Shared Protection for Multicast Sessions in Mesh Networks

Narendra K. Singhal\(^1\), Canhui Ou\(^1\), and Biswanath Mukherjee\(^1\)
\(^1\)Department of Computer Science, University of California, Davis, CA USA. 
\(^1\)SBC, 2600 Camino Ramon, San Ramon, CA USA. 
singhal@cs.ucdavis.edu, Tel: (530) 752-5129, Fax: (530) 752-4767

Abstract: We investigate optimal sharing among backups of several multicast sessions in an optical mesh network. Our cross-sharing approach provides significant cost-savings of backup resources relative to arc-disjoint and self-sharing approaches, especially for a large number of sessions.
©2004 Optical Society of America
OCIS codes: (060.4290) Networks; (060.4510) Optical communications

1 Introduction
Recent advances in networking – particularly high-capacity optical networking employing wavelength-division multiplexing (WDM) technology – have made bandwidth-intensive multicast applications such as HDTV, interactive distance learning, live auctions, distributed games, movie broadcasts from studios, etc. widely popular [1]. However, link failures in a communication network occur often enough to cause service disruption, and lead to significant information loss in the absence of adequate backup mechanisms. Hence, researchers have investigated efficient approaches such as arc-disjoint and self-sharing for protecting multicast trees, called “light-trees” in the literature [2].

Consider a small example network shown in Fig. 1 and two multicast sessions, \(S_1 = \{A, B, D\}\) and \(S_2 = \{B, C, D\}\), requiring protection. In general, a multicast session \(S_i\) is denoted as a set \(\{s_i, d_{i1}, d_{i2}, \ldots, d_{i\kappa}\}\) where \(s_i\) is the source node and \(d_{ik}\) are the destination nodes.

![Fig. 1. Approaches for protecting multicast trees: (a) arc-disjoint; (b) self-sharing; and (c) cross-sharing.](image)

1.1 Arc-Disjoint Trees
The approach of arc-disjoint trees (proposed in [3], [4]) constructs a backup tree which is arc-disjoint to the primary tree. Arcs are unidirectional edges and links are bidirectional with an edge in each direction. A cut on a link disrupts the traffic flow in both directions. In Fig. 1(a), the primary edges of session \(S_1\) – namely, \(A \rightarrow B\) and \(A \rightarrow D\) – are shown in solid lines, and they form a primary tree. Similarly, the backup edges of session \(S_1\) – namely, \(A \rightarrow C, C \rightarrow B,\) and \(C \rightarrow D\) – are shown in dashed lines, and they form a backup tree. The primary and backup trees of session \(S_1\) are arc-disjoint, and they can protect the session from a single fiber cut [4]. The cost of protecting the primary tree for \(S_1\), measured as the sum of the weights on the edges of the backup tree, is 24 units.

1.2 Self-Sharing Trees
In this approach, paths to each destination node (on the primary tree) are protected by discovering a backup path which is link disjoint to the corresponding primary path. A backup path can share not only with other backup paths but also with other edges on the primary tree. This self-sharing approach is shown to be more efficient than arc-disjoint trees in [5]. Figure 1(b) shows session \(S_2\) for which the primary paths (shown in solid lines) to the destination nodes, viz. \(B \rightarrow C\) to destination node \(C\) and \(B \rightarrow A \rightarrow D\) to destination node \(D\), form a primary tree. The backup path to the destination \(C\) is \(B \rightarrow A \rightarrow C\) and the backup path to destination \(D\) is \(B \rightarrow C \rightarrow D\). Two backup edges (shown in dotted lines) – \(B \rightarrow A\) and \(B \rightarrow C\) – are sharing wavelength channels with the primary of the same session. This is possible because, in a multicast session, all primary and backup paths carry replicas of the same information to the destination nodes. Backup edges which share a channel with the primary of the same tree are called self-sharing edges. (Corresponding primary and backup trees are called self-sharing trees.) The remaining backup edges, which do not carry traffic, but are reserved to be used when failure occurs, are idle-backup edges, e.g., \(A \rightarrow C\) and \(C \rightarrow D\) shown in dashed lines in Fig. 1(b). Observe that the backup paths are link-disjoint to their corresponding primary paths and, hence, they can protect a multicast session from a fiber cut. The cost of additional backup resources for protecting \(S_2\) is 15 units.
1.3 Cross-Sharing Trees

