Online Routing of Bandwidth Guaranteed Paths with Local Restoration using Optimized Aggregate Usage Information

Saqib Raza, Faisal Aslam and Zartash Afzal Uzmi
Department of Computer Science
Lahore University of Management Sciences, Pakistan
Email: {saqibr,faisal,zartash}@lums.edu.pk

***Abstract*— We investigate the problem of distributed online routing of bandwidth guaranteed paths with local restoration. A unified model is proposed that captures the bandwidth sharing characteristic of backup paths that provision local restoration, corresponding to different fault models. We apply the model to describe bandwidth sharing on backup paths for varying degrees of network state information. The extent of backup bandwidth sharing depends on the amount of network state information made available through routing protocols. A key design criterion for traffic engineering schemes is to maximize the sharing between backup paths, while minimizing this protocol overhead. Kodialam et al. demonstrated in [1] that propagating a constant amount of aggregated information per link leads to cost effective bandwidth sharing. We propose oAIS, a new aggregate information scenario, in which we judiciously select the propagated information, such that the protocol overhead is identical to that in [1]. Simulations show that oAIS outperforms other information scenarios with comparable protocol overheads.**

I. Introduction

The destination based forwarding paradigm employed in plain IP routing does not support routing network traffic along explicit routes determined through constraint based routing [2]. The emergence of *Multi-Protocol Label Switching* (MPLS) has overcome this limitation of traditional shortest path routing, by presenting the ability to establish a virtual connection between two points on an IP network, maintaining the flexibility and simplicity of an IP network while exploiting the ATM-like advantage of a connection-oriented network [3]. Ingress routers of an MPLS network classify packets into forwarding equivalence classes and encapsulate them with labels before forwarding them along pre-computed paths [4]. The path a packet takes as a result of a series of label switch operations in an MPLS network is called a *label switched path* (LSP). LSPs may be routed through constraint based routing, that adapts to current network state information (e.g. link utilization) and selects explicit routes that satisfy a set of constraints. The ability to explicitly route network traffic using constraint based routing enables service providers to provision QoS for network traffic, and also leads to efficient network utilization [5].

The provisioning of bandwidth guaranteed LSPs has been the subject of recent research [1], [6]–[9]. Another important QoS objective is *restoration routing*. Restoration routing involves fault-persistent LSP setup such that the guaranteed bandwidth remains provisioned even if links or network nodes fail. This means that resources for backup paths are allocated during the initial routing of the LSP such that upon detection of failure, traffic is promptly switched on to preset backup paths. A key QoS parameter is *restoration latency*, which is the time that elapses between the occurrence of a failure and the diversion of network traffic to a preset backup path. Restoration latency should be kept low to minimize disruption to network traffic. Optimal restoration latency ensues if the network node immediately upstream the point of failure along the primary path is able to switch the network traffic onto the preset backup path. Such a model for restoration is called *local restoration*, also referred to as fast restoration, with restoration latencies comparable to those in SONET rings. We focus on routing of bandwidth guaranteed paths with local restoration.

One of the major costs of constraint based routing is the protocol overhead that stems from the need to periodically disseminate network state information so that distributed network elements may compute constrained routes. Increased protocol traffic incurs significant costs in terms of network utilization, memory, update-processing and associated context switching overheads [5]. In a practical setting, efficiency gains of constrained routing must be weighed against the magnitude of additional information that has to be propagated through routing protocols. Kodialam et al. have shown how aggregated link usage information can be used to achieve cost-effective routing of locally restorable bandwidth guaranteed paths [1]. We improve upon these results by judiciously selecting the aggregated link usage information circulated across the network without increasing the protocol overhead.

The rest of the paper is organized as follows: Sections II, III, and IV provide the problem definition and the relevant background. We describe a unified bandwidth sharing model in section V. Section VI details various information scenarios pertaining to network state, including the new optimized aggregate information scenario, within the framework of our model. Our simulation setup and results are given in section VII, while we draw our conclusions in section VIII.

II. Routing Paradigm

Trends in backbone and carrier networks towards the fast online provisioning of bandwidth guaranteed paths necessitate

routing of requests as they arrive [1], [8], [9]. We, therefore, consider the problem of online routing of restorable bandwidth guaranteed label switched paths. Online routing implies that routing requests arrive one at a time and there is no a priori information about future requests. In contrast, an offline algorithm that has advance knowledge of the entire set of LSP requests can optimally place the associated traffic onto the network. In the context of routing bandwidth guaranteed LSPs each LSP request has an associated bandwidth demand. In order to serve an LSP request a bandwidth guaranteed primary path is setup. In addition, a set of one or more bandwidth guaranteed backup paths are setup to provide connectivity in event of failure of network elements along the primary path. The characteristics and requirements for the backup paths are detailed in section III. We do not consider the case where new routes are computed for LSP requests that have already been routed, to optimize certain network state parameters or to accommodate new LSP requests.

