Protecting Multicast Services in Optical Internet Backbones

Long Long Ahmed E. Kamal
Dept. of Electrical and Computer Eng., Iowa State University, Ames, IA 50011, U.S.A.
E-mail: {longlong, kamal}@iastate.edu

Abstract

Many applications in the future Internet will use the multicasting service mode. Since many of these applications will generate large amounts of traffic, and since users expect a high level of service availability, it is important to provision multicasting sessions in the future Internet while also providing protection for multicast sessions against network component failures. In this paper we address the multicast survivability problem of using minimum resources to provision a multicast session and its protection paths (trees) against any single-link failure. We propose a new, and a resource efficient, protection scheme, namely, Segment-based Protection Tree (SPT). In SPT scheme, a given multicast session is first provisioned as a primary multicast tree, and then each segment on the primary tree is protected by a multicast tree instead of a path, as in most existing approaches. We also analyze the recovery performance of SPT and design a Reconfiguration Calculation Algorithm to compute the average number of reconfigurations upon any link failure. By extending SPT to address dynamic traffic scenarios, we also propose two heuristic algorithms, Cost-based SPT (CB_SPT) and Wavelength-based SPT (WB_SPT). We study the performance of the SPT scheme in different traffic scenarios. The numerical results show that SPT outperforms the best existing approaches, optimal path-pair-based shared disjoint paths (OPP_SDP). SPT uses less than 10% extra resources to provision a survivable multicast session over the optimal solution and up to 4% lower than existing approaches under various traffic scenarios and has an average number of reconfigurations 10-86% less than the best cost efficient approach. Moreover, in dynamic traffic cases, both CB_SPT and WB_SPT achieves overall blocking probability with 20% lower than OPP_SDP in most network scenarios.

Keywords: Optical Internet backbone networks; Multicasting; Routing; Protection.

I Introduction

The future Internet is a high performance communication network, that is capable of supporting large amounts of individual and aggregate traffic, in order to meet the explosive increase in bandwidth demand in the Internet, and to support a greater variety fo network applications. For this reason, the optical fiber is the physical medium of choice in both the access and distribution networks of the future Internet. Wavelength-division multiplexing (WDM) technology is used on optical fibers in order to allow an aggregate traffic on the order of Tbps to be carried on a single fiber, with each wavelength carrying traffic in the tens of Gbps order. Many of the applications that will be supported by the future Internet will employ the multicasting service mode [2]. These include high-definition video distribution, online gaming, e-Science applications, etc. To implement multicasting, a node should have the capability to replicate an incoming packet into multiple copies. In the context of optical networks, there are two ways to implement the multicast function at a node, unicast and multicast. In unicast mode, traffic duplication can only be implemented in the electronic domain, whereas in multicast mode, traffic duplication can be done in the optical domain by using optical splitters [6]. If a multicast session is provisioned as a tree in the optical domain, it is called a "light-tree" which originates at a source node and delivers the same data to a number of destination (leaf) nodes [4].

As the capacity of fibers keeps on increasing, a fiber cut caused by an accident or a failure of a switch port or a node interface may lead to loss of tremendous amounts of data. In the scenario of multicasting service, data loss on one fiber may cause the disruption of delivery to multiple nodes. Since users expect a high level of network and service availability, protection of multicast session against network component failures must be provided. The most common type of failures in optical networks is the link failures, and this is why a number of strategies have been proposed in the literature to provide this type of protection. One approach that was proposed in [3] is to find two light trees, where both of them start from the source and end at the destination nodes, but they are routed in a link disjoint manner. It is clear that this method is not capacity efficient since it is not always possible to find two link disjoint trees in a network. In [5], the authors introduced a number of protection schemes: link-based, segment-based and path-based. In link-based and segment-based approaches, a multicast session is routed first to construct a multicast tree, and then each link or segment on the tree is protected by a path starting at the tail node and finishing at the head node of the link or segment it protects. Alternatively, a path-based protection scheme, named optimal path-pair-based shared disjoint paths (OPP_SDP) algorithm, achieves the best result in terms of network resource consumption in [5] by self-sharing primary and spare capacity [7]. The idea is to find two shortest link disjoint paths for each source and destination pair. Recently, a couple of new technologies were applied to the survivability problem, p-cycle [9] and network coding [10]. These techniques do have some nice features such as the fast recovery of p-cycles or high bandwidth utilization of network coding. However, p-cycle-based schemes are not efficient and flexible to protect dynamic traffic, especially multicast traffic, while network coding introduces extra computational cost as well as O-E-O conversion since network coding can only be performed in the electronic domain in current optical networks, which may introduce an additional expense.

A path-based scheme, called multicast protection through spanning paths (MPSP), proposed in [8], outperforms OPP_SDP under both static and dynamic traffic patterns. It first provisions a primary multicast tree and then establishes a number of paths to protect each path between any pair of leaf nodes on the primary tree, called spanning path. Each path is link disjoint from the spanning path it protects. However, this scheme relies on the assumption that wavelengths reserved in a fiber can be used in two opposite directions by reconfiguring the switches at two end nodes. However, this feature cannot be achieved in practice. Between two connected nodes, there are usually two physical fibers set up and each of them works in one direction. The switches at end nodes use input and output ports to connect incoming and outgoing fibers, respectively [5, 6]. Reserved capacity (wavelength) in a fiber cannot be used in both directions by simply reconfiguring the switches at end nodes due to the fixed switching ports. One way to enable this feature is to change the physical infrastructure by deploying a pair of circulators between two nodes as shown in Fig. 1. The fiber is connected to the circulators instead of switching ports on the switches. The circulators connect to both input and output ports on the nearby switches and can configure the fiber to connect to either input port or output port. Only changing the configuration of both switches and circulators will make the transmission in both directions on the same fiber possible such that one unit of capacity reserved in a directed link can be shared by primary and protection paths in MPSP scheme. Due to the infrastructure of current Internet backbone networks, the lack of support for this functionality and the restrictions this imposes on other modes of communication, we do not take this assumption into consideration in our proposed scheme.

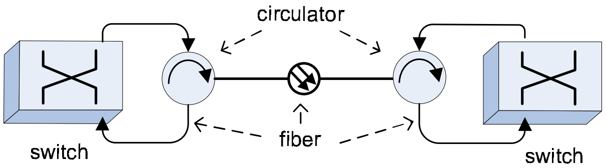


Figure 1: Additional depolyment of circulators enables capacity sharing in opposite directions of a fiber

A tree-based protection scheme, segment-based protection tree (SPT) algorithm, is proposed in this paper to provision a multicast request and protect it against any single link failure. We first provision the multicast session on a light tree and then construct protection multicast trees instead of paths to protect the primary light tree. Each protection tree, similar to primary tree, is rooted at the source and reaches every destination in the session. Each segment on the primary tree is protected by a protection tree. A protection tree can share any link with the primary tree as well as other protection trees. The uniqueness of our schemes is that each protection tree is a complete multicast tree from source to destinations. It does not have to traverse the end nodes of a segment it protects. In this case, multiple segments may share one protection tree, which potentially improves the efficiency of the bandwidth utilization.

The rest of this paper is organized as follows. In Section II, we present the assumptions and statement of the problem addressed. The proposed scheme, SPT, will be introduced in Section III. The method of computing the average number of reconfigurations will be presented in Section IV. We further study the dynamic multicast cases by proposing two heuristic algorithm extended from SPT in Section V. Numerical results will be presented and explained in Section VI. Finally, we conclude this paper in Section VII.

