Seamless Detection of Link and Node Failures in Label Switched Networks with Local Restoration

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

***Abstract*— We consider a label switched network in which backup paths are provided using a local restoration scheme [1], [2]. A node receives a failure notification if either an adjacent link or a neighboring node fails. Such a node, however, cannot distinguish between link and node failures and must do one of two things. 1) Activate backup paths corresponding to both the link and the node suspected to have failed, or 2) Employ a mechanism to establish whether it is a link or a node that has failed, and activate the requisite backup paths once the failure event has been identified. Incase backup paths are activated without disambiguating between link and node failure, bandwidth sharing estimates for a backup path must be revised to take into account concurrent activation of certain additional backup paths. Consequently, a greater amount of bandwidth has to be provisioned when the failure detection is ambiguous. In the case wherein a node waits to identify the exact failure before activating the requisite backup paths, there is increased switchover latency subsequent to the network failure. Evidently, the increased switchover latency translates into a greater traffic disruption following a network failure.**

We present a simple solution to this problem. Our solution eliminates the need to over-provision backup bandwidth. Moreover, our solution makes possible immediate activation of backup paths without waiting to disambiguate between link and node failure. The key idea is that if an intermediate node along the activated backup paths encounters a resource reservation violation, it can infer the exact type of failure that has transpired. It may then abort the traffic corresponding to the erroneously activated paths while the network traffic that requires restoration can flow through without being disrupted.

I. Introduction

The emergence of mission critical multimedia applications such as Voice over IP, videoconferencing, e-commerce, VPNs etc. has translated into stringent real-time QoS requirements for carrier networks. Packet loss rates and data delivery rates constitute important metrics of Service-Level Agreements (SLAs) offered by today's Internet service providers [3]. Therefore, service providers strive to provide high service availability with an increasing demand for networks with five nines availability (99.999% uptime). However, failure of network elements is not an uncommon occurrence even in well-provisioned networks. A host of conditions including malfunctioning hardware, faulty interfaces, router crashes, software errors, routine maintenance, and accidental fiber cuts can result in outages of network elements [4]. The service availability of a network is, therefore, contingent upon the ability of the network to minimize the disruption of network traffic resulting from failure of network elements.

There have been a number of proposals for mitigating the impact of link and node failure on network performance [5]–[8]. Such proposals range from optimizing the convergence of link state protocols upon network failure to mechanisms that can locally reroute network traffic around the point of failure. A key consideration is restoration latency, which is the time it takes for the traffic traversing the failed network element/s to be redirected onto a viable backup route. Restoration latency should be kept low to minimize disruption to network traffic. Proposals that rely upon progressive global or local convergence of forwarding information, subsequent to network failure, fall short of SONET (less than 100ms latency) restoration standards. They may also cause network instability, routing loops and link congestion. Significant research [1], [9]–[13] has been directed to leverage the capability of label-switched networks to provide fast restoration in event of network failure. Label switched networks provide the ability to pre-provision backup paths such that network traffic is immediately diverted onto backup paths upon detection of network failure. 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. This node, which redirects traffic onto the preset backup path, in case of failure, is called the Point of Local Repair (PLR). Such a model for restoration is called *local restoration*, also referred to as *fast restoration*.

Since resources are set aside for backup paths, backup paths are bandwidth intensive. Network bandwidth may be conserved by allowing backup paths that protect independent network elements to share bandwidth. This follows from the observation that the possibility of multiple simultaneous network failures can be discounted. A recent study on the characterization on failures in an IP backbone revealed that more than 85% of more of the unplanned network failures affect a single link or a single node [4].

A node receives a link failure notification if either an adjacent link or a neighboring node fails [14]. Such a node, however, cannot distinguish between link and node failure and must do one of two things. It may activate backup paths corresponding to both the link and the node suspected to have failed. Alternatively, such a node may employ a mechanism to establish whether it is a link or a node that has failed, and activate the requisite backup paths once the failure event has been identified. Incase backup paths are activated without disambiguating

between link and node failure, bandwidth sharing estimates for a backup path must be revised to take into account concurrent activation of certain additional backup paths. Consequently, a greater amount of bandwidth has to be provisioned to preempt oversubscription of resources resulting from ambiguous failure detection. In the case wherein a node waits to identify the exact failure before activating the requisite backup paths, there is increased switchover latency subsequent to the network failure. Evidently, the increased switchover latency translates into a greater traffic disruption following a network failure.

