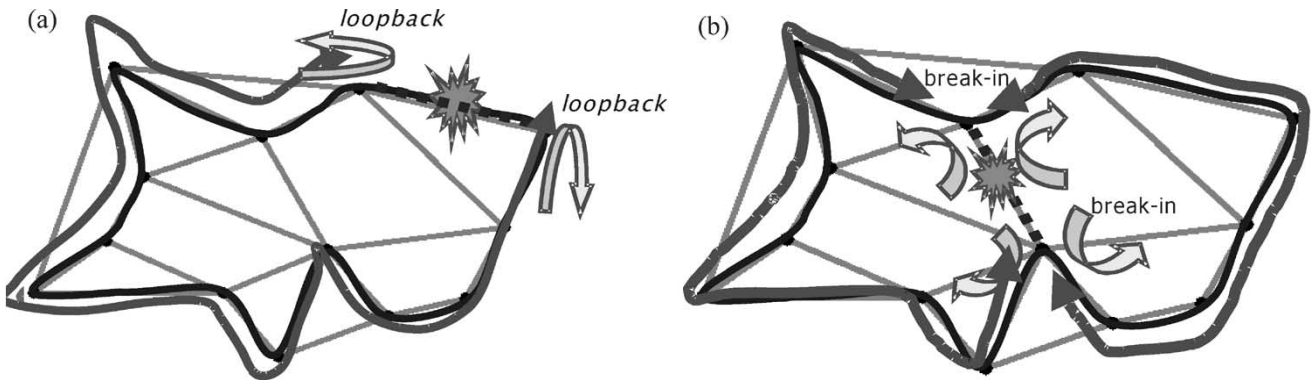

Fig. 2. Basic concept of p-cycles (a) BLSR-like action for on-cycle span failure, (b) mesh-like action for protection of "straddling" span failure.

from the need in every node for global capacity, topology, and backup-sharing relationships to support dynamic provisioning with SBPP. In addition, SBPP depends on real-time assembly of an actual backup path, which implies signaling, length dependence of the restoration time,[5] and as argued above, uncertainty about optical-path integrity for the dynamically assembled backup path.

C. p-Cycles

p-Cycles are a fairly new basic approach for transport network protection. p-Cycles are bidirectional line switched ring (BLSR) ringlike in structure and switching characteristics, but achieve meshlike spare-capacity efficiencies [1], [4]. They share the ringlike characteristic of protecting against on-cycle span failures through BLSR-like switching at the end nodes, but also protect against straddling span failures. p-Cycles provide a fully preplanned and pre-cross-connected span-protection mechanism which is essentially the same as in a BLSR ring. When a span failure occurs, the end nodes of a failed span have completely prearmed switching reactions. The protection of straddling-span failures is a unique property of p-cycles that enable networks to be designed with essentially the same capacity efficiency as a span-restorable mesh network. This allows p-cycle-based networks to be, essentially, as capacity efficient as span-restorable mesh networks, while still providing BLSR switching speeds. Overall efficiency relative to ring-based networks is even greater than what the simple ratio of protection to working capacity would suggest, because p-cycles are "spare-capacity only" structures that, unlike rings, do not constrain the routing of working paths to coincide with the layout of the cycles themselves. Working paths are free to take the shortest meshlike routes between their endpoints on the graph. In fact, it is more advantageous for working capacity to "straddle" p-cycles than to underlie the p-cycles themselves, because, in this case, each unit of protection capacity on a p-cycle can protect two units of working capacity on a straddling span.

Fig. 2 illustrates the operation of p-cycles. A single unit-capacity p-cycle is a closed path composed of one spare channel on each span around its cycle. When a failure occurs on a span covered by the cycle, the p-cycle provides one protection path for the failed span, as shown on Fig. 2(a). In this aspect, p-cycles operate like a unit-capacity BLSR. But p-cycles also protect so-called "straddling spans"—spans that have end nodes on the cycle, but are not themselves on the cycle, as shown in Fig. 2. Significantly, because the p-cycle itself remains intact when a straddling span fails, it provides two protection paths for each straddling-span failure scenario, and straddling spans themselves require no spare capacity. This apparently minor difference actually has a great impact on the capacity requirements of p-cycle protection. When optimized for spare capacity, p-cycle network designs are often exactly as efficient (or within a few percent) as a similarly optimized span-restorable mesh network. In fact, it has been shown theoretically that p-cycles are the most efficient preconfigured individual pattern possible for network protection [5], [6]. If hosted on optical cross connects (OXCs), they have the added advantage that they can be created and updated on a unit-capacity basis, without the inherent modularity present in rings. These combined attributes have made p-cycles an option that is of considerable recent interest and study.

Since first described in 1998 [1], there have been studies on the self-organization of p-cycle sets [4], development of an add drop multiplexer (ADM)-like nodal device for p-cycle networking, application of p-cycles to the MPLS/Internet Protocol (IP) layer [11], application to DWDM networking [25], proposing the p-cycle-protected working-capacity envelope (PWCE) [22], [23] and studies on joint optimization of working paths and spare capacity [26], applications to ring-network evolution that describes the use of an ADM-like device for implementing p-cycles in ring networks [24]. This creates the option to implement p-cycle-based span protection without relying on cross connect systems via a "capacity slice" nodal-equipment architecture with most of the desirable cost and "pay as you grow" characteristics of BLSR-ring ADMs. p-Cycles have also been considered as MPLS-layer protection structures, including protection of transiting flows against node failure, via node-encircling p-cycles (NEPCs) [11], [13].

[5]Each additional hop in the path implies additional signaling to seize and cross connect the respective shared spare channel, including checks that it are not already in use. A preconnected protection path has no such delays. The only delays are the time to switch the failed signal path over into the protection structure at the end nodes, followed by physical propagation and reframing times. However, the latter are common to any possible protection scheme other than dedicated $1 + 1$ APS.

More closely related to the present effort was past work on path-segment protecting p-cycles [7]. That work, loosely called "flow p-cycles," was the first extension of the p-cycles concept toward a path orientation. It significantly extended the ability of p-cycle methods to include node-failure protection (aside from the separate method of NEPCs), and it also gave a significant further increase in spare-capacity efficiency over regular span-protecting p-cycles. The key idea in flow p-cycles is to address the mutual capacity constraints by relaxing the end-to-end protection requirement to allowing protection of arbitrarily defined path segments. This improves on the spare-capacity efficiency over span p-cycles, but the operational aspects are complicated by failure specificity and the simplicity of strict end-to-end switchover to a predefined backup path is not achieved. The main complexity of the effort on flow p-cycles remained the struggle with the "mutual capacity" issue. As a result, flow p-cycles do not obtain the failure-independence property, nor do they incorporate the end-to-end path-switching properties that are the current goal.

Another recent development to which FIPP p-cycles will be relevant is the concept of a PWCE for simplified dynamic provisioning of protected services [22], [23]. Under the PWCE concept, one provisions service paths over inherently protected capacity, as opposed to explicitly provisioning protection for every service. When we route the service through the available channels of a PWCE, it is inherently protected. Provisioning protected services through the envelope looks the same as point-to-point routing over a nonprotected network. One does not have to make any explicit arrangements for the protection of every individual path or globally update the network state for every individual path setup (or takedown). Under PWCE, as developed for span-based survivability schemes, once the graph and the vector of spare-channel quantities on the network spans are given, there is a unique maximum number of protected working channels available on each span. Thus, a given distribution of spare capacity on a graph creates a uniquely determinable envelope of protected working capacity on each span. Within this operational envelope, a vast number of simultaneous service-path combinations are feasible and all are inherently protected. Provisioning of a new protected service path is, then, only a matter of routing over the shortest path through the envelope using only spans that currently have one or more protected working channels available. FIPP p-cycles lend themselves to PWCE-type operation as well, but in an even more desirable way, because entire routes between O–D node pairs will become predefined structurally protected entities.

D. Prior Work Towards the Property of Pre-Cross Connection

In the recent paper on PXTs [2], it was observed that (paraphrasing) "p-cycles are fast not because they are cycles, but because the protection paths they provide are fully preconnected before failure." This correctly reiterates one of the original aims of the work in [1] and [6] and adds a renewed emphasis on preconnection as a paramount property of interest in an optical network. In fact, it is the efficiency of p-cycles that is derived from their cyclic nature [1], [4]. Their speed arises from the fully preconnected property. Pre-cross-connected linear path segments or trails were initially studied in [4]–[6], where it became understood that cycles were inherently more capacity efficient than any acyclic protection structure because they can provide up to twice the number of protection relationships per unit of spare capacity. An important difference in motivation, however, between [6] and the more recent work [2], was that the latter sought a path-oriented activation model, whereas the previously developed method for PXTs used the preconnected trails by breaking into them as needed to effect SR. The basic concept in [2] is the same as in [6], except that the intent is to break into the PXTs that are present to replace failed working paths on an end-to-end basis. The preconnection property is also a primary motivation in recent work on preconnected trees as protection structures [8]. Although [8] is predated by Stamatelakis [27] and the technical report in [6], which extensively studied linear segments, trees, cycles, and even arbitrary mixtures of preconnected structures of spare capacity, it, too, shows renewed emphasis on trying to achieve a fully preconnected protection scheme. Regarding trees, however, one of the clearest findings of [27] and [6] was that cycles are both theoretically and experimentally more inherently efficient as protection structures than any acyclic structure can be.[6] The basic reason is an effect that very strongly separates preconnected cycles from any acyclic protection structure once a structure is closed on itself, that it can provide two protection paths per unit of its own capacity for each failure that straddles the cycle, and one for each failure on the cycle. In contrast, any acyclic structure can provide at most one path for a failure that is off the structure itself and none for failures that are on the structure itself. Thus, preconnection itself is not a new topic. It was extensively studied for linear segments, trees, and arbitrary patterns, including cycles as far back as 1997, and before in work such as [3], [5], and [6]. But the renewed general interest reflected by [2] and [8] in placing fully preconnected structures of spare capacity for network protection has highlighted the importance and desirability of possibly extending the preconnected nature of p-cycles to a truly end-to-end-protection switching mechanism, and not one that is span- or segment-based.

