PESO: Low Overhead Protection for Ethernet over SONET Transport

Swarup Acharya *Bhawna Gupta* *Pankaj Risbood* *Anurag Srivastava*
Network Software Research Department, Optical Networking Division
Bell Laboratories, Lucent Technologies, Inc.
{acharya,bhawna,risbood,anurag}@research.bell-labs.com

***Abstract*— This paper addresses the issue of reliable transport of emerging data services in Ethernet over SONET (EoS) networks that require protection guarantees beyond standard best effort delivery. We argue that the current consensus of using Ethernet spanning tree and a SONET 1+1 protection, while providing reliability, is an inefficient use of resources. Instead, we claim that EoS opens novel opportunities for protection heretofore unavailable in other environments. In particular, the deployment of Virtual Concatenation and LCAS protocols enables "route splitting", creating a *fundamentally new routing paradigm* for circuit-switched environments. We propose a scheme called PESO, appropriate for EoS, with innovative routing, failure notification and switching components. More importantly, it is competitive with SONET protection without its 100% bandwidth overhead. We also suggest a enhancement in LCAS that can further improve PESO's switching time. PESO leverages the underlying protocols, making it extremely attractive to implement and use in practice.**

I. Introduction

Ethernet over SONET (EoS) is increasingly being deployed as the foundation for next-generation data services in service provider networks. Both Ethernet and SONET are the dominant transport technologies — Ethernet for data transport in local area networks (LANs) and SONET for reliable voice transport in metropolitan and wide-area networks (MANs and WANs). EoS also makes good business sense – providers leverage their SONET infrastructure to deliver new services and thereby, create new revenue streams from their legacy hardware[1]. EoS is being also driven today by the availability of multi-service switches that can support both (Gigabit) Ethernet and SONET. In addition to traditional SONET rings, these switches also support more efficient mesh topologies. As providers deploy next-generation SONET, it is expected that mesh architectures will increasingly become commonplace.

Riding the SONET infrastructure over MANs and WANs, it is thus possible to deliver Ethernet data services seamlessly over regional and national geographic areas. This includes Ethernet private-line services providing dedicated bandwidth and *virtual* private-line services that use statistical multiplexing to share bandwidth among various streams. Applications for these services include voice and other enterprise applications such as storage networks and Transparent LANs (TransLANs).

[1]This work is equally applicable to SDH systems, the predominant optical transport system outside the United States. Our use of the term SONET is for simplicity and unless otherwise stated, implies SONET/SDH.

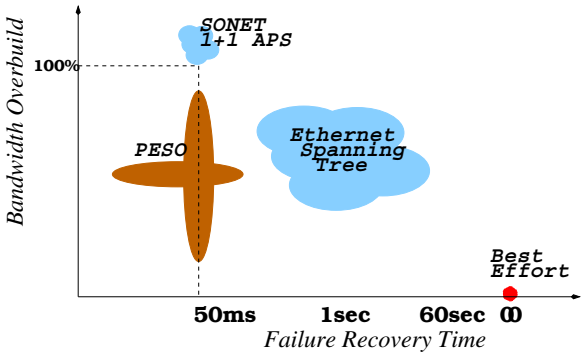

Fig. 1. Bandwidth Overhead - Switch Time Tradeoff

Clearly, these applications highlight the need to provide reliable data transport beyond what standard Ethernet best effort offers.

Ethernet and SONET use very different mechanisms for data protection due to their packet (or, frame) and circuit switched nature. Ethernet uses (Rapid) Spanning Tree Protocol [1], [2] as a protection mechanism. Since Ethernet switches (or bridges) use a spanning tree to forward frames, these protocols dictate how to reconfigure the tree as quickly as possible after a failure. In practice, depending on the size of the network, the reconfiguration can take 10-60 secs during which time there may be traffic disruption. SONET, on the other hand, uses some variant of the 1+1 Automatic Protection Switch (APS) such as UPSR and BLSR, where primary and backup paths are preprovisioned. On failure, the system switches from one path to the other [3]. SONET APS typically takes about 50 ms and is considered the gold standard of reliability. However, the preprovisioning of two paths imposes at least a 100% protection bandwidth overhead.

Both the Ethernet and SONET protection schemes play the same overprovisioning-reliability tradeoff as shown in Figure 1. It presents a schematic of this tradeoff with the x-axis loosely being the failure recovery time and y-axis the typical protection bandwidth required to meet that time. Clearly, best effort (no overprovisioning, no reliability) and SONET APS (100%+ overhead, 50ms protection) are at the two extremes.

Thus, the natural question to answer is what should be the protection mechanism for EoS traffic that requires reliable transport? The current view is to protect at the SONET layer

using a 1+1 APS [4]. In this paper, we argue that while achieving the reliability goals, APS makes very inefficient use of network resource for EoS Services. We show that from a service routing, provisioning and management perspective, EoS should be considered a fundamentally new paradigm. The key novelty is the introduction of new protocols such as Virtual Concatenation (VC) and Link Capacity Adjustment Scheme (LCAS) in next-generation SONET. VC and LCAS enable "traffic splitting" when routing traffic heretofore unavailable in any prior circuit switched environment. In fact, we show that for data applications, we can leverage traffic splitting in conjunction with these two protocols to provide reliability competitive with SONET APS but without its bandwidth overbuild. We call this scheme PESO (**P**rotection Scheme for **E**thernet over **SO**NET) and Figure 1 shows the favorable space for PESO in the tradeoff.

While this paper addresses the issue of Ethernet transport, the concepts are broadly applicable to any data payload (e.g., ATM) over the SONET infrastructure that uses (or, will eventually use) the VC protocol.

A. Contributions and Outline

The key contributions of this paper are as follows:

- This paper recognizes the ability to do traffic splitting in EoS environments and leverages this to provide novel reliability options.
- We propose a reliable transport mechanism for EoS called PESO that provides protection competitive to SONET APS without its high 100%+ bandwidth overhead.
- PESO introduces innovative flow-based routing algorithms that require low bandwidth overhead to provide competitive protection.
- We highlight a key shortcoming of the LCAS protocol and present a critical fix. This enhancement called FLCAS significantly improves its failure notification time. We leverage FLCAS to provide an enhanced variant of PESO.
- Finally, the goal of this work is to develop a solution that is not only novel but is also implementable within the existing EoS protocols and infrastructure. The PESO scheme proposed in this paper meets this requirement making it easy to integrate into currently deployable network hardware.

The rest of the paper is as follows. In Section III and IV, we present technical details on the VC and LCAS protocol as a requisite background for the PESO algorithm. Readers familiar with intricacies of these two protocols can directly skip to Section V that gives the PESO overview. Section VI explains the routing component of PESO. Section VII proposes a modification of LCAS protocol to achieve 50ms restoration. We summarize the performance of PESO in Section VIII and present related work in Section IX. Finally we conclude.

