IP Layer Restoration and Network Planning Based on Virtual Protection Cycles

Demetrios Stamatelakis, *Member, IEEE,* and Wayne D. Grover, *Senior Member, IEEE*

***Abstract*—We describe a novel restoration strategy called *virtual protection cycles* (p-cycles, patents pending) for extremely fast restoration in IP networks. Originally conceived for use in WDM and Sonet transport networks, we outline the adaption of the p-cycle concept to an IP environment. In an IP router-based network, p-cycles are implemented with virtual circuits techniques (such as an MPLS label switched path, or other means) to form closed logical loops that protect a number of IP links, or a node. In the event of failure, packets which would normally have been lost are encapsulated with a p-cycle IP address and reenter the routing table, which diverts them onto a protection cycle. They travel by normal forwarding or label switching along the p-cycle until they reach a node where the continuing route cost to the original destination is lower than that at the p-cycle entry node. Diverted packets are deencapsulated (dropped from the p-cycle) at that node and follow a normal (existing) route from there to their destination. Conventional routing protocols such as OSPF remain in place and operate as they do today, to develop a longer term global update to routing tables. Diversionary flows on the p-cycle inherently cease when the global routing update takes effect in response to the failed link or node. The p-cycle thus provides an immediate real-time detour, *preventing packet loss*, until conventional global routing reconvergence occurs. The aim of the paper is to explain the basic p-cycle concept and its adaptation to both link and node restoration in the IP transport layer, and to outline certain initial results on the problem of optimized design of p-cycle based IP networks.**

***Index Terms*—Internet, network design, network fault tolerance, network reliability, optimization methods, routing.**

I. INTRODUCTION

A. Motivation and Objective

ONE OF THE architectures of prime interest for the "next generation" networks is "IP over WDM."[1] Given the role that current and future networks play in our society, it is axiomatic that fast, accurate, and efficient means for restoration are of central importance in the design of IP–WDM networking technologies. Considerable prior work has focused on restoration in the physical layer, including WDM, with the primary aim of fast 100% restoration of any single fiber optic transmission span cut [6]–[13]. The main characteristic of restoration in the physical layer, whether by ring, mesh, or protection switching based methods, is that prefailure transmission capacity is directly replaced with equal bandwidth transmission path substitutions. Because of the "direct bandwidth replacement" model when rerouting carrier signals, as opposed to the service-layer traffic flows themselves, the effect on users and client networks is almost negligible: a very short transmission disruption, and an increase of a few tens of milliseconds at most in physical propagation delay.

However, physical layer (WDM or Sonet) restoration has some limitations and drawbacks. First, the total capacity investment for restorability can be expensive. With rings, or 1+1 diverse routing, there will generally be an investment of at least 100% in transmission capacity redundancy. With restorable mesh alternatives, this may be reduced, but 60%–80% physical redundancy levels are still typical. Second, node failures within a service layer can only be dealt with by the actions of peer-level network elements. For example, no reconfiguration of the underlying WDM light-paths in the physical layer can address the failure of a single network interface card on a router. Affected IP traffic flows can only be restored by some form of dynamic routing in the router network layer itself. Nor would it be desirable to have the many other services borne in the physical layer undergoing restoration "hits" due to reconfiguration requests passed down to the physical layer from a higher layer node element. There are also network operating contexts where physical layer restoration is not technically or economically an option. For instance, a wide-area or metro area Internet backbone operator may have a logical network comprised in part or wholly from carrier signals (STSn's or wavelengths) leased from facilities-based operators. Such an operator will be naturally interested in restoration strategies that are within their own sphere of control, i.e., their own logical IP transport network, either because control (or even knowledge of) the true physical layer is not available to them, and/or because it may be more economic to restore in their own IP logical environment than to lease the physical carrier signals at a premium for assured physical-layer restorability.

A further advantage of restoration at the IP layer is that there need not be any rigid distinction between working and "spare" capacity. Extra capacity allocations still have to be engineered to ensure some target level of quality of service during restoration, but during normal (nonfailure) operation, this extra capacity is available for improved working service performance. In addition, capacity-design for controlled oversubscription [22], [28] during restoration is a strategy that is available in a packet or cell-based layer but not in a Sonet or WDM environment.[2] Such strategies use all of the unutilized bandwidth available at the

Manuscript received October 15, 1999; revised May 15, 2000.

D. Stamatelakis is with TRLabs, Edmonton, AB, Canada (e-mail: dstamate@edm.trlabs.ca).

W. D. Grover is with TRLabs Edmonton, AB, Canada, and also with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada T6G 2E1 (e-mail: grover@edm.trlabs.ca).

Publisher Item Identifier S 0733-8716(00)09001-6.

[1] More precisely, this may actually be IP in a lightweight Sonet frame, or gigabit Ethernet, etc., over WDM.

[2] One can argue conceptually that the corresponding strategy also exists for a circuit-managed layer. Statistical muxing for circuits is otherwise known as engineering to a target blocking probability. The important difference is that under packet oversubscription of capacity allocations, all affected flows continue with a shared congestion-delay penalty whereas circuit-oriented oversubscription of capacity implies perfect recovery for some, and complete outage for those that are blocked.

time of failure. In contrast, WDM or Sonet restoration involves discrete assignments of working and spare capacity. The unused portion of a low-utilization "working" channel is not inherently available for restoration, as it is in a stat-muxed environment. Thus, there is both need and rationale for considering restoration techniques at both WDM and IP layers. In particular, it may turn out that WDM is a preferable layer in which to deal with physical span cuts, whereas the IP layer addresses recovery from loss of a router node.

In this work, we propose an approach based on p-cycles (to be explained), with the aim of providing restoration capabilities that are self-contained within an IP transport layer but operate on time scales much closer to that of WDM or Sonet rings. We also show how the method of p-cycles is amenable to capacity planning and optimization to give assurances of worst-case restored-state performance with near-minimal capacity cost. The strategy we describe is based on the concept of *virtual protection cycles* (p-cycles) [1]–[5]. p-cycles were originally conceived as a way to speed up the restoration of failures within mesh-based Sonet or WDM transport networks. The most significant aspect of the p-cycle concept as applied to Sonet/WDM mesh networks is that it permits ring-like switching speeds (because only two nodes do any real-time actions) and yet it exhibits the capacity efficiency characteristic of a span-restorable mesh network. This remarkable combination of properties has been reported and explained by the authors in a series of prior works, primarily considering Sonet/WDM type applications [3]–[5], [23]–[25]. The task now is to consider adaptation of the p-cycle concept to an IP environment.

B. Outline of Paper

The paper follows these logical steps. First, to be relatively self-contained, and because the p-cycle concept is relatively recent, we begin with a review of the basic p-cycle concept as developed for span restoration in a WDM or Sonet context. That comprises the remainder of this section. The second major section is devoted to considering IP p-cycles for inter-router link restoration. Section III introduces an important extension, that of a *node-encircling* p-cycle for restoration of transiting traffic flows affected by a router node failure. Section IV is devoted to a design formulation to support p-cycle based link restoration and an algorithm for configuring node encircling p-cycles. Section V is a concluding discussion.

C. Background on p-Cycles

In the arena of WDM or Sonet networking, p-cycles are an exciting recent advance because they promise the best properties from each of the basic prior alternatives for restoration: ring and mesh. These are, specifically, the rapid restoration speed of rings and the high capacity efficiency of mesh. Obviously this is an important claim. This section is therefore devoted to substantiating that motivating aspect of the present work, and to provide background about the basic p-cycle concept, before going on to consider p-cycles in the IP layer.

This section can be supplemented for interested readers by references [3], [4], [23]–[25]. Reference [3] is the basic report of our first results with p-cycles where we found the total spare capacity required for 100% restoration in five test networks to be within 0 to 9% in excess of an optimal span-restorable mesh, while the real-time restoration switching remained BLSR-like at only two nodes. Reference [4] describes a distributed autonomous protocol through which a network can self-organize a near optimum set of p-cycles within itself, as an interesting alternative to centralized control of p-cycle configuration (which remains an option, of course). Reference [23] is a more theoretical analysis substantiating the prior experimental findings of such high capacity efficiency and also proving that p-cycles are as efficient as any class of preconfigured spare capacity structure that can exist for restoration. Conference papers [24], [25] overlap somewhat with the present journal paper but [25] also includes material on a nodal capacity-slice device for p-cycle based WDM networking, not published elsewhere, that offers an ADM-like alternative to optical cross-connects for implementation of a WDM p-cycle based network. We now give a brief overview on rings, mesh, and p-cycles to set the stage for the rest of the work.