III. FIPP p-CYCLES

A. FIPP p-Cycle Concept

Although it has eluded ourselves and others until recently, it has occurred to us that there is one potentially very simple principle through which ordinary span-protecting p-cycles can be extended to provide an end-to-end path-protection technique without floundering on the complexity of the mutual capacity and failure-specificity issues, as (in hindsight) we did in [7]. The key is not to try to find the p-cycle equivalent to failure specific PR per-se, but to ask instead what is

[6] An acyclic structure within a graph contains no closed path. Both trees and trails are acyclic. Trails, as in [2], may exist with loops in their route, but remain as acyclic structures, because they are not preconnected at such looping points along their route. When unraveled, the trail is always a single linear segment with no contained cycles.

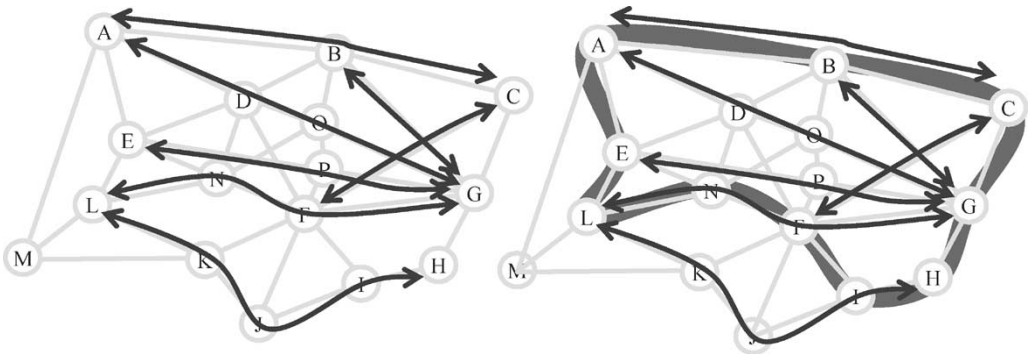

Fig. 3. Example of a FIPP p-cycle protecting a set of seven mutually disjoint primaries. The whole preconnected protection structure is shared intact.

the p-cycle equivalent of a failure-independent path-protection scheme such as SBPP. Implicit in this is the willingness to settle for less than the best possible capacity efficiency, by adopting a failure-independent protection reaction. Once this path of thinking is followed, it leads to realization that the key to failure-independent path-oriented p-cycles is just to enforce an *a priori* disjointness requirement on the end-to-end paths that share any p-cycle structure, just as SBPP enforces this on any primaries that share protection channels. Thus, the key principle is: Let the cycles act as p-cycles for end-to-end paths between nodes on the cycle, but only allow each cycle to provide protection relationships to a group of paths whose routes are all mutually disjoint.

To illustrate, let us return to the network backdrop from Fig. 1(a) and show how a certain set of working routes can be arranged to share a single FIPP p-cycle for their end-to-end protection, without any dependence on failure location. Fig. 1(b)–(d) showed how a certain set of four primary paths that are disjoint from each other can share spare channels on several spans to help form any of their backup paths. Fig. 3(a) shows an augmented set of end-node pairs (seven in total, for the example) and routes between them, which includes the same subset of four demands from Fig. 1, all of which can share the same FIPP p-cycle on their end nodes.

The important property that allows these seven working routes to be protected by one p-cycle in common is that they are all mutually disjoint. This is what leads to failure-independent end-to-end path protection via fully preconnected structures of spare capacity. In a sense, this draws directly from the key property underlying SBPP that in order to share spare channels on a span, all participating primaries must be mutually disjoint. What is different and quite significant, is that by defining a p-cycle with respect to a group of mutually disjoint primaries, those primary paths are all enabled to share a fully preconnected protection structure, not individual spare channels that still have to be cross connected to form backup paths. Note also in the example that some of the routes in the group of disjoint routes selected for the example fully straddle the respective FIPP p-cycle. These are the routes between end nodes B–G, A–G, and E–G. On these routes, it will be possible to protect two working paths per unit of capacity on the p-cycle. One route, A–C is fully on cycle and routes between end nodes L–G and L–H are partially on cycle and partially straddling. The capacity and switching implications of the latter two types of relationship between p-cycle and its protected routes are discussed further below. At this point, however, note from the description so far, including Fig. 3, that all of the following properties arise in conjunction.

1) No cross connections will be needed in real time to form the protection paths themselves (cross connection for traffic substitution to break into and use the protection paths that the cycle provides, at the two end nodes, is still required, however, as it is for any scheme not involving dedicated duplication such as 1 + 1 APS). This means that the utmost in speed (again, 1 + 1 APS aside)[7] and certainty that the optical path works when needed is possible. As with any preconnected protection structure, only two nodes (here the path end nodes) need to act in real time, to switch the affected traffic into and out of the backup path.
2) Protection switching is entirely failure-independent end-node controlled, reacts to either span or node failure along the path, and only a single advance-switching action is preprogrammed at each end node of the path.
3) Routes that straddle the cycle can each bear two working paths for each unit of spare capacity from which the p-cycle is formed.
4) Protection of paths that transit a failed node is obtained if the group of working routes is required to be fully link and node disjoint. Otherwise, node disjointness can be relaxed to link disjointness (end nodes common to both the working route and the protecting FIPP p-cycle is, of course, the exception, but under any scheme of network survivability, traffic is lost when the ultimate source-sink node of a path is itself failed). These properties are identical to SBPP. For SBPP to provide node protection (as well as link) all primaries that share spare capacity on their backup routes must be fully disjoint, but otherwise can be only link disjoint.

[7]In 1 + 1 APS, the signal feed into the backup path is always present. Upon failure, only the receiver has to perform switchover to select the alternate signal path. This is strictly the ultimate in speed of restoration. In fact, with differential delay equalization (DADE), it is strictly possible to achieve hitless switching (1 bit-time restoration) under 1 + 1 APS. (See [13, p. 123]). Our comments on real-time cross-connection requirements and speed of FIPP p-cycles above are relative to all other schemes, except schemes involving 100% dedicated duplication of resources such as 1 + 1 APS.

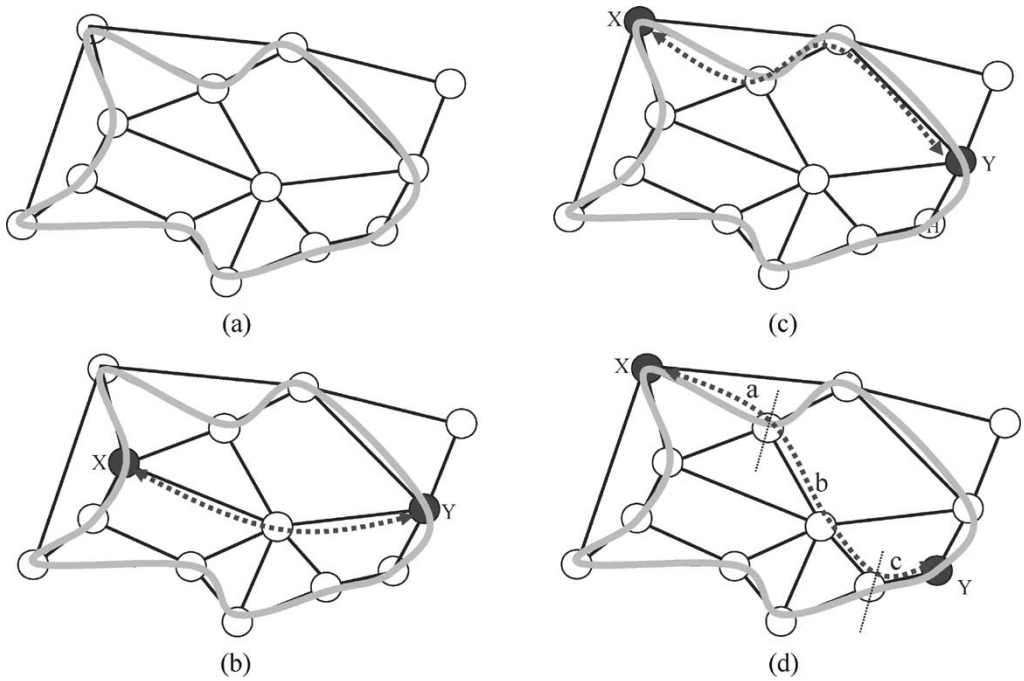

Fig. 4. Different path to p-cycle topological relations for the description of the operation of FIPP p-cycles: (a) network context and a FIPP p-cycle example; (b) pure straddling path relationship; (c) pure on-cycle relationship; and (d) partially straddling and partially on-cycle relationship.