We present a simple solution to this problem. Our solution eliminates the need to over-provision backup bandwidth. Moreover, our solution makes possible immediate activation of backup paths without waiting to disambiguate between link and node failure. The key idea is that if an intermediate node along the activated backup paths encounters a resource reservation violation, it can infer the exact type of failure that has transpired. It may then abort the traffic corresponding to the erroneously activated path/s while the network traffic that requires restoration can flow through without being disrupted.

The rest of this paper is organized as follows: Section II details the relevant background associated with MPLS local restoration. Section III describes the notion of backup activation sets and backup bandwidth sharing. Section IV describes a model to capture the bandwidth sharing characteristic of backup paths; this section also delineates the bandwidths for various backup paths used in our model. Section V highlights various inequalities for backup bandwidths. In section VI, we demonstrate how intermediate nodes along backup paths can be configured to infer the precise type of failure, thereby eliminating the failure identification and suboptimal sharing overheads. Our conclusions are presented in Section VII.

II. MPLS Local Restoration

A. MPLS and explicit constraint-based routing

The destination based forwarding paradigm employed in plain IP routing does not support routing network traffic along explicit routes determined through constraint based routing [9]. 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 [15]. Ingress routers of a MPLS network classify packets into forwarding equivalence classes and encapsulate them with labels before forwarding them along pre-computed paths [16]. 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 [10].

B. Restoration Routing

A key QoS objective is restoration routing. Restoration routing involves provisioning bandwidth guaranteed and fault-persistent LSP setup such that the guaranteed bandwidth remains provisioned even if links or network nodes fail. This means backup paths are allocated during the initial routing of the LSP such that upon detection of failure, traffic is promptly switched onto preset backup paths. Backup paths are spatially disjoint from the failed element/s. It is obvious that switching from the primary path to a backup path in the event of failure must occur at a node that is upstream from 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.

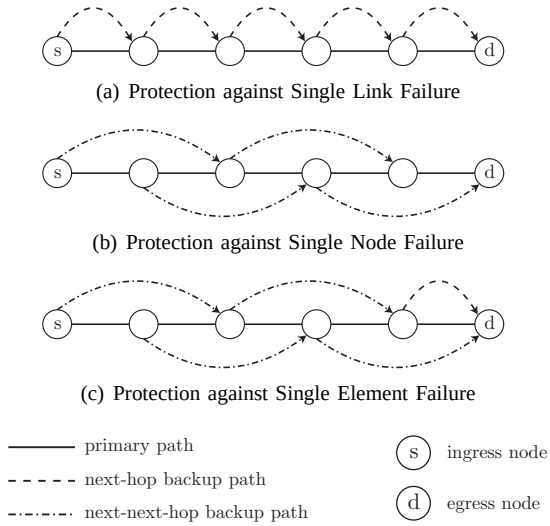
C. Restoration Levels

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 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 onto preset backup paths in the event of network failure and, therefore, results in restoration latencies comparable to those in SONET rings.

D. Single Element Protection

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, network measurements reveal that chances of multiple simultaneous failures are extremely low. A recent study on the characterization on failures in an IP backbone revealed that more than 85% of the unplanned network failures affect a single link or a single node . 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 [17]. The probability of multiple failures in the window of time it takes to setup reoptimized paths is negligible. A prudent restoration objective is to provide protection against failure of a single link or a single node.

Protection against single element failure (single link failure or single node failure) may be provided by virtue of two types of backup paths: *next-hop* paths and *next-next-hop* paths.



(a) Protection against Single Link Failure

(b) Protection against Single Node Failure

(c) Protection against Single Element Failure

Fig. 1. Fault Models

Definition 1: *A next-hop path that spans a link* (i,j)[1] *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.*

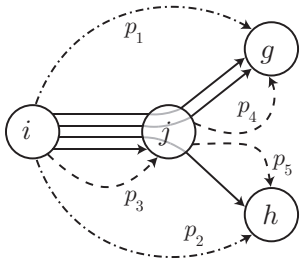
Definition 2: *A next-next-hop path that spans a link* (i,j) *is a backup path which*

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 and the merge point depends upon whether the restoration mode is one-to-one or many-to-one [2]. 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.