II. Routing Impact of VC and LCAS

The SONET standard imposes the well known STS-$< 1, 3c, 12c, 48c, \cdots >$ hierarchy for bandwidth demands. An underlying requirement when provisioning new circuits is that all the n slots of the STS-nc circuit be assigned contiguously. This property has been referred to as *contiguous concatenation* and has worked well for voice traffic. However, contiguous concatenation introduces serious inefficiencies for EoS systems as the data rates do not match up with the SONET rates. For example, the closest SONET hierarchy to Gigabit Ethernet (1Gb) is STS-48 (2.5Gb), a 60% wastage of bandwidth. Moreover, the contiguous requirement also causes bandwidth fragmentation [5], [6] further lowering network utilization — even though n slots may be available on a link, it may not be adjacent causing demands to be rejected.

The VC protocol was introduced to addressed both these problems. It allows a STS-nc demand to be split into k pieces of bandwidth n/k. The source node sends traffic down these k *members of the Virtual Concatenation Group*(VCG) and the sink node reconstructs the data stream back. Consequently, a 1GB demand can be mapped to 7 STS-3s (or, 21 STS-1s), imposing only a 8% overhead. More importantly, for this discussion, the standard allows each of these VCG members to be routed *independently*. Figure 2 shows the contrast between VC and contiguous concatenation, with the example on the right demonstrating a VCG of four members. We provide some specifics of the VC protocol in Section III and for a deep exploration the reader is referred to [7].

The availability of VC and the unique requirements for protection of data traffic combine to make EoS a fundamentally new paradigm for routing and service management. In nearly all circuit switched technologies (e.g., SONET, ATM, MPLS), it has never been possible to satisfy a bandwidth demand by splitting it into smaller-sized pieces and allowing the traffic to follow different paths across the network. In its absence, either a circuit is fully backed up causing a 100% overhead (or, more if the backup route is longer) or, not at all. Once a traffic is allowed to be split among different VCG members, it provides the luxury of being creative with the protection mechanism. For example, by backing only *some* of the members, one can provide a probabilistic guarantee of reliability with lower bandwidth overhead.

Voice and data services also have fundamentally different reliability requirements. While voice generates constant bit rate traffic, data traffic is bursty giving the advantage that data applications can continue operation, possibly at a lowered performance, even if the capacity along the path is reduced. For example, a wide-area enterprise storage network, while slowing down, can still function if failures reduce the underlying network capacity by 50%. In other words, unlike voice which has a binary service up or down condition, data services have a gradual degradation in "quality" as the available bandwidth reduces. Indeed, if the circuit was provisioned for the peak rate, the impact of lowered capacity along the path may not even be noticeable in many cases.

We aim to exploit this observation in PESO. We argue that the SONET 1+1 APS was designed for the voice *all-or-nothing* protection and thus, an overkill for data traffic. Instead, we claim that as networks start to get more and more reliable, baring catastrophic problems, link failures are often transient. Thus, using VC, we aim to route VCG members in a manner

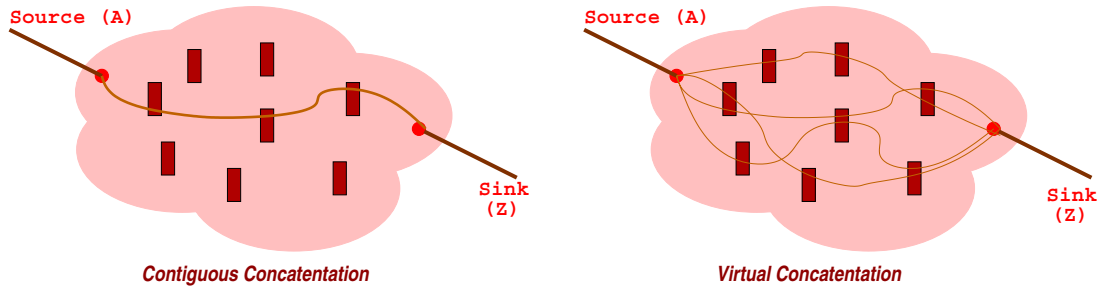

Fig. 2. Routing with Contiguous and Virtual Concatenation

such that link or, node failure(s) does not shut down the entire stream but allows some amount of traffic to still continue to flow. More importantly, this creates a novel way to look at the bandwidth overbuild-reliability tradeoff — *given that the operator is willing to run a data service at some diminished rate for a short period of time, what is the* minimal overbuild *that will meet those requirements?*

Indeed, the LCAS protocol in next-generation SONET allows us to actually do this seamlessly. LCAS supplements VC by allowing VCG members to the dynamically added or deleted without impacting service. The key benefit of LCAS for our discussion is that it enables a circuit to continue operation, albeit at a lower capacity, even after a failure of one (or, more) of the VCG members. Clearly, a failure causes a short service hit; however, the protocol has an in-band mechanism to work around this failed member and continue operation on the active ones. In effect, LCAS already offers an in-built resiliency mechanism after failure. Thus, using LCAS for protection, reduces the problem to finding an efficient routing strategy that minimizes the volume of traffic loss due to any failure. The routing component of the PESO algorithm achieves exactly that. We will present the specific details of the LCAS protocol in Section IV. We highlight there that as currently standardized, the LCAS algorithm has some handicaps in its ability to provide protection. In Section VII, we show how LCAS may be enhanced to meet out reliability goals.

A. Novel EoS Services Options

In this section, we list three novel routing strategies that are applicable to an EoS setting. Each of these three play the overbuild-reliability differently. The first two scenarios require no overprovisioning of bandwidth but have strict limitations on the acceptable "loss" in service on failure. The last option follows the more traditional approach of overprovisioning to achieve protection.

- **Scenario** $\mathcal{A}$: *Route a data service(e.g. 1GbE circuit) such that a single node or, link failure does not affect more than some* $X\%$ *of the total bandwidth.* This reflects the case when the traffic is provisioned for the peak rate but operator needs to ensure that the average rate, say 30% below the peak rate, is maintained even after failures.
- **Scenario** $\mathcal{B}$: *Route a data service such that a single node or, link affects the* minimum *bandwidth.*
- **Scenario** $\mathcal{C}$: *Route a data service with* overprovisioning *such that* minimum overbuild *is required to protect against a single node or a link failure.*

Going forward, we will refer to the service requirements as one with *No Overprovisioning* or, $\mathcal{NOP}$ such as Scenarios $\mathcal{A}$ and $\mathcal{B}$ and those that *Require Overprovisioning*, or, $\mathcal{ROP}$ such as Scenario $\mathcal{C}$. We will present routing algorithms for each of these cases in Section VI. Clearly, the $\mathcal{NOP}$ category is a novel addition to bandwidth "guaranteed" EoS services.

III. Virtual Concatenation Protocol

In this section, we provide details of the VC protocol relevant to our discussion. As we briefly introduced earlier, transporting various data services via contiguous concatenation creates inefficient mapping of their frame rates to the closest SONET equivalent. Moreover, it leads to bandwidth fragmentation requiring expensive defragmentation operation to recover capacity [5], [6].