5) The route taken by any signal in a failure-protected state can be fully known in advance. By the same token, it should be no harder than in regular p-cycle design to limit cycle size to limit the length of any restoration path, or even specifically limit the length of any individual protection path.
6) The protection structures in the network are a relatively small number of cycles, and are as easily visualized, changed, and managed as a set of conventional cross-connect-managed p-cycles for span-oriented protection.
7) It is immediately apparent that dynamic demand can be just as easily handled under the PWCE concept using path-protecting p-cycle structures, as it is with conventional p-cycles.

B. FIPP p-Cycle Operation

Basic operation is almost unchanged relative to ordinary p-cycles. To illustrate, Fig. 4(a) gives a network context and an illustrative cycle for discussion. In Fig. 4(b), the route is in a pure straddling relationship to its protecting p-cycle. In this case, two working paths can be supported on the route (per unit of p-cycle capacity) and only the end nodes of the demand have to do any switching. The assignment of each working path to the left- or right-going protection path can be based on an odd–even assignment, or any other criteria. The preassigned switching direction is stored at the end nodes, in the same way that the path to the p-cycle protection relationship itself is stored at each path end node. Under optimized design, the bulk of the relationships between paths and chosen p-cycles will tend strongly to be of this type, because it is twice as efficient as fully or partially on-cycle relationships, and the solver is at liberty to chose cycles to relate to the routes to be served most often in this preferential way. This is the simplest case to appreciate the failure-independence property because the prearmed end-node switching action does not depend on the exact failure location or even the type of failure (i.e., span or node failure). By making sure that we consider only protection of disjointly routed (i.e., "compatible") demands under any single p-cycle, we avoid the need for any failure information dissemination, as no failure can possibly strike two demands protected by the same FIPP p-cycle. We also address the mutual-capacity problem, because no two members of a compatible demand set can fail simultaneously, and there cannot be any contention for spare capacity on a p-cycle.

In Fig. 4(c), the route is entirely on cycle. Such cases are also simple: Only one protection path is provided, and it is unambiguously determined as the other part of the cycle itself. Thus, at the path end nodes, the protecting-cycle number is stored and only one path is associated with it, in the one unique direction that is usable.

Fig. 4(d) shows the most general case, and one for which several approaches suggest themselves. It is the situation where the route-bearing paths to be protected is partly on cycle to the associated p-cycle. As mentioned, this case is limited to providing a single protection path per unit of p-cycle capacity, but we also need to address the switching logic. Let us say, for instance, that the "default" protection switching preplan for the assigned path on the route shown is to switch to the "lower" path segment of the cycle between nodes X and Y. Then, if the path fails on segment a along its route, everything is fine because the lower protection path survives the failure. But if the path fails on segment c, then the default switching direction is no longer viable because a failure on this segment also fails the related p-cycle itself in the "default" direction for that path. In this situation, we need some way to know to change behavior

and to restore the affected path now over the "upper" arc of the cycle. However, this is trivially realized locally within the end nodes (both X and Y) just by adding a rule that says "if the span in which the failed path arrives is the same as the span in the default direction on the p-cycle for that path, then use the other direction." In other words, the coincidence of failure states [or appearance of alarm-indication signal (AIS)] on both a path and its respective protecting p-cycle, as seen at the paths end nodes, is sufficient (and completely local) information to guide the choice of protection path. No special-case signaling is thus required as it might seem. Of course, in all cases, it is assumed and understood that the optical paths between nodes X and Y on the p-cycle are all optically feasible by design, otherwise, such cycles would not be admissible to the network design.

One further possibility not shown in Fig. 4, but logically similar to Fig. 4(d), is where the route has one or more spans on cycle to the FIPP p-cycle, but none of these are adjacent to the end nodes. In this case the preplanning information still reflects that only one protection path can be offered along with information about its default direction. The need to change from the default direction can still be deduced locally at failure time, however, because either loss of light (in an optically transparent path) or AIS (in an opaque path) will propagate on the failed arm of the FIPP p-cycle to the respective end nodes controlling the switching. Thus, the failure is not limited to being in a span or path segment immediately adjacent to the end node to realize the p-cycle itself has been affected by the failure too.

Finally, in discussing the preplanning information to be kept in nodes and their logical switching behavior, another overall strategy is worth noting. As so far described, there is a presumption of specific preplanned configuration data being downloaded into each end node, indicating which path is associated with which p-cycle and in what direction. This is still not very complicated and may lead to the fastest possible switching times. It is also suitable for a completely transparent optical network where no channel-associated signaling between nodes is supported. But a more general and more self-managing approach is also possible, as follows: if we assume that path end nodes can monitor channel-associated overhead bytes on both the working paths they terminate and on FIPP p-cycles passing through them. The first assumption is assured because the working paths are, by definition, terminated at their end nodes. The second assumption may engender some extra cost at the cross-connect nodes to electrically monitor FIPP p-cycle channels. This may not, however, be much more expensive than the basic capability already required at any end node, i.e., the ability to break into the optical p-cycle and both launch and receive a new electrical payload signal to/from the protection paths that the p-cycle is already providing. Assuming such channel-associated overhead signaling, every topologically distinct FIPP p-cycle is numbered when created. If multiple unit-capacity copies of the same cycle are built, they all bear the same unique FIPP p-cycle number, because this number is really just an identifier for a specific group of mutually disjoint routes in the network, which are all mutually disjoint and have end nodes on that common topological cycle. This number and a list of end-node pairs with protection relationships on the cycle and a list of the spans in the cycle are applied to the overhead bytes and forever "march around in a circle" on each p-cycle itself, advertising its existence and the set of node pairs (whose routes are all mutually disjoint) that are compatible with its use. In addition, each working path (while still in electrical form at its source) is labeled with the number of the FIPP p-cycle with which it is associated for protection. With this in place, even as network-configuration changes, every node can easily learn and update the inventory of FIPP p-cycles currently available to it and can self-preplan the associations and default directions of each path for which it is an end node to the available FIPP p-cycles. Any node can also know in advance of a failure if by some chance the current network configuration (i.e., set of working paths and configured p-cycles) did not actually provide for 100% restorability of any of its terminating paths.

But even more generally, another option is to collect the p-cycle number information off each terminating path, and observe the FIPP p-cycles that are passing through the node and then just wait until failures arise. Once a failure has manifested itself, the exact situation of failed working paths and surviving FIPP p-cycle segments is uniquely defined and known at all end nodes by virtue of the alarm status in each port at each end node, along with the prefailure information collected. At this point, a simple matching algorithm can be applied to assign each failed working-path signal into an appropriate port where a surviving FIPP p-cycle segment is known to be present and offering a protection path to the desired peer end node. By matching each failed working path to one unit of surviving capacity on a locally accessible FIPP p-cycle of the same topological number, the overall restoration action is inherently (although indirectly) cognizant of the mutual-capacity coordination issue at the level of the network as a whole, and also takes into account any nondefault routing requirements arising from partially on-cycle relationships, as in the case of Fig. 4(c) above. In this approach of generalized node-local matching of failed working paths to available protection paths, a multiple quality-of-protection priority scheme is easily derived. In addition, the scheme continues to work, making best use of available resources, even when more than one failure arises, because it is self-updating to the onset of additional working-path failures and/or failure-related removal of FIPP p-cycles already in use from a first failure. The key to the generality is that each end node continually knows the other end-nodes to which it has restoration needs, and knows the inventory of currently surviving protection paths to those other nodes, and the topological compatibility of each failed path with respect to shared use of the same FIPP p-cycle capacity.

C. FIPP p-Cycle Network Design

As the reader can see from above, once a set of FIPP p-cycles and protected paths are deployed, the switching mechanism is relatively autonomous and simple. Let us, therefore, next ask how we would correspondingly design networks that operate based on this type of protection structures. To do so, let us first define the concept of a group of "compatible routes" to represent any set of working routes that are all mutually disjoint.

The mutual disjointness applies to spans if the design aim is to protect only against span failures, otherwise the disjointness applies to both nodes and spans of the routes in the group. Any set of primary paths in SBPP that share at least one spare channel amongst their backup routes could be said to form such a group of compatible routes. The significance is that any protection resource shared by paths routed over the members of a compatible group of routes will never be in conflict with each other under any single failure scenario. In this framework, we can restate SBPP as a scheme for defining compatible route sets for the sharing of individual spare channels, and FIPP p-cycles becomes a scheme for defining compatible route sets for the sharing of entire preconnected protection structures. This orientation to viewing demands in groups based on their routing leads to the following basic ideas, through which one can approach the FIPP p-cycles network-design problem.

One principle can be stated as: "Identify groups of routes over the graph that are all mutually disjoint. Then, define a path-protecting p-cycle by routing a cycle through the collected set of end nodes of these routes. Allow that p-cycle to be capacitated, so as to protect all the working paths that the network's demand matrix requires to be routed over those routes." The main problem with this approach (at least for attempting optimal-design studies for research) is that enumerating all groups of compatible routes is the combinatorial complexity. If $Y = N(N-1)/2$ is the number of end-node pairs between N nodes in a network, then there are $Y(Y-1)/2$ possible combinations of two-route groups, and "Y choose 3" possible three-route groups, and so on. Enumerating compatible groups and selecting them based on the cost of the cycle they could all share for protection is, therefore, an approach that could be revisited later, especially for heuristics approaches, but which we will leave alone for now.