The optimality of the primary and backup paths computed to serve an LSP request is a function of the network state information available during path computation. A centralized path computation server can maintain the exact network state since such a server is essentially responsible for all updates to network state. However, a centralized path computation server incurs additional costs in terms of high processing power and high bandwidth control channels [10]. Distributed control entails autonomous path computation by a router, based on the router's view of the network state. Network state information may be periodically distributed by other routers via link state routing protocols. In our case the ingress node of the LSP being routed computes the primary and the backup path(s).

III. Restoration Features

This section delineates the features and options that constitute our restoration model.

A. Restoration Level

Note that when an element (a link or a node) along the primary LSP fails, the traffic traversing the LSP must be switched onto a preset backup path that is routed divergently from the failure element. It is obvious that switching from the primary path to a backup path in event of failure must occur at a node that is upstream the point of failure along the primary path. The backup path should merge with the primary path downstream the point of failure. We refer to the node at which a backup path merges with the primary path as the *merge point* for that backup path. There are different restoration levels based on how further upstream along the primary path is the node that switches the LSP traffic onto the backup path.

In *end-to-end restoration*, also known as *path restoration*, a single backup path that is link and node disjoint with the primary path is used in event of any failure on the LSP's primary path. Thus, the head-end of the backup path is the LSP ingress node and the merge point is the LSP egress node. In *local restoration*, separate backup paths are computed to protect individual network elements along the primary LSP, such that the network node immediately upstream a point of failure along the primary path switches the LSP traffic onto the backup path. In the context of local restoration we will refer to the node immediately upstream the failure element along the primary path as the *point of local repair*. The merge point, in the case of local protection, is a node downstream the failure element in the primary path. Local restoration enables prompt switchover of network traffic unto preset backup paths in event of network failure and, therefore, results in optimal restoration latency. We only consider local restoration in this paper.

B. Fault Models

We cannot guarantee bandwidth restoration for all failure scenarios. It is possible to conceive a situation in which multiple failures in a network may disable the entire set of primary and the backup paths for an LSP. However, link or node failure is a low probability event. Network measurements reveal that chances of multiple failures are even lower. Furthermore, upon failure along the primary path new reoptimized primary and backup paths may be provisioned, with local restoration serving only as a temporary measure [11]. The probability of multiple failures in the window of time it takes to setup reoptimized paths is negligible. A more realistic restoration objective is to provide protection against failure of a single link or node. We consider local protection against three fault models: *single link failure*, *single node failure*, and *single element failure* (link failure or node failure). In order to elucidate local recovery for each of the three fault models we distinguish between two types of backup paths: *next-hop* paths and *next-next-hop* paths.

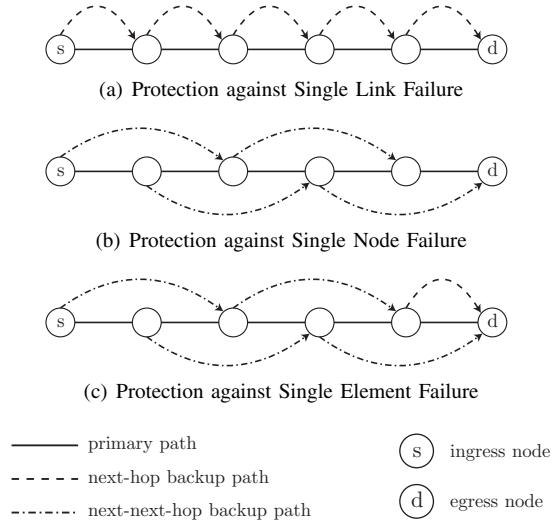
Definition 1: *A next-hop path that spans a link* (i, j) *is a backup path which*

a) *originates at node* i*, and*
b) *provides restoration for one or more primary LSPs that traverse* (i, j)*, if* (i, j) *fails.*

Definition 2: *A next-next-hop path that spans a link* (i, j) *is a backup path which*

a) *originates at node* i*, and*
b) *provides restoration for one or more primary LSPs that traverse* (i, j)*, if either* (i, j) *or node* j *fails.*

Fig. 1 delineates how local restoration may provide recovery for each of the three fault models. The figure shows backup paths merging with the primary path at the node immediately downstream the point of failure. This may not necessarily be the case as will be explained later. As obvious from fig. 1(a), establishing next-hop paths spanning every link along the primary path provides restoration in event of single link failure. Fig. 1(b) shows that setting up next-next-hop paths spanning all except the last link along the primary path provides restoration in event of single node failure. Note that such a configuration also protects against the failure of all except the last link. In order to provision restoration in event of single element failure an additional next-hop backup path



(a) Protection against Single Link Failure

(b) Protection against Single Node Failure

(c) Protection against Single Element Failure

Fig. 1. Fault Models

is setup spanning the last link as depicted in fig. 1(c). Further note that the point of local repair for this next-hop backup path is the penultimate node along the primary LSP and the merge node is the egress node of the primary LSP.

