Dynamic Topology Overlays for Multipoint Ethernet-over-SONET/SDH

N. Ghani[*], Q. Liu[*], R. Wang[**], Y. Qiao[**], C. Xie[*], S. Seethamraju[*], A. Gumaste[‡]
[*]University of New Mexico, [**]Shanghai Jiao Tong University, [‡]IIT Bombay

Abstract**:** *Next-generation SONET/SDH technologies have fielded much-improved service mapping and provisioning capabilities. These features provide many exciting avenues for developing new Ethernet services, and various Ethernet-over-SONET studies have already been done. However these efforts have primarily focused on provisioning point-to-point demands, i.e., private line type offerings. This paper considers the broader and more challenging case of provisioning multi-point-to-multi-point Ethernet LAN services over SONET/SDH and presents novel strategies for bus and tree overlays. Detailed simulation results are also presented along with directions for future work.*

Keywords: *Ethernet-over-SONET; Carrier Ethernet, virtual concatenation; inverse multiplexing, next-gen SONET/SDH*

I. INTRODUCTION

Ubiquitous *time-division multiplexing* (TDM) SONET/SDH technology has continued to evolve over the years [1]. Today, revamped "*next-generation SONET/SDH*" (NGS) supports new capabilities for multi-service provisioning via standards for *generic framing procedure* (GPF), *virtual concatenation* (VCAT), and the *link capacity adjustment scheme* (LCAS) [2],[3]. GFP defines efficient mappings onto TDM channels for client interfaces such as Ethernet, Fibre Channel, ESCON, Infiniband, etc. Meanwhile VCAT allows operators to customize tributaries to match user demands, e.g., at STS-1 or VT1.5 levels. A key feature here is inverse-multiplexing [3] which enables demand resolution over multiple sub-connections. Finally LCAS supports dynamic "hit-less" adjustment of concatenated trails, furthering traffic engineering and service survivability provisions. NGS also allows interoperability with both legacy SONET/SDH and newer ITU-T *optical transport network* (OTN) standards [1].

As optical and SONET/SDH technologies have matured, the focus for many carriers has shifted towards services provisioning [4]. In particular, there is immense commercial interest in extending "Carrier Ethernet" services over metro and wide-area domains [5]-[7]. Here the *Metro Ethernet Forum* (MEF) has even moved to standardize a variety of service models for *Ethernet private line* (EPL), LAN (E-LAN), and tree (E-Tree) services [7]. Specifically, EPL offers point-to-point connectivity between users and the added *Ethernet virtual private line* (EVPL) variant allows multiple users to share a single *user network interface* (UNI). Meanwhile E-LAN pursues multi-point-to-multi-point (mp-2-mp) connectivity across larger geographic ranges.

Given these trends, there are many compelling reasons for designing new *Ethernet-over SONET/SDH* (EoS) extension schemes. Foremost is the fact that extensive TDM build-outs will likely remain in place for the foreseeable future [4]. Hence it is very desirable to re-use these infrastructures—in conjunction with advanced NGS capabilities—to build new Ethernet services. Additionally, nearly all optical *dense wavelength division multiplexing* (DWDM) networks use SONET/SDH framing of wavelengths (with future trends towards OTN-based framing). Given the larger granularity of wavelength channels, NGS provides a natural "sub-rate" grooming solution here. Along these lines, numerous EoS studies have been conducted, focusing on multi-path routing schemes for inverse multiplexing NGS networks, i.e., to improve efficiency and survivability [8]-[16].

However the above efforts have mostly focused on point-to-point EPL services with little consideration for mp-2-mp (multi-point) offerings such as E-LAN and E-Tree. Clearly the latter represent more lucrative "value-add" services and require further investigation. Now the authors in [17] have recently tabled basic multi-point EoS solutions using mesh and star overlays. This paper builds upon this effort by developing more advanced bus and *minimum spanning tree* (MST) overlays and is organized as follows. First Section II reviews existing research work on NGS/EoS survivability. Next, Section III introduces the EoS LAN problem and proposes novel bus and MST overlay algorithms. Section IV then presents commensurate connection group (overlay) provisioning schemes using inverse multiplexing and tiered (partial) protection. These solutions are evaluated in Section V using simulation and conclusions and directions for future work are presented in Section VI.

II. BACKGROUND

Various studies have looked at service provisioning in advanced SONET/SDH networks. A key focus here has been to leverage inverse multiplexing to design multi-path routing strategies to achieve more efficient survivability, e.g., versus legacy SONET/SDH 1+1/1:1 span protection, *bi-directional line-switched ring* (BLSR), etc. For example, [8] outlines several low-overhead *protection for Ethernet over SONET* (PESO) schemes for ensuring adequate immunity against single link failures by exploiting path diversity. Namely, sub-connection route overlaps are minimized to limit outages from single link failures. More recently, [9] also presents some strategies for "degraded-service-aware" provisioning to route sub-connections to ensure that no one carries more than a given fraction of the total flow. Multi-path load distribution is also used to minimize the maximum incremental link utilization.

To better address topological concerns, direct sub-connection protection/restoration schemes have also been studied. For example, [10] proposes a tiered partial protection scheme to protect a subset of working sub-connections. Simulations results show notable efficiency gains and very high recovery rates with the use of post-fault restoration. Meanwhile [11] proposes two inverse multiplexing *shared* protection schemes, *protecting*

individual virtual connection group members (PIVM) and *provisioning fast restorable virtual connection group* (PREV). The former allows backup capacity sharing between *link-disjoint* sub-connections whereas the latter limits sharing to link-disjoint sub-connections with the same source-destination. As expected, PIVM gives much higher efficiency whereas PREV gives much faster recovery. Meanwhile others have also studied differential delays in multi-path routing. For example, [12] defines the *cumulative differential delay routing* (CDDR) problem with the goal of resolving an integral number of sub-connection paths to limit destination (sink) memory requirements. Meanwhile [13] develops a modified link-weight *k*-shortest path algorithm to account for lower bounds on differential delay and results show improvement over more traditional algorithms.