The idea of link-vector has been widely applied in various studies (e.g., the conflict vector in [6]) to identify sharing potential among backup paths, especially in a wavelength-convertible network. We propose a modified link-vector model for determining the number of backups to be provisioned on an edge while sharing idle-backup edges of several trees. A link vector $\nu_e$ for edge $e$ can be represented as an integer set, \{\nu^e \subseteq \nu \mid \nu \in \mathbb{N}, 0 \leq \nu^e \leq \lambda(e)\},$ where $\nu^e$ is the number of backup channels needed on edge $e$ when link $\nu$ fails and $\lambda(e)$ is the number of wavelengths on edge $e$.

<table>
<thead>
<tr>
<th>$\nu^e$</th>
<th>A $\rightarrow$ C</th>
<th>B $\rightarrow$ C</th>
<th>C $\rightarrow$ D</th>
<th>D $\rightarrow$ A</th>
<th>A $\rightarrow$ C</th>
</tr>
</thead>
<tbody>
<tr>
<td>B $\rightarrow$ C</td>
<td>\begin{align*} 1 &amp; \rightarrow 0 \ 0 &amp; \rightarrow 0 \ 0 &amp; \rightarrow 0 \ 0 &amp; \rightarrow 1 \end{align*}</td>
<td>\begin{align*} 0 &amp; \rightarrow 0 \ 0 &amp; \rightarrow 0 \ 0 &amp; \rightarrow 0 \ 1 &amp; \rightarrow 1 \end{align*}</td>
<td>\begin{align*} 0 &amp; \rightarrow 0 \ 0 &amp; \rightarrow 0 \ 0 &amp; \rightarrow 0 \ 1 &amp; \rightarrow 0 \end{align*}</td>
<td>\begin{align*} 1 &amp; \rightarrow 0 \ 0 &amp; \rightarrow 0 \ 0 &amp; \rightarrow 0 \ 0 &amp; \rightarrow 1 \end{align*}</td>
<td>\begin{align*} 0 &amp; \rightarrow 0 \ 0 &amp; \rightarrow 0 \ 0 &amp; \rightarrow 0 \ 2 &amp; \rightarrow 0 \end{align*}</td>
</tr>
</tbody>
</table>

Table 1. Link-vector model for the two trees $S_1$ and $S_2$, respectively.

Table 1 shows the link-vector for idle-backup edges of the two primary trees, $S_1$ and $S_2$, respectively. Only those links which carry primary traffic make idle-backup edges active, e.g., links $A \leftrightarrow B$ and $D \leftrightarrow A$ for session $S_1$. Link vectors are computed for only the idle-backup edges. When a link fails, the idle-backup edges which feed the affected destination nodes become active. Observe that only one idle-backup edge needs to be reserved on $A \rightarrow C$, thus cross-sharing this edge between sessions $S_1$ and $S_2$. As a result, the total cost of backup edges for protecting both sessions $S_1$ and $S_2$ is 31 units, a savings of 8 units which is worth exploiting.

2 Problem Description

Given:
1. A topology $G = (V, E)$ consisting of a weighted directed graph, where $V$ is a set of $N$ network nodes, and $E$ is the set of links inter-connecting the nodes. Each link is assigned a weight ($c_{mn}$ between nodes $m$ and $n$) to represent the cost of moving traffic from one node to the other.
2. Nodes are equipped with multicast-capable opaque crossconnects which convert the signal from optical to electrical to optical (OEO), and hence allow full wavelength conversion.
3. A batch of $b$ multicast sessions.