C. Restoration Modes

There are two restoration modes for local protection: *one-to-one* restoration and *many-to-one* restoration.

Definition 3: *One-to-one local restoration is a restoration mode in which*

a) *for each primary LSP that traverses a link* (i, j) *and is protected by a next-hop path, a separate next-hop path that spans* (i, j) *is established, and*
b) *for each primary LSP that traverses a link* (i, j) *and is protected by a next-next-hop path, a separate next-next-hop path that spans* (i, j) *is established.*

Definition 4: *Many-to-one local restoration is a restoration mode in which*

a) *for all primary LSPs that traverse a link* (i, j) *and are protected by a next-hop path, a single next-hop path that spans* (i, j) *is established, and*
b) *for all primary LSPs that traverse links* (i, j) *and* (j, k)*, and are protected by a next-next-hop path, a single next-next-hop path that spans* (i, j) *is established.*

The many-to-one restoration mode takes advantage of the MPLS label stack capability [12], conserving label space and thus the amount of LSP state information maintained at each node. We mentioned earlier that in local protection the merge point of the backup path is a node downstream the failure element along the primary path. Note that when a backup path is protecting more than one primary LSP, as in the many-to-one restoration mode, the set of nodes downstream the failure element may differ for each protected LSP. However, each such set must contain the node immediately downstream the point of failure. Therefore, the merge point of a backup path in the many-to-one restoration mode is the node immediately downstream the point of failure. Backup paths in one-to-one restoration can intersect the primary LSP at any node that is downstream the failure element along the primary path. The one-to-one restoration mode places less constraint on the placement of backup paths as compared to the many-to-one mode and, therefore, results in more optimal routing of backup paths. This paper considers the one-to-one restoration mode.

IV. Problem Definition

Our network consists of n nodes and m unidirectional edges. LSP requests arrive one by one at the ingress node of the LSP, and the routing algorithm has no a priori knowledge of future requests. An LSP request is characterized by the parameters s_k, d_k and b_k which, respectively, specify the ingress node, the egress node and the bandwidth demand of the k^{th} LSP request. In order to serve an LSP request a bandwidth guaranteed primary path must be setup along with locally restorable backup paths that provide protection against the failure of primary path elements. The characteristics of the backup paths depend on the fault model and is explained in section V. If the routing algorithm is able to find sufficient bandwidth in the network for the requisite primary and backup paths, the paths are setup (see [13] and [14]), and the LSP request is accepted; otherwise, the LSP request is rejected. The $(k+1)^{th}$ LSP request arrives only after the k^{th} LSP request has either been accepted or rejected. We want to route LSP requests such that we optimize network utilization. A reasonable objective is to minimize the sum of bandwidth reserved for the set of primary and backup paths established to serve the LSP request [8]. We, therefore, route an LSP request such that the least amount of additional bandwidth is reserved.

V. Bandwidth Sharing Model

The assumption that no more than one link or node will fail simultaneously implies that only backup paths protecting the failed element will be activated at a time. This presents an opportunity to share bandwidth between backup paths, and such sharing is the key to efficient routing. There are two types of bandwidth sharing: *intra-demand* and *inter-demand* sharing [1], [6]. Sharing between backup paths protecting a single primary LSP is called intra-demand sharing. Sharing between backup paths protecting different primary LSPs is referred to as inter-demand sharing.

Backup paths must be routed divergently from the element they are protecting i.e., a backup path that protects a particular node or a link cannot traverse that node or link. For the rest of the links, the amount of additional bandwidth that needs to be reserved for a backup path traversing the link is given by the amount of bandwidth that is being protected, minus the amount of backup bandwidth already reserved on the link that can be shared by the backup path being routed. Therefore, modelling link cost for a backup path involves computing the amount of backup bandwidth reserved on the link which can be shared by the backup path. Specifically, we wish to know how much bandwidth may be shared on a link (u, v), by a

next-hop or next-next-hop backup path ρ, that spans a link (i,j). To this end, we define the following notation:

A_{ij} : Set of LSP requests having a primary path that traverses (i,j).
$nhop_{ij}^{k}$: Next-hop path corresponding to the k^{th} LSP request that spans (i,j).
$nnhop_{ij}^{k}$: Next-next-hop path corresponding to the k^{th} LSP request that spans (i,j).
B_{ij} : Set of next-hop and next-next-hop paths that traverse (i,j).
C_{ij} : Total bandwidth capacity of (i,j).
F_{ij} : Bandwidth reserved on (i,j) for primary LSPs.
G_{ij} : Bandwidth reserved on (i,j) for backup LSPs.
R_{ij} : Residual bandwidth on (i,j).
μ_{ij} : Set of next-hop paths that span (i,j); $\mu_{ij} = \bigcup_{k} nhop_{ij}^{k}$.
ω_{ij} : Set of next-next-hop paths span (i,j); $\omega_{ij} = \bigcup_{k} nnhop_{ij}^{k}$.
τ_{ij}^{uv} : Set of next-hop paths that span (i,j) and also traverse (u,v); $\tau_{ij}^{uv} = B_{uv} \cap \mu_{ij}$.
ψ_{ij}^{uv} : Set of next-next-hop paths that span (i,j) and traverse (u,v); $\psi_{ij}^{uv} = B_{uv} \cap \omega_{ij}$.
ϑ_{ij}^{uv} : Set of backup paths traversing (u,v) that are activated simultaneously with ρ if (i,j) fails.
υ_{j}^{uv} : Set of backup paths traversing (u,v) that are activated simultaneously with ρ if node j fails.
γ_{ij}^{uv} : Bandwidth reserved for backup LSPs on (u,v) that can not be shared by ρ.