Owing to the close coupling of SONET/SDH and optical DWDM technologies in the metro/core, researchers have also studied EoS grooming over DWDM networks. For example, [14] and [15] propose basic schemes to resolve incoming requests to the STS-1 level and table modified shortest-path heuristics for multi-path routing. Meanwhile, [16] has recently studied Ethernet grooming in optical networks with *mixed line rate* (MLR) links. The authors argue that MLR networks can yield better performance over *single line rate* (SLR) networks and table more efficient routing algorithms.

Although the above studies present some key innovations, new schemes for multipoint "value-add" Carrier Ethernet services need to be addressed, i.e., based upon connection group overlays. Indeed, the ability to split a LAN request into multiple sub-connections (via inverse multiplexing) can yield genuine *multi-tiered* full/fractional rate LAN services. This topic is now studied in detail.

III. TOPOLOGY OVERLAY SCHEMES

The extension of Carrier Ethernet LAN services over SONET/SDH domains mandates reliable *multi-point-to-multi-point* connectivity across dispersed metro/wide-area sites. Given the lack of timeslot multi-casting in SONET/SDH, such services will require coordinated setup of multiple point-to-point TDM connections, i.e., *connection groups* or *topology overlays*. Nevertheless, few studies have addressed Ethernet LAN services in the context of advanced SONET/SDH networks, i.e., most efforts surveyed in Section II are only applicable to point-to-point EPL services. Note, however that within the broader networking context, various "overlay" studies have still been done. For example the *resilient overlay network* (RON) [18] project builds a static Internet overlay to improve routing resiliency. Meanwhile the *service overlay networks* (SON) [19] study addresses *quality of service* (QoS) support using queuing theory and optimization techniques to achieve static partitioning and oversubscription. The further design of "dynamic" topology overlays has also been considered. For example, [20] proposes a *virtual network* (VN) assignment scheme for node/link selection with/without reconfiguration.

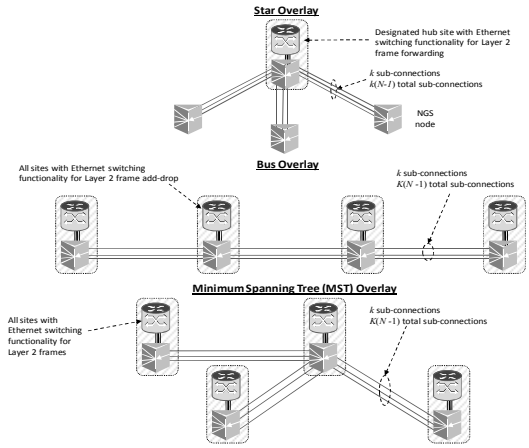

Figure 1: Ethernet LAN overlays: star, bus, tree

Albeit of some relevance here, the above efforts do not account for the specialized inverse-multiplexing capability of EoS settings. Hence a more targeted overlay approach is developed herein to support "dynamic/on-demand" multi-point Ethernet services in TDM networks. Namely, the first step focuses on building a connection group overlay for a LAN request, i.e., identifying point-to-point connections. Here three overlay provisioning schemes are considered based upon *star*, *bus*, and *minimum spanning tree* (MSF) topologies, Figure 1. The requisite notation is first introduced (all vector and set entities denoted in bold).

Consider a physical SONET/SDH network of N nodes and M links. This network is modeled as a graph $G(\boldsymbol{V},\boldsymbol{L})$, where $\boldsymbol{V}$ is the set of NGS nodes (vertices) and $\boldsymbol{L}$ is the set of SONET/SDH links (edges), i.e., $\boldsymbol{V}=\{v_1,v_2,...,v_N\}$ $\boldsymbol{L}=\{l_{12},l_{13},...,l_{ij}\}$. Here, link l_{ij} is the link between nodes i and j of C units capacity and c_{ij} is the available (free) capacity of this link. Necessarily, if there is a link from i to j, then there is also a link from j to i since SONET/SDH links are bi-directional. Now consider the i-th Ethernet LAN request between a subset of nodes given by the vector $\boldsymbol{v}_i=\{v_{i1},v_{i2},\ldots\}$, $\boldsymbol{v}_i \subseteq \boldsymbol{V}$. This request is used to build a *connection group* comprising of a set of n_i bi-directional point-to-point connections, $\{\boldsymbol{s}_i, \boldsymbol{d}_i\}$, where the vector $\boldsymbol{s}_i=\{s_{i1}, s_{i2}, \ldots\} \subseteq \boldsymbol{v}_i$ and $\boldsymbol{d}_i=\{d_{i1}, d_{i2}, \ldots\} \subseteq \boldsymbol{v}_i$ represent the source/destination end-points, e.g., individual connections denoted as s_{i1}– d_{i1}, s_{i2}– d_{i2}, etc. Assuming a requested LAN throughput of x_i STS-1 units, each individual connection can also be assumed to be of size x_i STS-1 units. Although the exact makeup of the connection group will depend upon the overlay topology chosen, each of the constituent point-to-point connections can still be inverse-multiplexed, i.e., "split", in to multiple "sub-connections". Namely, the latter is specified by an inverse multiplexing factor, K. Hence the LAN connection group LAN request is denoted as the tuple $(n_i,\{\boldsymbol{s}_i, \boldsymbol{d}_i\},x_i,K)$. Details on the overlay algorithms are now presented.