As obvious from Fig. 1(a), establishing next-hop paths spanning every link along the primary path provides restoration incase a single link fails. Fig. 1(b) shows that setting up next-next-hop paths spanning all except the last link along the primary path provides restoration incase a single node fails. Since our objective is to provide restoration if either a link fails or a node fails, we can merge the two sets of backup paths represented by Fig. 1(a) and Fig. 1(b). However, note that the configuration in Fig. 1(b) may also be used to protect against the failure of all except the last link. In order to provision restoration in event of single element failure, we only need to setup an additional next-hop backup path spanning the last link as depicted in Fig. 1(c).

[1]A bidirectional link between two nodes constitutes a single network element. However, traffic traverses a link in a specific direction. We, therefore, use $\{i,j\}$ to represent the bidirectional link between $node\ i$ and $node\ j$, and use the ordered pair (i,j) when direction is significant. Thus, (i,j) refers to the directed stem of $\{i,j\}$ from $node\ i$ to $node\ j$. Note that failure of the facility $\{i,j\}$ implies failure of (i,j) and (j,i).



Backup Paths*	Protects one or more LSPs Traversing
p_1	i-j-g
p_2	i-j-h
p_3	i-j
p_4	j-g
p_5	j-h

*Each of these represents a set of one or more backup paths not necessarily routed through same nodes

Fig. 2. Backup paths used in local restoration

III. Backup Activation Sets and Bandwidth Sharing

Since multiple simultaneous failures is an unlikely event, our restoration objective is to provide restoration in event of single link or single node failure. As a corollary of our single element failure assumption, the network must be in any one of three mutually exclusive states at any given point in time:

a) Default State (no failure)
b) Link Failure State (a link $\{i,j\}$ has failed)
c) Node Failure State (a $node\ j$ has failed)

Corresponding to each of these states, a specific subset of the backup paths established in the network is required to be active. We formalize this notion by defining *activation sets*. An activation set, corresponding to a given failure condition, is the set of backup paths that should be active under that failure condition. In other words, an activation set corresponding to a network element is the set of backup paths that are required to be active in order to provision restoration for traffic disrupted as a result of failure of that element.

When the network is in the default state, no network failure has transpired and therefore, no backup paths are activated. Thus the activation set of any link or node under default state is empty. The following enumerates the paths included in the activation sets for a link (i,j) and a $node\ j$:

Activation set for link $\{i,j\}$*:*

Recall from section II that a next-hop path protects against failure of a link, and a next-next-hop path protects against failure of both a link and a node. Incase $\{i,j\}$ fails all next-hop and next-next-hop paths protecting $\{i,j\}$ are activated. This activation set for $\{i,j\}$ comprises the following backup paths (see Fig. 2):

- all next-hop paths that span (i,j)
- all next-hop paths that span (j,i)
- all next-next-hop paths that span (i,j)
- all next-next-hop paths that span (j,i)

Activation set for $node\ j$:

Recall from section II, only next-next-hop paths protect against failure of a node. Thus, the activation set for $node\ j$ comprises all the next-next-hop paths that span $(x, j)\ \forall x \mid (x, j)$ exists. Fig. 2 shows the set of backup paths included in the activation set of $node\ j$.

Network bandwidth is a scarce resource. Efficient utilization of bandwidth necessitates sharing bandwidth reservations between paths that do not simultaneously require the reservation. Since only a single link or a single node may fail at a time, two backup paths are not simultaneously required unless they are in the same activation set. Therefore, under ideal conditions two backup paths that do not belong to the same activation set can share bandwidth with each other. Backup paths belonging to the same activation set must make bandwidth reservations that are exclusive of each other. Consequently, a next-hop path that spans (i, j) is activated if (i, j) fails, and therefore, cannot share bandwidth with other backup paths belonging to the activation set of (i, j). Similarly, a next-next- hop path that spans (i, j) is activated if either $\{i, j\}$ or $node\ j$ fails, and hence cannot share bandwidth with backup paths belonging to the activation sets of either (i, j) or $node\ j$.