Ring-based survivability involves the use of bidirectional line switched rings (BLSRs) or unidirectional path-switched rings (UPSRs) as self-protecting transmission systems overlaid on the network topology. Operation and planning of UPSR and BLSR based transport networks, and their more recent optical path protection ring (OPPR) and optical shared protection ring (OSPR) variants in a WDM context, is already well covered in the literature. The important point is that rings use a simple switching mechanism which permits restoration in about 50–60 ms, although by their nature they require at least 100% redundancy. In particular, the BLSR uses a working to spare loop-back switching mechanism at the two nodes adjacent to a failure, and this is essentially the identical switching mechanism that p-cycles (for WDM or Sonet) employ. In conventional multiring network designs, however, where the working fiber or channel groups themselves are not fully utilizable, effective spare-to-working capacity ratios (the capacity redundancy) can be 200–300%. Thus, rings are fast but not intrinsically capacity-efficient.

Mesh-based survivability is more capacity-efficient because each unit of spare capacity is reusable in many ways, across many different failures. Signals that traverse a failed span are rerouted through many diverse paths which, when considered in total, require smaller amounts of spare capacity to realize than are present in ring-based networks. Performance very close to idealized maximum-flow routing efficiency can be achieved. Mesh restoration has traditionally been based on cross-connect systems embedded in a mesh-like set of point-to-point transmission systems, under either centralized or distributed control. Because of this, and because of the more general nature of solving a discrete capacitated multiple-path rerouting problem, restoration is not as fast as with rings but permits a major reduction in the capacity required to serve the same set of demands. While ring-based networks always require 100% or more redundancy, a span-restorable mesh network may be as little as 50% redundant, depending on network topology and demand pattern [7], [8], [11].

Thus, each of these long-standing alternatives has strengths and weaknesses. Mesh networks tend to be economic in long

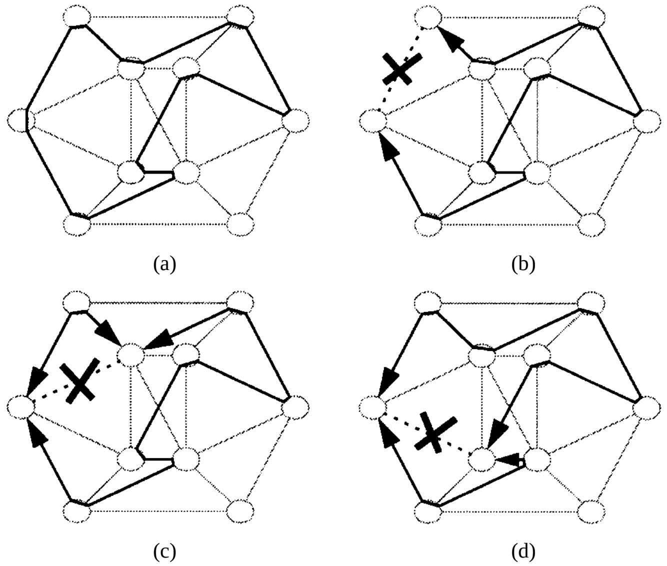

Fig. 1. Use of p-cycles in restoration (from [3]).

haul architectures where capacity efficiency correlates more directly to cost savings. Rings tend to be more cost-efficient in metro areas where cost is dominated by terminal equipment, not distance-related transmission cost. To date, the choice between a ring or mesh-based network has been essentially black or white, i.e., a one-or-the-other proposition. Nothing has previously emerged that can offer the best advantages of both of these extremes. This is the significance of p-cycles as first proposed in [3]. They combine the speed of rings with the efficiency of mesh-based networking. An appreciation of this is essential before continuing to IP p-cycles.

The method of p-cycles for Sonet or WDM is based on the formation of closed paths (elementary cycles in graph theoretic terms), called *p-cycles*, in the spare capacity of a mesh-restorable network. They are formed in advance of any failure, out of the previously unconnected spare capacity units of a restorable network. Despite similarity to rings—both use a cycle on the network graph for their topology—p-cycles are unlike BLSR/OSPR, UPSR/OPPR (or FDDI) logical rings in that they protect both *on-cycle* and *straddling* failures (to be explained). Fig. 1(a) shows an example of a *p-cycle*. In Fig. 1(b), a span on the cycle breaks and the surviving arc of the cycle is used for restoration. This is functionally like a unit-capacity BLSR. In Fig. 1(c) and (d), however, the same p-cycle is accessed to support restoration of working paths that are *straddling* the cycle. In fact, cases Fig. 1(c) and (d) are the more advantageous circumstances in general because *two* restoration paths are available from each p-cycle for such failures. In contrast, either type of conventional Sonet or WDM ring provides at most one restoration path per unit of ring protection capacity. Rings also protect only against failures on the spans of the same ring, not on “straddling” spans.

This makes a significant difference to the network restoration coverage provided by the same investment in spare capacity in a ring as opposed to in a p-cycle. For example, further examination of the single p-cycle in Fig. 1 shows that it can provide restoration path(s) to 19 potential span failures (ten straddling relationships, nine on-cycle relationships), while as a ring, protection is available only for the nine spans on the cycle. But, in addition, the p-cycle provides two restoration paths for each of the ten spans that are in a *straddling* relationship. Thus, spare capacity on a p-cycle is more widely accessible, i.e., more highly shared for restoration than in a BLSR or UPSR. Although it is not initially obvious, under the appropriate design optimization, this fact allows p-cycle based networks to be essentially as capacity-efficient as mesh networks. This was verified in [3] where fully restorable p-cycle capacity plans were generated and compared to conventional mesh restoration for five test networks. The worst test case required 9% additional spare capacity while the remaining cases required 0 to 3%. Reference [3] details the test case networks and the mixed integer programming formulation under which these results were obtained.

Although p-cycles seem, initially, to embody only one small difference relative to today’s well-known ring systems (the aspect of protecting straddling failures), there are many consequences from this difference when fully worked through. p-cycles (if based on cross-connects) can be formed from unit capacity channels of the point-to-point OC-n or DWDM systems present, whereas rings commit a whole OC-n module of working and spare capacity to the same cycle. Rings also have a structural association between the working demands which they protect and the protection bandwidth in the same ring, while p-cycles are formed only within the spare capacity layer of the network, leaving the working paths to be routed freely on shortest paths, or any other route desired. In other words, the working demands may be provisioned freely as growth arises, as if in a failure-free point-to-point network; the p-cycles formed in the sparing layer adapt to suit the working path layer. A deployed p-cycle design may also be easily modified by the cross-connects that form it, whereas Sonet ring placements are essentially permanent structural commitments of both working and spare capacity, to which the routing of new working paths must conform. Finally, the implication of protecting straddling failures is that a p-cycle spare capacity design takes little or no more capacity than a corresponding span-restorable mesh network [3]. Generally this will be substantially less than 100% capacity redundancy. This property, plus the fact that the switching operations for restoration with p-cycles is essentially just that of a BLSR, is the reason we say that p-cycles can offer “the speed of rings, with the efficiency of mesh.” The reason they are as fast as rings is that there are only two traffic-substituting connections to be made for any working path failure to be restored. Moreover, the two end-nodes that perform the switching only do a transmit signal bridging and receive direction transfer operation that is essentially BLSR-like in nature, and they know in advance exactly which working-to spare switching functions will be needed for any given failure.

II. IP Link Restoration Based on p-Cycles

IP networks are already restorable in the sense that OSPF and BGP routing protocols will, through dissemination of link-state and route advertisements, eventually update the routing tables network-wide to compensate for failure [14]–[16]. In terms of techniques usually considered for transport restoration, however, these are relatively slow processes, and existing methods have no direct regard for the capacity congestion effects that may arise from the purely logical (i.e., uncapacitated) routing policy changes that result.