A. Star Overlay

The star overlay scheme was recently tabled in [17] and is presented here for reference sake. This approach uses a

designated hub site to provide Ethernet connectivity to all other nodes, assuming that it has adequate Layer 2 switching capabilities, either provided by the client or carrier node. This essentially is equivalent to a single level tree overlay (single parent), Figure 1. In the former case, packets transiting the hub site cross the carrier-client boundary twice, whereas in the latter case they stay inside the carrier network. This overlay requires $n_i=O(|\boldsymbol{v}_i|-1)=O(|\boldsymbol{V}|)$ TDM connections or $O(K(|\boldsymbol{v}_i|-1))=O(K|\boldsymbol{V}|)$ VCAT sub-connections (Figure 1). As per [17], the scheme first selects a designated hub node, h_i, and then iterates and adds "overlay" links to the other LAN group nodes. The key design goal here is to minimize overall resource utilization and/or lower blocking. Hence three different hub selection strategies are tabled:

Random Hub Selection (Star-RS): This approach randomly selects h_i via a uniform distribution from 1 to $|\boldsymbol{v}_i|$.

Minimum Average Hop (Star-MAH): This scheme chooses the hub with the minimum average hop count to all other LAN nodes. The goal is to minimize resource utilization of the overall LAN group as follows:

$$h_i = v_{ij^*} \text{ where } j^* = \min_j \left(\sum_{k=1,k\neq j}^{|\boldsymbol{v}_i|} hop(v_{ij}, v_{ik}) \right) \quad \text{Eq. (1),}$$

where $1\leq j\leq|\boldsymbol{v}_i|$. As this scheme only uses *static* information, computational speedup can be achieved by pre-computing and storing inter-nodal hop-counts. Specifically, $O(|\boldsymbol{v}_i|(|\boldsymbol{v}_i|-1))=O(|\boldsymbol{V}|^2)$ shortest path computations are required prior to startup along with $O(|\boldsymbol{v}_i|^2)=O(|\boldsymbol{V}|^2)$ storage overheads. However at run-time $O(|\boldsymbol{v}_i|(|\boldsymbol{v}_i|-1))= O(|\boldsymbol{V}|^2)$ lookups will be required to select the hub site, yielding a total compute complexity of $O(|\boldsymbol{V}|^2)$.

Minimum Average Cost (Star-MAC): This scheme follows the same flow as Star-MAH with the exception that the hub-site is now chosen using the minimum average *cost* to all LAN group nodes. Namely, the link cost here is a dynamic value, computed as being inversely-proportional to the available link capacity on link l_{ij}:

$$\omega_{ij} = \frac{1}{c_{ij} + \varepsilon} \quad \text{Eq. (2),}$$

where ε is a small quantity chosen to avoid floating-point divide errors. Hence the overall hub selection is given as:

$$h_i = v_{ij^*} \textbf{ where } j^* = \min_j \left(\sum_{k=1,k\neq j}^{|\boldsymbol{v}_i|} \textbf{cost}(v_{ij}, v_{ik}) \right) \quad \text{Eq. (3),}$$

where $1\leq j\leq|\boldsymbol{v}_i|$, and cost($v_{ij}$,$v_{ik}$) is the sum of all link costs, Eq. (2), along the minimum cost path between nodes v_{ij} and v_{ik}. The goal here is to use dynamic link resource state to choose the hub site and avoid "congested" nodes with heavily loaded links. However, compute complexities are notably higher in this case, i.e., $O(|\boldsymbol{v}_i|(|\boldsymbol{v}_i|-1))=O(|\boldsymbol{V}|^2)$ shortest path computations per request, yielding $O(|\boldsymbol{V}|^2\cdot|\boldsymbol{V}|\log|\boldsymbol{V}|)=O(|\boldsymbol{V}|^3\log|\boldsymbol{V}|)$ compute complexity. Note that since this overlay scheme only requires Layer 2 processing at a single designated hub site, it can be problematic for lower topological node degrees, i.e., hub in-degree requirement of $|\boldsymbol{v}_i|$-1. It is here that alternative bus and MST overlays can be more beneficial and these are detailed next.

```
Given a LAN request between nodes in v_i, where |v_i|≥3
Initialize bus sequence vector r_i={}
Select first bus node pair (overlay link) in r_i={r_i1, r_i2} from v_i using min,
average hop count (Eq. 4), or minimum average cost (Eq. 5)

% Generate first connection in bus overlay group
s_i1 = r_i1
d_i1 = r_i2

% Loop and generate rest of bus overlay
for j=1 to |v_i|-2
  {
   Search for next candidate ordering node from first node in ordering
   vector r_i using min. hop or min. cost, i.e., v_ik*, x_1 (Eqs. 5, 6)

   Search for next candidate ordering node from last node in ordering
   vector r_i using min. hop or min. cost, i.e., v_im*, x_2 (Eqs. 8, 9)

   % Update bus sequence vector, generate overlay link connection
   if (x_1≤x_2)
       {
        % Minimum hop or cost is from first node
        r_i={v_ik*, r_i1, r_i2, …, r_ij-1}
        s_ij = v_ik*
        d_ij = r_i1
       }
   else
       {
        % Minimum hop or cost is from last node
        r_i={ r_i1, r_i2, …, r_ij-1, v_im*}
        s_ij = r_ij-1
        d_ij = v_im*
       }
  }

% Set LAN group connection count
n_i=|v_i|-1
```

