Optical Switching and Networking 7 (2010) 54–65

Contents lists available at ScienceDirect

Optical Switching and Networking

journal homepage: www.elsevier.com/locate/osn

Service overlays for Ethernet LAN over SONET/SDH

N. Ghani [a,*], M. Esmaeili [a], H. Wu [b], Y. Qiao [c], R. Wang [d], M. Peng [e]

[a] University of New Mexico, United States
[b] Shanghai Jiao Tong University, China
[c] University of Florida, United States
[d] University of California Davis, United States
[e] Wuhan University, China

ARTICLE INFO

Article history:
Received 3 July 2009
Received in revised form 8 January 2010
Accepted 5 February 2010
Available online 13 February 2010

Keywords:
Ethernet-over-SONET
Carrier Ethernet
Virtual concatenation
Inverse multiplexing
Next-generation SONET/SDH

ABSTRACT

Next-generation SONET/SDH provides carriers with new capabilities for building advanced Ethernet data services. Along these lines, numerous Ethernet-over-SONET studies have been done, with most focusing on point-to-point private line type services. This paper introduces the more challenging problem of provisioning multi-point-to-multi-point Ethernet LAN services over inverse-multiplexing networks and proposes novel overlay strategies based upon star, bus, and minimum spanning tree topologies. Multi-tiered protection schemes are also developed to help set up these respective overlay designs. Detailed simulation results are then presented along with directions for future work.

1. 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* (GFP), *virtual concatenation* (VCAT), and the *link capacity adjustment scheme* (LCAS) [2,3]. Namely, GFP defines efficient mappings onto TDM channels for client protocols 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].

Now, as these 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

* Corresponding author. Tel.: +1 505 277 1475.
E-mail address: ghanin@yahoo.com (N. Ghani).

1573-4277/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.osn.2010.02.001

all optical *dense wavelength division multiplexing* (DWDM) networks use SONET/SDH-based 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 over 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-added" services and require further investigation. Now, the authors in [17] have recently tabled basic multi-point EoS solutions using mesh and star overlays. However, this paper builds upon this initial work by presenting a more complete introduction to the problem and then proposing more advanced bus and *minimum spanning tree* (MST) overlays. The paper is organized as follows. First, Section 2 reviews the existing research work on NGS/EoS survivability. Next, Section 3 introduces the EoS LAN problem and proposes novel bus and MST overlay algorithms. Section 4 then presents commensurate connection group (overlay) provisioning schemes based upon inverse multiplexing and tiered (partial) protection strategies. Post-fault restoration mechanisms are also added here. These solutions are evaluated in Section 5 using simulation, and conclusions and directions for future work are presented in Section 6.

2. 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. The idea here is to ensure 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 tables some strategies for "degraded-service-aware" provisioning to route sub-connections to ensure that no one path carries more than a given fraction of the 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/inverse multiplexing. For example, [12] defines the *cumulative differential delay routing* (CDDR) problem to resolve an integral number of sub-connection paths to limit destination (sink) memory requirements. The CDDR problem is shown to be both NP-complete and provably hard to estimate within a constant factor, and hence two heuristics are proposed and studied using simulation. Meanwhile, [13] develops a modified link-weight k-shortest path algorithm to account for lower bounds on differential delay, and the results show improvement over more traditional algorithms. Specifically, link costs are computed as combinations of the original link weights (operator-specified) and their inverses, and the path cost functions are appropriately bounded.

Owing to the close coupling of SONET/SDH and optical DWDM technologies inside metro/core transport domains, researchers have also studied EoS grooming over DWDM networks. For example, [14,15] 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 multi-point "value-added" Carrier Ethernet LAN 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.

3. 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 set up of multiple point-to-point TDM connections, i.e., to "emulate" multi-point-to-multi-point connectivity via *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 2 are only applicable to point-to-point EPL services. However, within the broader networking context, some "overlay" design studies have been done. For example, the *resilient overlay network* (RON) [18] proposes a static Internet overlay to improve routing resiliency. Meanwhile, the *service overlay network* (SON) [19] addresses *quality of service* (QoS) support using queuing theory and optimization 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) scheme for node/link selection with reconfiguration. Albeit of some relevance here, the above efforts do

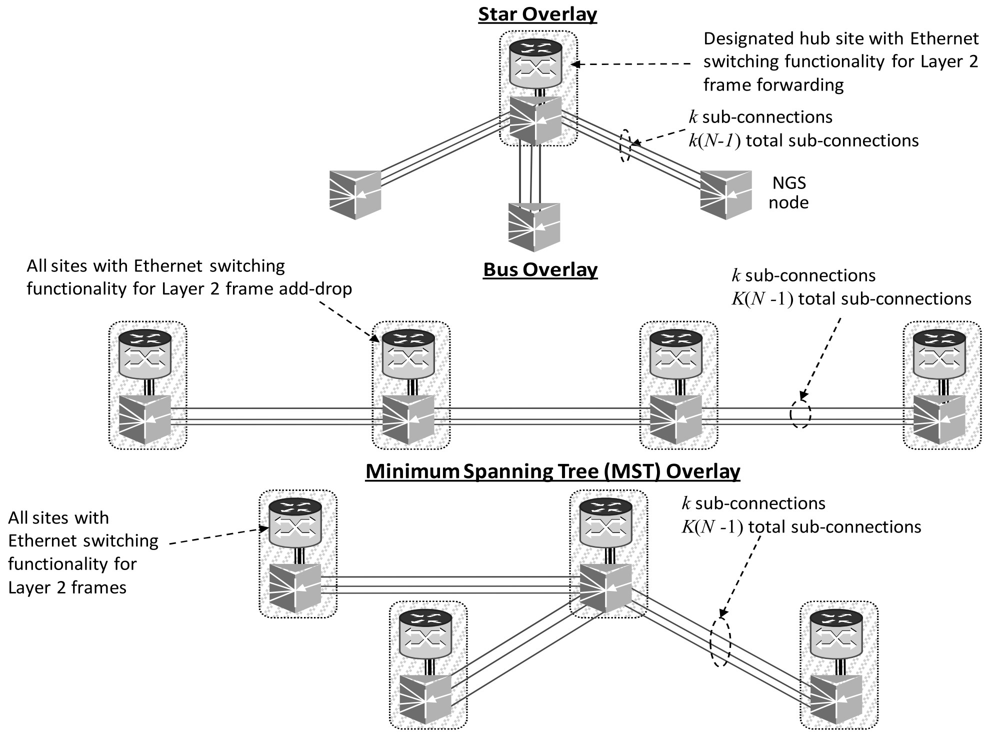

Fig. 1. Ethernet LAN overlays: star, bus, minimum spanning tree (MST).

not account for the specialized inverse-multiplexing capability of EoS settings.