II Preliminaries

In this section, we first describe multicast protection problem addressed in the paper with the corresponding assumptions. We then summarize a number of multicast provisioning methods, some of which will be used in the scheme proposed in the paper.

II.1 Multicast Protection Problem

A typical multicast session is unidirectional whereas the links of a typical WDM mesh network are bidirectional, since each link has two optical fibers transporting signals in two opposite directions with the same capacity. Each directed fiber is also called ”an arc” in [5]. Meanwhile, each arc is assigned a value to indicate the cost of transmitting the data from one end to the other. The cost usually refers to the length of the physical fiber. This cost metric is usually chosen since it has a direct relation to the cost of engineering the fiber spans. In particular, in addition to the cost of the fiber itself, the length of the fiber span is proportional to the number of optical amplifiers, as well as the number of regenerators, which some of the most expensive components in optical networks.

We make the following assumptions and present the formal statement of the multicast protection problem:

1. Given a weighted directed connected graph $G = (V, E)$ in which each directed link[1] $e = (u, v) \in E$ where $u, v \in V$ is assigned a weight (cost) c_e and a capacity with W wavelengths. The graph, G, is at least 2-connected.

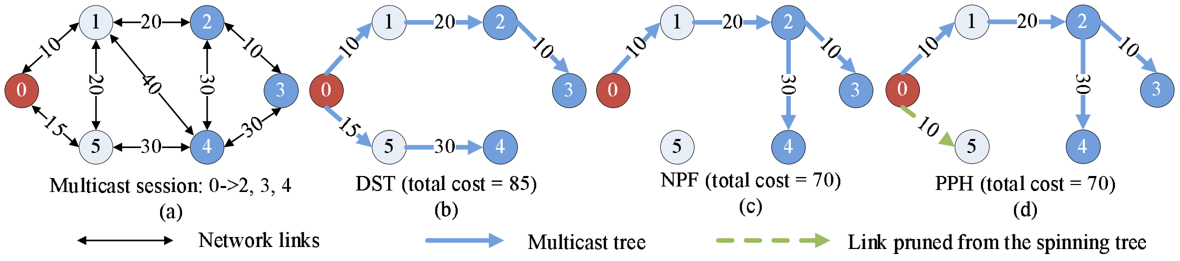
2. Given a directed multicast request d with a source node s and a set of destinations $\{t_1, t_2, ..., t_M\}$ where $s, t_i \in V$ and M is the number of destination nodes. The traffic requirements of the session is equal to one wavelength. d is expressed as $(s, \{t_1, t_2, ..., t_M\})$.

3. A single link failure will cut off the links in both directions such that traffic delivered in both fibers will be lost. Thus, when we claim two link-disjoint paths (trees) in this article, it indicates that two paths (trees) do not travel the links with the same end nodes in any direction.

4. In this article, we assume that each network node is equipped with an optical switch, optical splitters and wavelength converters if necessary.

The multicast protection problem is described as follow:

Given a weighted graph $G = (V, E)$ and a multicast request d, find a provisioning of the multicast session d such that the multicast service is survivable against any single link failure in G using the minimum cost.



Multicast session: 0->2, 3, 4
(a)

DST (total cost = 85)
(b)

NPF (total cost = 70)
(c)

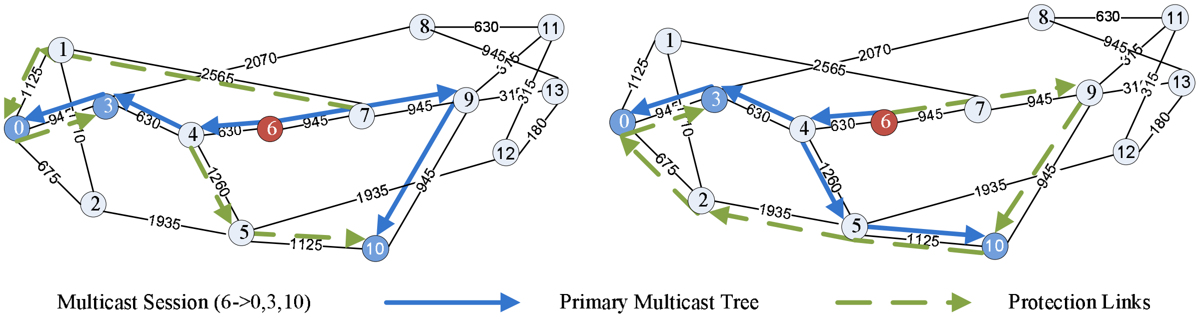
PPH (total cost = 70)
(d)

Figure 2: Demonstration of various multicast provisioning algorithms

[1] Here we use ”link” to represent ”arc” similar to [5] and therefore links (u, v) and (v, u) are two different links but have the same cost and capacity.

II.2 Multicast Provisioning Methods

In order to provision a survivable multicast session with minimum cost, it is essential to study how to provision a multicast request. In optical transport networks, multicast provisioning problem can be referred to as finding a light-tree that delivers data from the source to all the destinations with the minimum cost. Deployment of optical splitters at each network node enables multicast implementation in the optical domain. Thus, this problem turns out to be a classic graph theory problem, "Steiner tree problem", which has been proven *NP-complete* [23]. Hence, the multicast protection problem is also *NP-complete* in the general case and this is why we develop heuristic solution approaches in this paper.



Total cost of final multicast topology is 11115 where the cost of the primary multicast tree is 5040 obtained by using DST
(a)

Total cost of final multicst topology is 11160 where the cost of the primary multicast tree is 4590 obtained by using NPF
(b)

Figure 3: Comparison of the final topologies with the primary multicast trees constructed by using DST and NPF, respectively.

Many approximate algorithms have been proposed in the literature such as Nearest Participant First (NPF) algorithm [24], KMB algorithm [25], pruned Prim′s heuristic [26], referred to as PPH and so on. We actually consider three multicast schemes in the construction of multicast tree: NPF, pruned Prim′s heuristic and simply using Dijkstra's Shortest Path algorithm, namely, DST, to find the shortest path from the source to each destination and combining all the paths to construct a multicast tree.

The heuristic NPF is a greedy-based algorithm with time complexity $O(M|V|^2)$. The procedure is explained as follow:

1. start from the source node;
2. find a destination node that is closest to the current tree;
3. connect the closest destination node to the closest part of the tree;
4. repeat until all the destinations are connected in the tree.

Prim′s algorithm is a well known approach of finding the minimum spanning tree with time complexity $O(|V|^2)$. Based on the minimum spanning tree obtained, PPH trims the unwanted branches such that the resulting multicast tree only reaches the given destinations. The total time complexity is $O(|V|^2 + M|V|)$.

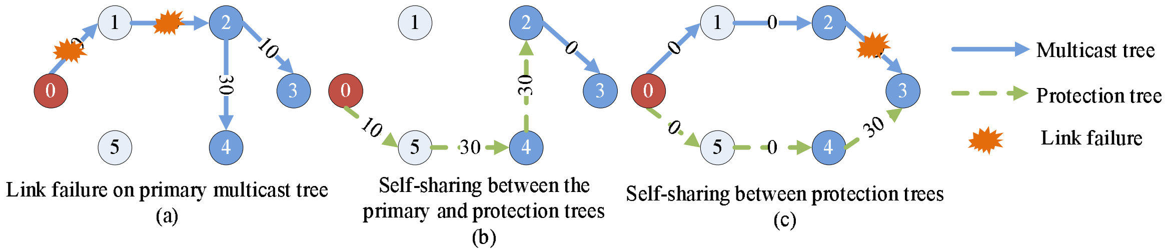
The algorithm DST, with time complexity $O(M|V|^2)$, is straightforward and is actually a special case of NPF by assuming that the source is the only node on the current tree. Thus, a multicast tree produced by DST always has equivalent or higher cost than what NPF produces.