Figure 2: Bus overlay

B. Bus Overlay

This overlay implements a linear inter-connection between the LAN sites, Figure 1, and requires a total of $n_i=O(|\boldsymbol{v}_i|-1)=O(|\boldsymbol{V}|)$ connections or $O(K(|\boldsymbol{v}_i|-1))=O(K|\boldsymbol{V}|)$ VCAT sub-connections. As such, it is equivalent to a specialized tree overlay with only one child per parent. Although this design has notably lower nodal in-degree requirements than the star overlay, i.e., 2, it requires Ethernet "add-drop" switching capabilities at all LAN group sites. Here, connection group selection is also more involved, as shown in the pseudocode listing of Figure 2. Specifically, the main goal here is to determine the node *sequence* that yields minimum overall bus resource utilization and/or lowers blocking. To compute this, the algorithm uses a node sequence vector, $\boldsymbol{r}_i=\{r_{i1}, r_{i2}, \ldots\}\subseteq\boldsymbol{v}_i$, to iteratively build the ordering of nodes in the bus. Initially this vector is initialized to a null value, i.e., all nodes in $\boldsymbol{v}_i$ are "unassigned", Figure 2. The initial overlay link in $\boldsymbol{r}_i$ is then computed, using a given selection strategy (detailed shortly). The algorithm iterates through the remaining "unassigned" nodes ($\boldsymbol{v}_i$-$\boldsymbol{r}_i$) to sequentially determine their ordering in the connection group overlay. In particular, this is done by

checking the first and last nodes in $\boldsymbol{r}_i$ to compute the next "closest" node according to a particular strategy:

Random Selection (Bus-RS): This scheme selects bus nodes in a random manner. Namely, the next node is chosen by applying a uniform distribution over "unassigned" nodes. This is the simplest of all of the bus assignment schemes.

Minimum Average Hop (Bus-MAH): This scheme chooses bus nodes to minimize resource consumption. The first two nodes in the bus (i.e., first overlay link) are "assigned" as the LAN group node pair with the minimum interconnecting hop count, i.e.,

$$r_i = \{r_{i1}, r_{i2}\} = \{v_{ik^*}, v_{im^*}\} \qquad \text{Eq. (4)},$$

where $k^*, m^* = \min_{k,m}(\text{hop}(v_{ik}, v_{im}))$. Next, the scheme iterates to select the remaining bus nodes by checking the "assigned" end-points (i.e., loop over index j, $j \geq 3$, Figure 2). As mentioned above, the first and last end-point nodes in $\boldsymbol{r}_i$, i.e., $\boldsymbol{r}_{i1}$ and $\boldsymbol{r}_{ij-1}$, are checked to determine the next "unassigned" node with the minimum average hop count. Specifically,

$$x_1 = \text{hop}(r_{i1}, v_{ik^*}) \qquad \text{Eq. (5)},$$

where $k^* = \min_k(\text{hop}(r_{i1}, v_{ik}))$, $v_{ik} \notin \boldsymbol{r}_i$, versus that from the last node in the current ordering vector:

$$x_2 = \text{hop}(r_{ij-1}, v_{im^*}) \qquad \text{Eq. (6)},$$

where $m^* = \min_m(\text{hop}(r_{ij-1}, v_{im}))$, $v_{im} \notin \boldsymbol{r}_i$. Hence the next bus node is determined and inserted at the head or tail of the ordering vector, i.e., after iteration j, as follows:

$$\boldsymbol{r}_i = \begin{cases} \{v_{ik^*}, r_{i1}, r_{i2}, \ldots, r_{ij-1}\} & if \quad x_1 \leq x_2 \\ \{r_{i1}, r_{i2}, \ldots, r_{ij-1}, v_{im^*}\} & if \quad x_1 > x_2 \end{cases} \qquad \text{Eq. (7)}.$$

Like Star-MAH, this approach also uses static hop-count information, i.e., $O(|\boldsymbol{v}_i|(|\boldsymbol{v}_i|-1)) = O(|\boldsymbol{V}|^2)$ shortest path computations prior to startup with $O(|\boldsymbol{v}_i|^2) = O(|\boldsymbol{V}|^2)$ storage overheads. However at run-time $O(|\boldsymbol{v}_i|(|\boldsymbol{v}_i|-1)) = O(|\boldsymbol{V}|^2)$ lookups are required per node selection, i.e., total complexity of $O(|\boldsymbol{V}|^2)$.

Minimum Average Cost (Bus-MAC): This scheme has the same overall flow as the Bus-MAH scheme, with the exception that hop counts are now replaced with minimum average costs, Eq. (2), as follows:

$$x_1 = \text{cost}(r_{i1}, v_{ik^*}) \qquad \text{Eq. (8)}$$

and

$$x_2 = \text{cost}(r_{ij-1}, v_{im^*}) \qquad \text{Eq. (9)}$$

where the cost() function is defined as in Section III.A, $k^* = \min_k(\text{cost}(r_{i1}, v_{ik}))$, $v_{ik^*} \notin \boldsymbol{r}_i$ and $m^* = \min_m(\text{cost}(r_{ij-1}, v_{im}))$, $v_{im^*} \notin \boldsymbol{r}_i$. Like Star-MAC, this scheme uses dynamic resource state and hence has higher compute complexity, i.e., compute complexity $O(|\boldsymbol{V}|^2 \cdot |\boldsymbol{V}| \log|\boldsymbol{V}|) = O(|\boldsymbol{V}|^3 \log|\boldsymbol{V}|)$.

```
Given a LAN request between nodes in v_i, where |v_i|≥3
Initialize MST nodes vector r_i={}
Select first MST node in r_i={ r_i1}randomly from v_i

% Loop and generate rest of bus overlay
for j=1 to |v_i|-1
  {
    Search for next candidate MST node by using min. hop or
    min. cost from existing nodes in r_i, i.e., v_ik* (Eqs. 10, 11)

    % Update ring sequence vector, generate overlay link connection
    r_i={r_i1, r_i2, …, r_ij-1, v_ik*}
    s_ij = r_im
    d_ij = v_ik*
  }

% Set LAN group connection count
n_i=|v_i|-1
```