Along these lines, this paper studies dynamic "on-demand" provisioning of multi-point Ethernet LAN requests in EoS networks. Specifically, the proposed solution makes minimal assumptions on demand arrivals and develops graph-based heuristics to provision random requests. Carefully note that alternate formulations can be developed for static or semi-static "batch arrival" demands, i.e., as these cases provide more a priori demand information. Sample approaches here include *integer linear programming* (ILP) optimization and genetic algorithms. However, these solutions are left for future study as the goal here is to basically introduce the multi-point EoS problem and present an initial solution. In general, "on-demand" provisioning, as studied herein, is quite germane to workflow and grid-computing environments in which applications require multi-point connectivity for limited durations.

The proposed solution herein assumes an advanced "NGS-capable" network layer supporting multiple overlying Ethernet switches and comprises two key phases, overlay topology design and connection group provisioning. The first phase (detailed in this section) builds a "Layer 2" connection group overlay for an incoming LAN request, i.e., specifies the group of point-to-point connections to set up the LAN. Next, the second phase (detailed in Section 4) resolves the individual connections in the overlay over the SONET/SDH layer using inverse multiplexing and tiered protection. From an implementation perspective, both of these steps are performed by a centralized entity with full topological and resource knowledge of the underlying SONET/SDH network. Hence the individual SONET/SDH switches need not be aware of the Layer 2 overlay topology, and instead only need to source/sink end-to-end inverse multiplexed connections. Conversely, the overlying Ethernet switches (connected to SONET/SDH switches) will become aware of this topology after running appropriate tree-based learning protocols.

Now, with regards to overlay topology design, three different schemes are proposed, based upon *star*, *bus*, and *minimum spanning tree* (MST) topologies; see Fig. 1. In order to introduce these designs properly, the requisite notation is first introduced (all vector and set entities in bold). Namely, 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., $\boldsymbol{V} = \{v_1, v_2, \ldots, v_N\}$ $\boldsymbol{L} = \{l_{12}, l_{13}, \ldots, l_{ij}\}$. Here, link l_{ij} is the link between nodes i and j of Cunits 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 endpoints, 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

```
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
```

Fig. 2. Bus overlay.

chosen, each of the constituent point-to-point connections can still be inverse-multiplexed, i.e., "split", into 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 schemes are now presented.

3.1. Star overlay

The star overlay scheme was recently tabled in [17] and is presented here for the sake of reference. This approach uses a designated hub site to provide Ethernet connectivity to all other nodes, assuming that it has adequate Layer 2 switching capabilities, provided by either the client or carrier node. This essentially is equivalent to a single level tree overlay (single parent); see Fig. 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, which translates into $O(K(|\boldsymbol{v}_i| - 1)) = O(K|\boldsymbol{V}|)$ VCAT sub-connections (Fig. 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 resource utilization and/or lower blocking. Hence three different hub selection strategies are tabled [17].

Random hub selection (Star-RS): This approach randomly selects the LAN group's hub node, h_i, by using 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 LAN nodes. The goal here is to minimize resource utilization of the overall LAN group as

$$h_i = v_{ij^*} \quad \text{where } j^* = \min_j \left(\sum_{k=1,k\neq j}^{|v_i|} \text{hop}(v_{ij}, v_{ik}) \right), \tag{1}$$

where $1 \le j \le |\boldsymbol{v}_i|$. Since this scheme only uses *static* information, computational speedup can be achieved by pre-computing and storing inter-nodal hop-counts. Namely, $O(|\boldsymbol{v}_i|(|\boldsymbol{v}_i| - 1)) = O(|\boldsymbol{V}|^2)$ shortest path computations are required prior to start up 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 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 and inversely proportional to the available capacity on link l_{ij}:

$$\omega_{ij} = \frac{1}{c_{ij} + \varepsilon}, \tag{2}$$

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

$$h_i = v_{ij^*} \quad \text{where } j^* = \min_j \left(\sum_{k=1,k\neq j}^{|v_i|} \text{cost}(v_{ij}, v_{ik}) \right), \tag{3}$$

where $1 \le j \le |\boldsymbol{v}_i|$, and $\text{cost}(v_{ij}, v_{ik})$ 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., $O(|\boldsymbol{v}_i|(|\boldsymbol{v}_i| - 1)) = O(|\boldsymbol{V}|^2)$ dynamic 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.

3.2. Bus overlay

This overlay implements a linear inter-connection between the LAN sites, Fig. 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. Moreover, associated connection group selection is more involved here, as shown in the pseudocode listing of Fig. 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 to reflect the fact that all nodes in $\boldsymbol{v}_i$ are "unassigned"; see Fig. 2. Subsequently, an initial node pair is "assigned" by generating the first overlay link in $\boldsymbol{r}_i$, i.e., as per a given selection strategy (detailed shortly). The algorithm then iterates through the remaining "unassigned" nodes ($\boldsymbol{v}_i - \boldsymbol{r}_i$) to sequentially determine their ordering (connection group overlay links). Specifically, at each iteration the first and last nodes in $\boldsymbol{r}_i$ are analyzed to "assign" the next 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 at each iteration via a uniform distribution over the remaining "unassigned" nodes.