A second, more immediately useful principle with the same aim is to say: "Given a cycle considered as a candidate FIPP p-cycle, identify a subset of routes between end nodes that are on the cycle x that never contend at the same time for restoration by the associated p-cycle (i.e., which form a compatible group on the end nodes of the candidate p-cycle)." This approach is more practically viable because it can be based on already well-developed ideas and methods for enumerating either all cycles of a graph or a reduced number of only high-potential candidate cycles. In the future, various combinations of these two basic approaches may suggest a variety of semiheuristic design techniques for FIPP p-cycle network design. For now, however, we will pursue an integer linear programming (ILP) design model to understand the theoretical performance of FIPP p-cycles relative to similarly optimized SBPP, and regular p-cycle network designs based on the second approach, using standard methods to enumerate cycles of the graph. We define the model entitled FIPP SCP as follows:

SETS

$\boldsymbol{S}$ Set of spans, indexed by i (failed) or j (surviving).
$\boldsymbol{D}$ Set of demand relations, indexed by r.
$\boldsymbol{P}$ Set of eligible cycles, indexed by p.

PARAMETERS

Δ A large positive constant (100 000).
∇ A small positive constant (0.0001).
c_j Cost of span j (can include all equipment costs and is proportional to length).
d_r Number of unit demand in the bundle of demand for relation r.
x_r^p Equal to 1 if the demand relation r is on-cycle p, 2 if the demand relation r is completely straddling cycle p; 0 otherwise.
π_j^p Equal to 1 if cycle p crosses span j, 0 otherwise.
$\partial_{m,n}$ Equal to 1 if demand relations m and n are "rivals." This means that the routes defined for routing demand between the end nodes of relations m and n are not mutually disjoint. They have one or more spans or nodes in common.

DECISION VARIABLES

s_j Spare capacity placed on span j.
n^p Number of unit-capacity copies of cycle p used as an FIPP p-cycle in the solution.
n_r^p Number of copies of cycle p used to protect demands on relation r.
γ_r^p Equal to 1 if cycle p does protect demand relation r, 0 otherwise.

$$\textbf{FIPP} - \textbf{SCP}: \quad \text{Minimize}: \sum_{\forall j \in S} c_j \cdot s_j \tag{1}$$

(Minimize total cost of spare capacity placed.)

CONSTRAINTS

$$\sum_{\forall p \in P} x_r^p \cdot n_r^p \geq d_r \qquad \forall r \in \boldsymbol{D} \tag{2}$$

(The entire demand quantity for relation r must be protected.)

$$n^p \geq n_r^p \qquad \forall r \in \boldsymbol{D} \tag{3}$$

(Place the maximum number of copies of cycle p required for any single demand.)

$$s_j \geq \sum_{\forall p \in P} n^p \cdot \pi_j^p \quad \forall j \in \boldsymbol{S} \tag{4}$$

(Place enough spare capacity to form all the p-cycles.)

$$\gamma_r^p \geq \nabla \cdot n_r^p \qquad \forall r \in \boldsymbol{D}, \forall p \in \boldsymbol{P} \tag{5}$$

(γ_r^p is 1 if n_r^p is greater than 0.)

$$\gamma_r^p \leq \Delta \cdot n_r^p \qquad \forall r \in \boldsymbol{D}, \forall p \in \boldsymbol{P} \tag{6}$$

(γ_r^p is 0 if n_r^p is 0.)

$$\partial_{m,n} + \gamma_m^p + \gamma_n^p \leq 2 \qquad \forall (m,n) \in \boldsymbol{D}^2 | m \neq n; \forall p \in \boldsymbol{P} \tag{7}$$

(Do not allocate the same cycle to protect two rival demands.)

Constraint (2) ensures that all demand for a particular O–D pair r is protected using p-cycles. Constraint (3) ensures that only the maximum number of instances of p-cycle p required for any single demand relation r is provisioned. Constraint (4) ensures that sufficient spare capacity exists to form all the

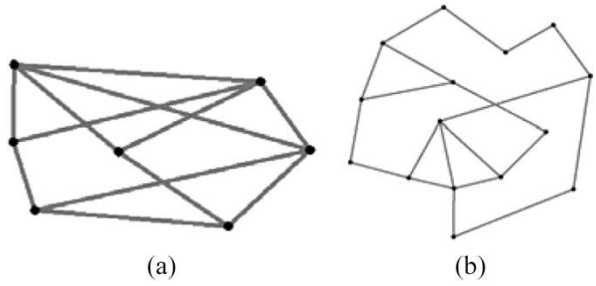

Fig. 5. (a) 7n12s; and (b) 15n20s test networks.

p-cycles selected by the design. Constraints (5) and (6) define the binary variable γ_r^p that simply defines, at run time, whether p-cycle p is indeed used to protect demand r. Constraint (7) is the key new FIPP specification that ensures that any individual p-cycle only protects a set of mutually disjoint (compatible) demands. If demands m and n are not disjoint (i.e, $\partial_{m,n} = 1$), then only one or none of γ_m^p or γ_n^p can be 1 at the same time—and consequently, p-cycle p can only be used to protect one of m or n. In solving this model, one relaxation we use is to drop the integrality requirement on n_r^p variables. This removes one family of integer variables, but does not affect solution quality of feasibility because the working flows (demand quantities) and themselves and n^p are both kept integral. This is an instance of flow variable relaxation under integer capacity and demand that is a commonly used technique that does not affect the ultimate solution, but can help speed up the solution.

IV. Test Networks and Experimental Methods

The FIPP p-cycle design model was coded in AMPL and solved using CPLEX on a dual Opteron Windows 2000 machine with 1 GB of RAM. We use three test network topologies shown in Figs. 5 and 6. Fig. 5(a) is a small, but richly connected, artificial test case of seven nodes and a span of 12 (denoted 7n12s), where every node is of degree 3 or higher. Although this network is not reflective of real networks in size, it plays a useful role in the research methodology, because it allows us to do a comparative study of different architectures on a test case where all eligible routes, all eligible cycles, and all possible demand pairs are simultaneously represented in the optimal-design problems. The Cost 239 network in Fig. 6 is a dense (average nodal degree is 4.7) real-world European network from [28]. Fig. 6 also indicates span numbers for reference in the table of shortest routes used (Table I). Fig. 5(b) shows another artificial and sparser network (15n20s) that has many degree-two nodes, and is more similar to typical North American transport networks. For all test-case designs, the cost of each span is assumed to be 1, so the minimum cost design is the same as the minimum capacity design.

The demand matrix is varied for different test cases. For the 7n12s test case, we consider a full mesh of demands for all test cases, i.e., a demand exists between every pair of end nodes. Therefore, although, this is a small network topologically, it is made larger, in the sense of its research value, by virtue of its full mesh-demand pattern. In test cases with the Cost 239 network, we vary the demand matrix from a sparse randomly

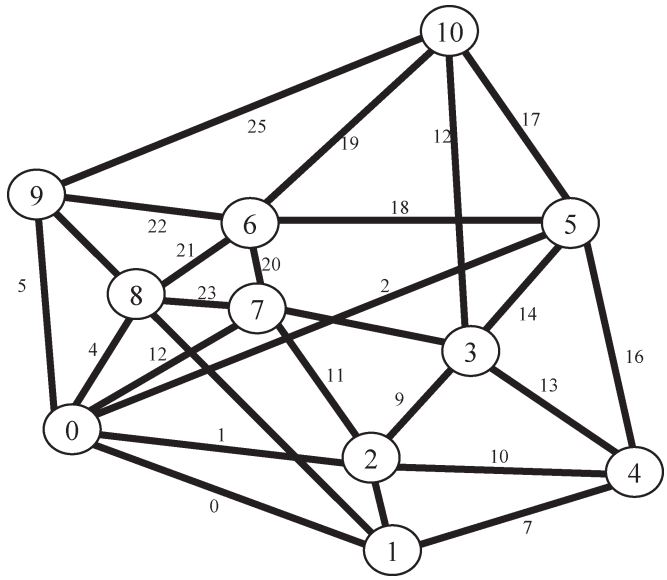

Fig. 6. Cost 239 test network indicating span numbers.

TABLE I
Working-Path Routing Used for Cost 239

Demand	End Nodes	Spans in Working Route
0	(0,1)	0
1	(0,4)	0,7
2	(0,7)	3
3	(0,10)	2,17
4	(1,4)	7
5	(1,7)	0,3
6	(1,10)	6,9,12
7	(2,5)	1,2
8	(2,8)	1,4
9	(3,4)	13
10	(3,7)	15
11	(3,10)	12
12	(4,7)	10,11
13	(4,10)	13,12
14	(5,8)	2,4
15	(6,7)	20
16	(6,10)	19
17	(7,10)	15,12
18	(9,10)	25

chosen set of 19 and 27 demands to the maximum possible of 55 demands. The nonzero demand pairs were chosen randomly, so not all demands of the 19-demand case are present in the 27-demand case. For the 15n20s test case, we randomly picked 21 out of the total possible 105 demand pairs. In all test cases, every nonzero demand bundle is exactly two units. We will later examine in detail the 19-demands test case, and, hence, we list all the 19 demands, their end nodes and the details of their working routing in Table I.