Figure 3: MST overlay based on Prim's algorithm

C. Minimum Spanning Tree (MST) Overlay

The MST overlay achieves a balance between the star and bus overlays by constructing a more generalized tree. This is achieved by adapting Prim's MST algorithm [21] for the subset of overlay LAN nodes. Namely, the initial MST node is selected randomly at first. The algorithm then loops to add nodes (links) until all LAN nodes are accounted for, similar to the Dijkstra's shortest-path search procedure.

The overall pseudocode for the MST overlay is shown in Figure 3. Akin to the bus overlay, a node sequence vector, $\boldsymbol{r}_i = \{r_{i1}, r_{i2}, \ldots\} \subseteq \boldsymbol{v}_i$, is used to track the nodes added to the MST overlay. Namely, the first MST node is "assigned" randomly, i.e., r_{i1}, Figure 3. Next, the algorithm iterates and adds new MST nodes to $\boldsymbol{r}_i$. Specifically, at each iteration all nodes in the tracking vector $\boldsymbol{r}_i$ are searched to find a new "unassigned" node from the set $\boldsymbol{v}_i - \boldsymbol{r}_i$ pursuant to a particular minimization strategy (akin to the star and bus overlays). Specifically, two strategies are tabled here:

Minimum Average Hop (MST-MAH): This scheme chooses MST nodes in order to minimize resource consumption. Namely, the scheme iterates (i.e., loop over index j, $j \geq 3$, Figure 3) to select an "unassigned" LAN node with the minimum hop count to a node in $\boldsymbol{r}_i$, i.e.,

$$r_{ij} = v_{ik^*} \quad \text{s.t.} \quad \min_{m,k}(\text{hop}(r_{im}, v_{ik})), \qquad \text{Eq. (10)}$$

where $1 \leq m \leq j-1$, and $v_{ik} \notin \boldsymbol{r}_i$. Akin to the Bus-MAH scheme, this approach only uses static information to choose the next MST node, i.e., $O(|\boldsymbol{v}_i|(|\boldsymbol{v}_i|-1)) = O(|\boldsymbol{V}|^2)$ shortest path computations needed prior to startup with $O(|\boldsymbol{v}_i|^2) = O(|\boldsymbol{V}|^2)$ storage overheads. However $O(|\boldsymbol{v}_i|(|\boldsymbol{v}_i|-1)) = O(|\boldsymbol{V}|^2)$ run-time lookups are required to select all MST nodes, yielding an overall complexity of $O(|\boldsymbol{V}|^2)$.

Minimum Average Cost (MST-MAC): This scheme follows the same overall flow as MST-MAH, with the exception that hop counts are now replaced with minimum average costs, Eq. (2), as follows:

$$r_{ij} = v_{ik^*} \quad \text{s.t.} \quad \min_{m,k}(\text{hop}(r_{im}, v_{ik})), \qquad \text{Eq. (11)}$$

where $1 \le m \le j-1$, $v_{ik} \notin \boldsymbol{r}_i$, and the cost() function is defined as in Section III.A. Since this scheme uses dynamic resource state, akin to Bus-MAC, it has higher compute complexity, i.e., total complexity $O(|\boldsymbol{V}|^2 \cdot |\boldsymbol{V}| \log|\boldsymbol{V}|) = O(|\boldsymbol{V}|^3 \log|\boldsymbol{V}|)$.

IV. MULTI-TIERED LAN GROUP PROVISIONING

Carrier Ethernet LAN service users will demand flexible, multi-tiered survivability support. For example most "regular" users will suffice with partial recovery against single faults. Alternatively a subset of users may demand much more stringent 100% recovery. To meet these requirements, the framework herein provisions LAN overlay connections (Section III) using inverse multiplexing and *tiered* protection algorithms. The aim here is to guarantee a minimum LAN throughput in the event of a single fault. To achieve this, a fractional protection factor, ρ ($0 \le \rho \le 1$), is used to specify a minimum *pre-provisioned* protection level for the LAN connection group. Namely a minimum level of ρx_i STS-1 units of dedicated protection capacity must be provisioned for all group connections. Consider the details.