The initial approach to use p-cycle ideas for more rapid IP network restoration, covered in this section, is to use p-cycles in much the same logical way that they would be used for Sonet or WDM networks, but with the "switching mechanism" and establishment of the logical p-cycle construct itself adapted to the IP router environment. In the IP context (of both this section and Section III), the p-cycles are envisaged as the "fast" part of a "fast plus slow" overall recovery process. This recognizes that if p-cycles can deal with the immediate real-time packet protection part of the problem, the ordinary routing update protocols (currently OSPF, BGP, for instance) will still proceed to develop global routing table updates as usual. But in the interval of real time before the update has completed, p-cycles will be in use to prevent the loss of packets. When the slower routing update process converges, the affected packet flows will return to normal routing procedures and traffic on the p-cycles will automatically drop back to zero. Thus, the IP p-cycle mechanism will serve as a fast-acting but temporary protective measure which secures network traffic while the routing tables adapt globally to the new network state.

The main extensions to be introduced for IP p-cycles are in the details of the IP addressing and routing environment to create "virtual circuit"-like p-cycle constructs. These adaptations are further extended in Section III with the addition of the concept of *node-encircling p-cycles* to deal with node (i.e., router) failure, in addition to normal p-cycles for link failures. This section focuses on the first step of moving p-cycles from the Sonet or WDM context for span restoration to the corresponding role of link restoration in an IP context.

A. Routing of IP Packets

Conventional IP is inherently connectionless. A packet is forwarded from a router in response to an appropriate routing table entry for the packet's destination address. Based on the destination address (or subnet address), an entry in the routing table points to the local port of egress to the next router toward the destination. In this manner, packets are routed, from source to destination, on a hop-by-hop basis following a series of local router table lookups. Each node's routing table is established by the network-wide execution of a routing protocol. A number of routing protocols exist, such as RIP, BGP, and OSPF [14], [17]–[19]. For each router, the protocol determines a set of least "cost" routes from the router to each destination. The route cost is determined by the sum of the individual costs of the links used to form the route. The link costs may be determined by the link's delay, the link's monetary cost to use, or commonly by the inverse of the link's available capacity, or any other measure assigned by a system administrator. Each table entry, for a given destination, contains fields for the next router along the determined least cost route and the associated cost of the route from the router to the destination. By accessing its tables, a router can rapidly determine the next router to which to forward a packet along its least cost route. Overall, this basic "forwarding engine" function of a router is to: inspect an IP address, enter the routing table at a match to this address (or subnet mask address), and forward the packet out the port indicated in the routing table. It is within this fast table-lookup forwarding environment that virtual p-cycles can be created with a small number of reserved IP addresses or any other kind of IP tunnel, tag, or label switching technique.

B. p-Cycle Implementation in an IP Environment

In an IP network, unlike Sonet or WDM, there is no distinct concept of identifiable spare capacity that is so designated and reserved for restoration. IP links simply have a high or low utilization relative to their current installed "pipe" bandwidth. It is therefore not initially clear how to exploit the p-cycle concept with its preconfigured circuit-like logical structures of protection bandwidth. To do so, we need to create some kind of virtual circuit construct within an IP router environment.

In this regard, although IP is far more widely used than ATM, ATM does embody a number of useful concepts which are being adapted by the IP standards community. Primary among these are quality of service aspects (QoS) and, relevant to our proposal, the support for creation of point-to-point multihop connections using virtual circuit-like IP tunneling or *label switching* methods (as, for example, in MPLS: multiprotocol label switching). MPLS [20], [21] is essentially a means to form what were called VCs or VPs (in ATM parlance) within an IP environment, mainly for the purpose of shifting the routing of IP packets from the core of the network to the edge. Irrespective of the original motivation for MPLS, or other tunnelling, label switching, or tag switching proposals, it is possible to realize p-cycles using any of them. Even in pure IP, one could realize virtual p-cycles with a set of corresponding routing table entries created with reserved or otherwise unallocated regular IP addresses, set aside to define the desired p-cycles through a form of IP tunneling. Thus, there is a range of technical means to effectively produce a virtual circuit (or virtual path)[3] construct in an IP environment. While having stressed that there any many techniques that could be used to create virtual p-cycles, we will henceforth adopt the terminology of recent MPLS developments and refer to a *label-switched path* (LSP) rather than the prior ATM terms of VC or VP. The importance of LSPs to our problem is that once a packet is entered onto an LSP, its subsequent routing is completely predefined by the label switching sequence which defines the LSP. In addition, bandwidth is consumed by an LSP only when traffic is being carried. When there is no traffic for an LSP, it employs only logical resources such as label allocations. Similarly a virtual p-cycle will consume bandwidth only when used in failure recovery, since this is the only time that it carries traffic. The additional traffic, introduced by the p-cycles' rerouting of packets during restoration, must, however, be taken into consideration in the capacity design of the network, or excessive congestion could result.

[3] While the generic technical term for this is the creation of a *virtual circuit*, in the specific language of ATM a p-cycle is more correctly analogous to a VP than a VC because a large number of individual network flows will cross the p-cycle together when it is in use. It is an intermediate common transport structure, not a logical structure associated with individual end node applications as is the ATM VC.

C. Restoration of IP Link Failures Using p-Cycles

The first class of failures which p-cycles can restore is the loss of a logical IP link (sometimes also called an IP trunk or IP logical span or just a "link") between a pair of adjacent routers. This could be caused by a failure at the physical/transport layer, such as a transmission span cut, or by the failure of an interface card. For physical layer failures, the failure will appear in the IP layer only if there is no physical layer restoration mechanism or its capability is exceeded for some reason. For example, a span failure occurring within an OSPR WDM ring should be restored in about 50 ms (i.e., at least as fast as a Sonet BLSR). This time is far less than any IP routing protocol time-outs, and so the failure and restoration event will not even be observed as such at the IP layer. In general, therefore, WDM physical layer restoration of span cuts remains a very fast and effective line of defense against span failure and may be preferred in practice over router-based recovery against physical span cuts. With this in mind, our explanation of router-based restoration against link cuts should be seen as providing an additional option for IP network operators, rather than being necessarily advocated to the exclusion of physical layer restoration for span cuts. (For node restoration, the logic is quite different, however.)

This said, p-cycle restoration of logical link failures in an IP network is envisaged as follows. When an IP link failure has been detected, the router ports which terminate the failed link will be marked as dead (and the usual link-state advertisement update process will be triggered). Until a global routing update is effected, any packet whose next hop, as indicated by the normal routing table entry for the packet's destination address, would have been directed into the dead port, is instead deflected onto a p-cycle which has been assigned to protect the link.

"Deflection onto a p-cycle" occurs by *encapsulating* the original IP packet in a "p-cycle packet" and reentering the routing table. When reentering the routing table, the encapsulation IP address matches to the surviving port where a virtual p-cycle has been previously established. The packet is forwarded into the corresponding surviving port at the initial encapsulating node and travels through the p-cycle, following either label switching or routing table entries at other nodes that have been preestablished on the p-cycle IP address or label sequence to define the logical p-cycle.

Eventually packets in the detoured flow arrive at the router on the other side of the dead link.[4] At this point, the original IP packet is removed from the encapsulating packet and continued on its prefailure route toward its final destination. The node that decapsulates the detoured packet knows it is the one to do this because it has a local routing table entry pointing to a nonfailed outgoing port for the original IP address in the p-cycle packet and, as it is located on the other end of the failed link, it also contains a port with an alarm condition.

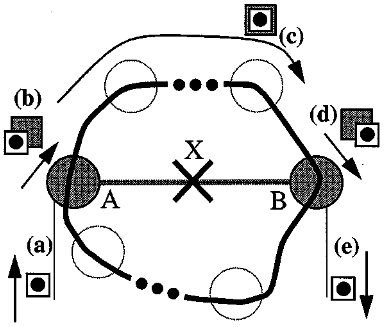

Fig. 2. p-cycle recovery of an IP packet from a span failure.