IV. Model for Bandwidth Sharing and Backup Bandwidth Computation

A node receives a link failure notification if either an adjacent link or a neighboring node fails. Such a node, therefore, cannot distinguish between link and node failure. Incase we do not wish to incur the latency overhead associated with disambiguating between link and node failure, the node has to activate all backup paths that are members of the activation set of either the link or the node suspected to have failed. This violates the ideal condition that only backup paths of a single activation set may be active at a time. This has implications on the extent of bandwidth sharing that may be achieved between backup paths. We present a comprehensive model that characterizes bandwidth sharing and backup path activation. To this end we define the following notation.

- B_{ij} : Set of next-hop and next-next-hop paths that traverse (i, j).
- $nhop_{ij}^{k}$: The k^{th} next-hop path that spans (i, j).
- $nnhop_{ij}^{k}$: The k^{th} next-next-hop path that spans (i, j).
- G_{ij} : Total bandwidth reserved on (i, j) for backup LSPs.
- 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 that span (i, j); $\omega_{ij} = \bigcup_{k} nnhop_{ij}^{k}$.
- τ_{ij}^{uv} : Set of next-hop paths that span (i, j) and 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); $\psi_{ij}^{uv} = B_{uv} \cap \omega_{ij}$.
- ι_{ij} : Set of backup paths activated if $\{i, j\}$ fails when PLRs can distinguish between link and node failure.
- ϱ_j : Set of backup paths activated if $node\ j$ fails when PLRs can distinguish between link and node failure.
- $\overline{\iota}_{ij}$: Set of backup paths activated if $\{i, j\}$ fails when PLRs cannot distinguish between link and node failure.
- $\overline{\varrho}_j$: Set of backup paths activated if $node\ j$ fails when PLRs cannot distinguish between link and node failure.
- ϑ_{ij}^{uv} : Set of backup paths traversing (u, v) that are activated if $\{i, j\}$ fails when PLRs can distinguish between link and node failure. $\vartheta_{ij}^{uv} = \iota_{ij} \cap B_{uv}$
- υ_{j}^{uv} : Set of backup paths traversing (u, v) that are activated if $node\ j$ fails when PLRs can distinguish between link and node failure. $\upsilon_{j}^{uv} = \varrho_j \cap B_{uv}$
- $\overline{\vartheta}_{ij}^{uv}$: Set of backup paths traversing (u, v) that are activated if $\{i, j\}$ fails when PLRs cannot distinguish between link and node failure. $\overline{\vartheta}_{ij}^{uv} = \overline{\iota}_{ij} \cap B_{uv}$
- $\overline{\upsilon}_{j}^{uv}$: Set of backup paths traversing (u, v) that are activated if $node\ j$ fails when PLRs cannot distinguish between link and node failure. $\overline{\upsilon}_{j}^{uv} = \overline{\varrho}_j \cap B_{uv}$

Furthermore, let $b_{\wp}$ be the bandwidth of a next hop or next-next-hop backup path $\wp$. The following defines the cumulative bandwidth of backup paths for various sets defined above.

- L_{ij} : Cumulative bandwidth of backup paths activated if $\{i, j\}$ fails when PLRs can distinguish between link and node failure. $L_{ij} = \sum_{\wp \in \iota_{ij}} b_{\wp}$
- N_j : Cumulative bandwidth of backup paths activated if $node\ j$ fails when PLRs can distinguish between link and node failure. $N_j = \sum_{\wp \in \varrho_j} b_{\wp}$
- $\overline{L}_{ij}$: Cumulative bandwidth of backup paths activated if $\{i, j\}$ fails when PLRs cannot distinguish between link and node failure. $\overline{L}_{ij} = \sum_{\wp \in \overline{\iota}_{ij}} b_{\wp}$
- $\overline{N}_j$: Cumulative bandwidth of backup paths activated if $node\ j$ fails when PLRs cannot distinguish between link and node failure. $\overline{N}_j = \sum_{\wp \in \overline{\varrho}_j} b_{\wp}$
- L_{ij}^{uv} : Cumulative bandwidth of backup paths traversing (u, v) that are activated if $\{i, j\}$ fails when PLRs can distinguish between link and node failure. $L_{ij}^{uv} = \sum_{\wp \in \vartheta_{ij}^{uv}} b_{\wp}$

N_j^{uv} : Cumulative bandwidth of backup paths traversing (u, v) that are activated if $node\ j$ fails when PLRs can distinguish between link and node failure. $N_j^{uv} = \sum_{\wp \in \upsilon_j^{uv}} b_\wp$