Mechanisms exist that enable the point of local repair to distinguish between link and node failure [15]. Since ρ spans (i,j), its point of local repair node i may only activate ρ in two situations: (i,j) fails or node j fails. Corresponding to either case, there is a set of backup paths established for previously routed LSP requests that are activated. Backup bandwidth that is consumed by this set of backup paths on a given link (u,v) cannot be shared by ρ. ϑ_{ij}^{uv} is the set of paths activated along with ρ on (u,v) if (i,j) fails, and υ_{j}^{uv} is the set of paths activated along with ρ on (u,v) if node j fails. Since either (i,j) or node j may fail, therefore, the backup bandwidth reserved on (u,v) that is not sharable by ρ, γ_{ij}^{uv} is given by the maximum of γ_a and γ_b, where:

$$\gamma_a = \sum_{nhop_{ij}^{k} \in \vartheta_{ij}^{uv}} b_k + \sum_{nnhop_{ij}^{k} \in \vartheta_{ij}^{uv}} b_k, \quad \text{and}$$

$$\gamma_b = \sum_{nhop_{ij}^{k} \in \upsilon_{j}^{uv}} b_k + \sum_{nnhop_{ij}^{k} \in \upsilon_{j}^{uv}} b_k$$

Thus, the bandwidth available for inter-demand sharing on (u,v) by ρ (a next-hop or next-next-hop backup path that spans a link (i,j)) is $G_{uv} - \gamma_{ij}^{uv}$. The constituents of the sets ϑ_{ij}^{uv} and υ_{j}^{uv} depend on the fault model, as is clarified by the following discussion.

A. Single Link Failure Fault Model

In the single link failure fault model only a next-hop path that spans (x,y) is setup for every link (x,y) along a primary LSP. Therefore, ρ is a next-hop path. As restoration is not provisioned in event of node failure, no backup paths are activated if node j fails. Thus, $\upsilon_{j}^{uv} = \emptyset$. In the event (i,j) fails, all next-hop paths that span (i,j) are activated along with ρ. Thus, $\vartheta_{ij}^{uv} = \tau_{ij}^{uv}$.

B. Single Node Failure Fault Model

In the single node failure fault model only next-next-hop paths are set up. Therefore, ρ is a next-next-hop path. As restoration is not provisioned in event of link failure, neither ρ nor any other backup paths are activated if (i,j) fails. Thus, $\vartheta_{ij}^{uv} = \emptyset$. In the event node j fails, ρ is activated along with all next-next-hop paths protecting node j. Thus, $\upsilon_{j}^{uv} = \bigcup_{x} \psi_{xj}^{uv}$.

C. Single Element Failure Fault Model

In the single element failure fault model, both next-next-hop paths and next-hop paths may be set up. A next-next hop path that spans (x,y) is setup for every link (x,y) along a primary LSP, where y is not the LSP egress node (see fig. 1). A next-hop path that spans (x,y) is setup for a link (x,y) along a primary LSP, if y is the LSP egress node. Since ρ spans (i,j), ρ is a next-hop path if j is the LSP egress node; otherwise ρ is a next-next-hop path. We consider both cases:

Case 1: ρ is a next-hop path. Both next-hop and next-next-hop paths protect against the failure of a link. In the event, (i,j) fails all next-hop and next-next-hop paths that span (i,j) are activated along with ρ. Thus, $\vartheta_{ij}^{uv} = \tau_{ij}^{uv} \cup \psi_{ij}^{uv}$ if ρ is a next-hop path. Recall that the merge point of a next-hop path that spans a link (x,y) in the single element failure fault model is node y. Therefore, such a next-hop path is not activated if node y fails. Correspondingly, ρ will not be activated if node j fails. Thus, $\vartheta_{ij}^{uv} = \emptyset$ if ρ is a next-hop path.

Case 2: ρ is a next-next-hop path. As in Case 1, if (i,j) fails all next-hop and next-next-hop paths that span (i,j) are activated along with ρ. Thus, $\vartheta_{ij}^{uv} = \tau_{ij}^{uv} \cup \psi_{ij}^{uv}$ if ρ is a next-next-hop path. Incase node j fails, ρ is activated along with all next-next-hop paths protecting j. Thus, $\upsilon_{j}^{uv} = \bigcup_{x} \psi_{xj}^{uv}$ if ρ is a next-next-hop path.