An example shown in Figure 2 illustrates the multicast trees constructed by employing various algorithms described above. Given a multicast session that sourced at node 0 and destined to node $\{2, 3, 4\}$, NPF and PPH construct the same multicast tree and achieve less cost than DST does.

III Tree-based Protection Scheme

In this section, we present the tree-based protection scheme, SPT, to provision a multicast request against any single-link failure using minimum cost.

SPT scheme consists of two phases. The first phase is to construct three primary multicast trees by using the three methods, NPF, PPH and DST, repectively, introduced in the section II. The second phase is to provision a protection structure to protect each primary multicast tree established in the first phase. A final survivable multicast session is established by combining the primary multicast tree and its corresponding protection structure. We choose the final topology with the minimal total cost among three methods. Although the multicast tree obtained by using DST has the same or higher cost than that achieved by NPF. The reason to consider DST in only phase one is that the objective considered in our problem is the total cost of the final survivable multicast session other than the primary multicast tree only. A primary multicast tree with higher cost in phase one may end up with requiring a protection structure with lower cost to protect it.



Link failure on primary multicast tree
(a)

Self-sharing between the primary and protection trees
(b)

Self-sharing between protection trees
(c)

Figure 4: An example of Self-sharing

An example shown in Figure 3 illustrates this property. We have a multicast session sourced at node 6 and destined at node 0, 3 and 10. The primary multicast tree constructed by using DST has cost 5040, which is higher than that achieved by NPF, which is 4590. However, the final survivable topology of this multicast session with the primary tree constructed by DST has lower cost than that with NPF, since it uses the protection link with lower cost. Therefore, the total cost of the combination of primary tree and protection structure can still be lower than the structure by using other multicast schemes in phase one.

In the second phase, we find a protection structure for each primary tree obtained in phase one. Each primary multicast tree is decomposed into a number of segments. Following the definition in [5], a segment is defined as the sequence of links from the source or any branch node (on a tree) to a leaf node or to a downstream branch node. For each segment of the tree, the SPT scheme establishes another multicast tree to protect it, called "protection tree". A protection tree is generated by running both NPF and PPH and selecting the one with the less network cost. We do not consider DST here because DST is a special case of NPF and can never produce better solution than NPF. Each protection tree must not traverse the segment it protects. However, it is not necessary for it to pass two end nodes of any segment it protects either. Any protection tree is a complete multicast tree rooted at the source and destined to all the destinations regardless of which segment it protects.

We apply the self-sharing introduced in [7] to our protection scheme. Self-sharing means that a backup path/tree can share capacity not only with other backup paths/trees but also with other edges on the primary tree. An example of self-sharing is illustrated in Fig. 4. If a link failure occurs on either link $(0, 1)$ or $(1, 2)$ as shown in Fig. 4(a), we need a protection tree to protect the segment $\{(0, 1), (1, 2)\}$, and the tree is shown in Fig. 4(b). Three links used by the protection tree are newly reserved and they are $\{(0, 2), (2, 4), (4, 3)\}$. However, one capacity reserved on link $(2, 3)$ for primary tree can be shared with this protection tree and thus the cost of this link in the protection tree is 0. The self-sharing between two protection trees are shown in Fig. 4(c). To protect against the failure of link $(2, 3)$ in the primary tree, the second protection tree is constructed. However, four links used by this tree have already been reserved by the primary and the previous protection trees, which can be shared. Therefore, only one link should be newly reserved, which is link $(4, 3)$. Accordingly, the cost of rest links are equal to 0, too.

Based on the self-sharing rule, we construct the algorithm, Segment-based Protection Tree (SPT). Before describing the algorithm, we introduced symbols used in the algorithm are explained in Table 1:

Given a multicast session, we call the combination of primary and all the protection trees as final survivable topology. Working traffic will be transmitted through the primary tree under the normal condition. The primary tree is divided into a number of segments and each of them is a basic protection unit. All the links reserved for protection other than the primary tree in the final topology is called "pure protection links".

Table 1: Notations used in the Algorithms

Notation	Meaning
T_m^k	the kth primary multicast tree obtained by heuristic k, where $0 \leq k < 3$ and $0, 1$ and 2 represents heuristic algorithm NPF, PPH and DST, respectively
$T_{p_i}^k$	the ith protection tree for primary multicast tree k
$\mathbb{P}_k$	the union of all the protection trees for kth primary tree, denoted by $\bigcup_i T_{p_i}^k$
$\mathbb{R}_k$	the union of all links used for the multicast session generated by heuristic k, denoted by $T_m^k \bigcup \mathbb{P}_k$
c_e	the cost of link $e \in E$

The algorithm SPT is to find the final topology with a cost that is as low as possible.

The SPT is presented in Algorithm 1. Each segment in the primary tree is denoted by $l \in T_m^k$. If any existing protection tree established earlier does not traverse l and its counterpart in the opposite direction[2], then l is protected by this tree upon any failure of link $e \in l$. If no such protection tree exists, a new protection tree needs to be provisioned. However, the new tree can share any link with all the established trees in $\mathbb{P}_k$ as well as the primary tree T_m^k in the modified graph G' with removal of l. Hence, we set the cost of all links available for sharing as 0. Then, algorithm NPF and PPH are executed to obtain the new protection trees $T_{p_i}^k$ and $T_{p_{i'}}^k$ and the one with the less link cost will be selected and added into the protection tree set $\mathbb{P}_k$ in which the links that do not exist in the final set $\mathbb{R}_k$ will also be added. In the final step, three final sets with three different primary trees are compared and we choose the one with the minimum cost, $\mathbb{R}_{\min}$, as the final survivable topology.

Since the number of links of a tree is less than $|V|$, in the worst case, the number of segments on a primary tree cannot exceed $|V|$. Therefore, the time complexity of the heuristic SPT is $O(M|V|^3)$.

IV Reconfiguration Calculation

Besides the network cost, the recovery time, referred to as the time period from the occurrence of the failure to the restoration of the traffic, is another important criterion to evaluate the performance of a protection approach. The recovery process consists of several stages: failure detection, signaling transmission and switch reconfiguration in order to reroute the traffic. The switch reconfiguration process consumes the most part of recovery time, since each reconfiguration takes 10 - 20 ms [27] depending on the technology used. Therefore, it is essential to figure out the average reconfiguration time upon any link failure in a network.

Based on the SPT approach proposed in Section III, a multicast tree is provisioned first and then each segment on the tree will be protected by a protection tree. Thus, given a failure in a network, if this link happens to be used by the multicast tree, a protection tree will be activated to protect it. Accordingly, some nodes on the protection tree may be required to reconfigure the switches to reroute the traffic. The rule to determine whether a node needs to reconfigure its switch is whether this node receives the incoming traffic from a different node or forwards it to a different output node in the protection tree compared to that in the primary multicast tree.