Minimum average hop (Bus-MAH): This scheme chooses bus nodes in order to minimize resource consumption. Namely, 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.,

$$r_i = \{r_{i1}, r_{i2}\} = \{v_{ik*}, v_{im*}\}, \tag{4}$$

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, $j \geq 3$; see Fig. 2). Specifically, the first and last "assigned" end-point nodes in $\boldsymbol{r}_i$, i.e., $\boldsymbol{r}_{i1}$ and $\boldsymbol{r}_{ij-1}$, are examined to determine the next "unassigned" node with the minimum average hop count. This is done by comparing the respective minimum hop count from the first node in the current ordering vector, i.e.,

$$x_1 = \text{hop}(r_{i1}, v_{ik*}), \tag{5}$$

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

$$x_2 = \text{hop}(r_{ij-1}, v_{im*}), \tag{6}$$

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

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

Akin to the Star-MAH scheme, this approach only uses static information to choose the bus ordering, i.e., $O(|\boldsymbol{v}_i|(|\boldsymbol{v}_i| - 1)) = O(|\boldsymbol{V}|^2)$ shortest path computations prior to starts 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 needed to select each node in the ordering, yielding a complexity of $O(|\boldsymbol{V}|^2)$.

Minimum average cost (Bus-MAC): This scheme follows 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 given by the following:

$$x_1 = \text{cost}(r_{i1}, v_{ik*}) \tag{8}$$

and

$$x_2 = \text{cost}(r_{ij-1}, v_{im*}) \tag{9}$$

```
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
```

Fig. 3. MST overlay based on Prim's algorithm.

where the cost() function is defined as in Section 3.1, $k^* = \min_k(\text{cost}(r_{i1}, v_{ik}))$, $v_{ik^*} \notin \boldsymbol{r}_i$ and $m^* = \min_m(\text{cost}(r_{ij-1}, v_{im}))$, $v_{im^*} \notin \boldsymbol{r}_i$. Since this scheme uses a dynamic resource state, akin to Star-MAC, it also has higher compute complexity, i.e., $O(|\boldsymbol{v}_i|(|\boldsymbol{v}_i| - 1)) = O(|\boldsymbol{V}|^2)$ shortest path computations for bus node selection, yielding a total compute complexity of $O(|\boldsymbol{V}|^2 \cdot ||\boldsymbol{V} \log |\boldsymbol{V}|) = O(|\boldsymbol{V}|^3 \log |\boldsymbol{V}|)$.

3.3. 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 Dijkstra's shortest-path search procedure.

The overall pseudocode for the MST overlay is shown in Fig. 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}; see Fig. 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, $j \geq 3$; see Fig. 3) to select an "unassigned" LAN node with the minimum hop count to a node in $\boldsymbol{r}_i$; i.e.,

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

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 start up 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:

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

where $1 \leq m \leq j-1$, $v_{ik} \notin r_i$, and the cost() function is defined as in Section 3.1. Since this scheme uses a dynamic resource state, akin to Bus-MAC, it has higher compute complexity, i.e., $O(|\boldsymbol{v}_i|(|\boldsymbol{v}_i|-1)) = O(|\boldsymbol{V}|^2)$ shortest path computations for bus node selection, yielding a total compute complexity of $O(|\boldsymbol{V}|^2 \cdot |\boldsymbol{V}| \log |\boldsymbol{V}|) = O(|\boldsymbol{V}|^3 \log |\boldsymbol{V}|)$.

Note that the above-detailed overlay schemes (star, bus, MST) represent increasingly complex topological choices. Now, it may be desirable to determine the "best" overlay type for a given underlying physical topology. In general, this is a rather difficult problem to solve formally, particularly for the case of random "on-demand" arrivals. However, it is postulated here that the MST overlay will likely give better results as it is a more generic representation which subsumes the other two; i.e., star and bus overlays are both special instances of MST overlay. Moreover, it is further postulated that a generic mesh is likely the "optimum" overlay type as it further generalizes upon the MST. Along these lines, future work can look at studying the optimal overlay topologies for a fixed, a priori set of LAN demands using techniques such as ILP modeling and genetic algorithms.

4. Multi-tiered LAN group provisioning

Carrier Ethernet LAN service users will demand flexible, multi-tiered survivability support. For example, most "regular" users will be satisfied with partial recovery against single faults. Alternatively, a subset of users may demand much more stringent 100% recovery, e.g., financial services, packet video transport, etc. To meet these requirements, the proposed framework provisions the LAN overlay connections (derived in Section 3) using inverse multiplexing and *tiered* protection and restoration algorithms. The overall aim here is to guarantee a minimum LAN throughput in the event of a single fault. To achieve this, a fractional protection factor, ρ ($0 \leq \rho \leq 1$), 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.

4.1. 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 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, $0 \leq j \leq n_i$. In turn, this connection can be "resolved" into multiple "sub-connections", up to a maximum of $K \leq x_i$, as per the inverse multiplexing factor. Although various policies are possible here, an "even" distribution approach is chosen in order to better distribute loads, akin to [10]. Specifically, consider integral division of x_{ij} by K, yielding

$$z = \left\lfloor \frac{x_{ij}}{K} \right\rfloor, \tag{12}$$

where the remainder term is given by

$$y = x_{ij} - Kz \tag{13}$$

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, $1 \leq k \leq K$. However, for the more general case of $y \neq 0$, the remainder term is simply distributed over the first r sub-connections. Hence, the resultant generic expression for the individual capacity for the k-th requested sub-connection, x_{ijk}, in STS-1 increments is

$$x_{ijk} = \begin{cases} z+1 & 1 \leq k \leq y \\ z & y < k \leq K, \end{cases} \qquad \sum_j x_{ijk} = x_{ij} = x_i; \tag{14}$$

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, as will be common in most operational settings. The use of inverse multiplexing to improve LAN overlay resiliency is detailed next.

4.2. 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-1 units 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 protection threshold of ρx_i STS-1 units. This approach ensures a "LAN-wide" throughput of at least ρx_i STS-1 units in the event of a single link failure. This tiered protection concept is shown in Fig. 4 for the case of bus overlay for the i-th LAN request of 12 STS-1 units (about 600 Mb/s fractional Ethernet) between three nodes, v_1, v_3, and v_9. (Note that similar diagrams can also be drawn for other overlays.) The request is mapped using an inverse multiplexing factor of $K = 3$ and a protection threshold of $\rho = 0.5$ (50%). Assuming a computed bus ordering of $\boldsymbol{r}_i = \{v_1, v_3, v_9\}$, this LAN request is transformed into a connection group containing two bi-directional SONET/SDH connections, $v_1 - v_9$ and $v_1 - v_9$, respectively. In turn, these connections are inverse multiplexed into $K = 3$ working sub-connections of 4 STS-1 units, e.g., routes $\boldsymbol{w}_{i11}$, $\boldsymbol{w}_{i12}$, and $\boldsymbol{w}_{i13}$ for LAN group connection $v_1 - v_3$; see Fig. 4.

