Photon Netw Commun (2016) 31:386–403
DOI 10.1007/s11107-015-0590-3

Static protection against single multicast resource failure in optical WDM networks

Dylan A. P. Davis[1] · **Vinod M. Vokkarane**[1]

Received: 29 July 2015 / Accepted: 6 December 2015 / Published online: 7 January 2016

Abstract The multicast paradigm offers tremendous benefits in efficiency for transmitting data across optical networks, allowing a single client to send information to an entire set of endpoints. A multicast request is most efficiently provisioned through the creation of a tree, with the endpoints, or resources, occasionally serving as branching points. This practice can lead to the source of the request becoming disconnected from the associated resources should one of those branching resources fail. In cases where a large amount of data are currently in transmission, the ramifications of this failure can be severe. We propose an optimal solution through integer linear programming for the static protected multicast routing and wavelength assignment problem, where an entire set of requests is provisioned with built-in redundancy against single resource node failure. We compare the optimal performance against several heuristics and find that protection against this type of failure can be provided with the trade-off of increased wavelength consumption, compared to less-protected solutions.

Keywords Network protection · Optical WDM networks · Resource node failure · Survivability · Multicast · Split-incapable networks

✉ Dylan A. P. Davis
Dylan_Davis@student.uml.edu

Vinod M. Vokkarane
Vinod_Vokkarane@uml.edu

[1] Electrical and Computer Engineering Department, University of Massachusetts Lowell, 1 University Ave, Lowell, MA 01854, USA

1 Introduction

The ever-increasing number of users and applications perpetually consuming and producing data across the Internet of today has produced a great demand which must be met. Extreme-scale science applications continue to grow in importance, and with those applications comes a need for a network that can support the high bit rate necessary for inter-laboratory cooperation and data storage at scale. A prime example can be found in the experiments performed using the Large Hadron Collider (LHC) run by the European Organization for Nuclear Research (CERN), which are expected to generate data ranging from petabytes to exabytes in scale throughout the project lifecycle [1]. Transmitting the resulting measurements and calculations both to storage facilities for data replication and to other laboratories for data verification in a timely fashion requires a medium with the ability to support a tremendous bit rate. Optical networks are such a medium.

In all-optical networks, data are transmitted in the optical domain on fibers which connect optical switches, and the fibers themselves are typically divided into several logical wavelength channels through the process of wavelength-division multiplexing (WDM). Optical Cross-Connects (OXCs) support all-optical WDM by demultiplexing optical signals and multiplexing them onto the correct fibers. A logical connection utilizing WDM between two endpoints is called a *lightpath*, made up of a combination of the physical links between the two nodes and the particular wavelength that carries that connection's traffic on each link. Wavelengths on the same link do not interfere with each other, so more than one lightpath can overlap and share a physical link. This flexibility is mitigated by the fact that without the presence of wavelength converters in the network, a lightpath must use the same wavelength along its entire physical

path. This restriction is known as the wavelength-continuity constraint.

While simple point-to-point connections can be easily supported through this process, modern applications may demand the ability to transmit to and from multiple points. Example use cases include real-time streaming or distributed storage and retrieval. These multiple points can be referred to as a generic set of *resources*, which could be a group of experiment laboratories, data centers in a Content Distribution Network, or user machines receiving a live stream of a presentation. *Multicast*, a widely used point-to-multipoint paradigm between a single source node s and an entire set of resource nodes D, is possible at the optical layer through the use of multicast-capable OXCs (MC-OXCs). MC-OXCs are equipped with power splitters, which enable an incoming optical signal to be replicated/split into some number of outgoing signals [2]. The source and the resources of a request are collectively referred to as *members*, and the logical end-to-end connections between them can be described as a conglomerate of lightpaths known as a *light-tree*. Multicast light-trees can also be supported without splitting technology through the use of a logical overlay, in which an optical signal carrying traffic is dropped to the electrical layer at particular nodes and converted back to optical to forward the traffic to one or more other nodes [3]. The establishment of an optical light-tree can be reduced to the Steiner minimal tree (SMT) problem, which is NP-complete [4].

Regardless of how efficient multicasting could be, multicast light-trees face the same vulnerability as simple point-to-point connections: The failure of a single physical-layer component, such as a fiber link or a switching node, could disconnect an entire session and render the established tree inoperable. An example is shown in Fig. 1a, where the removal of node 3 renders the provisioned connection futile. Survivability is the capability of a network to continue operation even in the presence of accidents, attacks, or equipment failures, which can all have detrimental effects on the integrity and performance of a network. Should a link carrying traffic be cut, any data propagating across the link at the time, which could number upward of 100 Gb depending on the medium, will be lost. Should a network node fail, not only can it no longer be used for forwarding traffic, but the data held in buffers at the time, and any data currently on links incident to the node, will be lost as well.

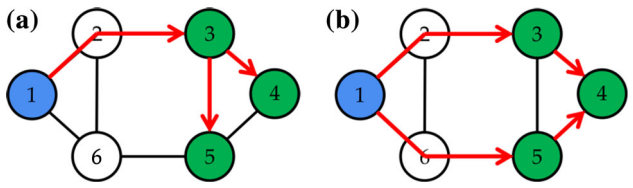

Fig. 1 Two solutions for a multicast request from source node 1 to resource nodes 3, 4, and 5. Should resource 3 fail, source 1 will become disconnected from resources 4 and 5 in solution (**a**), while solution (**b**) will remain viable should any one resource fail. **a** Unprotected multicast. **b** Protected multicast

Network survivability strategies tend to focus on dealing with link failure, which can be fairly common due to human error, such as accidental fiber cuts during construction projects [5,6]. Node failures typically require an extraordinary event to fail, such as a natural disaster [7] or a directed attack with an electromagnetic pulse [8]. The arrival of Hurricane Sandy in New York and New Jersey in October 2012 resulted in the failure of three hundred Verizon facilities along the eastern seaboard [9]. Other unavoidable natural disasters, such as the catastrophic destruction brought about by the 2011 T$\bar{o}$hoku earthquake and tsunami in Japan [10], can occur, often without warning. Researchers have proposed methods for designing networks to survive disasters, such as the authors of [11], who propose optimal disaster-aware methods for selecting locations for underwater fiber-optic cables. Survivability as a discipline has been well studied [12] and modeled [13], with strategies generally broken into two categories. Restoration involves computing and setting up an alternate path should a network failure occur, at the cost of greater downtime, while protection requires provisioning paths in advance and quickly switching traffic should something fail, with the cost of having to set aside resources which other active demands will be unable to use. Recent research in restoration includes work on improving efficiency through approaches that take advantage of the flexibility inherent in elastic optical networks [14], where the optical spectrum can be divided into slots that can be combined to meet demands. Protection can be further divided into two classes: dedicated, in which each demand has its own distinct backup path to switch to in the event of a failure, or shared, in which costs can be reduced by sharing a backup path between multiple demands, with only one demand able to switch should a failure occur.

Specialized methods for multicast survivability exist for all aforementioned subdivisions of the field. Preventing a disconnection in the event of a single link failure has been studied extensively [15,16]. The recent work proposed in [16] provides a tree-segment-based protection solution, lowering the blocking for dynamic traffic, where demands are satisfied as they arrive in the system and compete for resources, with a marginal increase in cost over earlier backup light-tree solutions [15]. An overlay dual-homing approach is used in [17] to protect client nodes against the failure of any one optical link or access network link, while others add additional paths onto a multicast tree to make established demands survivable against the failure of intermediate nodes between the source and the resources [18]. The problem of protecting multicast sessions across multi-domain optical networks is tackled in [19], where the authors propose two cost-efficient heuristics for building a survivable inter-domain multicast tree.