In order to obtain the average number of reconfigurations upon any link failure that disrupts a given multicast service, we assume that the primary tree T_m consists of L links and upon the failure of link $e \in T_m$, a protection tree T_{p_i} is activated and r_i nodes on T_{p_i} will reconfigure the switch. Therefore, the

[2] In the rest of the paper, when we say a tree does not travel a link or segment, it indicates that the tree does not travel the link or the segment in either direction

```
Algorithm 1: Segment-based Protection Tree Algorithm (SPT)
Input: $G(V, E), d = \{s, t_i\}$ $(1 \le i \le M)$
Output: $\mathbb{R}_{\min}$
for $k = 0; k < 3; k++$ do
    construct $T_m^k$ by running $k$th heuristic;
    foreach segment $l \in T_m^k$ do
        if $\exists T_{p_i}^k \in \mathbb{P}_k$, s.t. $l \notin T_{p_i}^k$ then
            continue;
        end
        else
            remove $e \in l$ from $E$;
            set $c_e = 0, \forall e \in \mathbb{R}_k$;
            run NPF and PPH to obtain protection trees $T_{p_i}^k$ and $T_{p_{i'}}^k$, respectively in $G$;
            select the $T_{p_i}^k$ with less cost and add it to $\mathbb{P}_k$;
            add $e$ to $\mathbb{R}_k$, $\forall e \in T_{p_i}^k$ and $e \notin \mathbb{R}_k$;
            recover $c_e$ where $e \in l$;
        end
    end
    if the cost of $\mathbb{R}_k$ is less than that of $\mathbb{R}_{min}$ then
        $\mathbb{R}_{\min} = \mathbb{R}_k$;
    end
end
```

average number of reconfigurations given any link failure is denoted by:

$$R_{\text{avg}} = \frac{\sum_{e \in T_m} r_i}{L} \text{, where } T_{p_i} \text{ protects } e \tag{1}$$

Based on the previous analysis, we propose Algorithm 2 to compute the average reconfiguration time with the application of SPT approach. Several symbols used in the algorithm are explained in Table 2:

Table 2: Symbols used in the Reconfiguration Calculation

Symbol	Meaning
L	total number of links in the primary tree T_m
X	the set of nodes that consists of $\{s, t_i\}$ and the nodes that have node degree more than 2 in the final survivable topology $\mathbb{R}$
R_{avg}	the average number of reconfigurations given any single link failure

In the Algorithm 2, the set X maintains all the potential nodes that may reconfigure the switch upon a link failure. Any node in the final survivable topology $\mathbb{R}$ has node degree at least two, since $\mathbb{R}$ is 2-connected. Except the source s, every node has at least a parent. If a node has nodal degree 2, the incoming and outgoing links that the traffic passes through will always be fixed and there is no need for reconfiguration. Therefore, we only consider the nodes with node degree at least 3 along with the source and the destinations as the potential nodes. In the algorithm, line 5 checks whether node v needs reconfiguration or not. If yes, line 6 increases the total number of reconfigurations. Therefore, the average number of reconfigurations is obtained by the total number divided by the total number of the links in the primary tree shown in line (11). The time complexity of Algorithm 2 is $O(L|V|^2)$.

Algorithm 2: Reconfiguration Calculation Algorithm of SPT

```
Input: $T_m, \{T_{p_i}\}, X$
Output: $R_{\text{avg}}$
$R_{\text{avg}} = 0$;
for $e \in T_m$ do
    if $\exists T_{p_i}$ protects $e$ then
        for $\forall v \in X$ do
            if $\exists (u, v)$ or $(v, u) \in T_{p_i}$ but $\notin T_m$ then
                $R_{\text{avg}} + +$;
            end
        end
    end
end
$R_{\text{avg}} = R_{\text{avg}}/L$;
```

V Dynamic Traffic Protection

In this section, we extend SPT algorithm to address dynamic traffic provisioning problem. We introduce another sharing method, cross-sharing, introduced by [7]. By applying both self-sharing and cross-sharing to deal with dynamic traffic, we propose two algorithms, Cost-based SPT and Wavelength SPT, by extending SPT algorithm such that not only the resources can be shared within a session but also among different sessions.

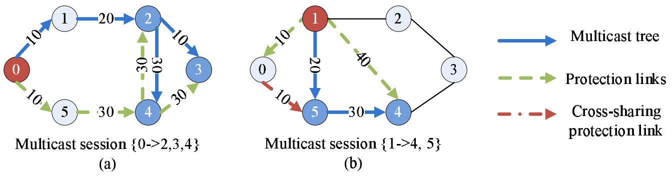

Multicast tree
Protection links
Cross-sharing protection link
Multicast session {0->2,3,4}
(a)
Multicast session {1->4, 5}
(b)
0
1
2
3
4
5
10
20
10
30
30
10
30
30
20
40
10
10
30

Figure 5: An example of Cross-sharing

The basic idea of cross-sharing is to share pure protection links among different multicast sessions. An example is shown in Figure 5 to illustrate cross-sharing. Two sessions arrive one after the other and the session $\{0{-}{>}2, 3, 4\}$ comes ahead of the session $\{1{-}{>}4, 5\}$. The final topology of the first session is shown in Fig. 5(a). As the second session comes, the primary multicast tree is $\{(1, 5), (5, 4)\}$ and the pure protection links are $\{(1, 0), (1, 4), (0, 5)\}$. Since the primary trees of session one and two are link disjoint, so the pure protection link $(0, 5)$ is shared such that only one protection capacity is required to protect both sessions. This feature will be used to provision dynamic traffic.

Based on the rule of cross-sharing, we combine both self-sharing and cross-sharing to provision each dynamic multicast session and protect them against single-link failure. We propose two heuristic algorithms, Cost-based SPT (CB_SPT) and Wavelength-based SPT (WB_SPT). Each heuristic is still based on SPT algorithm proposed in the section III for multicast protection. However, we also introduce cross-sharing such that each pure protection link can be used to protect multiple sessions. The distinction between CB_SPT and WB_SPT is that we use different measurements to choose the final topology for each incoming session. The former chooses the final topology with the minimum overall cost, whereas the latter chooses the final topology with the minimum number of wavelengths used.

The first heuristic, CB_SPT, is shown in as Algorithm 3. We still use three multicast algorithm, DST, NPF and PPH, to obtain three primary multicast trees. For each segment in a multicast tree, we try to find a protection tree to protect it as SPT does. The difference here is that any protection tree can not only use links in the primary tree and other established protection trees, but also pure protection links from other established multicast sessions.

In the algorithm, lines 1 and 2 construct three primary trees. For each segment, line 4-6 check whether this

Algorithm 3: Cost-based SPT Algorithm (CB_SPT)

```
   Input: $G, d = \{s, t_i\}$ $(1 \leq i \leq M)$, $\mathrm{Cost}_{\mathbb{R}_{\min}} = \infty$
   Output: accept or block?
1  for $k = 0; k<3; k{+}{+}$ do
2      construct $T_k^m$ by running $k$th heuristic;
3      for each segment $l \in T_m^k$ do
4          if $\exists T_{p_i}^k \in \mathbb{P}_k$, s.t. $l \notin T_{p_i}^k$ then
5              continue;
6          end
7          else
8              remove $e \in l$ from $E$ and set $c_e = 0, \forall e \in \mathbb{R}_k$;
9              check all the active sessions $d_j$ in $G$;
10             for each protection link $e_j$ on wavelength $\lambda_j$ do
11                 if the segments protected by $e_j$ are link disjoint with $l$ then
12                     set $c_{e_j} = 0$;
13                 end
14             end
15             run NPF and PPH to obtain protection trees $T_{p_i}^k$ and $T_{p_{i'}}^k$;
16             select the $T_{p_i}^k$ with less cost and add it to $\mathbb{P}_k$;
17             add $e$ to $\mathbb{R}_k$, $\forall e \in T_{p_i}^k$ and $e \notin \mathbb{R}_k$;
18             update the segments protected by each $e_l$ where $e_l \in T_{p_i}^k$;
19         end
20     end
21     if $Cost_{\mathbb{R}_k} < Cost_{\mathbb{R}_{min}}$ then
22         $\mathrm{Cost}_{\mathbb{R}_{\min}} = \mathrm{Cost}_{\mathbb{R}_k}$;
23     end
24 end
25 return ($\mathrm{Cost}_{\mathbb{R}_{\min}} == \infty$? block : accept);
```