Now, the protection threshold ρ mandates each individual connection to have at least $(0.5) \cdot 12 = 6$ STS-1 units of protection capacity. This is achieved by using dedicated protection on a *per-sub-connection* basis, similar to [10]. Namely, dedicated *link-disjoint* protection paths are computed for a minimal subset of working sub-connections until the desired threshold is achieved. Hence two protection sub-connections must be set up for each connection in Fig. 4, i.e., 4 STS-1 units each in routes $\boldsymbol{p}_{i11}$ and $\boldsymbol{p}_{i13}$ to protect the working routes $\boldsymbol{w}_{i11}$ and $\boldsymbol{w}_{i13}$ for the LAN

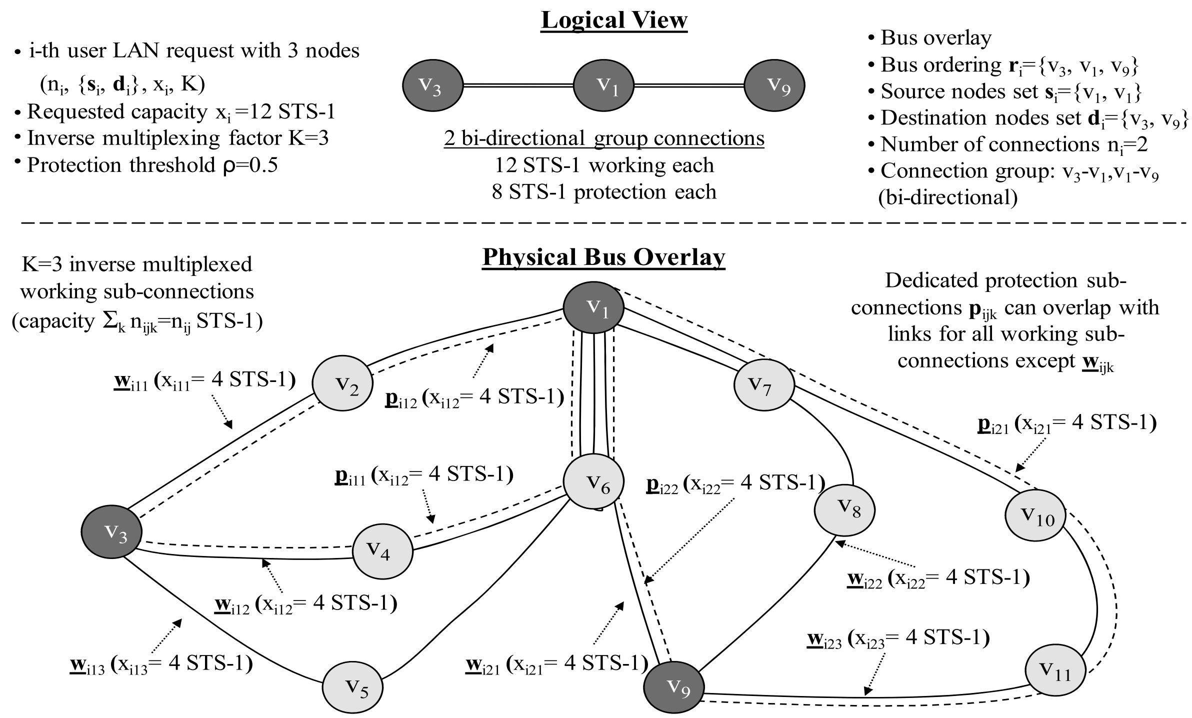

Fig. 4. Tiered LAN scheme for sample bus overlay.

connection $\boldsymbol{v}_1 - \boldsymbol{v}_3$. Note that, since protection is done on a per-sub-connection basis, the protection granularity is inversely proportional to the inverse multiplexing factor K. Hence it is possible for protection over-provisioning to occur for smaller values of K, e.g., a total of 8 STS-1 units of protection capacity is reserved for each connection in Fig. 4 even though the threshold is 6 STS-1, i.e., 33% over-provisioning. Although this inefficiency can be readily be resolved by appropriately "right-sizing" protection sub-connections, it is not considered here as equal-sized working/protection sub-connections simplify protection switchovers and mitigate edge buffering.

Finally, consider the actual working/protection provisioning algorithms for the LAN group connections. 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 set up *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 4.1), i.e., x_{ijk}, Eq. (14). Next, a modified *successive* Dijkstra's shortest-path computation scheme is used to iteratively compute the individual sub-connection route vectors, $\boldsymbol{w}_{ijk}$, for each requested sub-connection. Here, if a sub-connection is successfully routed, its capacity is pruned along all route links in $G'(\boldsymbol{V}, \boldsymbol{L})$. Furthermore, the algorithm only proceeds to the next group connection if the current connection is fully routed; otherwise, the LAN request is dropped. This scheme places no restrictions upon link overlap between sub-connections and only *feasible* links with sufficient capacity are considered; i.e., $c_{jk} \geq x_i$.

If all working group connections are successfully set up on the temporary graph $G'(\boldsymbol{V}, \boldsymbol{L})$, the tiered protection phase is initiated. Here the algorithm uses the "left-over" capacity in $G'(\boldsymbol{V}, \boldsymbol{L})$ and again iteratively tries to set up 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 set-up. 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. Since the maximum number of LAN group connections is $O(n_i) = O(|\boldsymbol{V}|^2)$, the working/protection provisioning algorithm is of $O(K|\boldsymbol{V}|^3 \log |\boldsymbol{V}|)$ complexity.

In addition, 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 3). 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 the 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. Finally, optional post-fault restoration of failed LAN connections (sub-connections) is also performed. Namely, all *non-protected* VCG members (e.g., $\boldsymbol{w}_{i13}$; see Fig. 4) traversing a failed link can be re-routed after failure notification by first pruning failed links on the network graph $G(\boldsymbol{V}, \boldsymbol{L})$,. The goal here is to achieve *full* recovery for partially protected LAN requests, thereby enabling carriers to achieve improved service recovery rates for lower-priced offerings.