Other methods include the use of predesigned cycles (p-Cycles), which have been proposed by multiple authors as a cost-efficient method for protecting against a wide variety of failures in optical networks [20]. Multicast survivability problems have been solved through overlapping p-Cycles to protect against intermediate node failure [21], or constructing trees for recovering from both node and link failure [22]. The authors of [23] take a different approach, focusing instead on a subset of node failure and proposing a solution for resource node failure protection in the context of virtual networks, provisioning an additional physical backup node for each virtual resource. Work in a similar vein is done in [24], where the authors provide a mixed ILP and heuristic algorithms.

With the currently explored approaches in mind, we aim to examine the static problem of protecting multicast requests in WDM networks against the failure of a single node out of the group of multicast resource nodes through a logical overlay. We have previously found solutions to this problem through both heuristics [25] and ILP [26] in the context of traditional optical WDM networks and have found that traditional SMT solutions for multicast often do not survive the removal of just one resource node from a tree. The removal of a branching, or Steiner, point can disconnect a branch from a multicast tree, rendering any resources on that branch unreachable through the established lightpaths. Resource nodes can, depending on the topology, often be used as Steiner points. Given the important role resource nodes serve in multicast communications, the loss of a resource node can result in both the disconnection of entire light-tree branches and the loss of any critical data at that node. This can make resource nodes a tempting target for anyone looking to cause maximum harm to an established multicast connection.

In order to mitigate this harm, we present protection methods that enable a connection between a source node and the resource nodes to remain intact should any one resource node fail. These strategies are not guaranteed to protect against the failure of any intermediate node, which can be costly [18], but rather act as a less-expensive compromise that will protect against targeted removal of these high-priority nodes. Aiming to protect only against resource failure allows algorithms to be designed with a focus on a clearly defined set of nodes, rather than attempting to tackle the problem of dealing with *any* node or Steiner point failing, which might be determined dynamically depending on how the light-tree is constructed. In this paper, we present an extended and improved ILP formulation for solving the static version of the resource-failure protection problem and compare the performance to heuristics.

A major motivation behind our approach is informed by the use case of large-scale science facilities, such as the previously mentioned LHC. These facilities may stream tremendous quantities of data at one time to multiple remote sites and cannot afford to lose both the data at a failed node and data en route to the other sites at the time of failure. Node in this case would refer to both a remote site and the closest major switch responsible for forwarding data to not only that site, but likely other repositories or laboratories as well. The demands in a scientific network can be large in terms of size and may be long-lasting, so the static approach of determining how efficiently a group of demands can be provisioned is appropriate. Each request receives dedicated backup paths, to ensure that every request is left with an option to switch to should a failure occur, which is a possibility when shared backup paths are utilized. We assume that the resource nodes are geographically distributed in such a way that a natural disaster, which would disable a swath of networking equipment simultaneously, would not affect multiple resources at a time. Finally, our protected solutions are constructed through the use of a logical overlay, as multicast-capable switches may often not be available.

The paper is structured as follows. The formal problem definition is given in Sect. 2, and the ILP solution follows in Sect. 3. Section 4 describes our proposed resource-failure protection heuristics, the performances of which will be quantitatively compared with the ILP in Sect. 5. Section 6 concludes the paper.

2 Problem definition

We are given the following inputs to the problem.

- A topology $G = (V, E)$, where V is a set of network nodes, and E is a set of unweighted, directed edges. The wavelength-continuity constraint is enforced.
- A set of wavelengths W, where $|W|$ is the number of wavelength channels supported by each fiber.
- A static set of immediate reservation multicast connection requests R, with $r \in R$. Each request $r = (s_r, D_r)$, where source $s_r \in V$ and the set of resources $D_r \subseteq V - s_r$, must be established while protecting it against the failure of any single node in D_r. Immediate reservation requests must be provisioned at the time of arrival, and the entire set arrives at once. The bandwidth granularity of each request is assumed to be equivalent to the capacity of a single wavelength, and no grooming is performed.

Defining the problem formally, the goal is to establish a protected solution G' for each multicast request r on topology G such that the removal of any one resource does not disconnect the remaining multicast members of that request, and the total number of wavelengths required to satisfy all requests is minimized. The protection requirement can be formulated as a bound of $|M| \geq 2$ for any minimal ver-

tex cut M, where $M \subseteq D_r$ for solution G'. A *cut* is a set of vertices from V, such that their removal causes the remaining graph G to become disconnected. Cuts may be of various sizes, but the *minimal cut* is that which contains the smallest set of nodes from V. Such a protected solution G' can be described as *biconnected*, meaning that it takes the removal of two elements (in this case, only resource vertices) to disconnect the solution. An example multicast solution is shown in Fig. 1a which, while using a minimum number of links, is unprotected should resource 2 be removed. A protected solution for the same multicast request is shown in Fig. 1b.

3 Integer linear programming solution

The Multicast Destination Failure Protection ILP formulated below is based in part on the Drop At Any Node (DAAN) multicast overlay ILP, presented in [27]. The DAAN approach to establishing multicast circuits in optical networks efficiently establishes a logical overlay over the underlying physical network. In these solutions, we provision requests by creating a set of lightpath routes in the overlay layer from the source node of a request to each resource member. Each lightpath route can terminate, or "drop," at any node to the electronic layer and can then return to the optical layer to forward the traffic toward another node. In this manner, a light-tree can be constructed without splitting hardware, at some cost to efficiency if a purely optical-level solution were possible. We build on this formulation to create resource-failure survivable overlays, providing protection for multicast sessions in any biconnected network. Our survivable solution combines multiple lightpaths to form a primary end-to-end "connection" from the source for each resource. Individual lightpaths can be shared between different connections. If there are any intermediate resource nodes present in a resource's primary connection, we provide a backup connection which does not share any intermediate resource nodes with the primary connection.

3.1 Minimum wavelengths required ILP formulation (ILP-MinWR)

3.1.1 Given

V is the set of nodes in the network.

A_{ij} is 1 if a physical link exists between $i, j \in V$.

R is the set of multicast requests, which are numbered 1 through R. For a given multicast request r, we denote the source node of the request as s_r and the set of resource member nodes is represented as D_r. The set of non-resource members is denoted as $X_r = V - D_r \cup s_r$.

W is the set of wavelengths available on each link.

H is the indexing set for variable $P_{u,v}^{r,d,h}$, where $H = \{1, 2\}$. This is used to indicate whether either one or two end-to-end connections are required between the source and a particular resource to provide resource-failure protection, with $h = 1$ indicating the primary path, and $h = 2$ the secondary.

Z is a very large number, used as an upper bound for inequalities.

3.1.2 Variables

The ILP will solve for the following variables:

$L_{u,v}^{r,w}$ is a binary variable, with a value of 1 if a lightpath is established for request r from node u to node v on wavelength w. It is 0 otherwise.

$F_{u,v,i,j}^{r,w}$ is a binary variable, with a value of 1 if there is a flow on the physical link from node i to node j on wavelength w, for a lightpath from node u to node v, for request r. It is 0 otherwise.

C_w^r is a binary variable, with a value of 1 if wavelength w is used to service multicast request r. It is 0 otherwise.

$P_{u,v}^{r,d,h}$ is binary, equal to 1 if there is an end-to-end connection (i.e., a series of lightpaths) from the source node s_r to resource $d \in D_r$ for request r, using lightpath (u, v) as a virtual link. These connections are indexed by $h \in H$. The value is 0 otherwise.