VC, standardized by ANSI and ITU-T [8], [9], addresses the rate mismatch problem by providing a finer granularity for concatenation. It enables multiple smaller rate circuits to be combined to create higher rates. For example, with VC, a Gigabit Ethernet circuit can be mapped to 21 STS-1 or 7 STS-3c signals, resulting in 92% efficiency. Since, VCG members are allowed to be on non-contiguous time slots, it also avoids the fragmentation problem. More importantly, they can also be independently routed. A VCG circuit is generally identified as STS-$X-Yv$, which describes a virtually concatenated circuit consisting of Y STS-Xc members.

A. Virtual Concatenation Operation

Currently, the VC protocol requires a VCG to be entirely composed of equal size members. It specifies two kind of concatenation, namely Higher Order (HO) and Lower Order (LO). HO concatenation is for transporting frame rates above STS-1 and requires VCG to be consist entirely of STS-1 (≈52Mbps) or STS-3c (≈156Mbps) members. The LO Concatenation is for carrying sub STS-1 rate services (10BaseT etc.), with members of size VT-1.5 (≈1.5Mbps) or VT-2.0 (≈2Mbps). Due to space constraints, we will focus only on the HO case.

SONET uses `H4` byte of Path Overhead (POH) to carry VC header for HO concatenation. Each SONET frame, sent every 125μs, carries one `H4` byte. The entire VC header information takes 16 consecutive SONET frames (or, one multiframe) and

thus, sent every 2ms (125μs * 16). The VC header information is carried on *all* the VCG members.

Since, VCG members can be diversely routed from one another, they can incur different delays and arrive at different times at the sink. The sink node uses Multiframe Indicator (MFI) field of the multiframe header for the phase alignment of members. This 12 bit field is a running frame number, which allows for a compensation of differential delay of 256ms among members. MFI is composed of two parts: MFI-1 (4 bits), which is incremented by 1 in every frame, and MFI-2 (8 bits), incremented every multiframe. VC also assigns a unique Sequence Number (SQ) to each member, which is used by sink node for reconstruction of original packet. For example, if a VCG has four members A,B,C and D; they'll be assigned SQ values of 0, 1, 2 and 3 respectively. MFI bit values are identical across all the members for a particular multiframe whereas SQ value is different for each member and refers to their relative position in the mapping of data to VCG. Since, SQ is a 8-bit field, VC protocol restricts the total numbers of members to a maximum of 256.

IV. LCAS Protocol

In this section, we explain a key shortcoming of the VC protocol and describe how LCAS address it. We also briefly describe some operational details of LCAS relevant to this work.

When one or more members of a VCG are adversely affected due to a network or link failure, the data can be corrupted even if some of the VCG members are still active. Consider an example where a packet stream of "123456781234..." was byte interleaved on to four members A,B, C and D. Now, consider a network element failure resulting in the failure of member D. Since, source node is not aware of the failure of member D, it'll keep mapping the data to all four members. However, failure of member D will force the sink node to perform packet assembly only with the three active members. Hence, the absence of member D will make the reconstructed data look like "123567123...", resulting in a malformed packet. Therefore, even though, VC theoretically provides resiliency by routing the members diversely, no practical benefits can be achieved by it.

LCAS protocol as described by the [10] remedies this problem. It provides a mechanism for the sink to notify the failure of a VCG member to the source using the still active members. After receiving such notification, the source node temporarily removes the failed member from the VCG group and starts sending data only on remaining active members. LCAS also enables scheduled addition and deletion of traffic from a VCG in a hitless manner. It can also be used to detect the restoration of a failed member and add it back to the VCG without requiring any operator intervention.

A. VCG Member Status and LCAS Refresh Time

LCAS uses the multiframe header structure from VC for carrying its header information and borrows some unused bits for its purpose. It utilizes the MFI and SQ fields of the VC header and 7 additional bytes from the 16 byte multiframe structure. Unlike VC, LCAS is a bidirectional protocol and the source and sink nodes continually exchange messages, monitoring all the VCG members.

The sink uses the MST field to transmit the status of a member to the source. It is set to OK for all members currently present and actively carrying the data and set to FAIL for all the failed and unused members. LCAS transmits status of all 256 potential members (maximum allowed size of VCG). As each multiframe can only contain the status of 8 members, it takes 32 multiframes to send all the 256 statuses. Since, it takes 2ms to transmit one multiframe, it requires a duration of 64ms (2*32) for a member's status to reappear in the refresh cycle. This duration of 64ms is defined as LCAS refresh time or, LRT. As, we show later that LRT plays a critical role in the failure notification component of PESO.

LCAS uses CTRL field for synchronizing the information between source and sink node. CTRL field carries the control command for the VCG members. The values it can take relevant to this discussion are:
NORM: This member is currently part of the VCG and is a normal member.
EOS: This member is also part of the VCG and its SQ number is highest among members
IDLE: This member is not part of the VCG.
DNU: DNU stands for *Do Not Use*. It informs sink not to use this member for the assembly.
These control commands together with the MST enable the addition and deletion of members to VCG. NORM control word is carried for all members except the member with the highest SQ number which carries EOS (End of Sequence). The sink node uses EOS field to determine the total size of the VCG as provisioned. We now describe the LCAS process for temporary member removal from the VCG that is a key component of PESO.

A VCG member is temporarily removed from the VCG when it incurs a failure. On a member failure, the sink node first detects it and drops the failed member for packet reassembly. In parallel, it notifies the source about the FAIL status of the member via the MST field. The source, on receiving the FAIL status, alters the CTRL field to DNU in the multiframe header of failed member and then stops putting data on it.

When the failed member recovers, the sink node detects this and sends a status of OK to the source. The source then changes the CTRL field from DNU to NORM and starts putting data on this member. After this the circuit goes back to its original capacity. The scheduled addition and deletion of members in the VCG follows a similar process and the reader is referred to the ITU-T standard for further details [10].

In the current LCAS implementation, LCAS Refresh Time (LRT) of 64ms may be unacceptable in many environments of tight reliability constraints. In Section VII, we present an enhancement to significantly reduce the LRT.

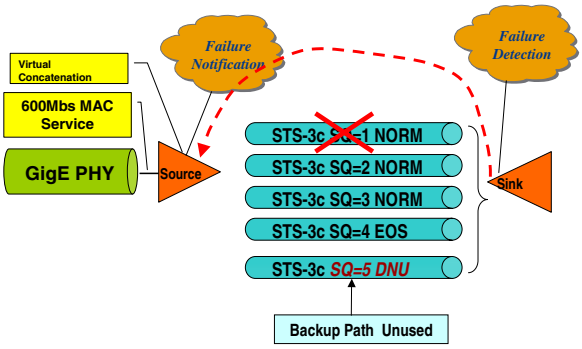

Fig. 3. PESO before failure

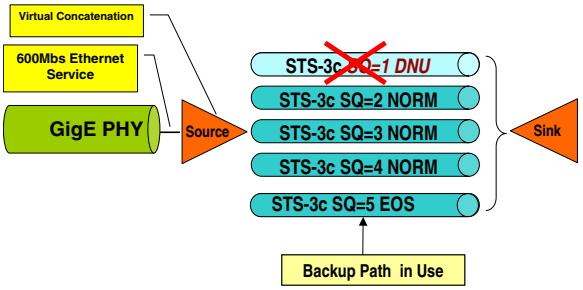

Fig. 4. PESO after failure

V. PESO Overview

In this section, we give an overview of the PESO protection scheme. PESO consists of three key components — a) routing, b) failure notification and c) protection switching. Depending on the reliability requirement being a $\mathcal{ROP}$ or, a $\mathcal{NOP}$ scenario as described in Section II-A, each of these components function slightly differently.