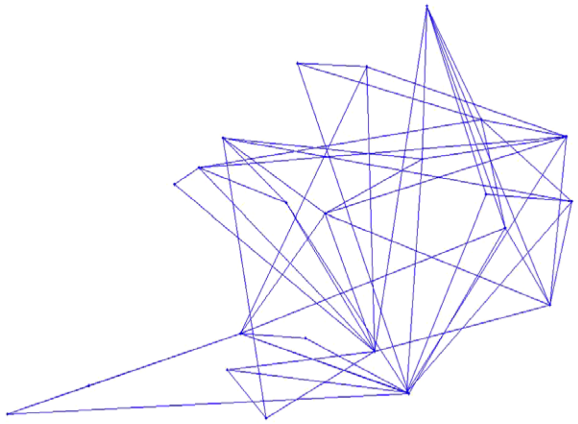

Fig. 5. Sample Waxman network topology (24 nodes, 43 links).

Hence this framework is very generic and can incorporate vanilla protection ($K = 1/\rho = 1$) and restoration ($K = 1/\rho = 0$ with post-fault restoration).

Carefully note that the above working/protection sub-connection provisioning schemes only address bandwidth constraints and do not incorporate any differential bandwidth delays. Indeed, the joint provision of bandwidth and delay constraints here is much more complex, and has been shown to be an NP-complete problem for even the simple single point-to-point provisioning case [12]. However, since the focus of this initial contribution is to introduce the multi-point EoS LAN problem, such further constraint formulations are left for future study. Moreover, many commercial SONET/SDH VCAT chipsets provide a decent level of buffering to handle differential delays. As such, these constraints will be most relevant in very long-distance networks and likely not metro and regional scenarios.

5. Performance analysis

The performance of multi-point EoS overlay schemes is studied by coding specialized simulation libraries in the *OPNET Modeler*™ simulation environment. Tests are done using two topologies, the ubiquitous NSFNET network (16 nodes and 25 links, node degree 1.56) and the randomly generated Waxman network (24 nodes, 43 links, node degree 1.79). Here, the latter topology is generated using a random graph generator (in MATLAB) based upon the Waxman model for computer network growth. Specifically, the number of nodes in a domain here is given by a Poisson process with intensity proportional to the area of the domain (set to 0.2 here). In addition, inter-nodal edges are assigned in proportion to their distances between nodes. In both topologies, all network elements here are generic NGS-capable *broadband digital cross-connects* (BB-DCS) nodes with STS-1 (51.84 Mb/s) switching granularity and OC-48 link speeds (see Fig. 5).

All 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 $\mu = 600$ s is used and the mean inter-arrival times are adjusted according to load (note that this scaled relative value is not necessarily representative of real-world holding times). All LAN group sizes are uniformly distributed from 3 to 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. All runs are averaged over 500,000 LAN requests and two modified tracking metrics are also introduced. First, a modified Erlang load is defined to account for varying LAN group sizes by scaling by the respective LAN connection counts, i.e.,

$$\text{Modified Erlang load} = \sum_{n=x_1}^{x_2} \frac{(n-1)}{3} \cdot \frac{\mu}{\lambda}. \tag{15}$$

In the above, the LAN sizes range from $x_1 = 3$ to $x_2 = 5$ nodes and $1/\lambda$ represents the mean inter-arrival rate. In addition, a modified *bandwidth blocking rate* (BBR) metric is also defined to account for varying LAN bandwidth request and group sizes. Consider the basic BBR definition, which is computed as the ratio of the aggregate bandwidth of failed requests to the aggregate bandwidth of all requests [10]. Along these lines, the modified LAN BBR metric scales this value by the individual LAN connection counts, e.g.,

$$\text{Modified LAN BBR} = \frac{\sum_{i=1}^{\#failed} (n_i - 1) \cdot x_i}{\sum_{j=1}^{\#requested} (n_i - 1) \cdot x_i}. \tag{16}$$

Initial tests are done to measure the carried load for non-protected LAN scenarios ($\rho = 0$) given a nominal LAN BBR of 5%. These scenarios are very relevant to carriers as they indicate the true "load-carrying" (i.e., revenue generation) capacity of the network at a "low-blocking" operating point. The results are shown in Figs. 6–8 and reveal some critical findings. Foremost, the results for the NSFNET topology are summarized in Figs 6(a) (star), (b) (bus), and 7(a) (MST). The findings here indicate that the more advanced bus and MST overlays both yield notably higher carried load than the star overlay. This improvement is due to the reduced in-degrees in the respective overlay topologies. Furthermore, it is seen that MAC overlay selection tends to yield the best performance for the bus and MST overlays. However, the opposite is observed for the more basic star overlay, where hop count overlay selection gives the highest carried load, i.e., MAH, load-balancing; see Fig. 6(a). Also, it is seen that the MST overlay slightly out-performs the bus overlay in the NSFNET topology, i.e., about 5% higher carried load, as it 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 (see Sections 3.2 and 3.3). Results with the bus and MST overlays for the denser Waxman network, Fig. 8, also show similar improvements with MAC overlay selection and load-balancing, i.e., a carried load about 5% higher than with competing bus overlay schemes. Furthermore, inverse multiplexing is seen to be beneficial for all overlay types, with increasing values of K yielding anywhere from 5% to 25% higher carried load versus "non-inverse multiplexed" operation ($K = 1$). However, the relative gains tend to decrease with