$LP_{u,v}^{r,h}$ is binary, equal to 1 if the lightpath (u, v) is a virtual link in a connection P from the source node s_r to *any* resource node. The value is 0 otherwise. A lightpath can act as a virtual link for several end-to-end connections between the source s_r and the resource nodes in R.

$I_{n,u,v}^r$ is binary, equal to 1 if node $n \in V$ is present in lightpath (u, v). The value is 0 otherwise.

$G_{n,u,v}^{r,d,h}$ is binary, equal to 1 if node $n \in V$ is present in lightpath (u, v), where (u, v) is a virtual link in request r's connection h to resource d. The value is 0 otherwise.

$N^{r,d,h}$ is an integer counter variable, equal to the number of resource nodes present in end-to-end connection P from s_r to d.

$B^{r,d}$ is binary, equal to 1 if any connection $P_{u,v}^{r,d,h}$ contains at least one intermediate resource node $\in D_r$, indicating that the connection from s_r to resource node d would become disconnected should another resource node fail. This variable determines whether more than one connection P is required to provide sin-

gle resource node failure protection for node d. The value is 0 otherwise.

$MaxIndex$ is an integer variable, representing the largest wavelength index used on any link network-wide. Minimizing this value is the objective.

3.1.3 Constraints

Objective function:

minimize: MaxIndex

Subject to:

$$MaxIndex \geq C_w^r \times w; \quad \forall\, r \in R, \quad w \in W. \tag{1}$$

$$\sum_{w}^{W} C_w^r \geq 1; \quad \forall\, r \in R. \tag{2}$$

$$L_{u,v}^{r,w} \leq C_w^r; \quad \forall\, r \in R, \quad w \in W, \quad u, v \in V. \tag{3}$$

A lower bound for the maximum wavelength index used is provided in Constraint (1). Constraint (2) ensures that at least one wavelength is used to satisfy each request and Constraint (3) that the set of established lightpath routes are bound by the number of wavelengths used.

$$\sum_{r}^{R}\sum_{u}^{V}\sum_{v}^{V} F_{u,v,i,j}^{r,w} \leq 1; \quad \forall\, i, j \in V, \quad w \in W. \tag{4}$$

$$F_{u,v,i,j}^{r,w} \leq A_{ij} \times L_{u,v}^{r,w} \quad \forall\, r \in R, \quad u, v, i, j \in V, \quad u \neq v, i \neq j, \quad w \in W. \tag{5}$$

$$\sum_{i}^{V} F_{u,v,i,j}^{r,w} - \sum_{k}^{V} F_{u,v,j,k}^{r,w} = \begin{cases} 0 & \text{if } j \neq u, v \\ L_{u,v}^{r,w} & \text{if } j = v \\ -L_{u,v}^{r,w} & \text{if } j = u \end{cases} \quad \forall\, u, v, j \in V, \quad w \in W, \quad r \in R. \tag{6}$$

Constraints (4) through (6) are the physical-layer constraints. (4) prevents any wavelength being used by more than one request on any particular link, while Constraint (5) allows lightpaths to be established only between nodes connected by a physical link in the topology. Constraint (6) is a flow conservation constraint, requiring the in-flow to equal the out-flow of any bypass, or non-endpoint, node. The lightpath sources or resources have either negative or positive flow, respectively.

$$\sum_{u}^{V}\sum_{w}^{W} L_{u,v}^{r,w} \geq 1; \quad \forall\, r \in R, \; v \in D_r. \tag{7}$$

$$\sum_{v}^{V}\sum_{w}^{W} L_{s_r,v}^{r,w} \geq 1; \quad \forall\, r \in R. \tag{8}$$

$$\sum_{u}^{V}\sum_{w}^{W} L_{u,s_r}^{r,w} = 0; \quad \forall\, r \in R. \tag{9}$$

$$\sum_{v}^{V}\sum_{w}^{W} L_{u,v}^{r,w} - Z \times \sum_{v}^{V}\sum_{w}^{W} L_{v,u}^{r,w} \leq 0; \quad \forall\, r \in R, \; u \in V, \; u \neq s_r. \tag{10}$$

$$\sum_{u}^{V}\sum_{w}^{W} L_{u,v}^{r,w} - \sum_{u}^{V} L_{v,u}^{r,w} \leq 0; \quad \forall\, r \in R, \; v \in X_r. \tag{11}$$

Lightpath establishment is covered through constraints (7) through (11). At least one lightpath must terminate at each resource node so the data can be received (7) and at least one lightpath must originate from the source node to carry the data (8). No lightpaths need to terminate at the source node (9), but lightpaths *are* allowed to terminate at any other node. Lightpaths can only originate at a non-source node if there is at least one terminating lightpath at the node (10). This is accomplished through summing up the number of terminating lightpaths at a node and subtracting the product of the number of lightpaths originating at the node and a large number. This ensures that the constraint can hold when there are a greater number of lightpaths originating from the node than terminating at it. There must be at least one lightpath originating from a non-resource node if a lightpath terminates there, so the data can be forwarded to resources (11).

$$I_{u,v}^{r,n} \times Z \geq \sum_{w}^{W}\sum_{i}^{V} F_{u,v,i,n}^{r,w} + \sum_{w}^{W}\sum_{k}^{V} F_{u,v,n,k}^{r,w}; \quad \forall\, r \in R, \; u, v, n \in V. \tag{12}$$

$$I_{u,v}^{r,n} \leq \sum_{w}^{W}\sum_{i}^{V} F_{u,v,i,n}^{r,w} + \sum_{w}^{W}\sum_{k}^{V} F_{u,v,n,k}^{r,w}; \quad \forall\, r \in R, \; u, v, n \in V. \tag{13}$$

A critical component of protecting multicast requests against the failure of a resource node is determining which nodes are physically present within a lightpath. The binary variable $I_{u,v}^{r,n}$ is set to 1 if there is at least one flow into or out of node n, indicating that a lightpath (u, v) either originates, terminates, or passes through n (12) and (13).

$$\sum_{h}^{H}\sum_{u}^{V\setminus\{s_r\}} P_{u,s_r}^{r,d,h} = 0; \quad \forall r \in R, \; d \in D_r. \tag{14}$$

$$\sum_{h}^{H}\sum_{v}^{V\setminus\{d\}} P_{d,v}^{r,d,h} = 0; \quad \forall r \in R, \; d \in D_r. \tag{15}$$

$$\sum_{v}^{V\setminus\{s_r\}} P_{s_r,v}^{r,d,1} = 1; \quad \forall r \in R, \; d \in D_r. \tag{16}$$

$$\sum_{v}^{V\setminus\{s_r\}} P_{s_r,v}^{r,d,2} = B^{r,d}; \quad \forall r \in R, \; d \in D_r. \tag{17}$$

$$\sum_{u}^{V\setminus\{d\}} P_{u,d}^{r,d,1} = 1;\ \ \forall r \in R,\ d \in D_r. \tag{18}$$

$$\sum_{u}^{V\setminus\{d\}} P_{u,d}^{r,d,2} = B^{r,d};\ \ \forall r \in R,\ d \in D_r. \tag{19}$$

$$\sum_{u}^{V\setminus\{v\}} P_{u,v}^{r,d,h} = \sum_{a}^{V\setminus\{v\}} P_{v,a}^{r,d,h};\ \ \forall r \in R,\ v \in V \setminus \{s_r, d\},\ d \in D_r,\ h \in H. \tag{20}$$