For comparison, we also generate optimal results for SBPP, PR, SR, and p-cycle span protection for all test cases. The SBPP-SCP design model is from [19], while the PR-SCP design model is from [12]. The SR design model is the standard Herzberg SCP model discussed in [13] and p-cycle design models are from [1]. We use a preprocessing program that first finds the shortest routes between the different O–D pairs and routes the corresponding demands along the single shortest

path. For eligible restoration routes, in the Cost 239 and 15n20s test cases, we enumerate all routes of length less than and equal to the tenth shortest route for the SBPP, SR, and PR solutions. In most of these problems, this amounts to providing virtually all existent routes in the graph. This automatically includes all routes that are part of the strictly optimal solution. This exhaustive representation of the eligible route set along with an allowed MIPGAP of only 1% as the CPLEX termination condition on these problems gives us confidence that the alternative architectures are being presented with virtually perfect design solutions for comparison against FIPP p-cycle designs. In the 7n12s test cases, we use the set of strictly all possible routes as the eligible route set. For use in the regular and FIPP p-cycle designs, the preprocessing program also finds candidate cycles using a standard-depth first-search algorithm and sorts the resulting cycles by hop count lengths. There are 40, 42, and 3530 distinct simple cycles possible in the 7n12s, 15n20s, and Cost 239 networks, respectively. For the Cost 239 test cases, we choose the 1000 shortest cycles (by hop counts) as candidates for FIPP p-cycles and for the span p-cycle network designs. For the remaining test cases involving the 7n12s and 15n20s networks, we also have the set of all cycles as the eligible cycle set. For the FIPP p-cycles designs, we precalculate the x_r^p, π_j^p, and $\partial_{m,n}$ parameters discussed in the previous section and provide this as input to the AMPL-CPLEX solver.

The demand matrix and working routing was the same for all the comparative architectures designed for each network test case. This means that schemes are being compared here strictly on the basis of "nonjoint" optimization, where shortest path working routes are employed in all cases. Under joint optimization of both working routes and protection-capacity considerations, all schemes will benefit somewhat further in efficiency, although it is known that schemes already close to the theoretical lower limits of PR do not improve much more, while initially less efficient designs under shortest path nonjoint optimization can improve very much more under joint optimization.

V. Results

A. Preliminary Comments

Overall, the FIPP designs took the longest time to solve and could not be solved to optimality for the larger networks. This is attributed to complexity arising primarily out of the γ_r^p 1/0 variables. In time, these problems may be overcome with suitable branching strategies and added relaxations or bounds. Often, in light of the difficulty of ILP problems, people dismiss the use of ILP for network design altogether. At present, however, our concern is not primarily about run time, but in trying to appraise the intrinsic properties of this new network architecture itself, and for this, ILP is an essential tool. In this regard, the role of ILP is as a research tool to try to understand the intrinsic efficiency of the new scheme compared to other schemes. The philosophy is that if the intrinsic capabilities or potential of the new architecture, as revealed by ILP studies, are promising, it then makes sense to work on faster heuristic algorithmic design strategies and so on. So both ILP and design heuristics have their roles to play. But the problem with using heuristics for the design of the corresponding test-case networks for research comparison is that the results then depend on both the intrinsic properties of the architectures being compared and the different suboptimalities inherent in the various heuristics used. However, even time-limited "best feasible" results from incomplete ILP runs can be valuable in research, as they usually still provide very high-quality existence-proof-type solutions, without even needing to develop a purpose-specific heuristic.

In addition, our philosophy is that differences in capacity efficiency of a few percent in are not really important in practice, if other important benefits such as full preconnection can be achieved. It is more a matter of theoretical understanding of the relative schemes being compared. In practice, our only real concern is to see if FIPP p-cycles are at least characteristically in the same range as SBPP for efficiency. Because if they are, it means their other advantages over SBPP can be exploited without practical penalty. Additionally, even if FIPP only performs better than span p-cycles and SR, it is still of interest, because now we have the p-cycle advantage of preconnection and the ability to do end-to-end service protection.

Our initial tests are for the 7n12s test case for which the solver terminated at a 3% MIPGAP[8] for the SR, PR, and SBPP runs with complete problem information as above. For the larger networks, Cost 239 and 15n20s, the number of eligible routes and cycles was limited as described above. CPLEX terminations with a 1% MIPGAP were obtained for all the reference architectures, but for these test cases, the FIPP design results are only the "best feasible" solutions found after about two days of run time. These best feasible solutions remain associated with gaps of 30% to 60% above the fully-relaxed LP lower bound. This means that the solver itself can only assure us that these designs are within 30% to 60% of optimal. It needs to be kept in mind, therefore, that in what follows for the two larger networks, feasible but suboptimal FIPP p-cycle designs are being compared to other designs that are known to be within 1% of optimum.

B. Discussion

We start with a discussion of the results for the 7n12s in Table II using all possible eligible routes for the SR, PR, and SBPP designs, and all possible cycles as eligible cycles for the FIPP and span p-cycle designs and a 1% MIPGAP. On 7n12s, we could only get within 3% of optimality for the test case where we used shortest path routing (first column of Table II).

[8]The MIPGAP is an allowed gap between the fully relaxed LP lower bound solution and the best currently found integer-constrained solution as a criteria for termination of the ILP run. The fully relaxed solution serves as a lower bound on the best possible feasible solution with integral decision quantities. Thus, if a problem terminates when running under 1% MIPGAP, it means that the solution found is provably within 1% of the true optimum. In some types of network design problem, the first feasible solution found may be close to optimal, but the fully relaxed LP version of the problem produces a very weak lower bound, so the solution is accepted when run-time limits are reached, even though the associated "gap" to the LP lower bound may remain as much as 60% to 70%.

TABLE II
TOTAL SPARE-CAPACITY RESULTS (CHANNELS REQUIRED) FOR 7n12s NETWORK DESIGNS

Protection Scheme	Shortest-Path Routing	Adjusted Routing
FIPP	32 (3% gap)	26
SBPP	30	26
SR	42	35
Span *p*-cycles	42	35
PR	25	23

Nonetheless, the FIPP design is within 7% of the corresponding SBPP-shortest path solution (32 versus 30).

Table I also shows that PR also outperforms both FIPP and SBPP, and this is in line with expectations, because as discussed, true path restoration is indeed the absolute lower bound on the capacity requirements for any single failure survivability technique. SR and span p-cycles perform equally well, but worse than SBPP, PR, and FIPP. This is also as expected from much prior research on these schemes.

Recognizing, that these are nonjoint designs, and because we have reason to suspect qualitatively that joint optimization may be relatively quite advantageous under the FIPP architecture, we allowed ourselves a further experiment on 7n12s, where we manually deviated four of the working routes from their shortest routes to routes that would (by inspection) be equivalent in hop count and more amenable to efficient FIPP implementation. We then resolved all the corresponding benchmarks for the new routing and the results are in the last column of Table II. Here, we found that the FIPP solution ran to a full termination under the 1% MIPGAP condition (in less than an hour) and is fully as good as the optimal SBPP design for the same set of demands and routes. Interestingly, all the other architectures also benefited, to varying degrees, from the adjusted working-path routing. We consider this an experimental validation of the qualitative reasoning, suggesting that FIPP and SBPP are at least categorically similar in the capacity efficiencies that can be achieved. By categorical similarity, we mean, for example, in the sense that SR and span p-cycles are also categorically similar. The existence of an identical result, and another that is only $\sim 3\%$ different suggests that there is no argument that the schemes lie in theoretically different categories of capacity efficiency, such as can be stated with confidence about, for example, SR and PR, which are clearly in different basic categories of achievable efficiencies.

Results for the larger Cost 239 and 15n20s test networks are summarized in Table III. In Table III, the FIPP results are only the best feasible solutions obtainable in the time available. The bracketed value shows the remaining gap from optimality. In practice, where the remaining gap on an FIPP solution suggests the true optimum solution could be below the PR solution value, PR more correctly provides the real lower bound on the possible solution value for FIPP. The last row of Table III is for the special case of SBPP designs, in which we allowed only routes equal in length to the single shortest backup path (disjoint from the working route) in the eligible route set. This represents a practical SBPP provisioning option (optimally designed nonetheless) wherein each working route takes the shortest path to the destination and its backup path is planned along one of the next shortest routes that is disjoint from the primary.

In Cost 239, the best feasible FIPP designs are similar or better than SBPP when the latter uses only the shortest alternate backup routes. In Cost 239 with 19 demands, the best feasible FIPP design with a gap of 30% is within 18% of fully optimal SBPP. Beyond this, the large remaining gaps on the best feasible FIPP p-cycle designs make it hard to say where the latter stands relative to fully optimal SBPP solutions. If we take the totality of results into account, however, we have one case in 7n12s with a full demand matrix, where FIPP is within 3% of SBPP in the initial designs, and matches SBPP when four working routes are adjusted. And in the larger test cases, the SBPP solution costs are all within the range of the gaps on the best feasible FIPP p-cycle designs. To this, we should add the considerations from first principles that the most closely related scheme of all those considered to FIPP, is SBPP. These are the only two path-oriented failure-independent schemes above, i.e., path-type schemes that do not benefit from stub release, as does PR. All told, therefore, it seems reasonable to expect that the FIPP architecture may be reasonably close in intrinsic efficiency to SBPP, just that we are not presently skilled enough at solving the FIPP p-cycle network-design problems. In fact, there is also one observation about FIPP and SBPP architectural properties that suggests that, in at least one aspect, FIPP could possibly be even more efficient than SBPP in some circumstances. Consider the following: From one point of view, it seems reasonable to surmise that an FIPP p-cycle could always be viewed as being formed from a specific choice of two backup routes, so that SBPP would have to serve as a lower bound for the spare-capacity results of FIPP p-cycle designs. However, there is one important respect in which it can be demonstrated that the FIPP solution space is not simply a subset of the SBPP solution space. To explain this, we need only look back at the case of the partially straddling path example in Fig. 4(d). SBPP has a fundamental requirement that every working path is fully disjoint from its own backup route (except at its end nodes), as well as fully disjoint from other working paths that share any spare capacity in their backup paths. There is no exception possible to this restriction under SBPP. But look again at Fig. 4(d). Here, we see that under FIPP, a working path can, in general, have path segments in common with its own protection structure. The switching behavior to allow this was explained when Fig. 4 was introduced. What is seen in Fig. 4(d) is equivalent to a limited type of stub release. Stub release is only otherwise found in true path restoration. In path restoration, restoration paths are allowed (when advantageous to the design) to reuse part of their own working paths within the protection path. In FIPP, as we can see in Fig. 4(d), we can allow the backup path to be co-routed with the working path, thus, effectively reusing the surviving component of the working route. This possibility separates the FIPP p-cycles architecture from SBPP and leaves it open that FIPP p-cycles might, in some cases, outperform SBPP in capacity efficiency, given suitable solutions to the optimization problem, because the behavior in Fig. 4(d) allows the optimizing design solver to occasionally save more capacity than would be possible under SBPP. The reader might also note that in the example where