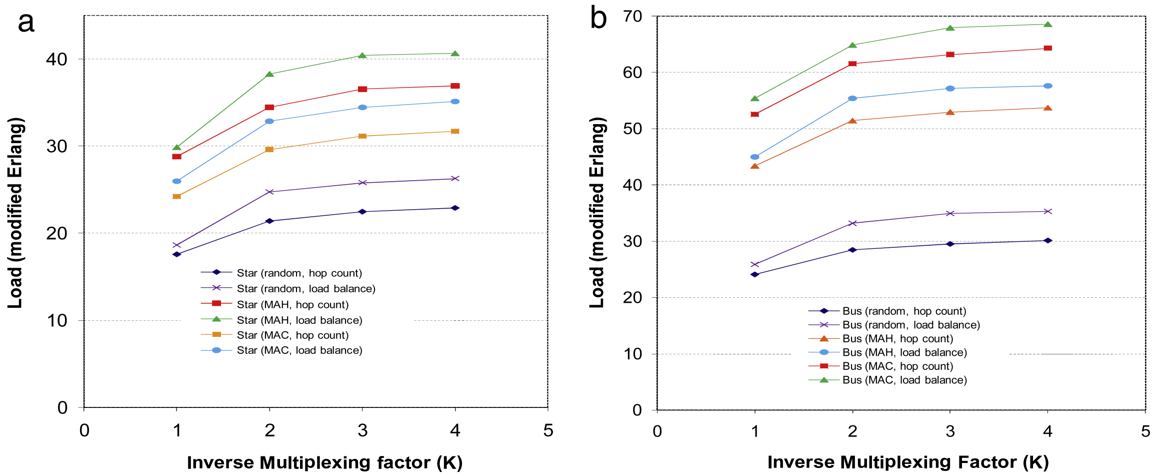

Fig. 6. NSFNET carried load for 5% LAN BBR and $\rho = 0$: (a) star overlay, (b) bus overlays.

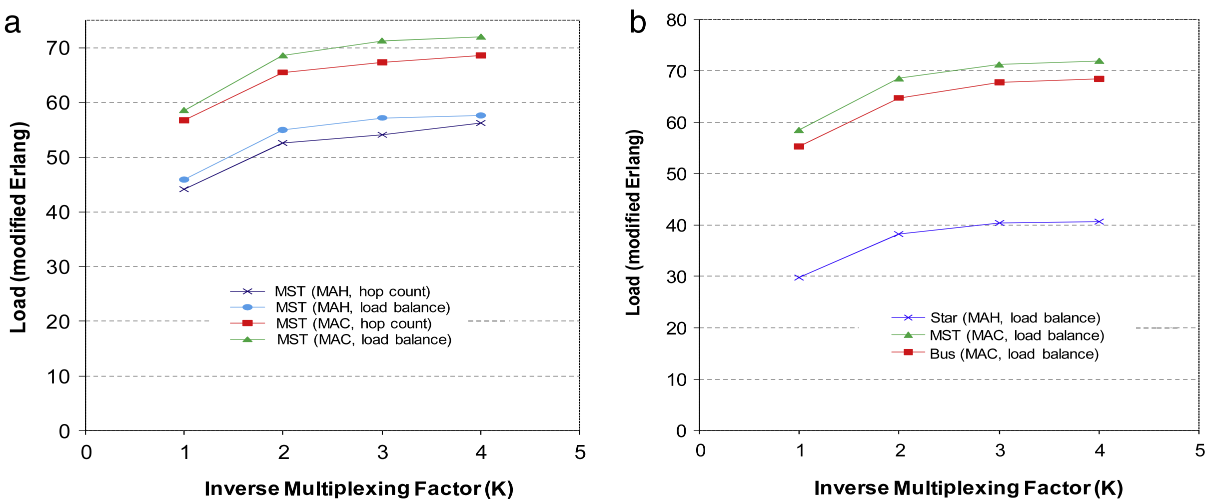

Fig. 7. NSFNET carried load for 5% LAN BBR and $\rho = 0$: (a) MST overlay, (b) comparison.

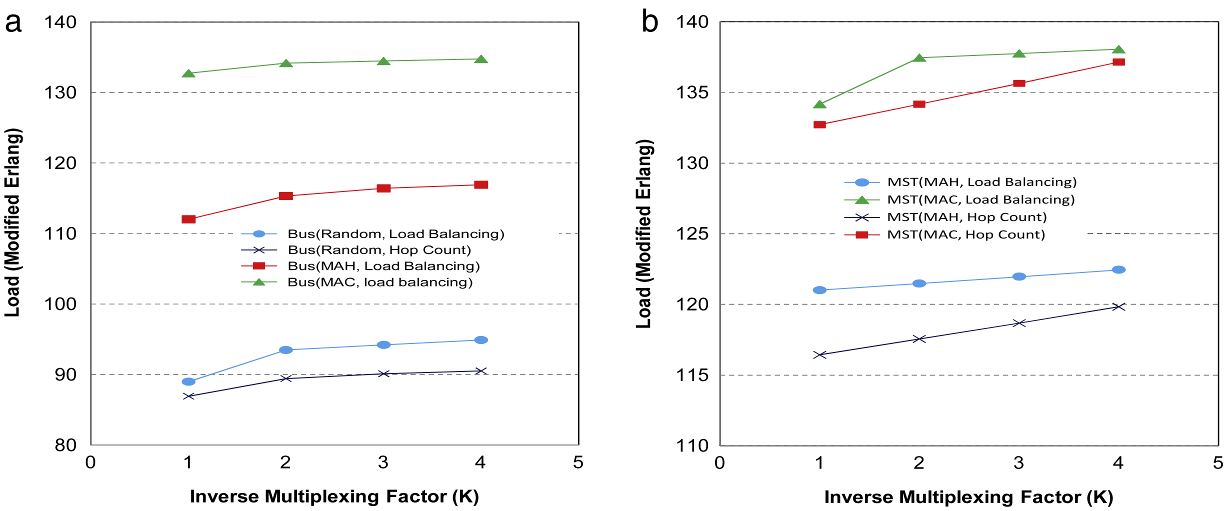

Fig. 8. Waxman network carried load for 5% LAN BBR and $\rho = 0$: (a) bus overlay, (b) MST overlay.