**Determine:** Optimal cost of backup resources used to protect the $b$ multicast sessions from any link failure.

2.1 Cross-Sharing Algorithm

1. For each multicast session $i \in b$, compute a minimum-cost primary tree using an Integer Linear Program (ILP). $P^i_{mn} = 1$, if edge $m \rightarrow n$ ($m$ and $n$ are indices for nodes) is on primary tree; otherwise $P^i_{mn} = 0$. $P^{i^t}_{mn} = 1$, if edge $m \rightarrow n$ carries traffic to destination $d_i$; otherwise $P^{i^t}_{mn} = 0$.
2. For each destination $d_i$, determine a backup path $B^{i^t}_{mn}$ link-disjoint to $P^{i^t}_{mn}$.
3. For each cut on link $\overline{\mathbb{m}} \leftrightarrow \overline{\mathbb{n}}$ ($\overline{\mathbb{m}}$ and $\overline{\mathbb{n}}$ are node indices between which a failure occurs), determine the unreachable destination nodes $d_i$, and assign $K^{i^t}_{mn} = 1$ if an idle-backup edge of tree $i$ on $m \rightarrow n$ becomes active for any destination node $d_i$. $K^{i^t}_{mn}$ is the link vector of tree $i$ for edge $m \rightarrow n$ (see Table 1).
4. For every edge $m \rightarrow n$, determine the maximum number of idle-backup edges ($\nu_{mn}$) required for a failure on any link $m \rightarrow n$ in the network.
5. Optimize the cost of idle-backup edges $\sum_{mn} c_{mn} \times \nu_{mn}$ using mathematical formulation as shown below and solve the ILP using an appropriate solver, e.g., CPLEX.

2.2 Integer Linear Program

Minimize:

$$\sum_{mn} c_{mn} \times \nu_{mn}$$

Constraints:

$$\forall i, d_i, m, s_i : \sum_n B^i_{mn} = 1$$

$$\forall i, d_i, m, d_i : \sum_n B^i_{mn} = 1$$

$$\forall i, d_i, m, d_i : \sum_n (B^i_{mn} - B^i_{mn}) = 0$$

$$\forall i, d_i, m, n : B^i_{mn} + P^i_{mn} \leq 1$$

$$\forall i, m, n, \overline{\mathbb{m}}, \overline{\mathbb{n}} : \sum_{d_i} (P^i_{mn} \times (B^i_{mn} - P^i_{mn})) \leq N \times K^{i^t}_{mn}$$

$$\forall m, n, \overline{\mathbb{m}}, \overline{\mathbb{n}} : \sum_{i} (P^i_{mn} \times K^{i^t}_{mn}) \leq \nu_{mn}$$
Explanation of Equations: Equations (1), (2), and (3) are flow equations to construct a backup path from root (the source of a multicast session or of a light-tree) to each destination node. Equation (4) ensures that this backup path is link disjoint to its primary counterpart. Equation (5) assigns $K_{m,n}$ to 1 whenever a cut on link $m \leftrightarrow n$ activates a backup edge on $m \rightarrow n$ for session $i$. Equation (6) computes the number of backup edges, $\nu_{m,n}$, to be reserved for any link failure $m \leftrightarrow n$ in the network.

3 Illustrative Numerical Results

A batch of $b = 25$ multicast sessions, each with destination set size 15, is randomly generated on a nationwide network shown in Fig. 2. The weights on the links of the network are used as cost metric. ILP formulations for the three approaches (arc-disjoint, self-sharing, and cross-sharing) are solved to reserve backup resources for the batch of sessions. Figure 3 compares the three approaches as the size of the batch increases from 1 to 25. For batch size of one, both self-sharing and cross-sharing trees require equal backup resource. However, as the batch size increases, so does the sharing potential of the backup among the sessions. As a result, cross-sharing performs significantly better than self-sharing for a large number of sessions in the network which has also been captured in the lowermost savings curve shown in Fig. 3. The cost savings of cross-sharing over self-sharing (shown in Fig. 3) for a batch size of 25 sessions is 39%.

4 Conclusion

In this paper, we proposed the use of a link-vector model for protecting several multicast sessions by maximizing sharing among their backup edges. Our cross-sharing approach produces significant savings in the cost of backup resources over existing self-sharing or arc-disjoint approaches. Cross-sharing is well applicable to other contexts, such as in SONET or Gigabit Ethernet (GbE), for lower-speed multicast trees.

References