$P_{u,v}^{r,d,h}$ is used to keep track of which (u,v) lightpaths are used for either the primary (h = 1) or backup (h = 2) connections from s_r to each $d \in D_r$. A connection from s_r to a d does not need a lightpath terminating at s_r (14). Constraint (15) similarly prevents lightpaths originating at node d from being used in connection from s_r to d. One lightpath originating from the source must be a part of the primary end-to-end connection to each d (16), and if the binary variable $B^{r,d}$ is equal to 1, there must also be a lightpath originating at the source for the backup connection as well (17). A similar set of constraints (18) and (19) is established for lightpaths terminating at resource nodes. Finally, the number of lightpaths in a connection from s_r to d terminating at a node v must equal the number of lightpaths originating at v, enforcing the continuity of connection traffic (20).

$$B^{r,d} \times Z \geq N^{r,d,1} - 2;\ \ \forall r \in R,\ \ d \in D_r. \tag{21}$$

$$B^{r,d} \leq N^{r,d,1} - 2;\ \ \forall r \in R,\ \ d \in D_r. \tag{22}$$

$B^{r,d}$ is a binary variable for determining when a backup connection is necessary for a particular resource d. Constraints (21) and (22), when combined, force B to equal 1 when there are at least two resource nodes in the primary connection, indicating that there is at least one intermediate resource node. Otherwise, $B = 0$.

$$LP_{u,v}^{r,h} * |D_r| \geq \sum_{d}^{D_r} P_{uv}^{r,d,h};\ \ \forall r \in R,\ \ u, v \in V,\ h \in H. \tag{23}$$

$$LP_{u,v}^{r,h} \leq \sum_{d}^{D_r} P_{u,v}^{r,d,h};\ \ \forall r \in R,\ \ u, v \in V,\ \ h \in H. \tag{24}$$

$$LP_{u,v}^{r,h} \leq \sum_{w}^{W} L_{u,v}^{r,w};\ \ \forall r \in R,\ \ u, v \in V,\ \ h \in H. \tag{25}$$

$$\sum_{h}^{H} LP_{u,v}^{r,h} \geq \sum_{w}^{W} L_{u,v}^{r,w};\ \ \forall r \in R,\ \ u, v \in V. \tag{26}$$

Each lightpath is a component of a connection, so the variable $LP_{u,v}^{r,h}$ is necessary for indicating when a lightpath is used in a connection. It is important to note that a lightpath can be used for multiple connections in a request simultaneously. LP is equal to 1 when it is both used in at least one connection (23) and (24), and the lightpath L is established for the solution (25) and (26).

$$G_{n,u,v}^{r,d,h} \geq P_{u,v}^{r,d,h} + I_{u,v}^{r,n} - 1;\ \ \forall r \in R,\ n, u, v \in V,\ d \in D_r,\ \ h \in H. \tag{27}$$

$$G_{n,u,v}^{r,d,h} \leq P_{u,v}^{r,d,h};\ \ \forall r \in R,\ \ n, u, v \in V,\ d \in D_r,\ h \in H. \tag{28}$$

$$G_{n,u,v}^{r,d,h} \leq I_{u,v}^{r,n};\ \ \forall r \in R,\ n, u, v \in V,\ d \in D_r,\ \ h \in H. \tag{29}$$

$$N^{r,d,h} = \sum_{n}^{M_r}\sum_{u}^{V}\sum_{v}^{V} G_{n,u,v}^{r,d,h};\ \ \forall r \in R,\ d \in D_r,\ h \in H. \tag{30}$$

$$\sum_{h}^{H}\sum_{u}^{V}\sum_{v}^{V} G_{n,u,v}^{r,d,h} \leq 1;\ \ \forall r \in R,\ n \in D_r \setminus \{d\}. \tag{31}$$

Variable I keeps track of when a node n is in a lightpath (u,v), and variable P indicates when a lightpath (u,v) is used in a connection, so indicator variable G can be used to show when node n is an intermediate node in an end-to-end connection. The value of G is determined through the equivalent of the logical operation $I \wedge P$ in Constraints (27), (28), and (29). Variable $N^{r,d,h}$ is then used to store the number of resource nodes in an s_r to d connection by summing up the value of $G_{n,u,v}^{r,d,h}$ across each lightpath and node. The N variable is used for determining when a backup connection is necessary in constraints (21) and (22). Variable G is finally then essential for determining whether a backup connection is survivable; if a resource node $d' \neq d$ is in both connections to d, then removing it will cause d to become disconnected from the source. This is prevented through constraint (31), which restricts every other resource to appearing at most once across every connection to resource d.

3.2 Minimum wavelength-links ILP formulation (ILP-MinWL)

While the presented ILP does minimize the number of wavelengths required on any one link in the network, it does not necessarily minimize alternative costs. Such a cost could be minimizing the number of wavelengths used across the entire network (i.e., the number of wavelengths used on each link, summed over each link in the network), or the *wavelength-links*. We present an alternative formulation for the objective of minimizing this cost, while satisfying a static set of multicast requests with logical overlays protected against single resource failure.

3.2.1 Variables

The ILP utilizes Eq. (2)—through (31), and requires an additional variable, replacing $maxIndex$:

WL is an integer variable, representing the number of wavelength-links network-wide. Minimizing this value is the objective.

3.2.2 Constraints

In addition, an alternative objective function is required, along with a new constraint to replace Constraint (1).

Objective function:

$$minimize: WL$$

Subject to:

$$WL = \sum_{r}^{R} \sum_{w}^{W} \sum_{u}^{V} \sum_{v}^{V} \sum_{i}^{V} \sum_{j}^{V} F_{u,v,i,j}^{r,w}. \tag{32}$$

Constraint (32) sets the value of WL equal to the number of flows across all links, summing up the value of $F_{u,v,i,j}^{r,w}$ across all requests, lightpaths, and links. As $F_{u,v,i,j}^{r,w}$ is a binary variable, with a value of 1 only when wavelength w is used on link (i, j) for some lightpath (u, v) for a request r, performing this summation is sufficient for determining the number of wavelength-links.

4 Heuristics

While the ILP provides an optimal solution in terms of minimizing required wavelengths, the run-time growth is exponential, making the ILP infeasible to run on large or well-connected topologies. We briefly describe two heuristics proposed in [25], which provide resource-failure protection approaches for each multicast request with a more reasonable time complexity.

4.1 Steiner minimal tree with failure-avoidance backup (Steiner-FAB)

A traditional SMT provides a minimal solution, in terms of hops or links used, for a multicast request, and while NP-Complete, it can be approximated in $\Theta(|V|^3)$ time [28]. While a SMT efficiently connects a source node to all of its resources, the solution found is in no way guaranteed to survive the failure of a resource node. Depending on the paths chosen to connect the member nodes, it is possible that a resource node is chosen as a branching point, disconnecting any nodes along one of the branches should it fail. Due to the minimal nature of a SMT for satisfying multicast requests, the approach is a logical starting point for developing a resource-failure survivable multicast heuristic. *Steiner minimal tree with failure-avoidance backup (Steiner-FAB)* builds upon a request's primary SMT, with the addition of backup paths to every *vulnerable* resource that could become disconnected due to another resource's failure, and is described by Algorithm 1. The approximation algorithm in [28] is used to build the original SMT in polynomial time.