An example of this process is given in Fig. 2. The example shows a link failure, X, its associated p-cycle, and routers, A and B, which were adjacent through the link X. In Fig. 2(a), an IP packet is arrives at router A with a next hop, indicated by the routing tables, that would normally be into the failed link X. In Fig. 2(b), router A, seeing the failure, instead encapsulates the packet in an appropriate p-cycle packet (one available locally that is known to protect link X) and inserts it into the p-cycle. The p-cycle packets traverse all intervening routers along the p-cycle, as in Fig. 2(c), until, at Fig. 2(d), it arrives at router B, where the original IP packet is decapsulated from the p-cycle packet. Finally, at event Fig. 2(e), the IP packet is routed normally toward its final destination. Note that although Fig. 2 shows only one arc of the p-cycle being used, it is possible to split the total flow to be detoured by some classification function on the destination IP address (even–odd, for example) so that the capacity around both directions of the p-cycle between nodes A–B is actually used during restoration. When this consideration is added, each virtual p-cycle is actually defined by a pair of routing table or label switching entries: one points to the local port which corresponds to entry of the p-cycle in the clockwise direction, the other is the counterclockwise usage of the same logical p-cycle.[5]

There are also two classes of link failure which a p-cycle can restore: failures which occur *on* a link of the p-cycle itself, and straddling failures which occur off the p-cycle, leaving it intact. Fig. 2 is an example of a straddling failure relationship. Fig. 1(b) shows what we mean by an on-cycle failure case as well. In this case, the local nodal action for rerouting into and from the p-cycle at the two end nodes is identical, but one of the two possible directional uses of the p-cycle itself is failed (showing a bad port indication), so inherently all flows are redirected into the one surviving direction of the logical p-cycle. Straddling failures, however, allow the restoration traffic to be split (by IP address or flows) as described in two directions around the failure. It is also possible to split traffic from a single failure among multiple different logical p-cycles that may be accessible at a given node. (The number of distinct p-cycles containing the required peer node for restoration that are accessible at a given node is something that the following design formulation can decide for the planner.) Both these measures distribute the restoration load more evenly, reducing possible congestion (or requiring less capacity in design.)

[4] Or, more generally, at any node where the continuing route cost of the original packet is lower than that where it entered the p-cycle. For simplicity, however, this generalization is omitted for the time being for the basic explanation of IP p-cycle link restoration. We return to it in node restoration.

[5] Shortly prior to press time for this paper, colleagues at Nortel Networks have advised us of completion of a prototype implementation of the proposed p-cycle scheme, providing restoration in 110 ms *without any packet loss* to either on-cycle or straddling link failures with hardware level detection of link failure [27]

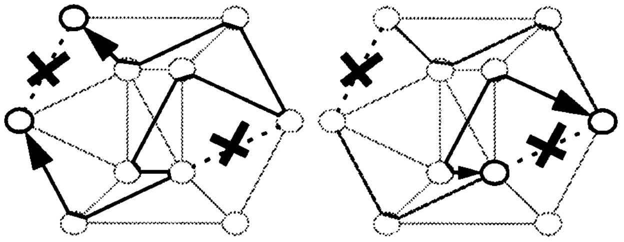

Fig. 3. Multiple failures restored simultaneously using the same p-cycle.

In Fig. 3, we illustrate some of the multiple failure properties of the p-cycle concept. The p-cycle shown has itself been disrupted by one on-cycle link failure, while another straddling link failure is also present. Straddling link failures do no damage to the p-cycle itself, on-cycle failures "open" the p-cycle into a surviving linear virtual-path segment. In the case of Fig. 3, the p-cycle shown can still offer restoration to both failures, however, because—although it has been disrupted—it still offers one restoration path in the other direction around the failure. The restoration path for the on-cycle (disruptive) failure utilizes the intact portion of the remaining p-cycle, while the straddling (nondisruptive) failure utilizes only a surviving arc of the remaining p-cycle. Both restoration paths connect the end routers of their respective failures and provide an alternate routing for affected packets.

III. "Node-Encircling" p-Cycles for Recovery from Router Failure

Unlike experience with Sonet or WDM transmission networks where node failures are rare compared to cable cuts or other line-related failures, IP networks reportedly suffer "node failures" as frequently or even more often than link outages. Router restarts, due to the routine application of software patches/upgrades, or due to router crashes, apparently generate the majority of router outages. These failures are in principle addressed by existing routing protocols which eventually disseminate the news of the router disappearance through link state advertisements by all adjacent nodes of the failed node, and the network as a whole globally converges to a revised routing plan. In practice, however, the volume of link state update flooding messages, the time required for stable complete reconvergence, and the resulting congestion effects could all be improved upon by a fast and more localized restoration response to router outage. Note, of course, that the term "node restoration" is really a misnomer. It is only prefailure *transiting* flows through a node that can be restored by any type of network-level response. Source-sink flows at the failed node itself are inherently unrestorable by any type of network-level rerouting. We therefore refer to this problem as network recovery from a node loss, or restoration of the transiting flows, rather than "node restoration" per se.

Adapting p-cycles to the restoration of transiting flows when a router fails involves extension of the p-cycle concept to that of *node-encircling* p-cycles. In link restoration, a given failure must either be on a link that lies directly on the p-cycle or, if not, it must have both end nodes on a respective p-cycle for that p-cycle to assist in its restoration. In addition, a link failure only affects a single hop along the original route of the IP packets affected, so both end nodes have intact prefailure knowledge of how packets are to be routed once they complete their p-cycle detour around the failure, and each node adjacent to the failure knows precisely which "next node" was in the original route.

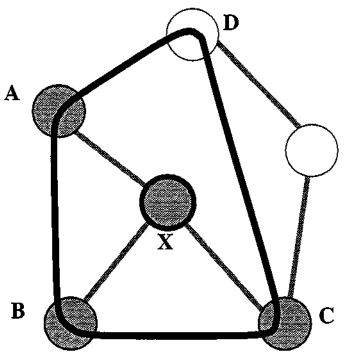

Fig. 4. Concept of a node-encircling p-cycle.

But in a node failure, a surviving neighbor node knows only that the next surviving node on the original route of the affected packets, after the failed node, must have been one of the other surviving neighbors of the failed node. It does not, however, know which other neighbor node of the failure node the prefailure route had transited. (To know this, it would have to have access to the failed nodes routing table or compute it from the failed nodes standpoint assuming a synchronized link state database.) A *node-encircling* p-cycle can, however, cope with this aspect of a router failure by *providing an alternate path among all of the routers which are adjacent to the failed router*. (As before, this will actually be a directionally opposite pair of logical paths on the same p-cycle.) Therefore, each node encircling p-cycle can provide a readily available replacement detour path for up to $n(n-1)/2$ router-pairs, where n is the number of routers adjacent to any node to be considered as a prospective failure node.

Thus, for a p-cycle to provide restoration for all of the prefailure flows affected by a router failure, *it must contain all of the routers that were adjacent to the failed router, but not the failed router itself*. The idea is that a node-encircling p-cycle constitutes a kind of "perimeter fence" which is assured to be intersected at ingress and egress by all (transiting) flows that may be affected by the given node failure. It must contain all adjacent routers, otherwise it cannot substitute routes for all of the possible prefailure flows that traversed them via the failure node. But unlike p-cycles for link restoration, it must *not* contain the one node that it is protecting, so that it is not itself disrupted when the router fails. These are the properties of what we call a "*node-encircling p-cycle*." Fig. 4 gives one example, others follow. In Fig. 4, node X is protected by the node-encircling p-cycle shown. Node X is adjacent to (has a link to) nodes A, B, C but, as shown, a node-encircling p-cycle may have to visit nonadjacent nodes (here, node D) to include all adjacent nodes. In general, a p-cycle which encircles one node would not encircle other nodes as well unless a "region-encircling" effect was desired, wherein flows would be protected on their transit through any desired subnetwork as a whole.