The PESO protection scheme can be designed to handle any specific failure model, however, for this discussion we assume a single link or node failure.

Consider a $\mathcal{NOP}$ scenario (e.g., Scenario $\mathcal{A}$ and $\mathcal{B}$). In these cases, PESO routing determines the number of VCG members necessary to provide the appropriate reliability and suggest routes for them. On failure, LCAS resizes the bandwidth in LRT time. Even though no additional bandwidth is overprovisioned, a weaker form of reliability is achieved by enabling data flow even after network failure. This is in contrast to a traditional all-or-nothing protection where the circuit would go down without any additional bandwidth overprovisioned.

The $\mathcal{ROP}$ scenario (e.g., Scenario $\mathcal{C}$ in Section II-A) is similar to the more traditional reliability requirements — the operator is willing to overprovision in order to be continue at full throttle even after a failure. The PESO approach in this case is to preprovision additional bandwidth as "backup" members in the VCG in addition to the "primary" members that would normally carry traffic. The PESO routing component provides the necessary routes for all the members. On failure, LCAS is used to switch traffic from the primary to the backup members. Assuming the backup bandwidth suffices, after a disruption of the LRT time, the circuit is restored.

The effectiveness of PESO, in terms of protection bandwidth overhead, is dependent upon the availability of diverse routes in the network. In the traditional SONET network where most of the deployment is in UPSR/BLSR rings, the network is limited to two diverse routes. As a result, PESO also will require a 100% protection bandwidth. However, PESO scheme will be extremely effective in mesh architectures, where availability of diverse paths is high. As service providers move to next-generation SONET networks, they are migrating towards mesh due to the efficiencies it provides over rings. In Section VI, we propose a novel routing scheme which minimizes the protection bandwidth overhead requirement.

PESO's protection speed is dependent upon the failure notification mechanism used by sink to notify source node about the failure of a primary member. In the current LCAS standard [10], time taken by the sink to detect and report failure is of the order of 64ms/128ms for Higher Order and Lower Order concatenation respectively. PESO proposes a faster version of protocol, FLCAS, presented in Section VII, which substantially brings down failure notification time, to less than 50ms, for most of the cases.

A. Protection Switching

In this section, we describe how PESO recovers from a network or link failure. As described above, in the $\mathcal{NOP}$ case, PESO simply uses the LCAS protocol for the member failure detection and their removal. Thus, we do not describe the details further and refer the reader back to Section IV.

The novelty in the switching component of PESO is for $\mathcal{ROP}$ scenarios such as the Scenario $\mathcal{C}$ in Section II-A. For such cases, VCG members are partitioned into primary and backup members. Once the primary and backup members have been identified and routed, the source node starts sending traffic on the primary members. The backup members do not carry any traffic during normal operation. The primary members carry `NORM` in their `CTRL` field while backup members carry `DNU` to ensure that the sink does not pickup any data from them. When a link or network element failure results in failure of a primary member, the LCAS protocol at sink detects and reports the failed member's status `FAIL` back to the source.

The PESO protection switching kicks in after the source node receives the notification of a member failure. It ensures that the failed primary members are temporarily removed from the VCG and instructs already provisioned backup members to take over. Upon notification of a member failure, PESO redirects the source to start sending normal (`NORM`) in the `CTRL` field of backup member and `DNU` on the failed member. This swapping of `CTRL` fields is achieved in same multiframe header. Once this multiframe header information is completely transmitted, the source switches the data previously transmitted on failed primary members to backup members.

Since, primary members can share routes, a single failure (network or link) can affect several primary members. Again, it is the responsibility of the routing algorithm to ensure that sufficient number of backup members are setup to support any failure. Figure 3 and 4 shows an example of the PESO

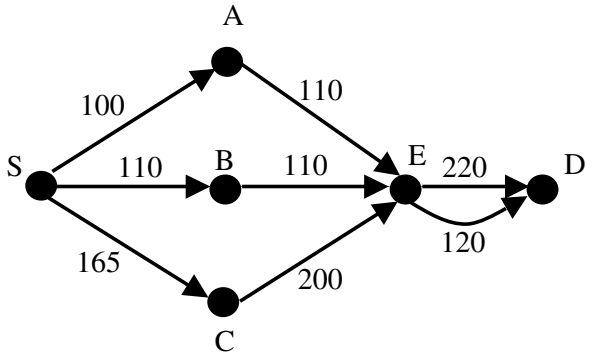

Fig. 5. Original network

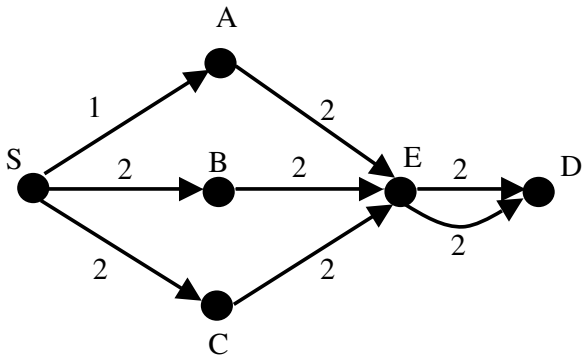

Fig. 6. Transformed network

protection scheme. To transport a 600M Ethernet service, a 4 member VCG is setup. Each VCG member is a STS-3c circuit routed diversely from one another. The backup member(SQ 5) is also diversely routed from primary members(SQ1-4). The protection bandwidth required to protect this VCG is 25%, an extremely low bandwidth overhead. As shown in Figure 3, backup members do not carry traffic during normal operation and instead have DNU in their CTRL field. Figure 4 shows the swapping of the CTRL field of primary and backup members after failure.

Clearly, PESO requires some support from the network element to enable this protection switch mechanism. The element has to provide some means to mark members as primary and backup and have the necessary logic to do the switch on failure. However, this is a relatively minor requirement compared to the complexity of supporting VC and LCAS.

One may also conceive of a variation of PESO scheme for the $\mathcal{ROP}$ case. Instead of partitioning into primary and backup members, one can spread the traffic equally among all the members. On failure of a member (active or backup), it is simply removed from the VCG using LCAS. The advantage of this scheme is that there is no special processing required once the source is notified — it simply follows the LCAS protocol on failure. However, it has a fundamental drawback. In PESO, traffic is not impacted if a backup member goes down since it does not carry any valid data (has a DNU flag). However, in this case, any member going down has a LRT time hit in traffic. Thus, PESO provides an extra level of reliability which plays a critical role since the diversity of VCG members increase the probability of introducing failures.