VI. Network State Information

Minimizing the total bandwidth reservation needed to route a locally restorable bandwidth guaranteed LSP, necessitates sharing backup bandwidth. The notion of how much bandwidth can be shared on a given link, is a function of the network state information present at the node involved in path computation. A significant cost of distributed constraint based routing is the protocol overhead that is incurred by the need to disseminate network state information through link state protocols. Network state information in the context of routing bandwidth guaranteed paths refers to link usage information. Our objective is to maximize the efficiency gains of constraint based routing while minimizing the protocol overhead.

Consider the k^{th} LSP request with bandwidth demand b_k. We wish to compute θ_{ij}^{uv}, which is the additional bandwidth that needs to be reserved on (u,v), for a potential backup path ε that spans a link (i,j) and traverses a link (u,v). ε will be setup if the primary LSP corresponding to the k^{th} LSP request traverses (i,j). Let ς_{uv}^{k} be the amount of backup bandwidth available for intra-demand sharing on (u,v), corresponding to the k^{th} LSP request. Recall from section II that, for a given LSP request, the ingress node computes the primary as well as all locally restorable backup paths. Therefore, the ingress node knows the precise value of ς_{uv}^{k} $\forall uv$. The extent of inter-demand sharing achievable is a function of the available network state information. We consider four information scenarios: complete information scenario, minimum information scenario, aggregate information scenario, and optimized aggregate information scenario.

A. Complete Information Scenario (CIS)

The values F_{ij}, G_{ij} and R_{ij}, as well as the sets ϑ_{ij}^{uv} and υ_{j}^{uv} are known for every pair of links (i,j) and (u,v) in the network. We can, therefore, compute the precise value of γ_{ij}^{uv} as detailed in section V. Thus, θ_{ij}^{uv} is computed as:

$$\theta_{ij}^{uv} = \begin{cases} 0 & \gamma_{ij}^{uv} + b_k \leq G_{uv} + \varsigma_{uv}^{k}, \\ & (i,j) \neq (u,v) \\ \gamma_{ij}^{uv} + b_k - G_{uv} & \gamma_{ij}^{uv} + b_k > G_{uv} + \varsigma_{uv}^{k}, \\ \quad -\varsigma_{uv}^{k} & \gamma_{ij}^{uv} + b_k - G_{uv} - \varsigma_{uv}^{k} \leq R_{uv}, \\ & (i,j) \neq (u,v) \\ \infty & \text{otherwise} \end{cases} \tag{1}$$

B. Minimum Information Scenario (MIS)

In the minimum information scenario only R_{ij} is known for every link (i,j) in the network. No inter-demand sharing is possible in MIS, since we lack knowledge of the exact characteristics of the reserved bandwidth that are significant from a sharing perspective. In MIS, θ_{ij}^{uv} may be computed as:

$$\theta_{ij}^{uv} = \begin{cases} b_k - \varsigma_{uv}^{k} & b_k - \varsigma_{uv}^{k} \leq R_{uv}, \\ & (i,j) \neq (u,v) \\ \infty & \text{otherwise} \end{cases} \tag{2}$$

C. Aggregate Information Scenario (AIS)

Kodialam et al. have proposed the aggregate information scenario in which the values F_{ij}, G_{ij} and R_{ij} are made known for every link (i,j) in the network [1]. This information can be disseminated using traffic engineering extensions to existing link state routing protocols [16], [17]. The availability of aggregated link usage information allows some intra-demand sharing. Since we do not know the sets ϑ_{ij}^{uv} and υ_{j}^{uv} for a pair of links (i,j) and (u,v), we can not precisely compute γ_{ij}^{uv}. However, we can use the available information to compute a conservative value for γ_{ij}^{uv}.

For example, consider routing the next-hop path ε under the single link failure fault model. Recall from section V that such a backup path is only activated in case (i,j) fails, along with all other next-hop paths that span (i,j). We want to find out the minimum additional bandwidth that needs to be reserved on (u,v), if ε traverses (u,v). To this end, we estimate the bandwidth on (u,v) reserved by backup paths that would be simultaneously activated if (i,j) were to fail. In the worst case all next-hop paths that span (i,j) may traverse (u,v). The amount of backup bandwidth required by these next-hop paths on (u,v) is given by F_{ij}, which is the total amount of primary bandwidth on (i,j) and, therefore, is the maximum bandwidth switched onto next-hop paths if (i,j) fails. Since the total backup bandwidth reserved on (u,v) is G_{uv}, the amount of backup bandwidth reserved by next-hop paths that span (i,j) is the minimum of F_{ij} and G_{uv}. Incase $F_{ij} < G_{uv}$ then (u,v) has at least $G_{uv} - F_{ij}$ amount of backup bandwidth not reserved by backup paths that will be simultaneously activated along with ε. Accordingly, ε has at least $\max(0, G_{uv} - F_{ij})$ bandwidth available on (u,v) for inter-demand sharing, in the single link failure fault model. If M_{ij}^{uv} denotes $\max(0, G_{uv} - F_{ij})$, then θ_{ij}^{uv} is computed as:

$$\theta_{ij}^{uv} = \begin{cases} 0 & b_k \leq M_{ij}^{uv} + \varsigma_{uv}^{k}, \\ & (i,j) \neq (u,v) \\ b_k - M_{ij}^{uv} - \varsigma_{uv}^{k} & b_k > M_{ij}^{uv} + \varsigma_{uv}^{k}, \\ & b_k - M_{ij}^{uv} - \varsigma_{uv}^{k} \leq R_{uv}, \\ & (i,j) \neq (u,v) \\ \infty & \text{otherwise} \end{cases} \tag{3}$$

The single node failure and single element failure fault models may also be dealt with in a similar fashion.

D. Optimized Aggregate Information Scenario (oAIS)

We define the optimized aggregate information scenario, which involves propagation of a similar volume of information as does the aggregate information scenario. However, by propagating a slightly different set of information increased inter-demand backup bandwidth sharing is achieved in oAIS as compared to AIS. As in the case of AIS, the values G_{ij} and R_{ij} are known for every link (i,j) in the network. However, instead of propagating F_{ij}, we choose to propagate H_{ij}, which is given by $\max_{uv}(\sum_{nhop_{ij}^{k} \in \tau_{ij}^{uv}} b_k + \sum_{nnhop_{ij}^{k} \in \psi_{ij}^{uv}} b_k)$. Note that H_{ij} is the maximum backup bandwidth held in reserve on any link by next-hop and next-next-hop paths that span a link (i,j).

Consider the single link failure fault model. Then, H_{ij} gives the maximum amount of backup bandwidth that is simultaneously active on any single link, if (i,j) fails. In case the entire set of backup paths that span (i,j) traverse a single link, $H_{ij} = F_{ij}$; otherwise, $H_{ij} < F_{ij}$. Therefore, H_{ij} gives a tighter upper bound for γ_{ij}^{uv} than is given by F_{ij}. It follows that, in oAIS, the amount of inter-demand sharing achieved by a next-hop path that spans (i,j) is greater than in AIS. That

is, $\max(0, G_{uv} - H_{ij}) \geq \max(0, G_{uv} - F_{ij})$. If N_{ij}^{uv} denotes $\max(0, G_{uv} - H_{ij})$, then θ_{ij}^{uv} may be computed as:

$$\theta_{ij}^{uv} = \begin{cases} 0 & b_k \leq N_{ij}^{uv} + \varsigma_{uv}^{k}, \\ & (i,j) \neq (u,v) \\ b_k - N_{ij}^{uv} - \varsigma_{uv}^{k} & b_k > N_{ij}^{uv} + \varsigma_{uv}^{k}, \\ & b_k - N_{ij}^{uv} - \varsigma_{uv}^{k} \leq R_{uv}, \\ & (i,j) \neq (u,v) \\ \infty & \text{otherwise} \end{cases} \quad (4)$$

The single node failure and single element failure fault models may also be dealt with in a similar fashion and, as for the single link failure fault model, lead to better inter-demand sharing with oAIS as compared to AIS. The details are omitted for the sake of conciseness.

It is obvious that CIS results in maximum inter-demand sharing. Kodialam et al. claim that CIS requires propagation of per path information, and it is not scalable to disseminate such a large volume of information [1]. Norden et al. [7] have introduced a new form of state called *Backup Load Distribution Matrix* (BLDM) that can be used in addition to the values F_{ij}, G_{ij} and R_{ij} to achieve maximum inter-demand sharing. In the context of the single link failure fault model, the BLDM maintained by each node is an $m \times m$ matrix, where m is the number of links in the network. The matrix element BLDM$[(i,j)][(u,v)]$ is equal to γ_{ij}^{uv}. In order to maintain the BLDM, γ_{ij}^{uv} $\forall ij$ has to be computed locally and propagated for every link (u,v) in the network. Thus, the total protocol overhead incurred by the BLDM approach for the single link failure fault model is $O(m^2)$. Similarly, the protocol overhead for the BLDM approach corresponding to the single node failure and single element failure fault models are $O(mn)$ and $O(m^2 + mn)$ respectively, where n is the number of nodes in the network. Furthermore, the BLDM approach for the single node failure and single element failure fault models, incurs additional signaling costs. The details are omitted to conserve space. The other three information scenarios involve propagation of a constant amount of information per link, irrespective of the fault model. Therefore, the protocol overhead for MIS, AIS, and oAIS is $O(m)$. Kodialam et al. [1] have demonstrated how AIS leads to cost effective bandwidth sharing while keeping the protocol overhead low. oAIS outperforms AIS in terms of bandwidth sharing, while incurring an identical protocol overhead.