The algorithm first constructs an empty set of $Vulnerable$ nodes and builds a SMT for the given request r and topology G (Lines 1–2). With the SMT established, Steiner-FAB evaluates the route between the source and each resource (Line 3) d_i in the tree, finding $path_i$ to resource d_i through the $route$ function, a depth-first-search (Line 4). Then, the length of $path_i$ is stored in the variable len (Line 5). The empty set V_{d_i} is instantiated, along with a variable $d_{nearest}$ for storing the nearest resource to resource d_i (Lines 6–7). Then, each resource d_j other than d_i is checked (Line 8), and if $path_i$ contains d_j (Line 9), the distance between d_i and d_j (Line 10) is stored in set V_{d_i} with d_j as tuple (d_j, len_j) (Line 11). With this loop through all other resources d_j, all resource nodes along the path to d_i are found. Then, as long as at least one intermediate resource node was found (Line 12), the node closest to d_i is found using the stored distance (Line 13), and a tuple $(d_i, d_{nearest}, len)$ is stored in the set of $Vulnerable$ nodes.

With that process repeated for each resource d_i, the procedure for protecting each vulnerable node is repeated until there are no vulnerable resources remaining (Line 15). Then, using the len value stored in each tuple in $Vulnerable$, the $mostVulnerable$ resource node tuple is determined (Line 16). Then, a subgraph G' is constructed, using all nodes in V except for $d_{nearest}$ stored in the $mostVulnerable$ tuple (Line 17), and the $ShortestPath$ from s to d_i is found and added to the original SMT t_r (Line 18). The $mostVulnerable$ tuple is removed from the $Vulnerable$ set (Line 19), and then for each tuple v remaining in the $Vulnerable$ set (Line 20), find all paths $Paths_i$ within the tree t_r to the resource node d_i (Line 21). If there is more than one path to node d_i (Line 22), then for each $path_p$ in that set of paths (Line 23), if the path does not contain the previously identified dangerous node $d_{nearest}$, then the resource d_i is considered safe and the associated tuple v is removed from the $Vulnerable$ set (Lines 24–27). Finally, now that the $Vulnerable$ set is empty, all resource nodes have been protected through the addition of backup paths, and the survivable modified tree t_r is returned (Line 28). An illustrative example of how Steiner-FAB would establish such a protected solution on a SMT with one vulnerable member is shown in Fig. 2. The run-time complexity of Algorithm 1 is bounded by the time taken to build the initial SMT, $O(|V|^3)$, the time taken to identify vulnerable nodes $O(|V|^3)$, and the time taken to route the set of $O(|V|)$ backup paths $O(|V|log|V|)$. Therefore, the total worst-case

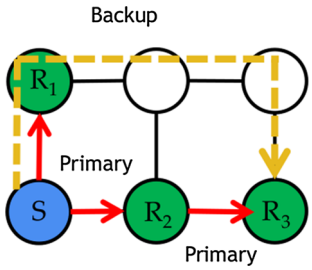

Fig. 2 Steiner-FAB algorithm first constructs a SMT (*solid red*) from source S to resources $R1$, $R2$, and $R3$. The algorithm then checks each resource node for vulnerability and identifies the single vulnerable multicast resource, $R3$. The backup lightpath (*dashed gold*) directly from S to node $R3$ is established to avoid the potential failure node $R2$ that, when removed, would disconnect S from $R3$. If either $R1$ or $R2$ fails, $R3$ will remain connected to source S (Color figure online)

run-time complexity is $O(|V|^3 + |V|^3 + |V|(|V|log|V|)$, or simply $O(|V|^3)$.

4.2 Critical resource biconnective survivability (CRB)

While Steiner-FAB does not add significant complexity to the SMT approach while providing the desired protection, it is possible to establish a more efficient solution through constructing a series of paths that prevents disconnection in the event of a member failure, without requiring additional backup paths after construction. Our *Critical Resource Biconnective Survivability* (CRB) solution, proposed in [25], aims to establish a subgraph G' from physical topology G for each request r in such a way that there is an alternate path to every $d \in D_r$, the set of resources, from s_r, the source, should it be impossible to establish a direct path from s_r to d without traversing another member of D_r. With these alternate paths, should any one resource fail, there will always be a path available to every remaining d from s_r. It is important to note that there is no guarantee that a protected solution G' can be found if the underlying physical topology G is itself not biconnected.

The CRB algorithm begins by creating a set of $Shortest\,Paths$ (Line 1). Then, for each possible member node pair, a subgraph (Line 3) is found that does not contain the set of $NonPairMembers = D_r \cup \{s_r\} \setminus \{m_i, m_j\}$ (Line 2). With this subgraph, the $Shortest\,Path$ between the node pair (m_i, m_j) can be found, if one exists, providing the shortest path between those two nodes that does not contain any other member node (Line 4). With this set of $Shortest\,Paths$, a new logical topology H is constructed, where the new set of nodes V' is made up of all multicast members for the request r, and there is a link $(i, j) \in L$ between each member that has a corresponding shortest path SP in $Shortest\,Paths$ (Line 5). A weight is assigned to each link in L equal to the length of the corresponding shortest path in $Shortest\,Paths$ (Lines 6–7). Following that, each logical edge adjacent to source s_r is added to the logical solution set, $LogicalSolutionEdges$ (Lines 8–10).

Algorithm 1: Steiner Minimal Tree with Failure Avoidance Backup (Steiner-FAB)

input : Multicast Request: $r = (s_r, D_r)$, $D_r = \{d_1, d_2, \ldots, d_K\}$
: Topology: G = (V,E)
output: SMT with backup paths protected against single critical resource failure

```
1  Vulnerable ← ∅
2  build SMT t_r for r
3  foreach d_i ∈ D_r do
4      path_i = t_r.route(s_r, d_i)
5      len = path_i.length
6      V_di ← ∅
7      d_nearest = NULL
8      foreach d_j ∈ D_r | d_j ≠ d_i do
9          if path_i.contains(d_j) then
10             len_j = route(d_i, d_j).length
11             V_di.add(d_j, len_j);
12     if V_di ≠ ∅ then
13         d_nearest = min(V_di) by len_j
14         Vulnerable ← Vulnerable ∪ {(d_i, d_nearest, len)}
15 while Vulnerable ≠ ∅ do
16     mostVulnerable = max(Vulnerable) by len
17     G' = (V', E') | V' = V − mostVulnerable.d_nearest
18     t_r.add(G'.ShortestPath(s, d_i))
19     Vulnerable = Vulnerable − {mostVulnerable}
20     foreach v ∈ Vulnerable do
21         Paths_i = t_r.findAll(route(s, v.d_i))
22         if Paths_i.size > 1 then
23             foreach path_p ∈ Paths_i do
24                 if path_p.contains(v.d_nearest) then
25                     continue
26                 else
27                     Vulnerable = Vulnerable − {v}
28 return t_r
```

Adding those logical edges ensures that there is an uninterrupted physical path between source s_r and those logically adjacent resource nodes, but there may be resources that cannot be reached without traversing another resource node. Those nodes are considered $Vulnerable$ (Line 11), and the resource nodes which could cause a disconnection will be put into the set $Failure$ (Line 12). For every resource node d (Line 13), if there is no logical edge directly connecting the source to that resource node (Line 14), then that resource node is marked as $Vulnerable$ (Line 15), a logical path is found to that node d through depth-first search (Line 16), and each resource node in that path is added to the set of $Failure$ nodes (Line 17–18). A $BiconnectedSolution$ is then found through running the *Minimum-Cost 2-Vertex Connected* (MC2VC) approximation algorithm on a subgraph where the nodes are $\{s_r\} \cup Vulnerable \cup Failure$ (Line 19). This gives you a minimum-cost biconnected subgraph,

Algorithm 2: Critical Resource Biconnectivity (CRB)

input : Multicast Request: $r = (s_r, D_r)$, $D_r = \{d_1, d_2, \ldots, d_K\}$
: Topology: G = (V,E)
output: Pruned physical topology G' protected against single critical resource failure