A. Inverse Multiplexing Considerations

Inverse multiplexing facilitates multi-path routing of flows, and within the context of a LAN overlay, this concept can be applied to individual group connections. Namely, consider the i-th LAN requesting x_i STS-1 units (mapped from Ethernet bandwidth equivalent). Here, each individual connection in the LAN group between node s_{ij} and d_{ij} will also require $x_{ij}=x_i$ STS-1 units, $0 \le j \le n_i$. In turn, this connection can be "resolved" into multiple "sub-connections", up to a maximum of $K \le x_i$, as given by the inverse multiplexing factor. Here, an "even" distribution approach is used to distribute load, akin to [10]. Specifically, consider integral division of x_{ij} by K yielding:

$$z = \left\lfloor \frac{x_{ij}}{K} \right\rfloor \qquad \text{Eq. (12)},$$

where the remainder term is given by:

$$y = x_{ij} - Kz \qquad \text{Eq. (13)}$$

and $0 < y < k$. For the special case of $x_{ij}=Kz$ (i.e., $r=0$), all requested sub-connections are sized at x_{ijk} STS-1 units, $1 \le k \le K$. However for the more general case of $y \ne 0$, the remainder term is distributed over the first r sub-connections, i.e., individual capacity for the k-th requested sub-connection, x_{ijk}, in STS-1 increments is:

$$x_{ijk} = \begin{cases} z+1 & 1 \le k \le y \\ z & y < k \le K \end{cases}, \quad \sum_j x_{ijk} = x_{ij} = x_i \quad \text{Eq. (14)}$$

i.e., the first y connections may receive an extra STS-1 unit. Note that the above formulation assumes that the inverse multiplexing factor K is pre-specified, reasonable for most operational settings. LAN overlay resiliency over SONET/SDH networks is detailed next.

B. Path Computation and Protection Strategies

The LAN provisioning solution operates in two phases. First, all *working* overlay connections (sub-connections) are routed with x_i STS-1units of capacity each. Next, each of these connections is protected by provisioning a subset of its sub-connections with dedicated protection sub-connections, i.e., to achieve a minimum "LAN-wide" protection of ρx_i STS-1 units. Namely dedicated *link-disjoint* protection paths are computed for a minimal subset of working sub-connections until the desired threshold is achieved. This approach ensures a "LAN-wide" throughput of at least ρx_i STS-1 units in the event of a single link failure. Carefully note that since protection is done on a per-sub-connection basis, protection granularity is inversely proportional to the inverse multiplexing factor K. Hence it is possible to have protection over-provisioning with smaller K, as noted in [10]. However this inefficiency can be easily be resolved by appropriately "right-sizing" protection sub-connections. Overall, the above approach simplifies protection switchovers by mandating equal-sized working and protection VCG members.

Now consider the actual working/protection provisioning algorithms for the LAN group connections (see [17] for pseudocode for hub overlay scheme). Here, the working phase first computes routes for all LAN connections/sub-connections. Namely, the algorithm makes a temporary copy of the network graph, $G'(\boldsymbol{V},\boldsymbol{L})$, and then iterates to setup *working* routes for all n_i connections in the LAN using inverse multiplexing. Each connection is resolved into sub-connections using the above-described "even" distribution approach (Section IV.A), i.e., x_{ijk}, Eq. (14). Next, Dijkstra's shortest-path computation scheme is applied in an iterative manner to route the individual sub-connection paths, $\boldsymbol{w}_{ijk}$. Here, the capacity of all successfully-routed sub-connections is pruned along the respective route links in $G'(\boldsymbol{V},\boldsymbol{L})$. Setup for the next group connection is only attempted if the current connection is fully routed otherwise the LAN request is dropped. Carefully note that there are no restrictions on link overlap between sub-connections and only *feasible* links with sufficient capacity can be treated, i.e., $c_{jk} \ge x_i$.

If all working group connections are successfully setup on the temporary graph $G'(\boldsymbol{V},\boldsymbol{L})$, the tiered protection phase is intiated. Here the algorithm uses the "left-over" capacity in $G'(\boldsymbol{V},\boldsymbol{L})$ and again iteratively tries to setup protection routes. To achieve this, a running count of the aggregate "connection-level" protection capacity, *protection_capacity*, is maintained. Specifically, this value is used to check against the desired minimum protection threshold (ρx_i) after each successful protection sub-connection setup. Overall, the LAN connection is deemed protected only if the ρx_i threshold is crossed, and then all related sub-connection protection paths, $\boldsymbol{p}_{ijk}$, are stored. Otherwise, the LAN request is dropped.

Finally, two different link cost "routing" metrics are used by the Dijkstra scheme when routing the working/ protection sub-connections, e.g., *hop count* and *cost* (as used in the

overlay computation schemes, Section III). Specifically, the latter weights all links equally and chooses the shortest feasible path. Conversely the latter metric, Eq. (2), distributes loads across lightly-loaded links and thereby increases multi-path routing diversity between individual sub-connections. Overall, these two strategies are used to achieve a balance between resource minimization and multi-path diversity.

V. PERFORMANCE ANALYSIS

The performance of multi-point EoS overlay schemes is studied by coding specialized simulation libraries in the *OPNET Modeler*™ simulation tool environment. All tests are done with the NSFNET topology with 16 nodes and 25 links. The network elements are generic NGS-capable *broadband digital cross-connects* (BBDCS) nodes with STS-1 (50 Mb/s) switching granularity and OC-48 link speeds. The LAN requests follow random exponentially-distributed holding and inter-arrival times, with means μ and λ, respectively. In particular, a scaled mean holding time of μ=600 seconds is used and the mean inter-arrival times are adjusted according to load. Carefully note that this is just a scaled relative value and is not necessarily representative of real-world connection holding times. All of the LAN group sizes are uniformly varied from 3-5 nodes and bandwidth requests are varied from 200 Mb/s to 1.0 Gb/s in 200 Mbps increments (4 STS-1 units) to model fractional demands. An average of 500,000 LAN requests are measured per run and a modified Erlang loading metric is introduced to account for connection group sizes:

$$\text{Modified Erlang load} = \sum_{n=x_1}^{x_2}(n-1)\cdot\frac{\mu}{\lambda} \qquad \text{Eq. (15),}$$

where the LAN groups range in size from x_1=3 to x_2=5 nodes and the 1/λ represents the mean inter-arrival rate.