segment is protected already by an existing protection tree. If yes, continue to the next segment. Otherwise, a new protection tree that is link disjoint to the segment has to be provisioned. In order to cross share as many links as possible. We need to record every session in the network and the detailed information of which protection link is used to protect which segments. Thus, we iterate every session and every pure protection link in it. If the segment protected by a protection link is link-disjoint to the segment we are trying to protect, this link is a potential free link to use. We set its cost to 0. The process is described by lines 9-14. Then we run PPH and NPF to obtain the protection tree with the lower cost. After each protection tree is established, we update the final topology and overall cost, which is depicted by lines 15-18. By comparing three topologies, we choose the one with the minimum cost as the final survivable topologies.

At any given time, we assume the average number of sessions in the network is denoted by $|d_j|$, which can be calculated by $\lambda\mu$, in which λ is the average arrival rate of multicast sessions and μ is the average holding time of each session. To check each pure protection link of each session in d_j, it takes $O(|V|^2)$. Therefore, the total time complexity of CB_SPT is $O((|V|^2 + |V||d_j||V|^2)) = O(\lambda\mu|V|^3)$.

The second heuristic is WB_SPT shown in Algorithm 4. Instead of looking for the survivable topology with the minimum cost, we try to minimize the total traffic flow and choose each session with the minimal number of wavelengths actually reserved every time. Thus, by extending SPT, the first several steps remain the same, in which we use three different multicast algorithm to construct three primaries trees and obtain the segments for each tree. The difference from SPT starts from line 6. Line 6-8 actually summarizes the total number of links used by the combination of primary and all the protection trees. The next step is to find all the pure protection links in the survivable topology, which can be cross sharing with other pure protection links in any existing sessions. If this link is found, the corresponding link in the new session does not need to be provisioned, which can be subtracted from the total number of wavelengths. This process is

Algorithm 4: Wavelength-based SPT Algorithm (WB_SPT)

```
Input: $G, d = \{s, t_i\}$ $(1 \leq i \leq M), W_k, W_{\min}{=}\infty$
Output: accept or block?
1  for $k = 0; k{<}3; k{+}{+}$ do
2      construct $T_k^m$ by running $k$th heuristic and set $W_k{=}0$;
3      for each segment $l \in T_m^k$ do
4          construct a protection tree $T_{p_i}^k$ for each segment $l$, similar to line 2-14 in Algorithm SPT;
5      end
6      if $\exists T_m^k$ and $T_{p_i}^k$ then
7          set $W_k = |e|,\ e \in \mathbb{R}_k$;
8      end
9      for each protection link $e \in \mathbb{R}_k$ do
10         if $\exists$ protection link $e_j \in d_j$ and segments protected by $e_j$ are link disjoint with the segments
           protected by $e$ then
11             $W_k{-}{-}$;
12         end
13     end
14     if $W_k < W_{min}$ then
15         $W_{\min} = W_k$;
16     end
17 end
18 return ($W_{\min} == \infty$? block : accept);
```

described by line 9-13. We choose the final topology out of the three as the one with the minimum number of wavelengths used. The total time complexity is the same as CB_SPT, which is $O(\lambda\mu|V|^3)$.

VI Numerical Results

In order to investigate the overall performance of the proposed multicast protection schemes in our study, we consider two network topologies: NSF network [28] and USNET [5]. Each link is assigned a certain cost determined by the distance between two end nodes, in terms of kilometers. USNET has a greater number of nodes, links and average node degree than NSF network.

The results consist of three parts. In the first part, we calculate the average cost of provisioning a given multicast session by using SPT in both network topologies. We will compare them with the best existing heuristics, OPP_SDP, as well as the optimal solution developed in [5]. In the second part, we compare the average number of reconfigurations between SPT and OPP_SDP upon any single-link failure. In the last part, we study the performance of two extended heuristic algorithms, CB_SPT and WB_SPT, for dynamic traffic scenarios and compare them with OPP_SDP in terms of blocking probability in various traffic scenarios.

We investigate the total link cost to route one multicast session and its protection trees in this part, under the following assumptions:

1. A network scenario is defined by one source and M destinations and the source and destinations are randomly generated for each network scenario;
2. The bandwidth requirement of each multicast session is one wavelength and the links of network topologies are uncapacitated;
3. For each network scenario, we run the simulation 200 times and take the average value.

Given fixed traffic pattern, we compare the average cost, in kilometers, achieved by SPT scheme to that obtained by with OPP_SDP algorithm as well as the optimal solution solved by formulating the problem using Integer programming, which is also proposed in [5]. Tables 3 and 4 illustrate the average cost of provisioning a multicast session obtained by the different approaches in NSF and USNET networks, respectively, in which the session size denotes the number of destinations in a session and saving ratio reflects the cost saving ratio of heuristic SPT over OPP_SDP and is defined by $(C_{\text{OPP\_SDP}} - C_{\text{SPT}})/C_{\text{OPP\_SDP}}$.

VI.1 Single Multicast Session

Table 3: The comparison of average network cost (in kilometers of fibers) of provisioning a survivable multicast session in NSF network

Session Size	2	3	4	5	6	7	8	9	10	11	12
Optimal	8835.5	10804.1	12537.2	13810.5	15097.2	16240.6	17152.1	18224.8	18984.2	19720.4	21164.9
SPT	8904	11021.8	13274.1	14563.4	15833.7	16899.3	17871.3	19415.3	19876.5	20938.9	22491.9
OPP_SDP	8922.2	11292.3	13383	14757.7	16262.6	17420.4	18432	20039.9	20572.7	21648.6	23351.6
Saving (%)	0.202	2.395	0.814	1.316	2.637	2.991	3.042	3.116	3.379	3.278	3.682

Table 4: The comparison of average network cost (in kilometers of fibers) of provisioning a survivable multicast session in USNET network

Session Size	2	4	6	8	10	12	14	16	18	20	23
Optimal	10839	15909.1	19696.7	22698.9	25518.4	28126.7	30491.4	32935.1	35209.61	37461.0	40838.
SPT	11076	16233.5	19974	24002.5	27212.5	29638.5	32199	34914	36607.5	39493	42761
OPP_SDP	11393	16234	20319.5	24224.5	27649.5	30132.5	32830.5	35526.3	37366.5	39770.6	43307.
Saving (%)	2.782	0.003	1.700	0.916	1.580	1.639	1.923	1.723	2.031	0.698	1.262

It is clear that results produced by both SPT and OPP_SDP are close to the optimal solutions within 10% in NSF network and 15% in USNET. However, SPT produces lower total cost than OPP_SDP approach in both network topologies. The saving ratio of SPT over OPP_SDP in NSF network is between 0.202% and 3.682%, and the maximum saving ratio in USNET is around 2.782% with various session sizes. In NSF network, the advantage of SPT over OPP_SDP gradually increases as the session size increases, which is not the case in USNET. One of the reasons is that NSF network has a smaller average nodal degree such that finding two link disjoint paths for each pair of source and destination conducted in OPP_SDP scheme may end up with long paths. However, SPT is not affected as much since different segments may share the same protection tree. The larger the session size is, the higher the possibility that segments will share protection with one another. However, this feature cannot be applied to OPP_SDP scheme.