Finally, we briefly focus on the protection switch time. For both the $\mathcal{NOP}$ and the $\mathcal{ROP}$ case, the failure detection and notification is via LCAS requiring the LRT of 64ms. Swapping the backup and primary members takes another 2ms for the $\mathcal{ROP}$ case and the same for the removal of failed members in $\mathcal{NOP}$ case when done by LCAS. So, total worst case switch time can be 66ms which is acceptable for most data applications. Note that we are ignoring signal propagation delay($\approx 5\mu s/km$) as they are negligible compare to switching times and are fixed for all the protection schemes. For the services which have even more stringent requirements, Section VII proposes an enhancement to lower the switch time.

VI. PESO Routing

In this section, we propose a novel routing scheme to enable the routing for virtually concatenated circuits. As mentioned in previous section, VC provides an unique opportunity of splitting the traffic flow on multiple paths (VCG members) carrying smaller rate traffic. PESO routing algorithms are intended to exploit this flexibility. The PESO routing algorithm accounts for both the $\mathcal{NOP}$ and $\mathcal{ROP}$ scenarios. We describe the routing to address the scenarios introduced in Section II-A and consider them below in order.

A. Routing for Scenario $\mathcal{A}$ (Algorithm α)

This is very likely be a very common case as service providers may not be willing to put additional bandwidth to protect data services. However, they may be interested in limiting the extent of the damage on failures. Moreover, critical services tend be provisioned at their peak rates and thus, a temporary failure may not necessarily impact the end user performance.

Algorithm α shown on next page addresses this scenario. Consider the network in Figure 5, where the requirement is to transport a 120Mbps Ethernet Service from source S to sink D such that single failure does not impact more than $2/3^{rd}$, or, 67% of the traffic. Transporting a 120Mbps Ethernet service requires a STS-3c($\approx$156Mbps) equivalent frame rate on SONET side. As per the VC standard, it can be achieved by either one STS-3c circuit or three STS-1 circuits. For the moment, we assume this service is transported on a three member STS-1 VCG. We will highlight later the trade-offs involved in choosing between STS-3c or STS-1 as members. Since, the requirement is that at least 40Mbps traffic (33% of 120Mbps) is protected against one failure, it is necessary for at least one STS-1 member to survive any failure.

Now consider the network of Figure 6, which represents the network of Figure 5 with link capacities altered. These new link capacities reflect the largest SONET rate (STS-Nc) they can carry. For instance, link S-A has available bandwidth of 100Mbps, which makes it large enough to carry only a STS-1 ($\approx$52Mbps), hence a capacity of 1 unit. Thus, routing of a 120Mbps service in Figure 5 is equivalent to routing (or pushing) 3 units of flow in the network of Figure 6. However, to ensure that no link failure results in failure of more than

PESO Routing Algorithm α

INPUT:
Network $G(V, E)$, new demand D for bandwidth $\mathcal{B}$ and the maximum bandwidth $\mathcal{X}$ allowed to be impacted on failure.
PROBLEM:
Route D in G such that a single link failure does not affect more than $\mathcal{X}$ amount of the traffic.
OUTPUT:
A set of routes for members of the VCG carrying D.
ALGORITHM:

Let STS-$\mathcal{F}$c and STS-$\mathcal{Y}$c be the smallest SONET frame rate that can carry $\mathcal{B}$ and $\mathcal{X}$ respectively.
For all edges in E:
  Set their capacity to highest SONET rate (N units for STS-Nc) they can carry or, to $\mathcal{Y}$ units whichever is smaller.

Find minimum cost flow of $\mathcal{F}$ units in G.

PESO Routing Algorithm β

INPUT:
Network $G(V, E)$, a traffic demand D of bandwidth $\mathcal{B}$.
PROBLEM:
Route the demand D in G such that a single link failure affects the minimum amount of traffic.
OUTPUT:
A set of routes for members of the VCG carrying D.
ALGORITHM:

Let STS-$\mathcal{F}$c be the smallest SONET frame rate that can carry $\mathcal{B}$.
Choose $\mathcal{Y}$ between 1 and $\mathcal{F}$ by binary search.
  For all edges in E:
    Set their capacity to highest SONET rate (N units for STS-Nc) they can carry or, to $\mathcal{Y}$ units whichever is smaller.
  Find minimum cost flow of $\mathcal{F}$ units in G.

Smallest $\mathcal{Y}$ for which $\mathcal{F}$ units of flow can be routed in G, is the desired solution.

two members (or two units of flow), its necessary that no link is allowed to carry more than two units of flow. To capture this constraint, we restrict the link capacities to a maximum two units. For example, though the link S-C has three units of capacity (as it can support a STS-3c), it has been assigned two units.

For routing $\mathcal{F}$ (or, 3 units in this case) units of flow, any of the standard flow routing algorithms can be used. For example, path augmentation based maximum flow algorithms from Ford & Fulkerson [11] or, Edmonds & Karp [12] can be used to route the flow. As our requirement is to route only $\mathcal{F}$ units of flow, these algorithms can be stopped after the sufficient flow is routed. In any given network, there may be various distinct solutions for routing $\mathcal{F}$ units of flow, therefore it may desirable to obtain the smallest cost solution. Such smallest cost feasible flow solutions can be easily computed using *Minimum Cost Flow* algorithms [13]. In Section VI-D, we analyze the overall complexity of algorithm α based on one such min-cost flow algorithm. Once $\mathcal{F}$ units of flow is routed, $\mathcal{F}$ paths of unit flow are extracted and each path is used to route a VCG member made up of a STS-1 circuit.

Note that the Algorithm α only handles link failures and not node failures. For example, in Figure 6, failure of the node E will result in complete failure of the entire VCG. However, we can address this by doing a standard graph transformation [13] of splitting each node into an ingress and egress node and inserting a link of requisite capacity between them. Therefore, for rest of the paper we only address link failures and assume that node failures can be accounted for using standard transformations.

B. Routing for Scenario $\mathcal{B}$ (Algorithm β)

This problem is similar to Scenario $\mathcal{A}$ except that the requirement is to minimize the extent of damage on failure. In a network of high route diversity, all the flows can be routed on disjoint paths where any failure will affect only unit flow. On the other extreme, in a network with no diversity where the all flow is carried on one route, a failure will bring the entire traffic down. Therefore, the problem of minimizing the damage on failure requires finding a solution in between these two extremes. Algorithm β above achieves that. Once the required value of flow ($\mathcal{F}$) is determined from the bandwidth ($\mathcal{B}$), Algorithm β chooses a value of $\mathcal{Y}$ (damage on failure) by doing a binary search between 1 and $\mathcal{F}$. For each value of $\mathcal{Y}$, it first alters the link capacities as in algorithm α and then attempts to route the flow of $\mathcal{F}$ units. For each value of $\mathcal{Y}$, algorithm β finds a solution (if there exists one) where VCG circuit of bandwidth $\mathcal{B}$ can be routed such that no link failure will affect more than STS-$\mathcal{Y}$c (assuming STS-1 members) amount of bandwidth. The smallest value of $\mathcal{Y}$ for which $\mathcal{F}$ units of flow can be routed in G, is the best solution.