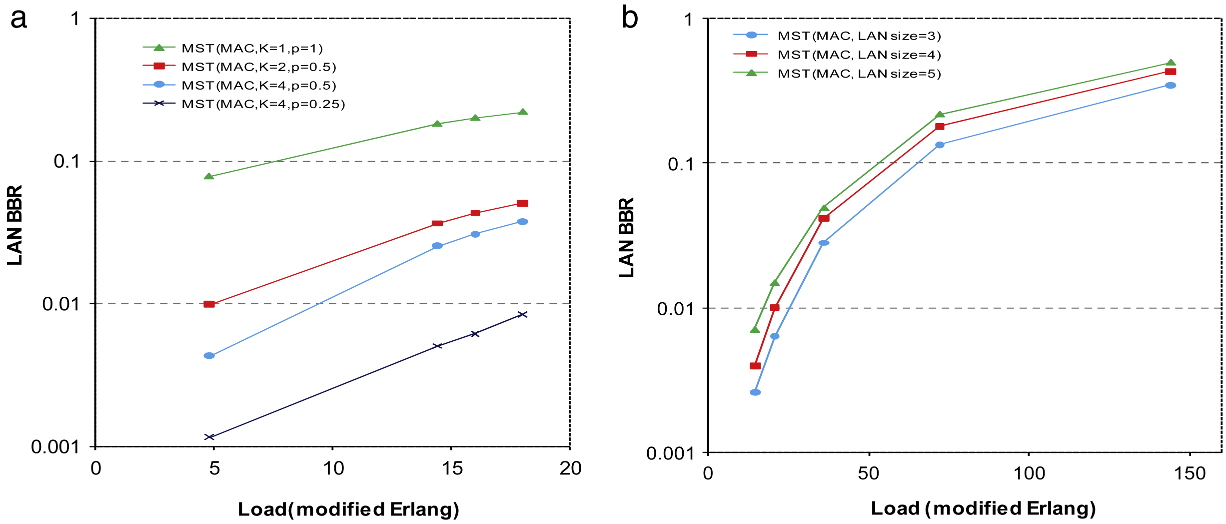

Fig. 9. NSFNET LAN BBR for MST overlay: (a) MST-MAC with varying K, ρ, (b) $K = 4$, $\rho = 0.25$.

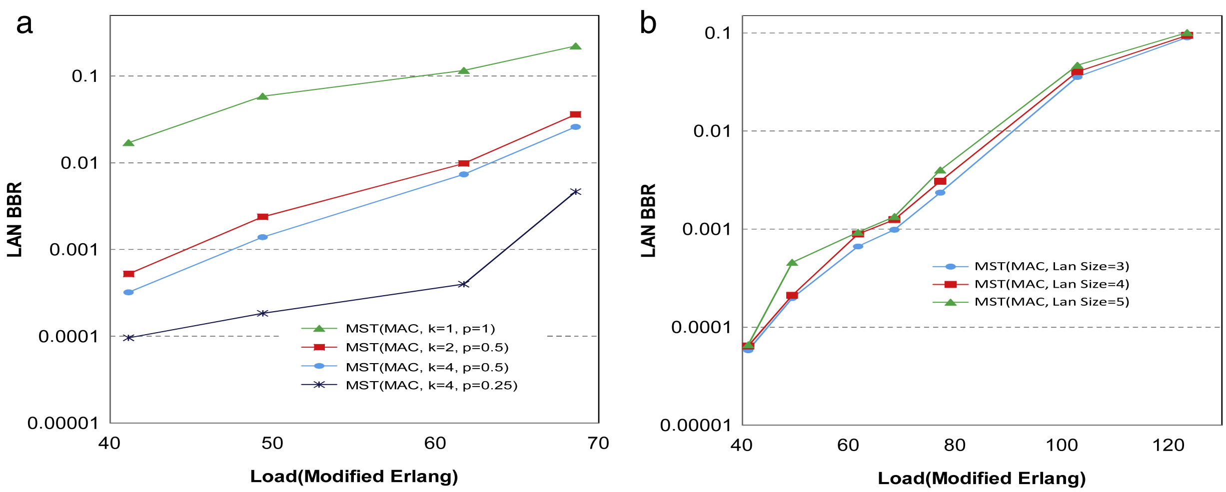

Fig. 10. Waxman network LAN BBR for MST overlay: (a) MST-MAC with varying K, ρ, (b) $K = 4$, $\rho = 0.25$.

increased node degree (connectivity), as observed in the Waxman network, averaging 5–10% (similar findings are obtained with other larger Waxman graphs as well). Added simulations with faster OC-192 links (not shown) also 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%).

As the above results show better performance with the MST overlay, the associated BBR performance is tested for this LAN overlay with tiered protection demands; see Fig. 9 (NSFNET) and 10 (Waxman network). Namely, Figs. 9(a) and 10(a) plot the LAN BBR results for varying protection thresholds ($\rho = 0, 0.25, 0.5$) and inverse multiplexing factors ($K = 1, 2, 4$). Here it is seen that the reduced protection factors give significantly lower LAN blocking, as expected. Also, larger inverse multiplexing factors yield decent gains for equivalent protection; e.g., $K = 4/\rho = 0.5$ gives about 10–30% lower blocking than $K = 2/\rho = 0.5$ (Figs. 9(a) and 10(a)). Furthermore, the *individual* blocking rates for different LAN request sizes (for $K = 4/\rho = 0.25$) are also plotted in Fig. 9(b) (NSFNET) and 10(b) (Waxman network). These results show that although the larger five node LAN requests experience higher blocking versus smaller three node requests, the differences are generally below 15% in NSFNET and even less, about 5%, in the denser Waxman network.

Finally, tiered LAN protection is tested in conjunction with post-fault restoration of failed *non-protected* working VCAT sub-connections. The goal here is to achieve *full* recovery for partially protected (lower tier) LAN services. In these tests, all link failures have exponentially distributed inter-arrival times and *mean-time-to-repair* (MTTR), averaging around 600 s (scaled). The MTTR values represent truck roll repair times and are typically much larger than protection switchover times. Furthermore, restoration performance is gauged by measuring *recovery rates* for carried loads up to 20% blocking, i.e., high load settings. Namely, the restoration rate is defined as the