1 $ShortestPaths \leftarrow \emptyset$; **foreach** $(m_i, m_j) \in D_r \cup \{s_r\}, m_i \neq m_j$ **do**
2 $\quad NonPairMembers = D_r \cup \{s_r\} \setminus \{m_i, m_j\}$;
3 $\quad find\ subgraph\ G_{i,j} = (V_{i,j}, E_{i,j}) | V_{i,j} = V \setminus NonPairMembers$;
4 $\quad ShortestPaths.add(G_{i,j}.ShortestPath(m_i, m_j))$;
5 $construct\ subgraph\ H = (V', L)\ |\ V' = D_r \cup \{s_r\}, L = \{(i, j) | (i, j)\ SP \in ShortestPaths\}$;
6 **foreach** $(i, j) \in L$ **do**
7 $\quad (i, j)[WEIGHT] = length(\ (i, j)\ SP \in ShortestPaths)$;
8 $LogicalSolutionEdges \leftarrow \emptyset$;
9 **foreach** $(s_r, j) \in L$ **do**
10 $\quad LogicalSolutionEdges.add((s_r, j))$;
11 $Vulnerable \leftarrow \emptyset$;
12 $Failure \leftarrow \emptyset$;
13 **foreach** $d \in D_r$ **do**
14 $\quad$ **if** $(s_r, d) \notin LogicalSolutionEdges$ **then**
15 $\quad\quad Vulnerable.add(d)$;
16 $\quad\quad Path_d = H.findPath(s_r,\ d)$
17 $\quad\quad$ **foreach** $v \in Path_d,\ v \neq s_r, d$ **do**
18 $\quad\quad\quad Failure.add(v)$;
19 $BiconnectedSolution = MC2VC(\ (\{s_r\} \cup Vulnerable \cup Failure),\ L)$;
20 **foreach** $edge\ (i, j) \in BiconnectedSolution$ **do**
21 $\quad$ **if** $(i, j) \notin LogicalSolutionEdges$ **then**
22 $\quad\quad LogicalSolutionEdges.add(\ (i, j)\ )$;
23 $H' = (V', LogicalSolutionEdges)$;
24 $PhysicalEdges \leftarrow \emptyset$;
25 $PhysicalNodes \leftarrow \emptyset$;
26 **foreach** $edge\ (i, j) \in H'$ **do**
27 $\quad PhysicalEdges.add(\{(i, j)\ |\ (i, j)\ SP \in ShortestPaths\})$;
28 $\quad PhysicalNodes.add(\{node\ v\ |\ v \in (i, j)\ SP \in ShortestPaths\})$;
29 **return** $G' = (PhysicalNodes,\ PhysicalEdges)$

constructed so that it can survive the removal of any one node. The edges in that $BiconnectedSolution$ are added to the set of $LogicalSolutionEdges$ if they are not already included in the set (Lines 20–22). A subgraph of logical graph H is then constructed, consisting only of member nodes V' and the $LogicalSolutionEdges$ (Line 23). The corresponding $PhysicalEdges$ and $PhysicalNodes$ (Lines 24–25) are then found through mapping the $LogicalSolutionEdges$ back to the physical topology (Lines 26–28). The physical subgraph $G' = (PhysicalNodes, PhysicalEdges)$ is then returned as a solution protected against the failure of any multicast resource node.

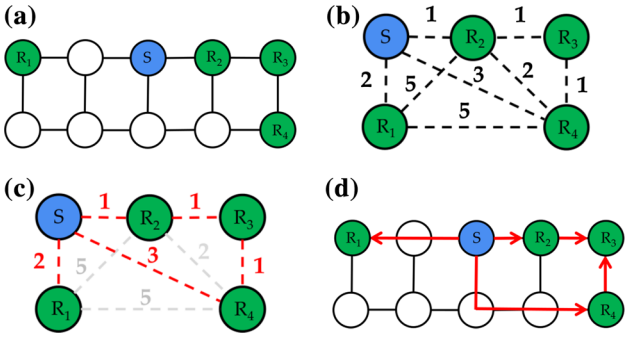

Fig. 3 Critical resource biconnective (CRB) survivability. **a** Physical topology G. **b** Logical topology H. **c** Pruned logical topology H'. **d** Pruned physical topology G'

An example conversion from physical topology G, to logical H, is shown in Fig. 3a, b. H is pruned in Fig. 3c and then mapped back to the physical topology as an established circuit in Fig. 3d. The time complexity of CRB has a lower bound of the optimized $O(|V|^3)$ complexity of the minimum-cost 2-vertex connectivity problem. Including the $O(|E|)$ complexity from each of the conversions from G to H and from H' to G', and the $O(|V|^2)$ time to identify vulnerable and potential failure nodes, the total complexity of this approach is $O(|V|^3 + |V|^2 + |E|)$, which can be reduced to $O(|V|^3 + |E|)$ [25].

5 Results and analysis

In this section, we quantitatively examine and compare the performance of the two presented resource-failure protection multicast ILPs (henceforth referred to as ILP-MinWR, for minimizing wavelengths required on any link network-wide, and ILP-MinWL, for minimizing the number of wavelength-links in the network) and both of the heuristics. In addition, the SMT approximation presented in [28] is considered alongside the heuristics and ILP. Even though the SMT solves only the multicast problem, not the survivable version, it is useful to compare this minimal multicast solution to the survivable solutions to give an approximate lower bound for wavelength consumption and other metrics.

The heuristic simulations, implemented in Python, and the ILPs, which were implemented with AMPL and solved using the Gurobi version 5.6.3 optimization software package, are run on both the 14-node National Science Foundation (NSFNet) topology shown in Fig. 4 and a symmetrical 25-node Manhattan topology depicted in Fig. 5. It is assumed that for each link (i, j) in the topology, there is a fiber available in both directions, each possessing its own set of available wavelengths. When comparing the ILP solutions and the heuristics, we generated 30 request sets, using 30

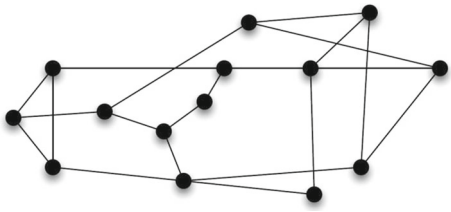

Fig. 4 Fourteen-node NSFNet topology. The topology has 21 edges, an average path length of 2.14 hops, a maximum nodal degree of 4, a minimum nodal degree of 2, an average nodal degree of 3, and a path diameter of 3 hops

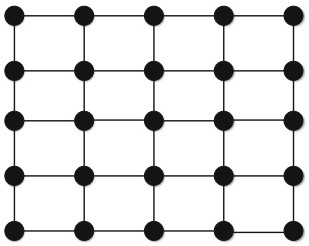

Fig. 5 Twenty-five-node Manhattan topology. The topology has 40 edges, an average path length of 3.33 hops, a maximum nodal degree of 4, a minimum nodal degree of 2, an average nodal degree of 3.2, and a path diameter of 8 hops

different seed values, all comprised of 5 requests, each with randomly selected source and two resource nodes following a uniform random distribution. The same was done to generate sets which required three, four, and five uniformly randomly selected resources. The ILPs and heuristics were evaluated only for sets of 5 requests, due to the high complexity of integer linear programming, with the values presented here averaged across the 30 seeds. The CRB, Steiner-FAB, and SMT algorithms were run on the same topology for the same sets of requests, the results of which are shown in Table 1 for the 14-node NSFNet and Table 2 for the 25-node Manhattan network. We additionally compared the performance between just the heuristics, with a much higher request set size of 1000 requests and with resource requirements of up to seven resources, on the same topologies. The results for NSFNet are shown in Table 3 and in Table 4 for the larger Manhattan network.