C. Routing for Scenario $\mathcal{C}$ (Algorithm γ)

This scenario permits an additional amount of bandwidth (in addition of $\mathcal{B}$) which can be used to completely restore the circuit after a failure. As indicated in the overview, PESO preprovisions additional VCG members for this case. To minimize the number of additional VCG members required for protection, its necessary that the minimum number of members (or units of flow) are affected on failure. In other words, if $\mathcal{Y}$ ($1 <= \mathcal{Y} <= \mathcal{F}$) members are allowed for protection bandwidth, no link should carry flows from more than $\mathcal{Y}$ members (or, $\mathcal{Y}$ units of flow). Thus, the problem of provisioning $\mathcal{F}$ members to transport a VCG of bandwidth $\mathcal{B}$ with complete protection (or additional members) can be mapped to following flow routing problem: *Route $\mathcal{F} + \mathcal{Y}$ units of flow in Graph G such that no link carries more than $\mathcal{Y}$ units of flow.*

Again as before, all link capacities in G reflect the largest SONET rate they can carry and are restricted to a maximum of $\mathcal{Y}$ units. The Algorithm γ executes this procedure for all values of $\mathcal{Y}$, between 1 and $\mathcal{F}$ by a binary search. And, the smallest value of $\mathcal{Y}$ which solves the above mentioned problem, is the best solution.

We now make a key observation. Algorithm γ does not require that *the primary and the backup members have to be diversely routed*. This is a fundamental difference from

PESO Routing Algorithm γ

INPUT:
Given a graph $G(V, E)$, a traffic demand D of bandwidth $\mathcal{B}$.
PROBLEM:
Route D in G with minimum additional protection bandwidth such that a single failure does not impact traffic.
OUTPUT:
A set of routes for members of the VCG carrying D.
ALGORITHM:

Let STS-$\mathcal{F}$c be the smallest SONET frame rate that can carry $\mathcal{B}$.
Choose $\mathcal{Y}$ between 1 and $\mathcal{F}$ by binary search.
  For all edges in E:
    Set their capacity to highest SONET rate (N units for STS-Nc) they can carry or to $\mathcal{Y}$ units whichever is smaller.
  Find minimum cost flow of $\mathcal{F} + \mathcal{Y}$ units in G.

Smallest $\mathcal{Y}$ for which $\mathcal{F} + \mathcal{Y}$ units of flow can be routed in G, is the desired solution.

standard protection algorithms that enforce this constraint. In fact, γ simply ensures that each link carries utmost $\mathcal{Y}$ units of flow without enforcing any diversity. Therefore, if a link failure affected $\mathcal{I}$ active and $\mathcal{J}$ backup members, then $\mathcal{Y} >= \mathcal{I} + \mathcal{J}$. Since, γ routed $\mathcal{Y}$ backup members (or, $\mathcal{F} + \mathcal{Y}$ units of flow) in total, $\mathcal{Y} - \mathcal{J}$ backup members definitely survived the failure. However, $\mathcal{Y} - \mathcal{J} >= \mathcal{I}$. Therefore, its guaranteed that at least $\mathcal{I}$ backup members are still present to support all the failed active members.

This loosening of the diversity requirement ensures that γ is also effective in environments that may have only limited connectivity among the nodes and in networks of smaller sizes. This make the algorithm extremely attractive in practice, particularly, as providers gradually build up their mesh infrastructure.

Finally, we briefly comment on the impact of the bandwidth of the VCG member. In all the three algorithms mentioned above, we assumed STS-1 based VCGs. But one may also consider STS-3c members as allowed by VC protocol. The tradeoffs are as follows. Use of STS-1s increase the probability of the requisite routes being found compared to STS-3c. However, they also incur higher network management overhead of provisioning three times as many members. Due to its lower granularity, STS-1 also enables a better match between the data rate of the service and the SONET rate for the VCG. Moreover, since PESO protects against a single failure, in the best case where the network admits high diversity, STS-1-based VCG will require a lower protection bandwidth compared to a STS-3c. Thus, it is preferred to build the VCG from STS-1 unless the management overheads are prohibitive.

D. Complexity Analysis

In this section, we summarize the complexity of the algorithms. The assignment step of link capacities to equivalent SONET rate takes order of $O(E)$. Minimum cost flow problems can be efficiently solved using [14]. [14] has a running time of $O((ElogV)(E + VlogV))$ and is known to be among the fastest available algorithms. However, for our application a simpler algorithm such as successive shortest path [15] will give better results. [15] is a pseudo-polynomial algorithm which computes a shortest path in each iteration and maintains an optimal solution. In worst case, it may require $\mathcal{F}$ shortest path computations to route $\mathcal{F}$ units of flow resulting in a running time of $O(\mathcal{F}ElogV)$. Thus, the worst case complexity of flow routing step and algorithm α is $O(\mathcal{F}ElogV)$. Algorithm β and γ, use binary search, which in the worst case may make $log(\mathcal{F})$ invocations of flow routing step. Thus, their worst case complexity will be order of $O(\mathcal{F}Elog(\mathcal{F})logV)$. Its important to note here that $\mathcal{F}$ can not be an arbitrarily large number. It refers to the SONET equivalent frame rate (STS-$\mathcal{F}$c) for traffic demand of bandwidth $\mathcal{B}$. The highest SONET frame rate currently defined by the standards is STS-768c and thus, $\mathcal{F}$ will never exceed that.

VII. FAST LCAS

In this section, we describe the details of *Fast LCAS* (FLCAS) protocol which proposes some critical enhancements to the existing LCAS protocol. FLCAS reduces the LCAS refresh time and provides some additional mechanisms to enable a faster failure notification. For space constraints, we describe FLCAS details only in context of HO concatenation. The LO case can be extended with minimum modification.

In its current implementation, the LCAS protocol does not account for the actual size of the VCG while sending the member statuses. It sends a complete cycle of 256 member statuses even when there may be fewer members present in VCG. Consequently, in cases of a member failure, the failed member's status cannot be sent back to source until its turn to send status arrives in the refresh cycle. Therefore, in the worst case, it could take a complete refresh cycle of 64ms (order of LRT) to send the status of a failed member. Hence, after a member failure, the traffic may be disrupted for at least that time. Since PESO is built on this component of LCAS, it imposes a similar traffic hit. For most data applications, this is well within acceptable limits (particularly, given the low protection bandwidth overhead). However, for mission critical services requiring protection switch time competitive with SONET APS, this may be prohibitive. FLCAS aims to address that.