TABLE III
RESULTS (SPARE-CAPACITY CHANNEL COUNTS) FOR COST 239 AND 15n20s NETWORKS

	Cost 239–19	Cost 239–27	Cost 239–55	15n 20s–21
Best Feasible FIPP (gap)	33 (30%)	44 (47%)	67 (66%)	173 (58%)
SBPP (opt)	28	30	46	134
Span Restoration (opt)	29	33	62	174
Span p-cycles (opt)	39	47	76	174
Path Restoration (opt)	21	22	35	111
SBPP (SP Alternate-opt)	46	44	64	174

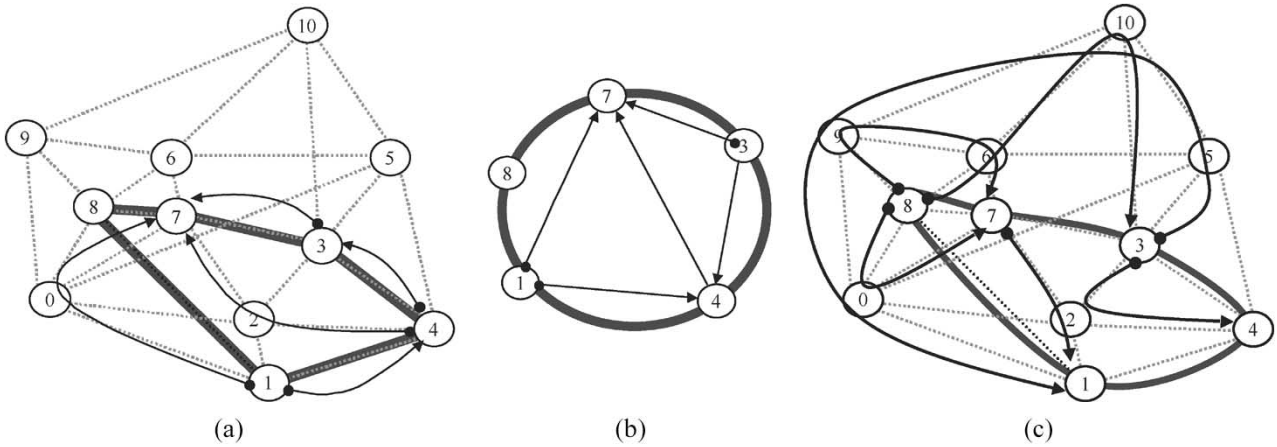

Fig. 7. (a) FIPP p-Cycle A in Solution for Cost 239 network with 19 Demands (SCP); (b) logical view; (c) example of potential additional loading capability.

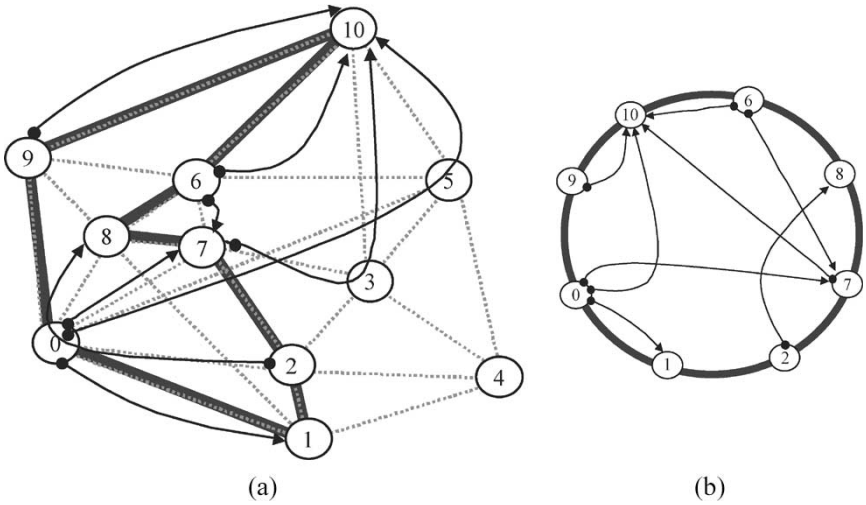

Fig. 8. (a) p-Cycle B FIPP p-cycle solution for Cost 239 network with 19 Demands (SCP); and (b) logical view.

FIPP p-cycles wind up reusing their "stub-path" segments, it is easy to construct an actual trap situation for SBPP. In SBPP, if no backup route exists that is disjoint from the chosen working route, then an infeasibility "trap" situation arises. In sparse networks, this may mean that working routes tend to have to be significantly longer for SBPP feasibility. In contrast (as the example shows), FIPP does not fall into such traps and allows working paths to take shortest routes.

C. Inspection of FIPP Design Characteristics

Both as a form of validation of the functional correctness (mainly the property of 100% restorability) and to further portray and understand the FIPP architecture, we picked the 19-demand Cost 239 test case to draw out all the FIPP p-cycles that are in the solution. The completely restorable design was based on only five FIPP p-cycles shown in Figs. 7–11. Each figure shows one of the actual FIPP p-cycles chosen in the best feasible design and the working demands associated with the cycle (shown using thin arrow-headed lines). We also show a simplified logical abstraction of each p-cycle and the end-to-end-node demand pairs that it protects.

Cycle A in Fig. 7 protects up to two units of demand for demand relations 9,5,4,12,10. The working routing is shown using thin arrow-headed lines. Notably, if p-cycle A were to be used as a span protecting p-cycle, its ratio of protected capacity

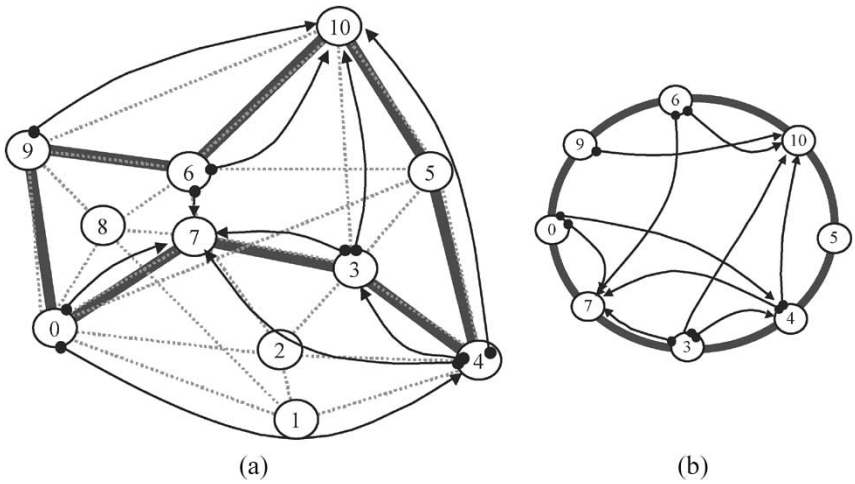

Fig. 9. (a) FIPP p-cycle C; and (b) logical view.