Eight metrics are compared: (1) the minimum number of wavelengths required on each link to provision all requests in a set; (2) the number of wavelength-links (the summation across all links, of the number of wavelengths provisioned per link) required network-wide to provision all requests in a set; (3) the average number of links utilized (had at least one wavelength allocated) per request; (4) the average diameter, or the minimum number of hops to reach the furthest resource from the source, per request; (5) the average path length difference between the source and the closest/furthest resource per request, also known as jitter, which can be important when considering delay for data arrival between all resource nodes; (6) the average number of resource nodes that must be removed from a provisioned request to disconnect the source from the remaining resource(s). If this average value is lower than 2, that indicates that some requests provisioned with that method can be disconnected with only 1 resource removed; (7) the number of requests in a set that are protected against single resource failure; (8) the running time, in seconds, for completing an entire request set given a certain algorithm.

The general ranking of the approaches is presented in Table 5. The ranking may change based upon the topology or the number of requests, with the SMT and Steiner-FAB solutions often swapping position, but the presented ordering generally holds. ILP-MinWR and ILP-MinWL score the best among the survivable approaches when it comes to the wavelengths required to provision all requests, and the number of wavelength-links consumed to provision a request set, respectively. The wavelengths required, wavelength-links, average links utilized, diameter, and jitter all increase as a greater number of resources must be reached per request. Each request must, on average, be provisioned a larger proportion of the network as the resource set size increases, so all approaches appropriately perform more poorly. Among the survivable heuristics, CRB outperforms Steiner-FAB in terms of wavelength-links and the average number of links utilized per request, while Steiner-FAB performs better when considering the number of hops between the source and the resources. CRB is less costly in terms of the number of wavelengths/links consumed, but the average time to transfer data to the furthest resources from the source will likely be lower with Steiner-FAB. CRB, as it utilizes the *Minimum-Cost-K-Vertex-Connected-Subgraph* algorithm as a component, can survive a greater number of resource node failures on average than any other approach, as CRB paths are often established to ensure that resources can connect not only to the source, but to each other. All survivable approaches (ILP-MinWR, ILP-MinWL, CRB, and Steiner-FAB) provide only protected solutions, while SMT is in no way guaranteed to provide survivable trees. SMT, given its minimal nature, solves the multicast problem in the shortest time, by far, and is followed by Steiner-FAB and CRB, which both have additional requirements beyond connecting the source to its resources. The ILP-MinWL and ILP-MinWR approaches both require significantly more time regardless of the topology compared to the heuristics, even when the heuristics have to handle a greatly increased number of requests in Tables 3 and 4.

Digging into the differences between NSFNet and the larger Manhattan network, the number of wavelengths required per link network-wide is slightly increased, but the number of wavelength-links consumed per request set is, in the worst-case with ILP-MinWR, almost doubled. The average number of links utilized per request experiences a slightly smaller growth, while the average request diameter and jitter scale with the increased average path length

Table 1 Comparison between the ILP and the CRB, Steiner-FAB, and SMT heuristics on the **NSFNet 14-node** topology

Num. resources	Approach	Wavelengths required	Wavelength-links	Avg. links utilized	*Avg.diameter*	Avg. jitter	Avg. failures to disconnect	Num. protected	Running time
2	ILP-MinWR	**1.8**	39.8	7.65	3.68	1.65	**2**	**5**	49,236
2	ILP-MinWL	2.17	20.9	4.18	2.86	1.06	**2**	**5**	18856.66
2	CRB	2.63	22.43	4.37	2.75	1.01	**2**	**5**	0.0042
2	Steiner-FAB	2.5	24.37	4.75	**2.62**	**0.88**	**2**	**5**	0.0033
2	SMT	2.23	**18.67**	**3.65**	2.77	1.03	1.65	3.23	**0.0025**
3	ILP-MinWR	**2.07**	56.27	11.76	3.77	2.06	2.74	**5**	105,305
3	ILP-MinWL	2.77	29.8	6.09	3	1.56	2.79	**5**	34,291
3	CRB	3.4	35.87	6.59	3.32	1.78	**2.98**	**5**	0.016
3	Steiner-FAB	3.7	41.43	7.39	**2.86**	**1.31**	2.78	**5**	0.0068
3	SMT	2.73	**25.5**	**4.86**	3.29	1.72	1.44	1.1	**0.0043**
4	ILP-MinWR	**2.4**	65	12.8	3.87	2.42	2.84	**5**	165345.83
4	ILP-MinWL	3.11	36.86	7.69	3.07	1.69	3.02	**5**	70,514
4	CRB	4.4	51.47	8.53	3.57	2.19	**3.81**	**5**	0.2366
4	Steiner-FAB	4.7	58.83	9.64	**2.93**	**1.54**	3.07	**5**	0.0116
4	SMT	3.17	**31.43**	**5.87**	3.53	2.11	1.14	0.23	**0.0066**
5	ILP-MinWR	**3.37**	87.33	15.27	3.72	2.45	2.96	**5**	344940.73
5	ILP-MinWL	3.62	48.77	9.42	3.28	2.05	2.94	**5**	121723.1
5	CRB	5.73	71.57	10.81	3.94	2.7	**4.31**	**5**	1.6922
5	Steiner-FAB	6.07	77.33	11.61	**2.99**	**1.75**	2.93	**5**	0.018
5	SMT	3.6	**37.27**	**6.83**	3.79	2.51	1.11	0.13	**0.0095**

All values represent an average across 30 distinct request sets generated by random seed values, with each set of **5** requests requiring either **2, 3, 4, or 5** resources. The *Avg. Links Utilized*, *Avg. Diameter*, *Avg. Jitter*, and *Avg. Failures to Disconnect* metrics each present a value averaged across the 5 requests in a request set, which are then averaged over the 30 request sets. The *Running Time* is in seconds. The best value for each metric is marked in bold

Table 2 Comparison between the ILP and the CRB, Steiner-FAB, and SMT heuristics on the **Manhattan 25-node** topology

Num. resources	Approach	Wavelengths required	Wavelength-links	Avg. links utilized	Avg. diameter	Avg. jitter	Avg. failures to disconnect	Num. protected	Running time
2	ILP-MinWR	**2.01**	59.19	10.98	5.38	2.68	**2**	**5**	432693.7
2	ILP-MinWL	2.47	31.16	6.04	4.24	**1.74**	**2**	**5**	263451.16
2	CRB	2.7	33.63	6.11	4.28	1.83	**2**	**5**	0.005
2	Steiner-FAB	2.87	36.17	6.69	**4.21**	1.77	**2**	**5**	0.0044
2	SMT	2.5	**27.2**	**5.33**	4.22	1.77	1.63	3.13	**0.0033**
3	ILP-MinWR	**2.23**	96.86	18.22	6.31	3.87	2.9	**5**	432705.25
3	ILP-MinWL	3.06	50.31	9.54	5.09	2.96	2.94	**5**	355,440
3	CRB	3.7	51.83	8.84	4.98	2.87	**3**	**5**	0.0089
3	Steiner-FAB	3.73	57.77	9.9	**4.74**	**2.63**	2.87	**5**	0.009
3	SMT	2.9	**36.43**	**7.02**	4.81	2.67	1.72	1.8	**0.0064**
4	ILP-MinWR	**2.57**	94.84	16.99	6.01	4.11	3.49	**5**	432912.5
4	ILP-MinWL	3.25	53.25	10.18	**4.95**	**2.95**	3.75	**5**	432678.5
4	CRB	4.67	70.63	11.09	5.41	3.56	**3.97**	**5**	0.032
4	Steiner-FAB	4.63	80.07	12.85	5.02	3.17	3.52	**5**	0.0166
4	SMT	3.2	**43.8**	**8.34**	5.21	3.33	1.48	0.8	**0.0101**
5	ILP-MinWR	3.71	156.45	24.79	6.09	4.36	3.9	**5**	432784.73
5	ILP-MinWL	4	86	15.17	5.35	3.7	3.9	**5**	432272.84
5	CRB	5.6	89.83	13.28	5.72	4.03	**4.93**	**5**	0.1781
5	Steiner-FAB	5.57	103.03	15.53	**5.25**	**3.56**	3.92	**5**	0.0264
5	SMT	**3.37**	**51.03**	**9.64**	5.53	3.83	1.21	0.27	**0.0146**