In USNET, the average nodal degree is higher and the distances between different pairs of nodes do not vary as much as in the NSF network. The shortest path pair established earlier in OPP_SDP scheme may be shared by other source and destination pairs with higher probability. Therefore, the advantages of SPT scheme is not as significant as that in NSF network.

VI.2 Average Number of Reconfigurations

Since the agility of recovering from failures through rerouting on backup paths, or trees, is decided by the number of switches that need to be reconfigured, we study the failure recovery performance in terms of average number of switch reconfigurations given any link failure in both NSF and USNET network topologies. The method of calculating the number of reconfigurations in the SPT scheme has been presented in Section IV. In OPP_SDP scheme, the shortest pair of paths between the source and each destination is constructed. We consider one as the primary path and another as the protection path. The combination of all the primary paths construct a primary multicast tree. We assume that when a link on the multicast tree fails, all the disrupted primary paths will be rerouted from the source to the corresponding destinations through the protection paths. Accordingly, the same reconfigurations rule described in STP can be applied here. Hence, we obtained the average number of reconfigurations of both protection schemes in NSF and USNET networks as shown in Fig. 6 and 7. Each value is obtained by taking the average over 200 independent cases for each network scenario.

It is obvious that the average number of reconfigurations increases as the session size increases in both topologies due to the fact that final topology gets larger and denser and link sharing becomes more prevalent between different source and destination pair. Therefore, the average nodal degree of the survivable multicast session gets higher and more nodes will become potential switch nodes. Thus, more nodes will actually reconfigure their switch upon a link failure. However, the increase in the number of reconfigurations under the OPP_SDP approach is much faster than SPT as the increase of the session size, because the larger number of destinations in the session results in a greater number of path pairs in the multicast topology and one link capacity may be shared by a large number of primary paths. Therefore, one link failure will disrupt more primary paths and cause more reconfigurations. As we can see in the figures, the performance of SPT and OPP_SDP are close when there are only two destinations. However, when they provision broadcast services, the advantages of SPT over OPP_SDP reaches almost 30% in NSF network and 86% in USNET.

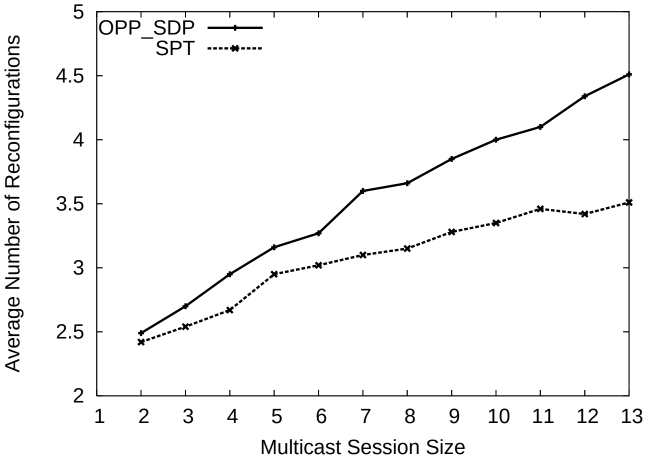


Figure 6: Comparison of Average Number of Reconfigurations, Per Single Link Failure, in NSF network

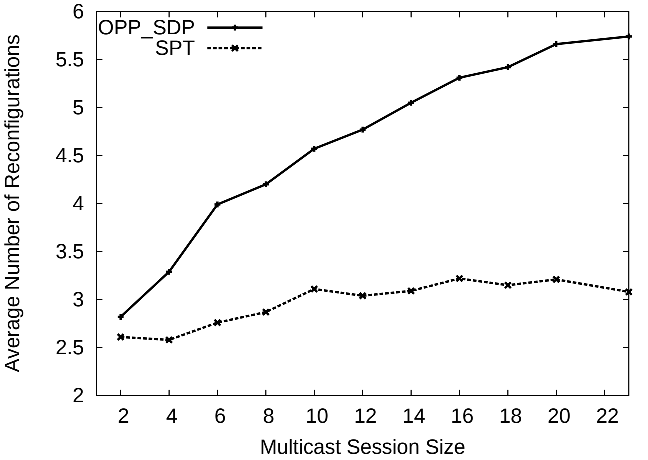


Figure 7: Comparison of Average Number of Reconfigurations, Per Single Link Failure, in USNET network

However, SPT performs very well in USNET since the average number of reconfigurations grows very slowly as the session size increases. Since each protection tree is independent from one another and also from the multicast tree, each protection tree can share a large number of links with the primary tree except the segment it protects, which means any link failure will not result in a significant change between the multicast tree and the corresponding protection, especially when the session size is very large. Therefore, only a limited number of nodes may need reconfiguration differing significantly from OPP_SDP scheme in the same scenario. In summary, SPT outperforms OPP_SDP in terms of the configuration time in all the network scenarios in our study and the advantages vary from 10% to 86%.

One may conclude from the above two sections that, while the cost advantages of SPT over OPP_SDP are modest, recovery from failures when using SPT can be much faster than OPP_SDP, especially if serial failure signaling is used.

VI.3 Dynamic Multicast Sessions

We also study the performance of two extended heuristic algorithms, CB_SPT and WB_SPT, and compare them with OPP_SDP for dynamic traffic scenarios. The simulation is conducted on two realistic networks, NSFNET and USNET. In each simulation run, 1000 randomly generated multicast requests are loaded to the network sequentially and the reject ratio is recorded.

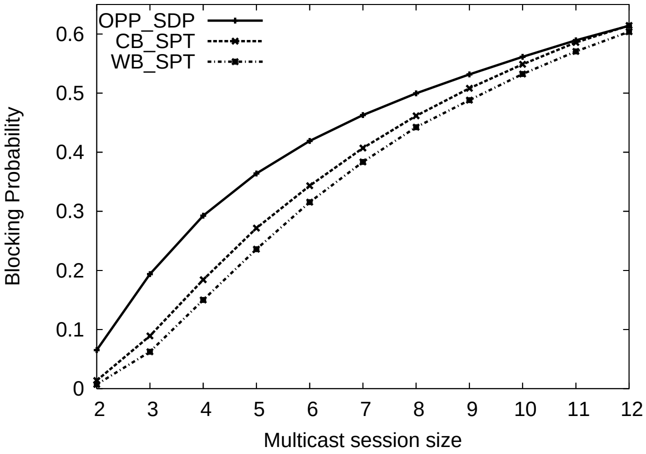


Figure 8: Blocking Probability of dynamic multicast sessions with Erlang=100 in NSFNET

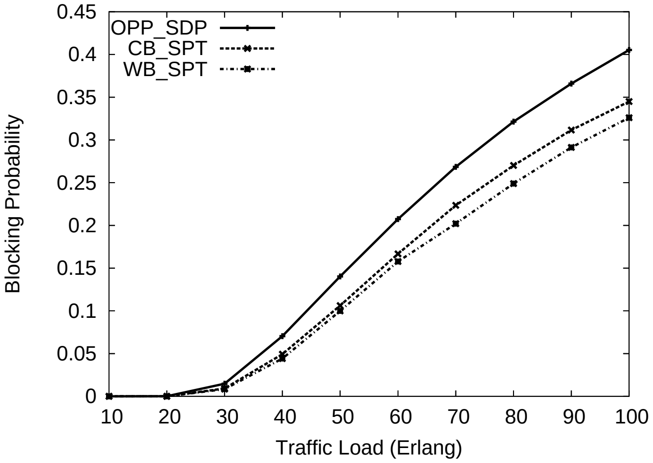


Figure 9: Blocking Probability of dynamic multicast sessions whose session sizes are uniformly distributed in [2, 12] in NSFNET