$\overline{L}_{ij}^{uv}$: Cumulative bandwidth of backup paths traversing (u, v) that are activated if $\{i, j\}$ fails when PLRs can distinguish between link and node failure. $\overline{L}_{ij}^{uv} = \sum_{\wp \in \overline{\vartheta}_{ij}^{uv}} b_\wp$

$\overline{N}_j^{uv}$: Cumulative bandwidth of backup paths traversing (u, v) that are activated if $node\ j$ fails when PLRs can distinguish between link and node failure. $\overline{N}_j^{uv} = \sum_{\wp \in \overline{\upsilon}_j^{uv}} b_\wp$

When a link or node fails, the node/s adjacent to the failed element immediately learn of the failure. A node receiving a failure notification activates backup paths that originate from it (the node is the PLR for the backup path), and provides restoration for the failed element. The backup paths activated depend on whether or not the node can distinguish between link and node failure. Table I details the backup paths activated in either case. The following inferences can be made from Table I:

$$\iota_{ij} = \mu_{ij} \cup \omega_{ij} \cup \mu_{ji} \cup \omega_{ji}$$
$$\varrho_j = \bigcup_x \omega_{xj}$$
$$\overline{\iota}_{ij} = \mu_{ij} \cup \omega_{ij} \cup \mu_{ji} \cup \omega_{ji}$$
$$\overline{\varrho}_j = (\bigcup_x \omega_{xj}) \cup (\bigcup_x \mu_{xj})$$
$$\text{Since} \quad \iota_{ij} = \overline{\iota}_{ij} \quad \text{therefore} \quad \vartheta_{ij}^{uv} = \overline{\vartheta}_{ij}^{uv}$$
$$\text{Also} \quad \varrho_j \subseteq \overline{\varrho}_j \quad \text{therefore} \quad \upsilon_{ij}^{uv} \subseteq \overline{\upsilon}_{ij}^{uv}$$

Let us define the variable I_{uv} to denote the actual backup bandwidth required to flow through a link (u, v). A resource reservation violation will occur if the network is in a state such that $I_{uv} > G_{uv}$. We now consider the implications of the ability to distinguish between link and node failure, on G_{uv} of any link (u, v). To this end we will consider each of the three network states defined in Section III.

V. Backup Oversubscription due to Ambiguous Failure Detection

We now consider the amount of bandwidth that is reserved within the network.

A. PLRs can disambiguate between link and node failure

In the default network state no backup paths are activated. Therefore, $I_{uv} = 0$ and hence the default state does not impose any constraint on G_{uv}.

If nodes receiving failure notifications are able to distinguish between link and node failure, failure of a link (i, j) means that L_{ij}^{uv} units of bandwidth are required to flow through (u, v). This means that $I_{uv} = L_{ij}^{uv}$. Therefore, in order to protect against a resource reservation violation the following invariant must always be true:

$$G_{uv} \geq L_{ij}^{uv} \ \forall \ i, j \tag{1}$$

If nodes receiving failure notifications are able to distinguish between link and node failure, failure of a $node\ j$ means that N_j^{uv} units of bandwidth are required to flow through (u, v). This means that $I_{uv} = N_j^{uv}$. Therefore, in order to protect against a resource reservation violation the following invariant must always be true.

$$G_{uv} \geq N_j^{uv} \ \forall \ j \tag{2}$$

B. PLRs cannot disambiguate between link and node failure

In the default network state no backup paths are activated. Therefore, $I_{uv} = 0$ and hence the default state does not impose any constraint on G_{uv}.

If nodes receiving failure notifications are unable to distinguish between link and node failure, failure of a link (i, j) means that $\overline{L}_{ij}^{uv}$ units of bandwidth are required to flow through (u, v). This means that $I_{uv} = \overline{L}_{ij}^{uv}$. Therefore, in order to protect against a resource reservation violation the following invariant must always be true:

$$G_{uv} \geq \overline{L}_{ij}^{uv} \ \forall \ i, j \tag{3}$$

If nodes receiving failure notifications are able to distinguish between link and node failure, failure of a $node\ (i, j)$ means that $\overline{N}_j^{uv}$ units of bandwidth are required to flow through (u, v). This means that $I_{uv} = \overline{N}_j^{uv}$. Therefore, in order to protect against a resource reservation violation the following invariant must always be true.