Initial tests are done to measure the carried load for non-protected LAN scenarios (ρ=0) given a nominal request blocking rate of 2%. These scenarios are very relevant to carriers as they indicate the true "load-carrying" (i.e., revenue generation) capacity of the network at "low-blocking" operating point. The results yield some critical findings and are shown in Figures 4 (star, bus) and 5 (MST, combined plot of best results from bus and MST overlays). Foremost, the findings show that the more advanced bus and MST overlays yield very similar carried loads, both notably higher than the hub overlay. In general, this improvement is due to the reduced in-degrees in the respective overlay topologies. Furthermore, it is seen that minimum cost-based overlay selection tends to yield the best performance, e.g., Bus-MAC, MST-MAC, Figure 5b. In particular, the MST overlay slightly out-performs the bus overlay, likely due to the fact that the former approach searches all "assigned" LAN nodes in the vector $\boldsymbol{r}_i$ to select the next overlay link whereas the latter only searches from the end-points (Sections III.B, C). Furthermore, load balancing routing, i.e., Eq. (2), coupled with inverse multiplexing is seen to give the best results for all overlay types. Specifically, increasing values of K yielding about 25-30% higher carried load than regular "non-inverse multiplexed" operation (K=1). However, added simulations with faster OC-192 links (not shown) indicate that inverse multiplexing is most effective when the ratio of the granularity of average LAN request sizes to the link rate is above 0.2 (i.e., 20%).

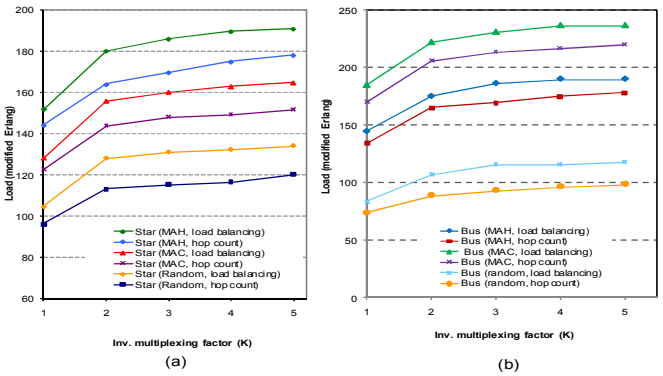

Figure 4: Carried load for 2% LAN request blocking, ρ=0: a) star overlay, b) bus overlays

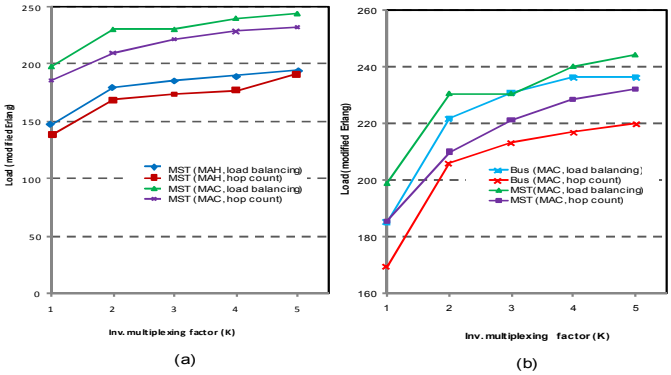

Figure 5: Carried load for 2% LAN request blocking, ρ=0: a) MST overlay, b) comparison

Next, LAN blocking is tested for the more capable bus and MST overlay schemes using the load balancing routing approach, i.e., Figures 6 and 7. First, Figure 6a plots the Bus-MAC request blocking for varying protection thresholds (ρ=0, 0.25, 0.5) and inverse multiplexing factors (K=1, 2, 4). Here it is seen that that reduced protection factors give significantly lower LAN blocking, as expected. Also, larger inverse multiplexing factors yield decent gains for equivalent protection, e.g., K=4/ρ=0.5 gives about 10-30% lower blocking than K=2/ρ=0.5 (Figure 7a). Similar findings are also observed for the MST-MAC scheme, as shown in Figure 7a. Furthermore, the *individual* blocking rates for different LAN request sizes (for K=4/ρ=0.25) are also plotted in Figures 6b and 7b. These results show that larger 5 node LAN sizes can experience almost half an order magnitude higher blocking than smaller 3 node LAN sizes. However, the improved MST-MAC scheme can deliver under 1% blocking for even mid-high range loads.

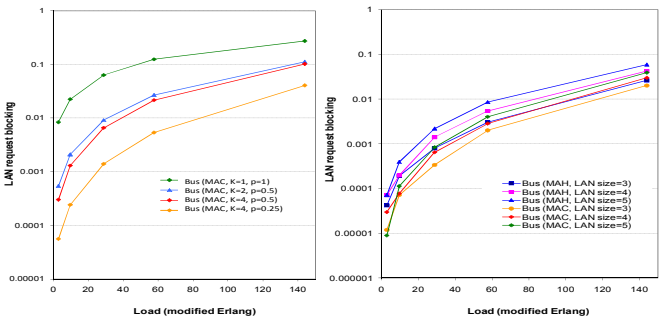

Figure 6: LAN blocking for bus overlays: a) Bus-MAC with varying K, ρ, b) K=4, ρ=0.25