It may initially seem onerous to create one node-encircling p-cycle for every network node, but the implications are not really that great. Obviously the method scales linearly with number of nodes. Any network of N nodes would be made fully recoverable against any single router outage by establishment of N p-cycles. For bidirectional use of every such p-cycle, this would still require no more than $2d$ additional routing table entries at each node, where d is the "degree" of the node in the graph-theoretic sense (i.e., the number of other nodes to which it has direct links). This is a very small overhead compared to the typical number of normal routing entries in a modern backbone router.[6]

One difference when an adjacent router fails, as opposed to an attached link, is that there may be no hard local alarm indication of the failure. Neighbor nodes may see only a disappearance of packet flows and eventual failure of the "hello" protocol, rather than a local hardware alarm. Conventionally it will take four missing 10-s "hello" packets for neighbors to detect the router loss. Much faster strategies for fast neighbor-node failure detection are easily conceived, however. An obvious step is to increase hello frequencies and reduce time-outs, or to create separate MPLS or carrier-signal overhead channels for direct supervisory contact between adjacent nodes to directly ascertain their neighbors "alive" status. Reducing time-outs alone, however, can result in false alarms. A more robust composite strategy would be to have a watchdog task in each router opportunistically insert "idle-fill-hello" packets in any outgoing port at any time the link is not in use with traffic and the watchdog task senses its local O/S and forwarding engine are still operating. This would be combined with an incoming detection of a sudden drop in packet arrival rates. The composite fast detection scheme would then be to note either the sudden cessation of idle-fill-hello packets (during times of nonpeak utilization), or (when link utilization is very high starving out the opportunistic hellos) to note any sudden total cessation of packet arrivals. This strategy could yield reliable router-failure detection in far less than the 40 s that would conventionally elapse to detect a soft-failed router by a neighbor node.

In any case, we presume a means of failure detection, and proceed from that point. Once failure has been detected, each adjacent surviving router will mark the port to the failed router as dead. Subsequently, packet arrivals which would normally be sent toward the failed router are forwarded within the local node to a p-cycle handler which encapsulates the original IP packet in a p-cycle packet, as already described for link restoration. Importantly, the routers do not have to actually distinguish between an adjacent link or node failure—the mechanisms are the same any time a port is marked as dead. As before, reentering the routing table with the encapsulated packet has the effect of injecting the encapsulated packet into the respective p-cycle. At this stage, we generalize the process for detecting when to strip an encapsulated packet off of the p-cycle (as alluded to in footnote 4 in Section II-C). The p-cycle packet format contains a *p-cycle ID* field, an *original route cost* field, a *destination address* field, and, as payload, the original IP packet, as shown in Fig. 5. The ID field contains a unique identifier for the p-cycle to which the packet belongs (this is the IP tunnel or label switched path info that guides packets around the p-cycle via forwarding at other nodes). The destination address field contains the IP address of the packet's destination, and the original router cost field contains the cost of the route the packet would have used prior to failure, as indicated by the local table entry for its destination address at the router where the p-cycle was entered. The route cost field will be used to determine when it is "safe" (i.e., assured to be loop-free) to remove the packet from the p-cycle, continuing it on a normal route to its destination.

p-Cycle ID	Destination Address	Original Route Cost	IP Packet

Fig. 5. p-cycle packet format.

As the encapsulating packet travels along the p-cycle, each router tests the packet (in the way to be described) to determine if it should remove it from the p-cycle (decapsulate it). The routers know to test these packets because they arrive on a p-cycle address or label. If the test result is negative, the router forward the packet out the local port indicated by the routing table entry for the p-cycle address, continuing it on the p-cycle. If the test is positive, the original packet is extracted from the p-cycle packet, and forwarded from that node using the local routing table entry for the original IP address of the decapsulated packet.

The test which the router applies is to compare the original route cost in the encapsulating p-cycle packet to the local route cost entry. If the latter is less than the original route cost, it will decapsulate the original IP packet from the p-cycle packet and forward it according to the preexisting route entry for that destination. If the local routing cost is greater than or equal to the original route cost, the router continues relaying the encapsulated packet along the p-cycle.

The test ensures that the packet will only be returned to normal forwarding at some point that is "downhill" from its entry into the p-cycle, i.e., that the p-cycle egress point must constitute a net advance toward the original destination before returning to normal routing. (Without this check to detect the first appropriate decapsulation node, a packet could get in a loop where it would continuously be introduced into a p-cycle at one point, be removed from the p-cycle at another point, and then be routed normally back to the first point where it could reenter the p-cycle.) After this, the packet will continue to move away from the failure point as it is routed toward its destination, and there will be no danger of it reentering the p-cycle.

Fig. 6 gives an example of an IP packet being automatically detoured without loss around a router failure, then restored to a normal continuing route. For the example, all links have a cost of 1, except two links which have a cost of 10 to permit illustration of the points above. In Fig. 6(a), a packet follows its normal route to its destination. The packet's prefailure route has a cost of 3. After being disrupted by a router failure, subsequent packets on that IP address are encapsulated within a p-cycle packet (with original route cost of 3) and injected into the p-cycle [Fig. 6(b)]. The p-cycle packet is relayed along the p-cycle until it reaches a router at Fig. 6(c). The router compares

[6] Which, today, may be on the order of 50 000–100 000 entries.

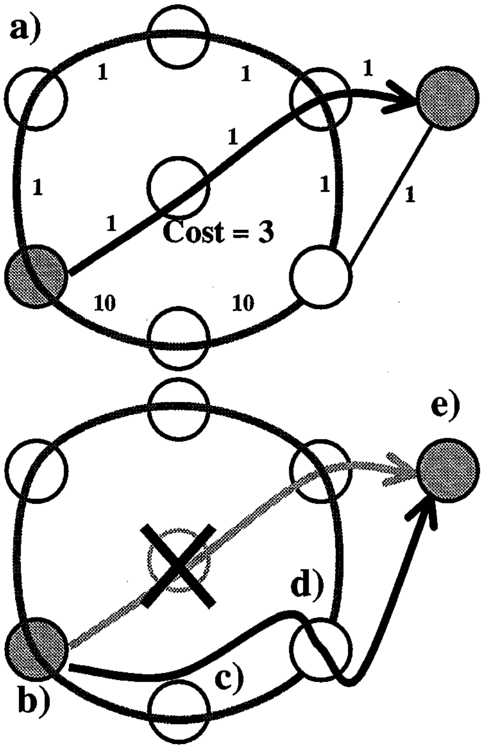

Fig. 6. p-cycle restoration of router failure.

the original route cost in the p-cycle packet to the route cost from its routing table (which, from this node, is 11). The local cost is not less than the original cost so it forward based on the p-cycle address and does not decapsulate (i.e., the packet continues on the p-cycle). At Fig. 6(d), the p-cycle packet reaches another router where the same cost comparison is performed. However, here the local cost of 1 is less than the original cost of 3; therefore, the router unencapsulates the original IP packet and routes it normally. Finally, in Fig. 6(e), the packet arrives at its destination, having survived the node failure on its original route.[7]

A. Types of Node-Encircling p-Cycles

A p-cycle, which protectively encircles a node, must be constructed within the subgraph that results when the protected node is itself removed from the network. It may or may not be possible to form a *simple* cycle within the resulting subnetwork (a simple cycle is defined as one which crosses each node and link only once.) There is, however, always a logically encircling p-cycle construct that is possible in any two-connected (prefailure) graph,[8] if two special considerations are dealt with. Fig. 7 gives an example of simple and nonsimple p-cycles which can result when protecting routers. The first is an example of a simple cycle, where the removal of the protected node does not disrupt the overall two-connectedness of the resulting network. In such cases, the node encircling p-cycle is usually visually apparent. The second example is a case where removal of the protected node results in a singly connected remaining network. The degree-one node can then only be included in the encircling p-cycle through a segment through which the cycle passes twice (or through a special relay interface arrangement to pass restored packets to surviving nodes on the stub [1]). The last example of a nonsimple p-cycle results when the removal of the protected node results in a subnetwork which has a bridge node; that is, a node whose removal would disconnect the graph. Here, the logically encircling p-cycle is formed as a "Fig. 8," as it is forced to pass twice through a node. In both of these special cases, the p-cycle does remain a cycle when viewed logically and is still able to protect against the loss of the node. In all cases, the p-cycles can also still be logically established by addition of normal extra routing table or label-switching entries although some routers would not allow the type of logical construct in Fig. 7(b) (hence the alternate arrangement described in [1] for that case).