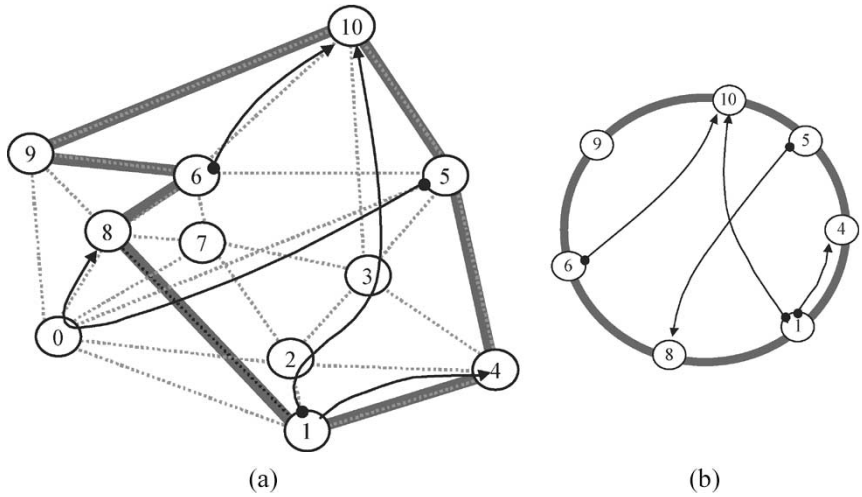

Fig. 10. (a) FIPP p-cycle D; and (b) logical view.

to its own consumed capacity is only 1, because it has no straddling spans. However, on using the same cycle as an FIPP p-cycle, this ratio increases to 2.6 (end-to-end demands protected per channel hop of the p-cycle) for the combination of demands shown in Fig. 7(c), a much higher efficiency. The chances of cycle A actually being selected as a span p-cycle would therefore have been low, but this same cycle becomes of much higher merit as an FIPP p-cycle, because it does have good straddling relationships when allowed to operate to protect paths solely on an end-to-end path basis. We note, therefore, that if we reuse the standard *a priori* efficiency [26] measure for span p-cycle selection to cut down the ILP problem size, we would tend not to include cycles such as this in the set of eligible cycles. This motivates further work on developing new heuristics and metrics for *a priori* cycle selection in FIPP. Another observation is that the number of demands that the FIPP p-cycle can protect can be increased, if the working routes are deviated from the shortest path. This suggests, not surprisingly, that a joint model that simultaneously optimizes the working capacity and p-cycle placements may yield considerably more efficient designs by having even greater latitude over the group of compatible routes that are chosen to associate with each candidate p-cycle. The same FIPP p-cycle A can "soak up" more demands without any increase in spare capacity. This also motivates further work in developing a working routing strategy that maximizes FIPP efficiency.

Similarly, p-cycle B protects up to two units of demand for node pairs 8,2,3,18,17,16,15,0 in Table I. FIPP p-cycle C protects demands on node pairs 9,2,18,16,15,12,11,10,1. FIPP p-cycle D protects demands on node pairs 6,4,16,14 and FIPP p-cycle E protects demands on node pairs 7,4,18,13.

On inspection of Figs. 10 and 11, we can see that the p-cycles do not give the impression of being very heavily loaded with protection relationships. In the present case, despite this seeming waste, the overall solution still requires less absolute spare capacity than the span p-cycle design. For interest and comparative purposes, the optimal span-protecting p-cycle solution for the same networks and set of working demands and routes is shown in Fig. 12.

By simple inspection, it is not clear if there are any strategies that could be used to convert span p-cycles to FIPP p-cycles. We ran a quick test case where we limited the FIPP-SCP solver to choosing from the four span p-cycles from Fig. 12. It did result in an optimal FIPP solution quickly, but did not improve on the

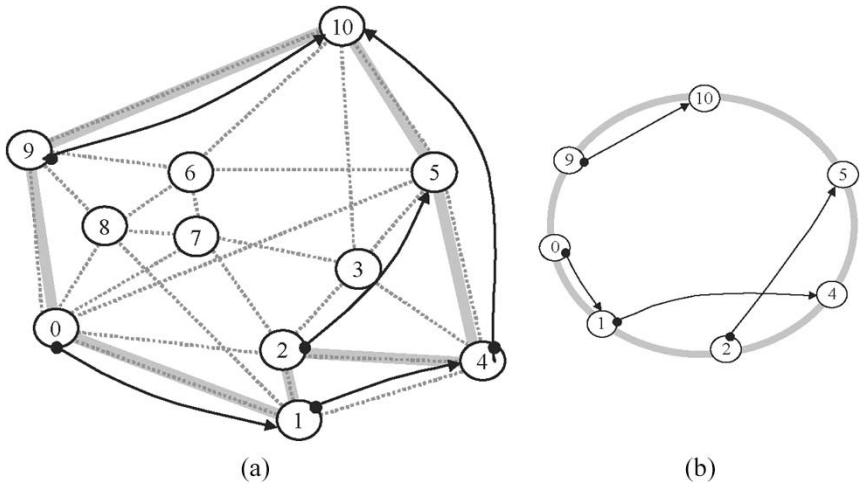

Fig. 11. (a) FIPP p-Cycle E; and (b) logical view.

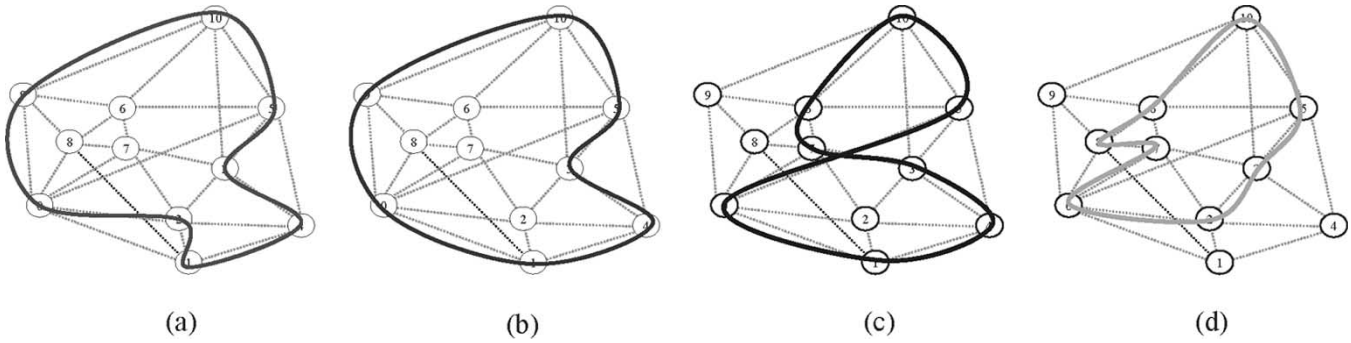

Fig. 12. Span p-cycle solution for Cost 239 with 19 demands and 1000 eligible cycles.

overall capacity efficiency relative to the span p-cycles design. There is no guarantee that a span p-cycle set can produce a feasible FIPP solution. This suggests, however, a possibly powerful strategy for assisting the ILP solution of the FIPP p-cycle network-design problem: First, solve the generally much easier regular p-cycle design problem and the corresponding PR problem and use its number of p-cycles and objective-function values as guiding additional upper and lower bounds, respectively, for the FIPP p-cycle ILP solution.

VI. Conclusion

We have proposed the concept of failure-independent path-protecting (FIPP) p-cycles, as a relatively simple scheme that extends the p-cycle concept into a path-oriented version that has an attractive combination of practical properties. The most important single property is that FIPP p-cycles appear both experimentally and theoretically to be similar in capacity efficiency to shared-backup path protection (SBPP), and both support completely failure-independent end-node activation and control against either span or node failure, but FIPP p-cycles do so with fully preconnected protection paths. This combination of features enables ringlike speeds, minimal real-time signaling, and the assurance of optical signal quality on the protection path when needed, in addition to inherently providing for node failures, as well as span failures. It may also be of practical value, in light of the conceptual complexity of other preconnected approaches [pre-cross-connected trails (PXTs) [2] and p-trees [8], for instance], that an entire FIPP p-cycle network design can be easily decomposed into representations, such as in Figs. 7–11 and viewed as a set of ringlike, independently operating substructures, each of which is associated with protecting the end-to-end working demands of a certain set of node pairs. On the other hand, we found even medium-size instances of the FIPP p-cycle network-design problem extremely hard to solve in integer linear programming (ILP) form, so ongoing research is warranted to look at heuristics for the FIPP p-cycle design problem, alternate approaches to posing or solving the FIPP ILP design model, and efforts at joint optimization of working routing while placing the FIPP p-cycles. We feel however that the demonstration that FIPP p-cycle designs are of even similar spare-capacity requirements to SBPP is significant, because it means that the other main advantage, full preconnection, can be exploited without any significant penalty ensuing regarding capacity requirements.

Application of the p-cycle-PWCE concept for simplified dynamic demand provisioning using FIPP p-cycles will also be interesting. One direct way to see the linkage to the PWCE concept for dynamic demand handling is that each FIPP p-cycle of a certain capacity, in effect defines a group of protected tunnels of routable working capacity between end-node pairs of the compatible group defined on that FIPP p-cycle. Within an environment of specifically capacitated FIPP p-cycles, each node that handles dynamic demand arrivals and departures will

see a corresponding "hard-wired" sort of protected tunnel to peer-provisioning nodes. For instance, a new demand arriving at node Y for destination X can be handled by node X simply by looking up a local table that tells it which FIPP p-cycle the route Y-to-X is associated with, and what the amount of currently unused protected capacity is to node X. From node Y's standpoint, the predefined and structurally protected end-to-end route to node X appears as a kind of direct pipe or channel group from route Y to X, which is protected end to end. Thus, it is not hard to see how FIPP p-cycles can be directly applied to support a version of PWCE that operates not on a span-by-span channel-protected basis, but on a direct end-to-end route-protected basis.

In closing, we think that FIPP p-cycles open up a new direction of research in survivable networks, and provides practical new options for network operators, especially those that were seeking ways to evolve towards meshlike spare-capacity efficiency in optical DWDM networks, but found that most shared mesh-type schemes require the assumption of on-the-fly cross connection of optical channels. With FIPP p-cycles, the characteristic efficiency benefits of a generic shared-mesh solution are obtained, but the protection paths are all completely pre-cross-connected and, hence, of known and tested optical-path integrity before they are needed.

REFERENCES

[1] W. D. Grover and D. Stamatelakis, "Cycle-oriented distributed pre-configuration: Ring-like speed with mesh-like capacity for self-planning network restoration," in *Proc. IEEE Int. Conf. Communications (ICC)*, Atlanta, GA, Jun. 7–11, 1998, pp. 537–543.