$$G_{uv} \geq \overline{N}_j^{uv} \ \forall \ j \tag{4}$$

C. Bandwidth Reservations

Since $\vartheta_{ij}^{uv} = \overline{\vartheta}_{ij}^{uv}$, as evident from Table I, $\overline{L}_{ij}^{uv} = L_{ij}$. Therefore, the inequalities 1 and 3 represent the same constraint. Similarly $\upsilon_j^{uv} \subseteq \overline{\upsilon}_j^{uv}$ implies that $\overline{N}_j \geq N_j$.

It, therefore, follows that the inequality 4 represents a tighter constraint on G_{uv} as compared to that represented by 2. Similarly, if it is possible to distinguish between link and node failure we require that $G_{uv} \geq N_j^{uv} \ \forall \ j$. However, incase it is not possible to distinguish between link and node failure, we require $G_{uv} \geq \overline{N}_j^{uv} \geq N_j^{uv} \ \forall \ j$. This is because the ideal condition stipulated in Section III does not hold if it is not possible to disambiguate between link and node failure. Backup paths not belonging to the same activation set may be simultaneously activated. The inflatory constraint is to compensate for such a simultaneous activation of backup paths belonging to different activation sets.

VI. Failure Detection via Control Plane

We propose a simple control plane mechanism to eliminate backup bandwidth oversubscription in a network wherein a node receiving failure notification cannot disambiguate between

When $\{i,j\}$ fails					**When $node\ j$ fails**				
	Link/Node Failure					Link/Node Failure			
	distinguishable		not distinguishable			distinguishable		not distinguishable	
Node notified of failure	Elements suspected to have failed	Backup paths activated by node to protect element	Elements suspected to have failed	Backup paths activated by node to protect element	Node notified of failure	Elements suspected to have failed	Backup paths activated by node to protect element	Elements suspected to have failed	Backup paths activated by node to protect element
$node\ i$	$\{i,j\}$	$nhop_{ij}^k\ \forall k$ $nnhop_{ij}^k\ \forall k$	$\{i,j\}$	$nhop_{ij}^k\ \forall k$ $nnhop_{ij}^k\ \forall k$	$node\ i$	$node\ j$	$nnhop_{ij}^k\ \forall k$	$\{i,j\}$	$nhop_{ij}^k\ \forall k$ $nnhop_{ij}^k\ \forall k$
			$node\ j$	$nnhop_{ij}^k\ \forall k$				$node\ j$	$nnhop_{ij}^k\ \forall k$
$node\ j$	$\{i,j\}$	$nhop_{ji}^k\ \forall k$ $nnhop_{ji}^k\ \forall k$	$\{i,j\}$	$nhop_{ji}^k\ \forall k$ $nnhop_{ji}^k\ \forall k$	$node\ g$	$node\ j$	$nnhop_{gj}^k\ \forall k$	$\{g,j\}$	$nhop_{ij}^k\ \forall k$ $nnhop_{gj}^k\ \forall k$
			$node\ i$	$nnhop_{ji}^k\ \forall k$				$node\ j$	$nnhop_{gj}^k\ \forall k$
					$node\ h$	$node\ j$	$nnhop_{hj}^k\ \forall k$	$\{h,j\}$	$nhop_{ij}^k\ \forall k$ $nnhop_{hj}^k\ \forall k$
								$node\ j$	$nnhop_{hj}^k\ \forall k$

TABLE I

FAILURE MODES FOR THE NETWORK OF FIGURE 2.

link and node failure. We do not compensate for fallacious activation of backup paths due to inability to distinguish between link and node failure. That is for any link (u,v) we set:

a) $G_{uv} \geq L_{ij}^{uv}\ \forall\ i,j$
b) $G_{uv} \geq N_j^{uv}\ \forall\ j$

We simply configure the network nodes to police traffic such that if more than the designated amount of backup traffic arrives for a link (u,v), $node\ u$ aborts all next-hop paths competing for the reserved bandwidth. We claim that the disrupted traffic is restored and backup resource reservations are not violated. Specifically, we will show that $I_{uv} \leq G_{uv}$ for any network state and that all disrupted traffic is restored in case a link fails or a node fails.