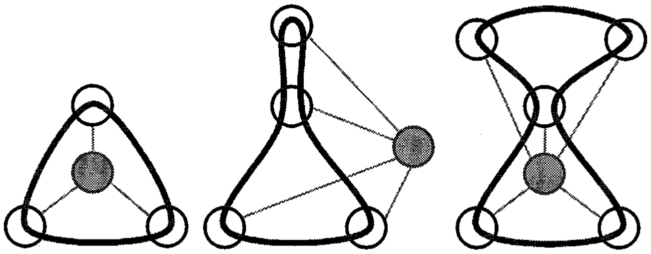

Fig. 7. Basic p-cycle topologies in router restoration.

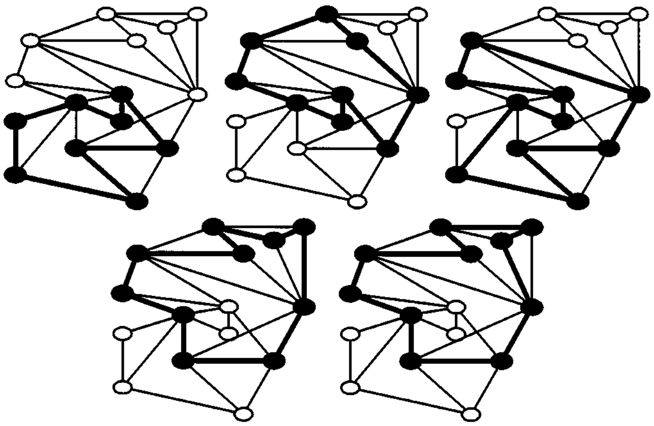

Fig. 8. Illustrative set of five p-cycles for solved for minimum peak oversubscription under 100% link restorability for test case Net 2.

IV. IP Network Design Using p-Cycles

A. Capacitated Network Design Using IP p-Cycles (for Link Restoration)

In the IP context, restorability design needs to consider the convergent flow effects arising from restoration. This is an aspect of restoration capacity design that is much simpler for Sonet or WDM p-cycle network design where every working signal unit is either exactly replaced (100% restorability) or not (<100% restorability.) In contrast, where stat-muxed flows are being redirected upon restoration, one can take an "oversubscription"-based view toward controlling (by design) the worst-case simultaneously imposed flows on any link during

[7] A multiflow animation of this process is available at www.ee.ualberta/~grover/pcycles/.

[8] That is, the topology of the prefailure network must contain no singly connected nodes. Networks containing singly connected nodes are not fully restorable by any rerouting method.

a restoration scenario. Moreover, capacity investment and worst-case restoration-induced oversubscription effects can be traded off against each other in a controlled manner [22], [28].

The aim in the following formulation is to determine a set of IP p-cycles that minimize the greatest oversubscription factor on any link, over all failure scenarios. This allows the network designer to accept a lowered (but assured worst-case) QoS during a restored network state in return for economic savings as less capacity is required to provision the network. This approach may also be an improvement over allowing ordinary routing protocols to be used solely for restoration since the latter do not take capacity/congestion effects into direct account at all. The formulation is a mixed integer program (MIP) through which the relationship between worst-case restoration-induced oversubscription and the corresponding capacity requirements of a network can be studied, dependent on the number of p-cycles that are allowed in the design optimization.

$$\text{minimize} \quad \eta_M \tag{1}$$

$$s.t. \sum_{i=1}^{N_C} \delta_i \le M \tag{2}$$

$$\alpha_{i,j} \le \delta_j \qquad \forall i = 1 \cdots N_S, j = 1 \cdots N_C \tag{3}$$

$$\sum_{j=1}^{N_C} \alpha_{i,j} = 1 \qquad \forall i = 1 \cdots N_S \tag{4}$$

$$l_{i,j} = w_i + \sum_{k=1}^{N_C} \beta_{i,j,k} \cdot w_j \cdot \alpha_{j,k} \qquad \forall i = 1 \cdots N_S, j = 1 \cdots N_S, k = 1 \cdots N_C, i \ne j \tag{5}$$

$$\eta_M \ge \frac{l_{i,j}}{c_i} \qquad \forall i = 1 \cdots N_S, j = 1 \cdots N_S, i \ne j. \tag{6}$$

The objective function, η_M [in (1)], is the maximum oversubscription ratio on any link during any restoration event. N_C is the number of graph cycles in the master set of cycles from which p-cycles can be chosen. N_S is the number of links in the network; δ_i is a binary decision variable which is 1 if cycle i is used in the design, and 0 otherwise; M is the maximum number of p-cycles which are permitted in the design (a user input, in this case); w_i is the amount of working traffic flowing through link i during normal operation; c_i is the total capacity on link $i; l_{i,j}$ is the total traffic flowing on link i during the restoration of link failure $j; \alpha_{i,j}$ is a nonnegative real variable that equals the fraction of the working traffic on network link i, which cycle j is assigned to restore; and $\beta_{i,j,k}$ is the link loading ratio on link i when cycle k is used to restore link j. Each $\beta_{i,j,k}$ is a constant which gives the fraction of the working flow from link j which is carried on link i using cycle k during j's restoration. It can either be 0, 0.5, or 1. It is zero for the case where the cycle k does not pass over link i, 0.5 for the case where the p-cycle offers two restoration paths and the traffic is split in half over each p-cycle segment, and 1 for the case where the p-cycle offers only a single restoration path and all traffic is carried over a p-cycle segment. Equation (2) limits the design to use, at most, M p-cycles. Equation (3) allows a link i to use cycle j for its restoration (as decided through the variable $\alpha_{i,j}$) only if cycle j has been used in the design (as decided by variable δ_j). Equation (4) requires each link failure to be fully restored by its assigned p-cycles. Note that the formulation can be modified so that the $\alpha_{i,j}$ are binary variables (equal to either 0 or 1): Equation (4) will then require that each link failure be completely restored with a single p-cycle. Equation (5) affects the measure of oversubscription restoration for all links, for each link failure, by summing the link's normal working flow with all fractional restoration flow(s) which are routed through p-cycles that traverse that link. Equation (6) asserts that all normalized restoration traffic flows are less than or equal to the oversubscription ratio η_M, which is being minimized by the objective function.

TABLE I
PROPERTIES OF THE TEST NETWORKS

	Net1	Net2	Net3
Nodes	10	15	20
Links	22	28	31

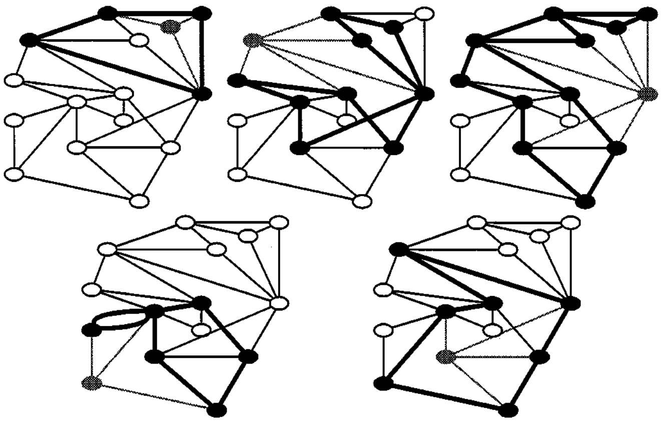

Fig. 9. Illustrative set of the first five node-encircling p-cycles for Net 2.