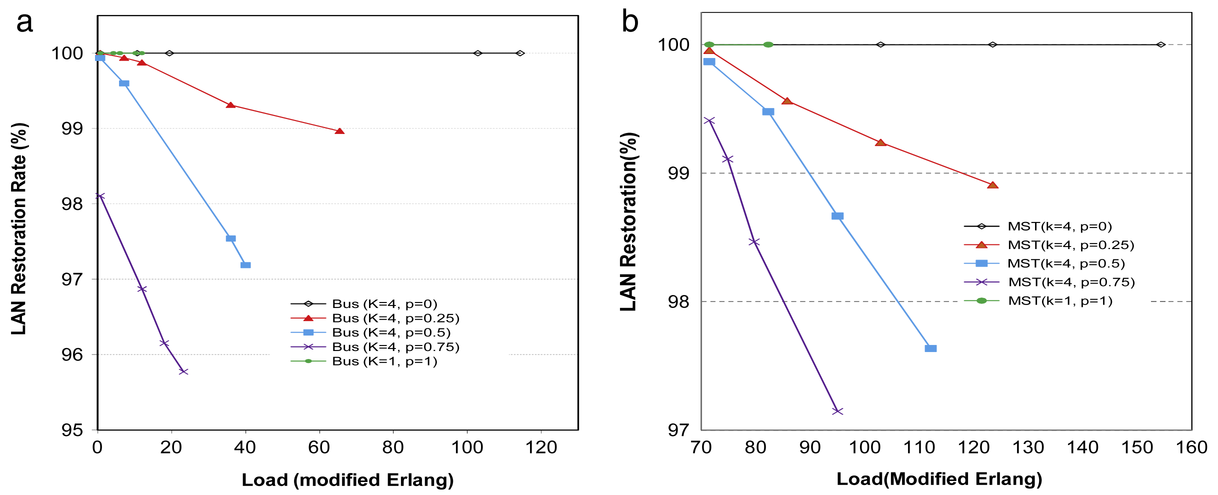

Fig. 11. Restoration with load-balancing routing: (a) NSFNET (Bus-MAC), (b) Waxman (MST-MAC).

percentage of LAN groups which recover *all* of their failed non-protected connections, i.e., regain full 100% pre-fault throughput.

The LAN restoration rates for bus and MST overlays with varying protection thresholds and inverse multiplexing factors are plotted in Fig. 11(a) (NSFNET) and (b) (Waxman network). Here, the $K = 1/\rho = 1$ case corresponds to regular non-inverse multiplexed protection (e.g., 1+1, 1:1) whereas $\rho = 0.25$ and 0.5 correspond to partial protection. Overall, these results reveal some very important findings with regards to partial protection (as enabled by inverse multiplexing). Namely, it is seen that increased levels of resource over-provisioning do not necessarily yield higher post-fault recovery. For example, tiered protection (i.e., $K = 4/\rho = 0.25$) yields over 99% recovery *and* gives almost four times the carried load versus full protection for 20% BBR in both networks (i.e., $K = 1/\rho = 1$). Results for pure restoration (i.e., $K = 4/\rho = 0$) are even better, showing full 100% recovery in all cases. These findings indicate that lower levels of pre-provisioned protection bandwidth actually leave more spare capacity in the network, and this is generally sufficient for recovering almost all failed sub-connections after single link failure events. As a result, post-fault sub-connection restoration is deemed a very effective means of achieving full LAN recovery.

6. Conclusions

This paper studies the performance of various topology overlay schemes for multi-point Ethernet LAN services, including star, bus, and minimum spanning tree. A comprehensive tiered survivability approach for provisioning these overlays is also detailed using both pre-fault protection and post-fault restoration algorithms. The overall findings show the lowest blocking and/or highest carried load with more advanced minimum spanning tree overlays, particularly when combined with load-balancing routing of sub-connections (i.e., active resource usage state). Finally, the use of post-fault restoration is particularly beneficial and is shown to yield very high recovery rates even with generally lower levels of pre-fault protection reservation. Building upon this work, future efforts will focus on more elaborate shared protection strategies as well as optimization formulations.

Acknowledgements

This research has been supported by the US National Science Foundation (NSF) Computer and Information Sciences Division under Award # 0806637. The authors are very grateful for this 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 40 (5) (2002) 104–112.
[3] G. Bernstein, et al., VCAT/LCAS in a clamshell, IEEE Communications Magazine 44 (5) (2006) 34–36.
[4] N. Ghani, et al., On IP-WDM integration: A retrospective, IEEE Communications Magazine 41 (9) (2003) 42–45.
[5] A. Kasim (Ed.), Delivering Carrier Ethernet: Extending Ethernet Beyond the LAN, McGraw Hill Publishers, 2007.
[6] L. Fang, et al., The evolution of carrier Ethernet services—Requirements and deployment case studies, IEEE Communications Magazine 46 (3) (2008) 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, in: IEEE Infocom 2004, Hong Kong, March 2004.
[9] R. Roy, et al. Degraded-service-aware multipath provisioning in telecom mesh networks, in: 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 13 (1) (2007) 79–92.
[11] C. Ou, et al., Survivable virtual concatenation for data over SONET/SDH in optical transport networks, IEEE/ACM Transactions on Networking 14 (1) (2006) 218–231.
[12] A. Srivastava, A. Srivastava, Flow aware differential delay routing for next-generation Ethernet over SONET/SDH, in: IEEE ICC 2006, Istanbul, Turkey, June 2006.
[13] S. Ahuja, et al. Minimizing the differential delay for virtually concatenated Ethernet over SONET Systems, in: IEEE ICCCN 2004, Chicago, IL, October 2004.

[14] R. Srinivasan, A. Somani, Dynamic routing in WDM grooming networks, Photonic Network Communications 5 (2) (2003) 123–135.
[15] K. Zhu, et al., Ethernet-Over-SONET (EOS) Over WDM in Optical Wide-Area Networks (WANs): Benefits and challenges, Photonic Network Communications 10 (1) (2005) 107–108.
[16] M. Bataynet, et al. Cost-Efficient Routing in Mixed-Line-Rate (MLR) optical networks for carrier-grade Ethernet, in: IEEE/OSA OFC 2008, San Diego, CA, February 2008.
[17] C. Xie, et al. Multi-point Ethernet over next-generation SONET/SDH, in: IEEE ICC 2009, Dresden, Germany, June 2009.
[18] D. Andersen, et al. Resilient Overlay Networks, in: 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 11 (6) (2003) 870–883.
[20] Y. Zhu, M. Ammar, Algorithms for assigning substrate network resources to virtual network components, in: IEEE INFOCOM 2006, Barcelona, Spain, April 2006.
[21] R. Prim, Shortest connection networks and some generalizations, Bell System Technical Journal 36 (1957) 1389–1401.