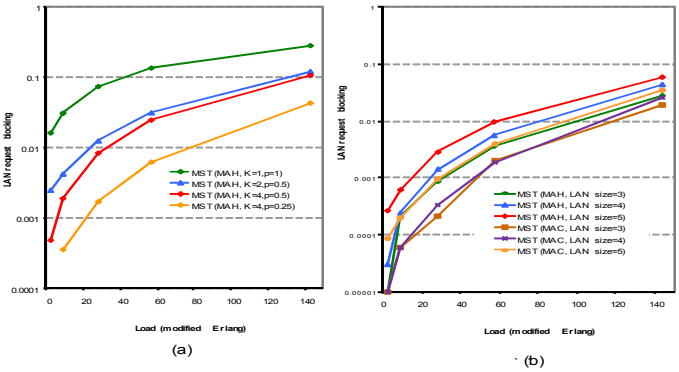

Figure 7: LAN blocking for MST overlays: a) MST-MAC with varying K, ρ, b) K=4, ρ=0.25

VI. CONCLUSIONS

This paper studies various topology overlay schemes for multi-point Ethernet LAN services, i.e., star, bus, and minimum spanning tree. A comprehensive tiered survivability approach for provisioning these overlays is also detailed. Findings show that more advanced bus and minimum spanning tree overlays yield best blocking and carried load performances. In addition, the use of active resource state in the overlay design can also improve performance. Future efforts will look to extend this work to study more elaborate shared protection strategies as well as post-fault restoration methods.

ACKNOWLEDGEMENTS

This research has been supported by the NSF *Computer and Network Systems* (CNS) division under award CNS 0806637 and the UNM ECE Department. The authors are very grateful to these sponsors for their generous support.

REFERENCES

[1] G. Bernstein, et al, Optical Network Control: Architectures, Standards, Protocols, Addison Wesley, Boston, 2003.

[2] E. Hernandez-Valencia, "Hybrid Transport Solutions for TDM/Data Networking Solutions," *IEEE Communications Magazine*, Vol. 40, No. 5, May 2002, pp. 104-112.

[3] G. Bernstein, *et al*, "VCAT/LCAS In A Clamshell," *IEEE Communications Magazine*, Vol. 44, No. 5, May 2006, pp. 34-36.

[4] N. Ghani, *et al*, "On IP-WDM Integration: A Retrospective," *IEEE Communications Mag.* Vol. 41,No. 9, Sept. 2003, pp. 42-45.

[5] A. Kasim (Editor), *Delivering Carrier Ethernet: Extending Ethernet Beyond the LAN*, McGraw Hill Publishers, Nov. 2007.

[6] L. Fang, *et al*, "The Evolution of Carrier Ethernet Services—Requirements and Deployment Case Studies," *IEEE Communications Mag*, Vol. 46, No. 3, March 2008, pp. 69-76.

[7] "Metro Ethernet Services-A Technical Overview," Metro Ethernet Forum, available at http://www.metroethernetforum.org.

[8] S. Acharya, *et al*, "PESO: Low Overhead Protection for Ethernet Over SONET Transport," *IEEE Infocom 2004*, Hong Kong, March 2004.

[9] R. Roy, *et al*, "Degraded-Service-Aware Multipath Provisioning in Telecom Mesh Networks," *IEEE/OSA OFC 2008*, San Diego, CA, February 2008.

[10] N. Ghani, S. Park, "Multi-Tiered Service Survivability in Next-Generation SONET/SDH Networks," *Photonic Network Communications*, Vol. 13, No. 1, January 2007, pp. 79-92.

[11] C. Ou, *et al*, "Survivable Virtual Concatenation for Data Over SONET/SDH In Optical Transport Networks", *IEEE/ACM Trans. on Networking*, Vol. 14, No. 1, February 2006, pp. 218-231.

[12] A. Srivastava, A. Srivastava, "Flow Aware Differential Delay Routing for Next-Generation Ethernet Over SONET/SDH," *IEEE ICC 2006*, Istanbul, Turkey, June 2006.

[13] S. Ahuja, *et al*, "Minimizing the Differential Delay for Virtually Concatenated Ethernet over SONET Systems", *IEEE ICCCN 2004*, Chicago, IL, October 2004.

[14] R. Srinivasan, A. Somani, "Dynamic Routing in WDM Grooming Networks," *Photonic Network Communications*, Vol. 5, No. 2, March 2003, pp. 123-135.

[15] K. Zhu, *et al*, "Ethernet-Over-SONET (EOS) Over WDM in Optical Wide-Area Networks (WANs): Benefits and Challenges," *Photonic Network Comm.*, Vol. 10, No. 1, Jan. 2005, pp. 107-108.

[16] M. Bataynet, *et al*, "Cost-Efficient Routing in Mixed-Line-Rate (MLR) Optical Networks for Carrier-Grade Ethernet", *IEEE/OSA OFC 2008*, San Diego, CA, February 2008.

[17] C. Xie, N. Ghani, "Multi-Point Ethernet Over Next-Generation SONET/SDH", to appear in *IEEE ICC 2009*, Dresden.

[18] D. Andersen, *et al*, "Resilient Overlay Networks," *ACM SOSP 2001*, October 2001, pp. 131-185.

[19] Z. Duan, et al, "Service Overlay Networks: SLAs, QoS, and Bandwidth Provisioning," *IEEE/ACM Transactions on Networking*, Vol. 11, No. 6, December 2003, pp. 870-883.

[20] Y. Zhu, M. Ammar, "Algorithms for Assigning Substrate Network Resources to Virtual Network Components," *IEEE INFOCOM 2006*, Barcelona, Spain, April 2006.

[21] R. Prim, "Shortest Connection Networks and Some Generalizations," *Bell System Technical Journal*, Vol. 36, 1957, pp. 1389–1401