The LCAS operation is shown in Figure 7 for a VCG with 30 members. The first column shows the multiframe number and the next two show the `MFI-2` field and the 8 member statuses (`MST` field). `MFI-2` bits are used by VC for frame alignment and reconstruction at the sink side and are sequentially incremented in every multiframe from 0 to 255. To determine which 8 members whose statuses (`MST`) are being sent, LCAS uses the lowest 5 bits of the `MFI-2`. Thus, the lowest 5 bits being 0(00000) implies that the status field carries the state of the first eight members (`SQ`=<0-7>) and the lowest 5 bits being 10 imply members <80-87>. Since in this example, there are 30 members, the first 4 multiframes capture all their statuses and the `MST` bits beyond the fifth

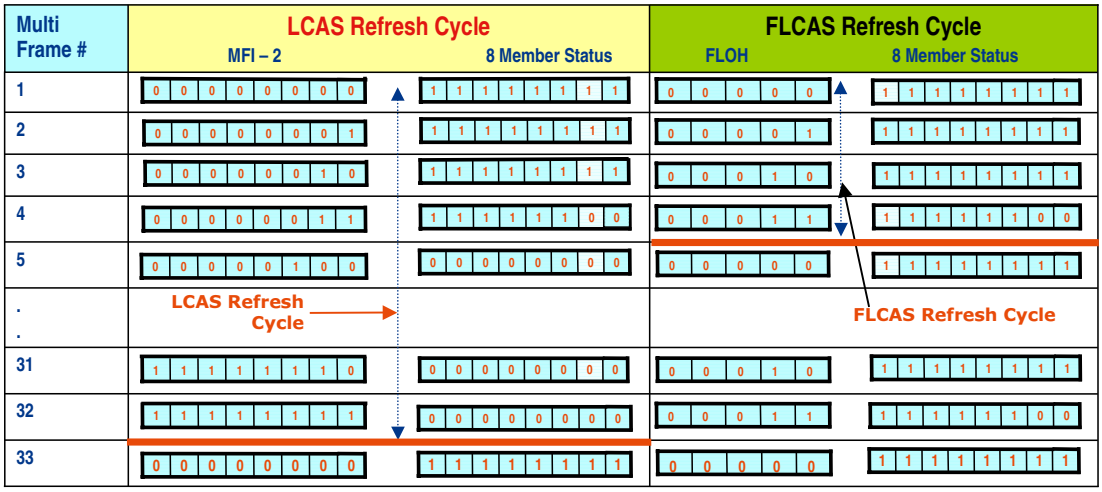

Fig. 7. Refresh Cycle for LCAS and FLCAS for a 30-member VCG

multiframe are wasted (and thus, 0). Thus, even though all the member statuses can be refreshed in four multiframes, or, 8ms (4*2ms), LCAS requires the entire 64ms cycle.

FLCAS solves this problem as follows. Instead of using, `MFI-2` bits to identify the member information, FLCAS uses *five extra bits* (from the ITU-T reserve bits) from multiframe header. These five bits are called *FLCAS Overhead*, or, `FLOH`. Since, `FLOH` bits are not used for frame alignment, they are not required to be sequentially incremented from 0 to 31. They are solely used to identify the members and serve the same purpose as the lowest five bits of the `MFI-2` field.

However, unlike `MFI-2`, the `FLOH` field takes values based on the number of members in the VCG. Values taken by `FLOH` bits depend upon the current size of VCG. Specifically, for a VCG with N member, `FLOH` bits take $\lceil N/8 \rceil$ values from 0 to $\lceil N/8 \rceil - 1$. `FLOH` bits just cycle through these values resulting in a shorter cycle and refresh time. A `FLOH` value of X specifies that the status of members with `SQ` number $X*8$ to $X*8+7$ is present in the multiframe header. For the 30 member VCG shown in the Figure 7, `FLOH` takes values between 0(00000) and 3(00011), covering status of 8 members for each value and then recycling back to 0(00000). Since, FLCAS only sends status of the members currently present in the VCG, it enables a faster refresh time. As in Figure 7, for this 30-member VCG, FLCAS refreshes every 4 multiframes, or 8ms, down from 32 multiframes, or, 64ms for LCAS.

Therefore, the refresh time of FLCAS protocol, called FLRT, is a function of the VCG size and given by 2ms*$\lceil N/8 \rceil$. For realistic VCG sizes ($N <= 128$), FLRT will be much smaller than 32ms. In fact, failure notification for most likely VCGs in the field ($N <= 32$) can be done within 8ms. This ability of FLCAS to support smaller worst case failure notification times for the most commonly occurring VCG sizes is an excellent improvement over the existing LCAS protocol which offers a worst case time of 64ms independent of the VCG size.

While FLCAS provides fast failure notification times for most of the reasonable size VCGs, for large VCG sizes($N > 200$) it can be still be in excess of 50ms. Though, it is unlikely that such capacity VCGs are provisioned in the near future, it may still be of interest to lower the theoretical worst case notification times.

FLCAS uses a simple, yet innovative idea of interrupting the refresh cycle in case of a failure. Since it already uses the `FLOH` bits to identify the specific members whose status is being passed, it can break the cycle to identify the failed member(s) in the status field and fill the `FLOH` appropriately. This is shown in Figure 8. It shows the values of the two fields when members 25 and 5 fail while multiframe 5 is being sent. FLCAS sends the notification for member 25 (`FLOH`=3(00011)) and member 5 (`FLOH`=0(00000)) in that order breaking the refresh cycle. The refresh cycle resumes normally after these notifications are sent.

Since, FLCAS does not wait for the failed member's turn to arrive in the `FLOH` refresh cycle (worst case time of FLRT), and instead breaks the cycle to send the failed member's status; it guarantees that the notification of *first failure* is sent in the first multiframe itself. This scheme further reduces the FLCAS worst case failure notification time for one member failure to 2ms (earlier 2ms*$\lceil N/8 \rceil$) making it independent of the VCG size.

We briefly consider the case when M members fail simultaneously such as the example above. Let us assume these M members map to F distinct `FLOH` values. Note that notification of multiple member failures that map to the same `FLOH` value go out in the same multiframe (e.g., member 25 and 26 both map to `FLOH`=3). In the worst case, failure of M members can map to all possible `FLOH` values (which is$\lceil N/8 \rceil$). Thus, the delay for the notifications will be $2F$ms (worst case of $2 * \lceil N/8 \rceil$ms). Note that for a 256 member VCG, worst case times are still order of 64ms. However, for FLCAS to ever achieve that worst case scenario, more than 32 members have to fail such that they map to 32 different `FLOH` values. Needless to mention, that is a rather unlikely possibility in

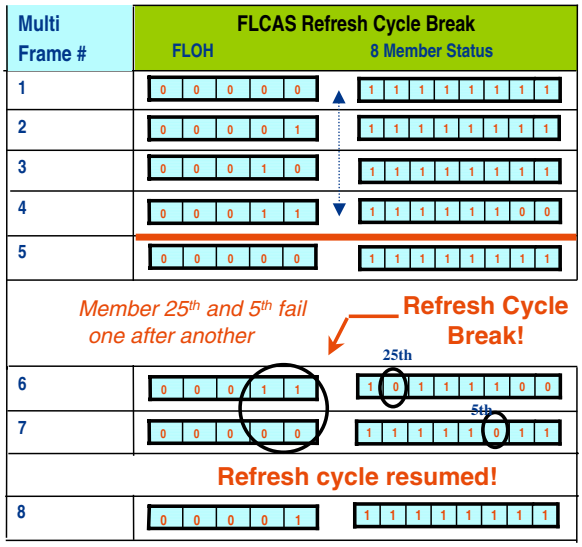

Fig. 8. Refresh Cycle Break in FLCAS for a 30-member VCG

practice.