Figure 8 and 9 show the comparison of blocking probability in NSFNET with various traffic scenarios. Each value in the figures is calculated by averaging the results of 200 independent cases at each traffic scenario. In Fig. 8, we simulate multicast sessions with different size, which vary from 2 to 12, but with the fixed traffic load, which is represented by 100 Erlangs. Since the traffic load in Erlangs = $\lambda\mu$, which means in a fixed time slot, the number of arrival sessions is 100 times of that of departure sessions. The blocking probability increases as the session size increases, since the larger the multicast session is, the more resources will be consumed. In Fig. 9, x axis represents the load in Erlangs which increases from 10 to 100 and we uniformly distribute the multicast session size between 2 and 12 for each scenario. In both figures, our proposed schemes, CB_SPT and WB_SPT, achieves lower reject ratio than OPP_SDP does, which was claimed as the most capacity efficient multicast protection scheme. Among them, WB_SPT achieves the best solution, since we try to use the minimum number of wavelengths to provision each session regardless of the cost of the session. In this case, more resources can remain for future sessions. More than half the sessions are rejected and three schemes behave almost the same. Due to the large size of each session, the network cannot benefit enough from cross-sharing to accept more multicast sessions.

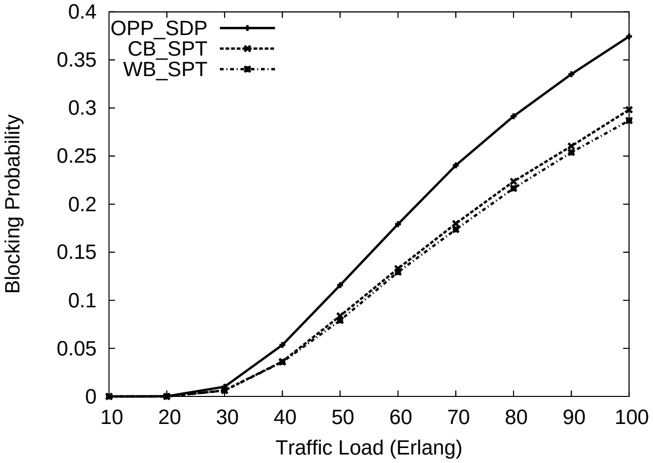


Figure 10: Blocking Probability of dynamic multicast sessions where session sizes are uniformly distributed in [2, 20] in USNET

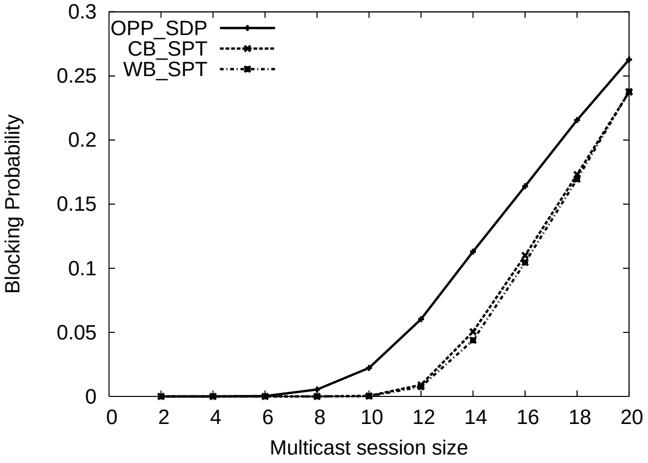


Figure 11: Blocking Probability of dynamic multicast sessions with Erlang=100 in USNET

We also study the dynamic traffic scenarios in USNET network as shown in Figure 10 and 11. In Fig. 10, the multicast session size is uniformly distributed between 2 and 20. In Fig. 11, the traffic load in terms of Erlang is fixed at 100. We can observe that comparison to the OPP_SDP without cross-sharing, both CB_SPT and WB_SPT have significant advantages over OPP_SDP in terms of blocking probability. The advantage reaches 20% as the traffic load or the multicast size increases. In such a dense network, WB_SPT still behaves better than CB_SPT, but the advantage is almost invisible. By combining the results from two networks, we can observe that WB_SPT achieves the best blocking probability in all the scenarios considered, which means minimizing the total number of wavelengths for each arrival multicast session is more efficient than minimizing the total link cost of each session. In fact, by making the cost of each link in the network equal to each other, WB_SPT is equivalent of CB_SPT. This means WB_SPT can be considered as a special case of CB_SPT.

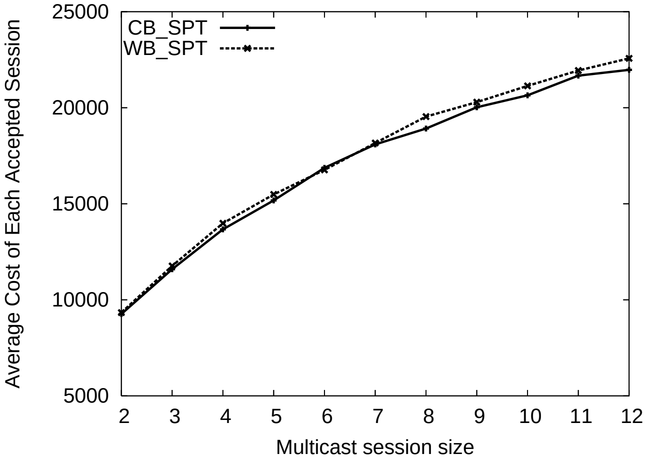


Figure 12: Average cost of each accepted session in NSFNET where traffic load equals 100 Erlangs and the number of wavelengths on each link equals 32

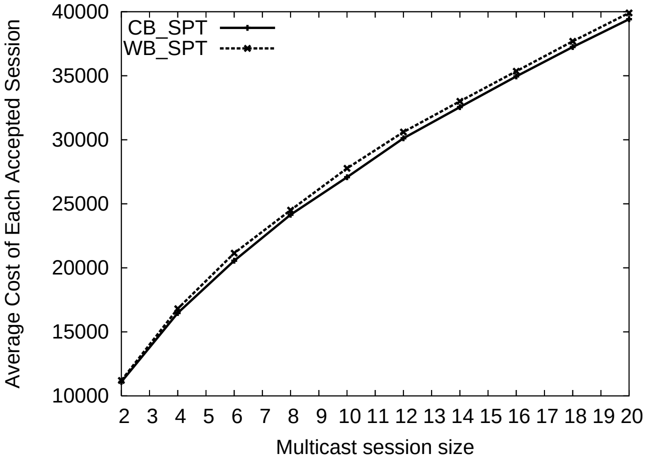


Figure 13: Average cost of each accepted session in USNET where traffic load equals 100 Erlangs and the number of wavelengths on each link equals 64

In addition, we further compare CB_SPT and WB_SPT protection schemes in terms of the average cost of each accepted multicast session. In practise, each multicast session may come from an independent user. The network operator may charge each user by the total cost of each session. Therefore, we study the average cost of each multicast session in NSFNET and USNET and results are shown in Figure 12 and 13, respectively. For each traffic scenario with a unique session size, we take the average value over 1000 independent cases. We can observe that CB_SPT achieves slightly lower cost than WB_SPT does. This is predictable because CB_SPT chooses the final topologies with the minimum cost out of three multicast provisioning schemes. However, the advantages are very small in both networks, especially in USNET. Consider the overall performance of both overall blocking probability and average cost, CB_SPT and WB_SPT perform very close in relatively large network with high nodal degree such as USNET. In sparse and medium size network, such as, NSFNET, WB_SPT can achieve better capacity efficient in terms of blocking probability, but CB_SPT has a small advantage in terms of the average cost of each accepted session.