A. Default State

Since there are no failures in this state, we have $I_{uv} = 0$. Therefore, $I_{uv} \leq G_{uv}$ which implies that the bandwidth reservations are not violated.

B. Link Failure State

If a link (i,j) fails, $\overline{L}_{ij}^{uv}$ units of bandwidth are required to flow through (u,v). Hence, $I_{uv} = \overline{L}_{ij}^{uv}$. Since $\overline{L}_{ij}^{uv} = L_{ij}^{uv}$ (as shown in Section V) and $L_{ij}^{uv} \leq G_{uv}$ therefore, $I_{uv} \leq G_{uv}$.

C. Node Failure State

When a $node\ j$ fails, normally $\overline{N}_j^{uv}$ units of bandwidth are required to flow through (u,v), i.e. $I_{uv} = \overline{N}_j^{uv}$. If the backup bandwidth reservations on (u,v) are such that $\overline{N}_j^{uv} \leq G_{uv}$, then it is trival to deduce that $I_{uv} \leq G_{uv}$. However, since we have set $G_{uv} \geq N_j^{uv}$, it is possible that $\overline{N}_j^{uv} \geq G_{uv}$. By employing our control mechanism, however, we ensure that I_{uv} remains smaller that G_{uv}. The proof is as follows:

Let η_u be the set of all next-hop paths that are incident on $node\ u$ for subsequent flow to $node\ v$, i.e., η_u denotes the set of all next-hop path traversing through (u,v), i.e., $\eta_u = \bigcup_{ij} \tau_{ij}^{uv}$. Note that η_u is the set of next-hop paths that will traverse (u,v) if none of the paths are aborted. Using our control plane mechanism, when $node\ u$ observes an oversubscription, it aborts all next-hop paths competing for the reserved bandwidth. When a $node\ j$ fails, the bandwidth corresponding to paths aborted by $node\ u$ is given by:

$$b_{\text{abort}} = \sum_{x} \sum_{\wp \in (\overline{v}_j^{xu} \cap \eta_u)} b_\wp$$

$$\text{Thus,} \quad I_{uv} = \overline{N}_j^{uv} - b_{\text{abort}}$$

$$\text{but} \quad \overline{N}_j^{uv} = \sum_{\wp \in \overline{v}_j^{uv}} b_\wp$$

$$\text{Therefore,} \quad I_{uv} = \sum_{\wp \in (\overline{\upsilon}_j^{uv} - \eta_u)} b_\wp$$

$$\text{but} \quad \overline{\upsilon}_j^{uv} - \eta_u = (\overline{\varrho}_i \cap B_{uv}) - \bigcup_{ij} \tau_{ij}^{uv}$$

$$= (\overline{\varrho}_i \cap B_{uv}) - \bigcup_{ij} (\mu_{ij} \cap B_{uv})$$

$$= (\overline{\varrho}_i \cap B_{uv}) - \{(\bigcup_{ij} \mu_{ij}) \cap B_{uv}\}$$

$$= (\overline{\varrho}_i - \bigcup_{ij} \mu_{ij}) \cap B_{uv}$$

$$= (\{(\bigcup_{x} \omega_{xi}) \cup (\bigcup_{x} \mu_{xj})\} - \bigcup_{ij} \mu_{ij}) \cap B_{uv}$$

$$\text{since} \quad \bigcup_{x} \mu_{xj} \subseteq \bigcup_{ij} \mu_{ij}$$

$$\overline{\upsilon}_j^{uv} - \eta_u = (\bigcup_{x} \omega_{xi}) \cap B_{uv}$$

$$= \upsilon_j^{uv}$$

$$\text{Therefore,} \quad I_{uv} = \sum_{\wp \in \upsilon_j^{uv}} b_\wp = N_j^{uv}$$

Since we have set $G_{uv} \geq N_j^{uv} \ \forall \ j$ we get $G_{uv} \geq I_{uv}$, which completes the proof. We further deduce that oversubscription for backup paths can be observed by $node\ u$ only in the case of node failures.