This is an NP-hard combinatorial optimization problem for which exact optimal solutions may be unobtainable for networks above a certain size. A number of relaxations and heuristics are therefore being considered for practical application on large networks. For research purposes, however, we have obtained completely optimal solutions for three networks, of up to 20 nodes and 31 links, given in Table I. The topology of Net 1 was used in the example for Fig. 1. Net 2 is the widely used metro LATA model published by Bellcore [6] and seen in Figs. 8 and 9. Net 3 is a topology from a western Canadian city, used before in [26]. Each test network was provisioned with link capacities that just met the working demand requirements of the shortest-path mapped demand matrix plus an amount of excess (or "spare") capacity given by an optimized "span restorable mesh" network. In other words, the formulation is being challenged to minimize the peak restoration- imposed oversubscription effect on any link, with varying numbers of p-cycles allowed, when all links have no more than the theoretical minimum capacity required for a corresponding span-restorable mesh (in Herzberg's sense [7]). The results were obtained with no hop limit on the size

TABLE II
MAXIMUM OVERSUBSCRIPTION RATIO VERSUS DESIGN NUMBER OF p-CYCLES

	Max *p*-Cycles in Design			
Network	**1**	**5**	**10**	**15**
Net1	1.625	1.0447	1.0417	1.0328
Net2	Infeasible	1.2	1.0448	1.0192
Net3	Infeasible	1.097	1.0449	1.0277

of cycle which could be used as p-cycles and the formulation was run with all possible simple cycles of the graph as p-cycle candidates. Table II shows the maximum oversubscription ratios on any link over all restoration events, dependent on the number of virtual p-cycles the design is allowed to use. As the number of design p-cycles is increased, it is seen that the maximum restoration-state link flows closely approaches the performance of a conventional mesh-restorable network in that 100% link restoration is achieved while approaching the stage where there is almost no impact from restoration on any other working flows of the network (i.e., a maximum oversubscription ratio of 1.0). This shows how application of p-cycles to IP restoration can ensure QoS limits under restoration conditions without requiring significantly greater capacity than does a link restorable mesh network. For illustration, the optimal set of five p-cycles for Net2 is shown in Fig. 8. These five p-cycles offer 100% link restorability with a peak restoration-induced oversubscription factor of 20% (i.e., 1.2 in Table II).

B. Network Design Using IP p-Cycles for Node Restoration

This subsection describes a preliminary design heuristic for node-recovery based on node-encircling virtual p-cycles. The heuristic itself is relatively simple and generates a protecting p-cycle for each network node. For each node, the first step is to mark each node which is adjacent to it. Next, in the subnetwork that remains when the protected node and all attached links are removed, any bridge nodes are discovered along with the associated subgraphs "hinged" on the bridge nodes. If there are no bridge nodes, the procedure described in the next paragraph is performed within the entire subnetwork.

Next, a cycle is found within each of these subgraphs such that the cycle traverses all of the previously marked nodes and bridge nodes, which the subgraph may contain. All the cycles found will be simple, as depicted in Fig. 7, except for the case where the resulting subgraph is a simple segment; for this case, the cycle will be "flattened" and pass through the segment twice.

The cycles from all the subgraphs are then merged to form the protecting p-cycle; if the network did turn out to have bridge nodes the resulting p-cycle will not be simple. Thus, the method operates by splitting the network into subregions within which simple cycles are possible, and merging these together to form a final, possibly nonsimple, cycle which has the desired p-cycle properties; i.e., it covers the nodes adjacent to the protected node, and does not touch on the protected node itself. Fig. 9 shows an example of node-encircling p-cycles which result when this algorithm is run in the Net 2 from the previous sections. Only the first 5 of the 15 for this network are shown.

C. Mappings of Physical-to-Logical Link Failures

The preceding methods expect that only a single failure needs to be restored at a given time. However, an IP network may be established in an environment in which multiple simultaneous logical link failures can arise from a single physical-layer span cut. For instance, in an IP–WDM network, lightpaths (contiguous wavelength paths) in the WDM layer will be used to set up logical link connections between IP routers. From the perspective of the IP network, these appear to be direct independent connections between routers, while in reality a single physical span may carry sections of multiple logical links between IP routers. The consequence of this is that the failure of a single physical span can translate into the functional failure of multiple simultaneous logical links in the IP layer. Fig. 10 gives an example. This must be taken into consideration when designing a set of p-cycles, so that any single physical span failure has a controlled or bounded maximum impact on the simultaneous failure of logical links within the same p-cycle.

One approach is to directly take the mapping of physical to logical failures into account when designing the set of p-cycles. (Note that this measure would not, however, be necessary for p-cycles used to protect against router failures or local port failures, if the underlying WDM transport is inherently restorable within its own layer, say by optical rings or an optical cross-connect based self-healing mesh.)

More generally, however, one can design with the aim being to assign p-cycles to protect links, in such a way that the mapping of single physical failures to multiple logical failures is considered; it can then be ensured that the p-cycle for a given logical link failure is either not disrupted by the simultaneous failure of any other logical link, or the disruption is taken into account by limiting the restoration use of the respective p-cycle. The latter case takes into account that a p-cycle is still functional or useful for a given restoration problem if at least one path remains through the remaining portion(s) of the p-cycle between the end routers of the protected logical link. If the physical to logical fault mapping data are available, this aspect can be added to the design methods of the previous two sections, but there is also a much simpler way to ensure the effectiveness of p-cycles under any unknown physical-to-logical fault escalation. That is simply to restrict the IP layer p-cycles to formation only over logical links which traverse a single underlying physical span, i.e., over hops between physically adjacent routers. In

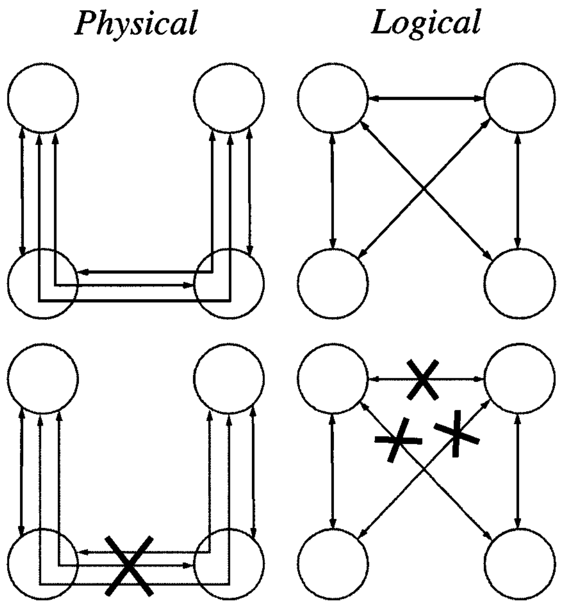

Fig. 10. Mapping of a single physical span failure to multiple logical link failures.

other words, in the graphs of all IP layer logical links, one can formulate the p-cycle design problem to form prospective p-cycles only out of logical links that have a 1 : 1 correspondence to a direct underlying transmission span. Such links represent adjacent routers that happen to enjoy a direct physical connection. This not only simplifies the design problems for both link restoration and node recovery, but it also ensures that for any single physical span failure, no p-cycle would suffer more than a single logical link failure and, so, would continue to be able to offer at least one restoration path.

V. Concluding Discussion

This paper proposes the concept of preconfigured virtual protection cycles (p-cycles) to provide faster restoration of both logical link and router (node) failures wholly within the IP router domain. The initial p-cycle encapsulating and deflection mechanism, p-cycle forwarding, and p-cycle decapsulation mechanism is not dependent on whether the failure is a link or node failure, although p-cycles can be planned separately for protection against either or both types of failure. Because p-cycles are a virtual path construct, they consume no capacity during normal operation, only when actively used in restoration. A related design formulation shows that in test cases where the excess capacity allocations are those that a corresponding optimal span-restorable mesh would have, a set of p-cycles can be obtained through which 100% restorability is provided with almost negligible restoration-induced performance impairment (i.e., max oversubscription close to 1.0).