VII Conclusions

This paper studied the problem of provisioning survivable multicast sessions in optical fiber based backbones, which are the backbones used in the future Internet. Protection against single link failures is provided such that minimum amounts of resources are used. The paper introduced a heuristic algorithm, Segment-based Protection Tree (SPT), to provision and protect a multicast session. In the SPT scheme, three primary multicast trees are established first by three different multicast provisioning approaches, NPF, PPH and DST, respectively, and then each segment of each primary tree is protected by a multicast tree, called protection tree, which is selected out of two candidates produced by NPF and PPH, respectively. Each primary tree and its corresponding protection trees compose a survivable topology. We choose the one with minimum network cost as the final topology. By extending SPT, we also proposed two schemes, CB_SPT and WB_SPT, to protect dynamic multicast sessions, in which we utilize the feature of self-sharing and cross-sharing to enable maximum protection capacity sharing within a multicast session as well as among different multicast sessions.

We studied the performance of SPT in terms of network cost and average number of reconfigurations. SPT uses no more than 10% extra cost over the optimal solution under all network scenarios considered and only 5% extra cost over the optimum when the session size is very small or large, such as unicast or broadcast, respectively. In terms of both cost and recovery performance, SPT achieves better than OPP_SDP, which was considered as the best capacity efficient scheme. We also studied the dynamic traffic scenarios, and the results show that our proposed schemes, CB_SPT and WB_SPT, also achieve better overall blocking probability than OPP_SDP in various network scenarios, in which WB_SPT achieves the better capacity efficiency but CB_SPT achieves lower average cost of each session.

References

[1] Long Long and Ahmed Kamal, “Tree-based Protection of Multicast Services in WDM Mesh Networks,” *in the proceedings of IEEE Globecom*, Nov. 2009.

[2] Hamad, A. M., T. Wu, A. E. Kamal and A. K. Somani, ”Multicasting Protocols For Wavelength-Routing Networks”, Elsevier Computer Networks, Volume 50, Issue 16 , 14 November 2006, Pages 3105–3164.

[3] M. Medard, S. G. Finn, R. A. Barry, and R. G. Gallager, “Redundant trees for preplanned recovery in arbitrary vertex-redundant or edge redundant graphs,” *IEEE/ACM Trans. Netw.*, vol. 7, no. 5, pp. 641-652, Oct. 1999.

[4] L. H. Sahasrabuddhe and B. Mukherjee, “Light trees: Optical multicasting for improved performance in wavelength routed networks,” *IEEE Commun. Mag.*, vol. 37, no. 2, pp. 67-73, Feb. 1999.

[5] N. K. Singhal, L. H. Sahasrabuddhe, and B. Mukherjee, “Provisioning of survivable multicast sessions against single link failures in optical WDM mesh networks,” *J. Lightwave Technol*, vol. 21, pp. 2587-2594, 2003.

[6] N. Singhal and B. Mukherjee, “Protecting multicast sessions in WDM optical mesh network,” *IEEE/OSA J. Lightwave Technol,*, vol. 21, no. 4, pp. 884-892, April 2003.

[7] N. K. Singhal, C. Ou, and B. Mukherjee, “Cross-sharing vs. self-sharing trees for protecting multicast sessions in mesh networks,” *Computer Networks,*, vol. 50, no. 2, pp. 200-206, 2006.

[8] H. Luo, L. Li, H. Yu and S. Wang, "Achieving shared protection for dynamic multicast sessions in survivable mesh WDM networks," *IEEE J. on Sel. Area in Commun.*, Vol. 25, No. 9, Dec. 2007.

[9] F. Zhang, W. Zhong, and Y. Jin, "Optimizations of p-Cycle-based protection of optical multicast sessions," *J. Lightwave Technol*, vol. 26, No. 19, Oct. 2008.

[10] R. Koetter, M. Medard, "An Algebraic Approach to Network Coding," *IEEE/ACM Trans. on Netw.*, Vol. 11, No. 5, Oct. 2003.

[11] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory,*, vol. 46, pp. 1204-1216, July 2000.

[12] S. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Info. Theory,* vol. 49, No. 2, pp.371-381, 2003.

[13] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. M. G. M. Tolhuizen, "Polynomial time algorithms for multicast network code construction," *IEEE Trans. Inf. Theory,* vol. 51, no. 6, pp. 1973-1982, 2005.

[14] T. Ho, M. Medard, R. Koetter, D.R. Karger, M. Effros, J. Shi and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inform. Theory,* vol. 52, no.10, pp. 4413-4430, 2006.

[15] T.Ho, B. Leong, Y. Chang, Y. Wen and R. Koetter, "Network monitoring in multicast networks using network coding," *in International Symposium on Information Theory (ISIT)* 2005.

[16] Al-Kofahi, O. and A. E. Kamal, "Network Coding-Based Protection of Many-to-One Wireless Flows," *IEEE J. Select. Areas Commun. special issue on Network Coding in Wireless Commun.,* Vol. 27, No. 5, June 2009, pp. 797-813.

[17] C. Gkantsidis and P. Rodriguez, "Network coding for large scale content distribution," *Proceedings of IEEE Infocom,* 2005.

[18] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard and J. Crowcroft, "XORs in the Air: Practical Wireless Network Coding," *in Proceedings of SIGCOMM,* 2006.

[19] M. Zhang, L. Wang and P. Ye, "All optical XOR logic gates: technologies and experiment demonstrations," *IEEE Communications magazine,* vol. 43, no.5, pp.s19-s24, May 2005.

[20] Eric D. Manley, Jitender S. Deogun and Lisong Xu, "Network coding for optical layer multicast," *Proc. Broadnets,* 2008.

[21] E.G. Coffman, M.R. Garey, D.S. Johnson, "Approximation algorithms for bin packing - A survey. In: Approximation Algorithms for NP-Hard Problems," *PWS Publishing Company*, Boston, pp 46-93, 1997.

[22] J.M.V de Carvalho, "Exact solution of bin packing problems using column generation and branch and bound," *Annals of Operations Research*, Springer, Volume 86, No. 0, January, 1999.

[23] S. L. Hakimi, "Steiners problem in graphs and its implications," *Networks*, vol. 1, no. 2, pp. 113-133, 1971.

[24] H. Takahashi and A. Matsuyama, "An Approximate Solution for the Steiner Problem in Graphs," *Mathematica Japonica 6* (1980), 573-577.

[25] L. Kou, G. Markowsky and L. Berman, "A fast algorithm for Steiner trees," *Acta Informatica 15* (1981), pp. 141-145.

[26] T. H. Corman, C. E. Leiserson, and R. L. Rivest, "Introduction to Algorithms," 2nd ed. Cambridge, MA: MIT Press, 2001; Chapter 23, "Minimum Spanning Trees".

[27] S. Thorpe, G. Edwards, and D. Stevenson, "Using Just-in-Time to Enable Optical Networking for Grids", *Proc. IEEE Broadnets Workshop*, Oct. 2004.

[28] S. Baroni, “Routing and wavelength allocation in WDM optical networks,” PhD Thesis, University College London, May 1998, pp. 118.