VII. Conclusion

We observed that, a node receives a link failure notification if either an adjacent link or a neighboring node fails. Such a node, however, cannot immediately disambiguate between link and node failure. Additional mechanisms employed to distinguish between link and node failure cause increased switchover latency subsequent to the network failure. Evidently, the increased switchover latency translates into a greater traffic disruption following a network failure. Controlling the switchover latency requires simultaneously activating backup paths corresponding to both the link and the node suspected to have failed. This adversely affects bandwidth sharing since backup paths that do not belong to the same activation set may be concurrently active. Consequently, a greater amount of bandwidth has to be provisioned to preempt backup resource reservation violation resulting from ambiguous failure detection.

We presented a simple solution to the problem. Our solution eliminates the need to over-provision backup bandwidth. Moreover, our solution makes possible immediate activation of backup paths without waiting to disambiguate between link and node failure. The key idea is that if an intermediate node along the activated backup paths encounters a resource reservation violation, it can infer the exact type of failure that has transpired. It may then abort the traffic corresponding to the erroneously activated paths. Specifically we show that backup resource reservation violation may only transpire if a node fails. Furthermore, we showed that incase of node failure all erroneously activated backup paths are next-hop paths. Therefore, in event of a backup resource reservation violation on a link, the head end aborts all next-hop paths competing for the resource. The next-next-hop paths may continue to provision the requisite restoration. As a result, no additional bandwidth must be reserved to compensate for erroneously activated backup path, and the network traffic requiring restoration can flow through without being disrupted.

Acknowledgment

This work was supported by a grant from Cisco Systems under the University Research Program.

References

[1] L. Li, M. M. Buddhikot, C. Chekuri, and K. Guo, "Routing Bandwidth Guaranteed Paths with Local Restoration in Label Switched Networks," *IEEE JSAC*, vol. 23, no. 2, pp. 437–449, 2005.

[2] S. Raza, F. Aslam, and Z. A. Uzmi, "Online Routing of Bandwidth Guaranteed Paths with Local Restoration using Optimized Aggregate Usage Information," in *Proceedings of the 2005 IEEE International Conference on Communications (ICC)*, Seoul, Korea, May 2005.

[3] R. Keralapura, C.-N. Chuah, G. Iannaconne, and S. Bhattacharrya, "Service Availability: A New Approach to Characterize IP-Backbone Topologies," in *Proceedings of IEEE IWQoS*, June 2004, pp. 232–241.

[4] A. Markopulu, G. Iannaccone, S. Bhattacharya, C.-N. Chuah, and C. Diot, "Characterization of failures in an IP backbone," in *Proc. IEEE Infocom*, March 2004.

[5] Z. Zhong, S. Nelakuditi, Y. Yu, S. Lee, J. Wang, and C.-N. Chuah, "Failure inferencing based fast rerouting for handling transient link and node failures," *IEEE Global Internet*, March 2005.

[6] C. Alattinoglu and S. Casner, "ISIS routing on the Qwest backbone: A recipe for subsecond ISIS convergence," NANOG meeting, Februry 2002.

[7] P. Narvaez, K.-Y. Siu, and H.-Y. Tzeng, "Local restoration algorithms for link-state routing protocols," in *Proc. ICCCN*, March 1999.

[8] S. Iyer, S. Bhattacharyya, N. Taft, and C. Diot, "An approach to alleviate link overload as observed on an IP backbone," in *Proc. IEEE Infocom*, March 2003.

[9] 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.

[10] 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.

[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] S. Kini, M. Kodialam, S. Sengupta, and C. Villamizar, "Shared Backup Label Switched Path Restoration," May 2001.

[13] J.-P. Vasseur, M. Pickavet, and P. Demeester, *Network Recovery: Restoration and Protection of Optical, SONET-SDH, IP and MPLS.* Morgan Kaufmann., 2004, ISBN:012715051X.

[14] M. Goyal, K. Ramakrishnan, and W. chi Feng, "Achieving faster failure detection in OSPF networks," in *Proceedings of the 2003 IEEE International Conference on Communications (ICC)*, May 2003, pp. 296–300.

[15] M. S. Kodialam and T. V. Lakshman, "Dynamic Routing of Bandwidth Guaranteed Tunnels with Restoration," in *Proceedings of Infocom*, March 2000, pp. 902–911.

[16] 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.

[17] P. Pan, G. Swallow, and A. Atlas (Editors), "Internet Draft: Fast Reroute Extensions to RSVP-TE for LSP Tunnels," August 2003.