The FLCAS approach can be extended to improve LCAS beyond just member failures. In fact, in general the FLCAS protocol can be used to notify any change in member status (e.g., member recovery, addition and removal) to the source node in the quickest manner possible. Also, FLCAS is designed such that a network element running it can interoperate with an element running standard LCAS with no additional overhead. Further details on the protocol is available in [16].

FLCAS when used in conjunction with PESO delivers excellent protection speed of well under 50ms. Infact, it achieves this bound for most of the practically likely VCG sizes and failure scenarios.

VIII. PESO Performance Summary

In this section, we summarize the contribution of PESO. Recall that PESO was designed to reliably transport EoS services with low bandwidth overhead and switching time competitive with SONET APS.

The bandwidth overhead imposed by PESO is dependent on the number of diverse paths available in the network. If the network is a SONET ring , there are only two diverse paths. In that case, PESO's overhead is the same as SONET 1+1. PESO's real gains arise in networks with three or, more diverse paths and these are increasingly available as next-generation SONET mesh topologies are deployed.

For PESO failure notification, one can either use LCAS or, the FLCAS protocol proposed in the paper. For a single failure, LCAS can cause a worst case service disruption of 64ms. The disruption time in FLCAS is dependent on the number of VCG members impacted by the failure, ranging from 2ms to upto $2 * \lceil N/8 \rceil$ms.

We highlight the benefits of PESO using an example of protecting a GigE (1Gb) circuit. Using VC, it can be routed by 21 STS-1s. If there are three diverse paths with requisite bandwidth (equivalent of STS-11), PESO routing will employ 11 STS-1s ($\lceil 21/(3-1) \rceil$) as backup members to completely protect against failure. Recall that it is not necessary for all the primary and backup members to be disjointly routed — PESO achieves reliability by limiting the flow on each link (or, node). In this case, the protection bandwidth overhead is only 53% (11/21), well below SONET.

If protected via LCAS, PESO can create a service disruption of 64ms on failure. Using FLCAS the same failure can be recovered in anywhere from 2ms to upto 8ms ($2*\lceil 21+11 \rceil/8$). Depending upon the reliability requirements, operator can choose the appropriate option.

Suffice it to say, PESO demonstrates for EoS environments, that it is indeed possible to create a mechanism within the SONET framework which meets the stated goal of lowering the protection bandwidth overhead while matching SONET APS.

IX. Related Work

Ethernet over SONET has received a lot of attention in the trade press lately [17]. Clearly, the inefficiency caused by the 100% overhead in SONET 1+1 APS is well recognized. In fact, analogous to this work, the SONET standard suggests the mechanism of shared protection as a means to minimize the protection bandwidth overhead. Also referred to as $m:n$ protection [3], it uses m additional backup paths to protect n primary paths (with $1:n$ being the special case). While similar in spirit, there is a key difference. Shared protection requires n different circuits to work together to share a common protection bandwidth and the first m circuits that fail, use up the protection bandwidth. Thus, it is superior to 1+1 only when n can be made larger than m. On the other hand, in PESO every circuit protects *itself* by overprovisioning more VCG members than the actual requirement. Thus, unlike shared protection, PESO can operate even when there is only one circuit between a source and sink node. Moreover, $m:n$ imposes a significant management overhead to implement in practice and thus, has limited support on commercially available network elements.

X. Conclusions

Service providers have a new lease of life with the deployment of EoS networks. In this paper, we argued that EoS is not just an evolution of Ethernet and SONET into a common framework but it instead provides fundamentally new options for reliable data transport that we exploit in the proposed PESO scheme. PESO provides fast, 50ms protection for data traffic in EoS without SONET's 100% overprovisioning requirement based on two key observations. Firstly, the VC protocol allows for traffic to be split and routed across multiple paths. Secondly, unlike the all-or-nothing protection required for voice, data traffic is amenable to operating at a lower bandwidth capacity for short periods of time. PESO leverages the LCAS mechanism for protection switching making it practical for deployment. More importantly, it proposes a fix to a known bottleneck in LCAS that limits its failure notification capabilities.

We believe that from the network management perspective, this paper presents a refreshingly novel view on the meaning and mechanisms for reliable EoS data transport. Furthermore, it opens up a number of new areas for research when one considers the numerous routing constraints one may impose on the VCG members. For example, minimizing differential delay among the members is a critical practical requirement. Alternatively, exploring restoration mechanisms that protect only a few members to get a favorable reliability-overhead tradeoff is an open challenge. Thus, we believe routing and network management will be critical areas of research going forward.

REFERENCES

[1] I. Standard, "Spanning Tree Protocol," *ANSI/IEEE Std 802.1D*, 1998 Edition.

[2] IEEE Standard, "Rapid Spanning Tree Algorithm and Protocol," *ANSI/IEEE Std 802.1W*, 2001.

[3] W. J. Goralski, *SONET/SDH*. Mcgraw-Hill, 2002.

[4] P. Joy, "Enabling Ethernet quality of service for mining new revenue from metro SONET networks," *NFOEC*, 2002.

[5] P. R. S. Acharya, B. Gupta and A. Srivastava, "Mobipack: Optimal Hitless SONET Link Defragmentation in Near-Optimal Cost," *Bell Labs. Technical Report*, 2003.

[6] S. Acharya, B. Gupta, P. Risbood and A. Srivastava, "Enabling Hitless Engineering of SONET Rings," *Proc. of Globecomm*, 2003.

[7] ITU-T Standard, "Virtual Concatenation Standard," *ITU-T standard G.707, pp.118-126*, 2000.

[8] I.-T. Standard, "Network node interface for the synchronous digital hierarchy (sdh)," *ITU-T standard*, 2000.

[9] ANSI standard, "Synchronous Optical Network - Basic Description including Multiplex Structure Rates, and Format," *ANSI T1.105-1995*, 1995.

[10] ITU-T Standard, "Link Capacity Adjustment Scheme for Virtual Concatenated Signals," *ITU-T standard*, 2001.

[11] J. L. R. Ford, "Flows in network," *Princeton University Press*, 1962.

[12] J. Edmonds and R. M. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems," *Journal of ACM, vol. 19, No. 2*, 1990.

[13] T. L. M. R. K. Ahuja and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

[14] J. B. Orlin, "A Faster Strongly Polynomial Minimum Cost Flow Algorithm," *Proc. of the 20th ACM Symposium on the Theory of Computing*, pp.377-387, 1988.

[15] W. S. Jewell, "Optimal Flow through Networks," *Interim Technical Report 8, Operations Research Center, MIT Cambridge, MA*.

[16] S. Acharya, B. Gupta, P. Risbood and A. Srivastava, "PESO: Low Overhead Protection for Ethernet over SONET Transport," *Bell Labs. Technical Report*, 2003.

[17] J. Conover, "Networking for the next generation," *Network Computing Magazine, http://www.networkcomputing.com*, 2001.