While this work provides new options for both link and node recovery in the IP layer, a most effective combination of layer-based responsibilities in a future IP over WDM network may be to use virtual p-cycles established in the IP layer to combat node failure and single link failures seen in the IP layer itself, on top of a ring-or mesh-based WDM layer that is self-restoring against physical layer span cuts. A primary advantage of leaving *physical span* restoration in the WDM layer is the practical issue of fault multiplication from one physical span cut to multiple logical link cuts in the IP layer. In a frequently changing network environment, one needs to assess the practicality of constantly monitoring the physical-to-logical fault mapping relationship and updating the virtual p-cycle design in the IP layer accordingly. The consideration of unknown and almost unbounded physical-to-logical layer fault escalation is a serious issue for any IP-layer recovery method, conventional routing table updates included, not just p-cycles. In contrast, if physical span restoration is assigned to the WDM layer, then physical span failure effects will essentially never be seen in the IP layer, and link-protection p-cycles in the IP layer need only deal with the single-logical link failure model. Node-encircling p-cycles are unaffected by the assignment of physical span restoration to either layer, as long as one's design aim is to withstand only any single node failure at a time. Thus, a composite view that we would suggest from a point of practical robustness and minimum information requirements would be threefold: 1) DWDM mesh or ring light-path layer span restoration (against cable cuts or lightpath link failures), 2) a set of about S/2 or fewer IP-layer p-cycles to protect against IP-layer link or interface failures,[9] and 3) a set of N node-encircling p-cycles to protect transiting flows against any single router node failure.

Acknowledgment

The authors would like to thank K. Felske and his group, Nortel Networks, Ottawa, ON, Canada, for making the original suggestion that they consider application of prior work on p-cycles for WDM/Sonet to the IP layer, and also for their recent implementation of the scheme which yielded the experimental result of 110 ms for IP router link restoration time [27].

References

[1] W. D. Grover and D. Stamatelakis, "Protection of routers in a telecommunication network," U.S. patent pending, Apr. 28, 1999.

[2] D. Stamatelakis and W. D. Grover, "Distributed preconfiguration of spare capacity in closed paths for network restoration," U.S. patent pending, July 11, 1997.

[3] W. D. Grover and D. Stamatelakis, "Cycle-oriented distributed preconfiguration: Ring-like speed with mesh-like capacity for self-planning network restoration," in *Proc. ICC'98*, 1998.

[4] ——, "Self-organizing closed path configuration of restoration capacity in broadband mesh transport networks," in *Proc. CCBR'98*, 1998, pp. 145–156.

[5] D. Stamatelakis, "Theory and algorithms for preconfiguration of spare capacity in mesh restorable networks," M.Sc. Thesis, Univ. Alberta, Edmonton, AB, Canada, 1997.

[6] "Digital crossconnect systems in transport network survivability," Bellcore Special Rep., SR-NWT-002514, no. 1, Jan. 1993.

[7] M. Herzberg and S. Bye, "An optimal spare-capacity assignment model for survivable networks with hop limits," in *Proc. IEEE GLOBECOM '94*, 1994, pp. 1601–1607.

[8] R. Iraschko, M. H. MacGregor, and W. D. Grover, "Optimal capacity placement for path restoration in mesh survivable networks," in *Proc. ICC'96*, 1996, pp. 1568–1574.

[9] J. Sosnosky, "Service applications for SONET DCS distributed restoration," *IEEE J. Select. Areas Commun.*, vol. 12, pp. 59–68, Jan. 1994.

[9] S/2 is rough measure of the number of p-cycles needed based on column 3 of Table II, where five logical p-cycles were found to protect networks of $S = 22$ to 31 interrouter links at under 5% oversubscription. The general point is that there are considerably fewer than S such p-cycles required.

[10] T.-H. Wu, H. Kobrinski, D. Ghosal, and T. V. Lakshman, “A service restoration time study for distributed control SONET digital cross-connect system self-healing networks,” in *Proc. ICC’93*, 1993, pp. 893–899.

[11] T. Chujo, H. Komine, K. Miyazaki, T. Ogura, and T. Soejima, “Distributed self-healing network and its optimum spare capacity assignment algorithm,” *Electron. Commun. Jpn.*, pt. 1, vol. 74, no. 7, pp. 1–8, 1991.

[12] M. Decina and T. Plevyak, Eds., *IEEE Communications Mag., Special Issue on Self-Healing Networks for SDH and ATM*, Sept. 1995, vol. 33.

[13] W. D. Grover, *Telecommunications Network Management into the 21st Century*, S. Aidarous and T. Plevyak, Eds. New York: IEEE Press, 1994, pp. 337–413.

[14] C. Huitema, *Routing in the Internet*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[15] J. T. Moy, *OSPF—Anatomy of an Internet Routing Protocol*. Reading, MA: Addison-Wesley, 1998.

[16] ——, “OSPF version 2,” RFC2178, Cascade Communications Corp., July 1997.

[17] G. Malkin, “RIP version 2—Carrying additional information,” Xylogics, Inc., RFC 1388, Jan. 1993.

[18] G. Malkin and F. Baker, “RIP version 2 MIB extension,” Xylogics, Inc., Advanced Computer Communications, RFC1389, Jan. 1993.

[19] C. Hedrick, “Routing information protocol,” Rutgers Univ., Piscataway, NJ, RFC 1058, June 1988.

[20] B. Davie, P. Doolan, and Y. Rekhter, *Switching in IP Networks*. New York: AP Professional, 1998.

[21] Multi-protocol label switching (mpls) charter. Internet Eng. Task Force. [Online]. Available: http://www.ietf.org/html.charters/mpls-charter.html

[22] W. D. Grover and Y. Zheng, “VP-based ATM network design with controlled over-subscription of restoration capacity,” in *Proc. 1st Int. Workshop Design Reliable Commun. Networks (DRCN’98)*, Brugge, Belgium, May 17–20, 1998, Paper O.33.

[23] D. Stamatelakis and W. D. Grover, “Theoretical underpinnings for the efficiency of restorable networks using pre-configured cycles ("p-cycles"),” *IEEE Trans. Commun.*, vol. 48, pp. 1262–1265, Aug. 2000.

[24] ——, “Rapid restoration of internet protocol networks using pre-configured protection cycles,” in *Proc. 3rd Can. Conf. Broadband Res. (CCBR’99)*, Ottawa, ON, Canada, Nov. 7–9, 1999.

[25] W. Grover and D. Stamatelakis, “Bridging the ring-mesh dichotomy with p-cycles,” in *Proc. Design Reliable Commun. Networks (DRCN 2000)*. Munich: Tech. Univ. Munich, Apr. 2000, pp. 92–104.

[26] W. D. Grover, J. B. Slevinsky, and M. H. MacGregor, “Optimized design of ring-based survivable networks,” *Can. J. Elec. Computer Eng.*, vol. 20, no. 3, pp. 139–149, 1995.

[27] K. Felske, “Group communication,” TRLabs Spring Res. Meet., Calgary, AB, Canada, May 3–4, 2000.

[28] W. D. Grover and Y. Zheng, “Dependence of network capacity requirements on the allowable flow convergence overloads in ATM backup VP restoration,” *Electron. Lett.*, vol. 33, no. 5, pp. 362–363, Feb. 1997.

Demetrios Stamatelakis (S’96–M’97) received the B.Sc. and M.Sc. degrees in electrical engineering from the University of Alberta, Edmonton, AB, Canada, in 1994 and 1997, respectively.

He joined TRLabs, Edmonton, as a full-time Research Engineer in the Network Systems Group. His main areas of research interest are in the design and optimization of restorable mesh and ring-based transport networks, clock signal distribution and synchronization, and, more recently, IP network restoration.

Dr. Stamatelakis is a P. Eng. in the Province of Alberta.

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

He was with BNR Ottawa and Edmonton in scientific staff and management before joining the senior management of TRLabs, Edmonton, in 1986. Here, he is currently Chief Scientist-Network Systems and has been instrumental in the development of TRLabs from a start-up to its present level. He is also a Professor in electrical and computer engineering at the University of Alberta. He has served on the on the Editorial Board for the *Journal of Network and Systems Management*. He has authored over 100 journal and conference publications and patents on 25 topics.

Dr. Grover received the TRLabs Technology Commercialization Award for the licensing of restoration-related technologies to industry, in 1996, the 1996/1997 McCalla Research Professorship in Engineering, the IEEE Baker Prize Paper Award for his work on Self-organizing Broadband Networks, in 1999, the IEEE Canada Outstanding Engineer Award, the Martha Cook-Piper Research Prize from the University of Alberta, and the Alberta Science and Technology (ASTECH) award for technology leadership. He is also an NSERC E.W.R Steacie Memorial Fellowship holder for the years 2001–2002. He has served as Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS. He is a P.Eng. in the Province of Alberta.