VII. Simulation Results

In this section, we describe our simulation setup and report results comparing the performance of oAIS with the previously defined information scenarios. Our criterion for routing each LSP request is to minimize the sum of bandwidth reserved for active and backup paths. The problem of joint optimization of a primary path and its associated set of backup paths is NP-hard [6]. We, therefore, use the heuristic routing algorithm proposed in [1] and evaluate its performance for the four information scenarios: MIS, oAIS, AIS, and CIS. We performed simulations for two different networks:

Network 1: Network 1 is a 15 node heterogeneous topology adapted from the network in [1] and is shown in Fig. 2. The light links have a capacity of 120 units in either direction, and the dark links have a capacity of 480 units in each direction.

Network 2: Network 2 is a homogeneous topology adapted from [7]. It represents the Delaunay triangulation for the twenty largest metros in continental US [7]. Each unidirectional link in the network has a capacity of 120 units.

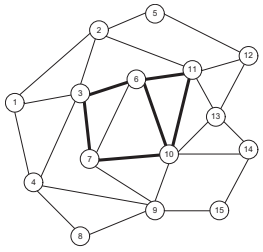


Fig. 2. Network 1

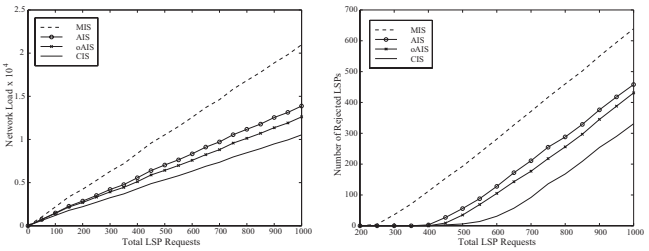
Each node in the network may be an LSP ingress or egress node. Therefore, there are 210 possible ingress-egress pairs in Network 1, and 380 such pairs in Network 2. We conduct simulation experiments similar to those given in [1]:

A. Network Loading Experiments

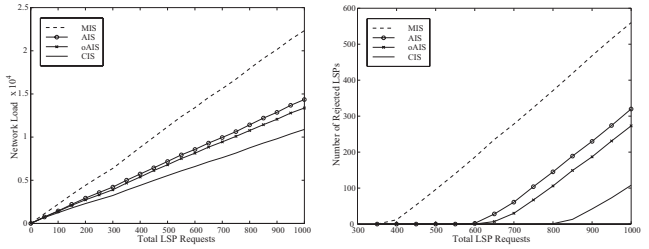
For these experiments, we set the link capacities to infinity. LSP requests arrive one by one, where the LSP ingress and egress nodes are chosen randomly from amongst all ingress-egress pairs. The bandwidth demand for an LSP request is uniformly distributed between 1 and 6 units, and the call holding time for each LSP request is infinite. Since link capacities are set to infinity and both our networks are strongly connected, all LSP requests are accepted. The network loading experiment measures the bandwidth efficiency of the four information scenarios. We conducted 100 experiments with different random seeds, for each of the three fault models. For each experiment, we measured the network load (the sum of bandwidth reserved on all links for primary and backup paths), corresponding to the four information scenarios. Note that better bandwidth sharing will result in lower network load.

B. Rejected Requests Experiments

As in the network loading experiments, LSP requests arrive one by one, where the LSP ingress and egress nodes are chosen randomly from amongst all ingress-egress pairs. The bandwidth demand for an LSP request is uniformly distributed between 1 and 6 units, and the call holding time for each LSP request is infinite. For each LSP request, if it is possible to route the requisite primary and backup paths, the LSP request is accepted and the associated bandwidth reservations are made on the network; otherwise, the LSP request is rejected. We count the number of rejected requests for each of the four information scenarios. Inefficient bandwidth sharing results in network overloading and hence, a greater number of rejected

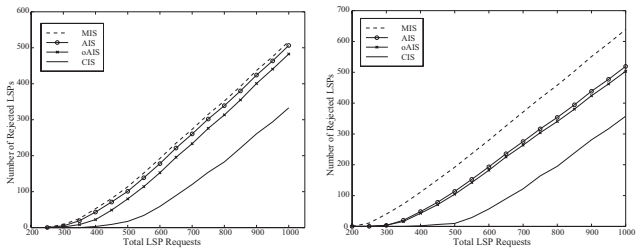


(a) Network 1 (b) Network 1



(c) Network 2 (d) Network 2

Fig. 3. Protection against single link failure



(a) Single Node Failure (b) Single Element Failure

Fig. 4. Rejected Requests Experiments for Network 1

LSP requests. We conducted 100 experiments with different random seeds, for each of the three fault models. Fig. 3 depicts the results for both the experiment sets corresponding to the single link failure fault model. The figure shows that for a given set of LSP requests:

a) the average network load is less for oAIS then AIS, in the network loading experiments, and
b) a greater number of requests are rejected in AIS as compared to oAIS, in the rejected requests experiments.