[2] T. Y. Chow, F. Chudak, and A. M. Ffrench, "Fast optical layer mesh protection using pre-cross-connected trails," *IEEE/ACM Trans. Netw.*, vol. 12, no. 3, pp. 539–547, Jun. 2004.

[3] R. L. Freeman, *Fiber-Optic Systems for Telecommunications*. New York: Wiley, 2002, ch. 6 and 10.

[4] D. Stamatelakis and W. D. Grover, "Theoretical underpinnings for the efficiency of restorable networks using preconfigured cycles ('p-cycles')," *IEEE Trans. Commun.*, vol. 48, no. 8, pp. 1262–1265, Aug. 2000.

[5] W. D. Grover and M. H. MacGregor, "Potential for spare capacity preconnection to reduce crossconnection workloads in mesh-restorable networks," *Electron. Lett.*, vol. 30, no. 3, pp. 194–195, Feb. 3, 1994.

[6] D. Stamatelakis and W. D. Grover, "Network restorability design using pre-configured trees, cycles, and mixtures of pattern types," TRLabs, Edmonton, AB, Canada, Tech. Rep. TR-1999-05, Oct. 30, 2000.

[7] G. Shen and W. D. Grover, "Extending the p-cycle concept to path segment protection for span and node failure recovery," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 8, pp. 1306–1319, Oct. 2003.

[8] S. Shah-Heydari and O. Yang, "Hierarchical protection tree scheme for failure recovery in mesh networks," *Photonic Netw. Commun.*, vol. 7, no. 2, pp. 145–159, Mar. 2004.

[9] J. Doucette and W. D. Grover, "Node-inclusive span survivability in an optical mesh transport network," in *Proc. 19th Annu. Nat. Fiber Optics Engineers Conf. (NFOEC)*, Orlando, FL, Sep. 7–11, 2003, pp. 634–643.

[10] J. Kang and M. J. Reed, "Bandwidth protection in MPLS networks using p-cycle structure," in *Proc. 4th Int. Workshop Design Reliable Communication Networks (DRCN)*, Banff, AB, Canada, Oct. 19–22, 2003, pp. 356–362.

[11] D. Stamatelakis and W. D. Grover, "IP layer restoration and network planning based on virtual protection cycles," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 10, pp. 1938–1949, Oct. 2000.

[12] R. R. Iraschko, M. MacGregor, and W. D. Grover, "Optimal capacity placement for path restoration in STM or ATM mesh survivable networks," *IEEE/ACM Trans. Netw.*, vol. 6, no. 3, pp. 325–336, Jun. 1998.

[13] W. D. Grover, *Mesh-Based Survivable Networks*. Upper Saddle River, NJ: Prentice-Hall, Aug. 26, 2003.

[14] G. Kaigala and W. D. Grover, "On the efficacy of GMPLS auto-reprovisioning as a mesh-network restoration mechanism," in *Proc. IEEE Global Telecommunications (GLOBECOM)*, San Francisco, CA, Dec. 1–5, 2003, pp. 3797–3801.

[15] R. R. Iraschko, W. D. Grover, and M. H. MacGregor, "A distributed real time path restoration protocol with performance close to centralized multi-commodity maxflow," presented at the 1st Int. Workshop Design Reliable Communication Networks (DRCN), Brugge, Belgium, May 17–20, 1998, Paper O.9.

[16] S. Kini, M. Kodialam, T. V. Laksham, S. Sengupta, and C. Villamizar. (2001, May). Shared backup label switched path restoration. [Online]. Available: http://search.ietf.org/internet-drafts/draft-kini-restoration-shared-backup-01.txt

[17] W. D. Grover and Y. Zheng, "VP-based ATM network design with controlled over-subscription of restoration capacity," presented at the 1st Int. Workshop Design Reliable Communication Networks (DRCN), Brugge, Belgium, May 17–20, 1998, Paper O.33.

[18] J. Doucette, M. Clouqueur, and W. D. Grover, "On the availability and capacity requirements of shared backup path-protected mesh networks," *Opt. Netw. Mag.*, vol. 4, no. 6, pp. 29–44, Nov./Dec. 2003.

[19] J. Doucette and W. D. Grover, "Comparison of mesh protection and restoration schemes and the dependency on graph connectivity," in *Proc. 3rd Int. Workshop Design Reliable Communication Networks (DRCN)*, Budapest, Hungary, Oct. 2001, pp. 121–128.

[20] C. Ou *et al.*, "Near-optimal approaches for shared-path protection in WDM mesh networks," in *Proc. IEEE Int. Conf. Communication*, Anchorage, AK, May 2003, pp. 1320–1324.

[21] Y. Xiong, D. Xu, and C. Qiao, "Achieving fast and bandwidth efficient shared-path protection," *J. Lightw. Technol.*, vol. 21, no. 2, pp. 365–371, Feb. 2003.

[22] G. Shen and W. D. Grover, "Design of protected working capacity envelopes based on p-cycles: An alternative framework for survivable automated lightpath provisioning," in *Performance Evaluation and Planning Methods for the Next Generation Internet*, A. Girard, B. Sansò, and F. Vazquez-Abad, Eds. Norwell, MA: Kluwer, 2005.

[23] ——, "Performance of protected working capacity envelopes based on p-cycles: Fast, simple, and scalable dynamic service provisioning of survivable services," in *Proc. Asia-Pacific Optical and Wireless Communication Conf. (APOC)*, Beijing, China, Nov. 7–11, 2004, vol. 5626, pp. 519–533.

[24] A. Kodian, W. D. Grover, J. Slevinsky, and D. Moore, "Ring-mining to p-cycles as a target architecture: Riding demand growth into network efficiency," in *Proc. 19th Annu. Nat. Fiber Optics Engineers Conf. (NFOEC)*, Orlando, FL, Sep. 7–11, 2003, pp. 1543–1552.

[25] D. A. Schupke, C. G. Gruber, and A. Autenrieth, "Optimal configuration of p-cycles in WDM networks," in *Proc. IEEE Int. Conf. Communication (ICC)*, New York, Apr. 28–May 2, 2002, pp. 2761–2765.

[26] W. D. Grover and J. Doucette, "Advances in optical network design with p-cycles: Joint optimization and pre-selection of candidate p-cycles," in *Proc. IEEE LEOS Summer Topicals*, Mont Tremblant, QC, Canada, Jul. 15–17, 2002, pp. 49–50.

[27] D. Stamatelakis, "Theory and algorithms for preconfiguration of spare capacity in mesh restorable networks," M.Sc. thesis, Dept. Elect. Comput. Eng., Univ. Alberta, Edmonton, AB, Canada, Spring 1997.

[28] P. Batchelor *et al.*, "Ultra high capacity optical transmission networks: Final report of action COST 239," Faculty Elect. Eng. Computing, Univ. Zagreb, Zagreb Croatia, 1999.

[29] A. Koster *et al.*, "Demand-wise shared protection for meshed optical networks," in *Proc. Design Reliable Communication Networks*, Banff, AB, Canada, Oct. 19–22, 2003, pp. 85–92.

[30] A. Groebbens *et al.*, "Efficient protection in MPλS networks using backup trees: Part one—Concepts and heuristics," *Photonic Netw. Commun.*, vol. 6, no. 3, pp. 191–206, 2003.

[31] ——, "Efficient protection in MPλS networks using backup trees: Part two—Simulations," *Photonic Netw. Commun.*, vol. 6, no. 3, pp. 207–222, 2003.

Adil Kodian (S'01) received the B.E. degree in electronics engineering from the National Institute of Technology, Surat, India. He is working toward the Ph.D. degree in electrical engineering with the Network Systems Group, TRLabs.

He has been with TRLabs, Edmonton, Alberta, Canada, since 2001. His research interests include failure-independent path-protecting p-cycles (coinvented with Wayne D. Grover), quality of protection service classes in p-cycle networks, and ring to p-cycle evolution. He is currently the Manager of the Business Development and Technical Services of Rohit Connexion.

Wayne D. Grover (S'74–M'76–SM'90–F'02) received the B.Eng. degree from Carleton University, Ottawa, ON, Canada, the M.Sc. degree from the University of Essex, Essex, U.K., and the Ph.D. degree from the University of Alberta, Edmonton, Alberta, Canada, all in electrical engineering.

He had 10 years of experience as scientific staff and manager at BNR (now Nortel Networks) working on fiber optics, switching systems, digital radio, and other areas before joining TRLabs as its founding Technical VP in 1986. In this position, he was responsible for the development of the TRLabs' research program and contributed to the development of the TRLabs' sponsorship base; he saw TRLabs through its early growth as a startup to the over-the-100-person level. He now functions as Chief Scientist—Network Systems, TRLabs, and as Professor, Electrical and Computer Engineering, at the University of Alberta. He has patents issued or pending on 26 topics to date.

Dr. Grover is a P.Eng. in the Province of Alberta and a member of the SPIE. He is a recipient of the IEEE Baker Prize Paper Award for his work on self-organizing networks, as well as the IEEE Canada Outstanding Engineer Award, the Alberta Science and Technology Leadership Award, and the University of Alberta's Martha Cook-Piper Research Award. In 2001–2002, he was also the holder of a prestigious NSERC E.R.W. Steacie Memorial Fellowship. He has also received two TRLabs Technology Commercialization Awards for the licensing of restoration and network-design-related technologies to industry.