Thus, oAIS performs better than AIS both in terms of network loading and the number of rejected requests. This is because increased inter-demand sharing is possible in oAIS as compared to AIS. Fig. 3 also includes the results for CIS and MIS to respectively mark the upper and lower bounds on inter-demand sharing performance. CIS clearly yields the best performance, both in terms of bandwidth efficiency and rejected requests. However, it incurs a protocol overhead that is $O(m^2)$. Of the other three information scenarios requiring a protocol overhead that is $O(m)$, oAIS yields the best performance in both experiment sets. Results reported in Fig. 3 are for the single link failure fault model. We obtained similar results for the single node and single element failure fault models. Fig. 4 shows the results for the rejected requests experiment set conducted for Network 1, for the single node and single element failure fault models.

VIII. Concluding Remarks

We considered the problem of distributed online routing of bandwidth guaranteed paths with local restoration. We proposed a unified model to capture the bandwidth sharing characteristic of backup paths and applied the model to describe bandwidth sharing on backup paths under various network information scenarios. We also proposed a new optimized aggregate information scenario (oAIS) that outperforms existing information scenarios with comparable protocol overheads.

We performed two categories of simulation experiments: network loading and rejected requests. Simulations on our networks indicate that for network loading experiments, oAIS on average, results in 10% lower load as compared to AIS, and 40% lower load as compared to MIS, for a set of 1000 randomly selected LSP requests. Furthermore, our simulation study reveals that oAIS on average, rejects 6% fewer demands as compared to AIS, and 33% fewer demands as compared to MIS, for the same set of LSP requests.

References

[1] M. S. Kodialam and T. V. Lakshman, "Dynamic Routing of Locally Restorable Bandwidth Guaranteed Tunnels Using Aggregated Link Usage Information," in *Proc. of Infocom*, April 2001, pp. 376–385.
[2] B. Davie and Y. Rekhter, *MPLS Technology and Applications*. Morgan Kaufmann, 2000, ISBN:1558606564.
[3] Alcatel, "White Paper: Traffic Engineering Solutions for Core Networks," July 2001.
[4] E. Rosen, A. Viswanathan, and R. Callon, "RFC 3031: Multi-Protocol Label Switching (MPLS) Architecture," January 2001.
[5] G. Apostolopoulos, R. Guerin, S. Kamat, and S. K. Tripathi, "Quality of Service Routing: A Performance Perspective," in *Proceedings of ACM SIGCOMM*, 1998, pp. 17–28.
[6] L. Li, M. M. Buddhikot, C. Chekuri, and K. Guo, "Routing Bandwidth Guaranteed Paths with Local Restoration in Label Switched Networks," in *Proceedings of ICNP*, November 2002, pp. 71–79.
[7] S. Norden, M. M. Buddhikot, M. Waldvogel, and S. Suri, "Routing Bandwidth Guaranteed Paths with Restoration in Label Switched Networks," in *Proceedings of ICNP*, November 2001, pp. 71–79.
[8] M. Kodialam and T. V. Lakshman, "Dynamic Routing of Restorable Bandwidth-Guaranteed Tunnels using Aggregated Network Resource Usage Information," *IEEE/ACM Trans. Networking*, vol. 11, no. 3, pp. 399–410, 2003.
[9] M. S. Kodialam and T. V. . Lakshman, "Dynamic Routing of Bandwidth Guaranteed Tunnels with Restoration," in *Proceedings of Infocom*, March 2000, pp. 902–911.
[10] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, "RFC 3272: Overview and Principles of Internet Traffic Engineering," May 2002.
[11] J.-P. Vasseur, A. Charny, F. L. Faucheur, J. Achirica, and J.-L. Leroux, "Internet Draft: MPLS Traffic Engineering Fast Reroute: Bypass Tunnel Path Computation for Bandwidth Protection," February 2003.
[12] P. Pan, G. Swallow, and A. Atlas (Editors), "Internet Draft: Fast Reroute Extensions to RSVP-TE for LSP Tunnels," August 2003.
[13] B. Jamoussi (Editor), "RFC 3212: Constraint-Based LSP Setup using LDP," January 2002.
[14] D. Awduche, L. Berger, D. Gan, T. Li, G. Swallow, and V. Srinivasan, "RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels," December 2001.
[15] A. Charny and J.-P. Vasseur, "Internet Draft: Distinguish a Link from a Node Failure using RSVP Hellos Extensions," October 2002.
[16] D. Katz, K. Kompella, and D. Yeung, "RFC 3630: Traffic Engineering (TE) Extensions to OSPF Version 2," September 2003.
[17] H. Smit and T. Li, "Internet Draft: IS-IS Extensions for Traffic Engineering," August 2003.