All values represent an average across 30 distinct request sets generated by random seed values, with each set of **5** requests requiring either **2, 3, 4, or 5** resources. The *Avg. Links Utilized*, *Avg. Diameter*, *Avg. Jitter*, and *Avg. Failures to Disconnect* metrics each present a value averaged across the 5 requests in a request set, which are then averaged over the 30 request sets. The *Running Time* is in seconds. The best value for each metric is marked in bold

Table 3 Comparison between the CRB, Steiner-FAB, and SMT algorithms on the **NSFNet 14-node** topology

Num. resources	Approach	Wavelengths required	Wavelength-links	Avg. links utilized	Avg. diameter	Avg. jitter	Avg. failures to disconnect	Num. protected	Running time
2	CRB	174.3	4449.13	4.29	2.75	1.04	**2**	**1000**	1.18
2	Steiner-FAB	186.07	4802.87	4.68	**2.6**	**0.89**	**2**	**1000**	0.78
2	SMT	**147.6**	**3687.4**	**3.59**	2.72	1	1.65	654.8	**0.59**
3	CRB	267	7027.6	6.36	3.23	1.77	**2.99**	**1000**	3.6
3	Steiner-FAB	293.7	7882.57	7.12	**2.8**	**1.33**	2.85	**1000**	1.87
3	SMT	**196.43**	**5008.7**	**4.77**	3.1	1.62	1.6	301.17	**1.22**
4	CRB	366.3	10140.13	8.38	3.64	2.33	**3.88**	**1000**	43.36
4	Steiner-FAB	410.53	11286.7	9.37	**2.9**	**1.58**	3.17	**1000**	3.21
4	SMT	**239.07**	**6215.4**	**5.8**	3.38	2.05	1.29	98.23	**1.97**
5	CRB	495.2	14011.33	10.67	3.89	2.68	**4.36**	**1000**	331.83
5	Steiner-FAB	537.3	14909.73	11.4	**2.95**	**1.74**	2.95	**1000**	4.95
5	SMT	**281.03**	**7344.9**	**6.73**	3.6	2.38	1.1	24.6	**2.71**
6	CRB	639.57	18423.07	13.32	3.92	2.79	**4.12**	**1000**	1878.37
6	Steiner-FAB	667.33	18704.53	13.25	**2.99**	**1.85**	2.51	**1000**	6.87
6	SMT	**318.67**	**8425.23**	**7.59**	3.79	2.65	1.02	3.47	**3.42**
7	CRB	803.27	24093.93	16.84	3.71	2.63	**3.43**	**1000**	5620.89
7	Steiner-FAB	809.07	22653.57	14.95	**3.01**	**1.93**	2.2	**1000**	7.69
7	SMT	**352.8**	**9470.57**	**8.41**	3.94	2.86	1	0.17	**4**

All values represent an average across 30 distinct request sets generated by random seed values, with each request in a set of **1000** requiring either **2, 3, 4, 5, 6, or 7** resources. The *Avg. Links Utilized*, *Avg. Diameter*, *Avg. Jitter*, and *Avg. Failures to Disconnect* metrics each present a value averaged across the 1000 requests in a request set, which are then averaged over the 30 request sets. The *Running Time* is in seconds. The best value for each metric is marked in bold

Table 4 Comparison between the CRB, Steiner-FAB, and SMT algorithms on the **Manhattan 25-node** topology

Num. resources	Approach	Wavelengths required	Wavelength-links	Avg. links utilized	Avg. diameter	Avg. jitter	Avg. failures to disconnect	Num. protected	Running time
2	CRB	155.6	6732.3	6.18	4.28	1.82	**2**	1000	1.52
2	Steiner-FAB	164.83	7043.17	6.58	**4.23**	**1.75**	**2**	1000	1.19
2	SMT	**130.73**	**5423.97**	**5.32**	4.26	1.76	1.66	661.6	0.95
3	CRB	222	10262.03	8.68	4.84	2.81	**3**	1000	3.19
3	Steiner-FAB	245.93	11163.37	9.66	**4.7**	**2.65**	2.92	1000	2.2
3	SMT	**167.07**	**7126.8**	**6.88**	4.81	2.74	1.69	345.87	2.06
4	CRB	298.7	14013.1	11	5.28	3.51	**3.97**	1000	11.53
4	Steiner-FAB	333.57	15581.2	12.54	**4.99**	**3.21**	3.58	1000	5.3
4	SMT	**199.83**	**8636.2**	**8.2**	5.19	3.39	1.47	158.1	3.27
5	CRB	377.33	18243.63	13.29	5.74	4.14	**4.88**	1000	75.49
5	Steiner-FAB	423.53	20216.27	15.21	**5.21**	**3.6**	3.83	1000	8.31
5	SMT	**228.93**	**10021.07**	**9.38**	5.48	3.86	1.25	62.5	4.5
6	CRB	469.23	23337.33	15.74	6.18	4.7	**5.59**	1000	558.53
6	Steiner-FAB	512.9	25034.07	17.71	**5.38**	**3.9**	3.71	1000	11.62
6	SMT	**254.63**	**11321.13**	**10.45**	5.72	4.24	1.11	21.4	5.77
7	CRB	570.07	29661.07	18.63	6.47	5.1	**5.86**	1000	3407.53
7	Steiner-FAB	599.33	29992.47	20.04	**5.51**	**4.14**	3.31	1000	10.92
7	SMT	**278.37**	**12555.9**	**11.44**	5.92	4.53	1.04	6.03	7.03

All values represent an average across 30 distinct request sets generated by random seed values, with each request in a set of **1000** requiring either **2, 3, 4, 5, 6, or 7** resources. The *Avg. Links Utilized*, *Avg. Diameter*, *Avg. Jitter*, and *Avg. Failures to Disconnect* metrics each present a value averaged across the 1000 requests in a request set, which are then averaged over the 30 request sets. The *Running Time* is in seconds. The best value for each metric is marked in bold

Table 5 General ranking in descending order of each approach for each metric considered

Wavelengths required	Wavelength-links	Avg. links utilized	Avg. diameter	Avg. jitter	Avg. failures to disconnect	Num. protected	Running time
ILP-MinWR	SMT	SMT	Steiner-FAB	Steiner-FAB	CRB	ILP-MinWR	SMT
ILP-MinWL	ILP-MinWL	ILP-MinWL	ILP-MinWL	ILP-MinWL	ILP-MinWL	ILP-MinWL	Steiner-FAB
SMT	CRB	CRB	SMT	SMT	Steiner-FAB	CRB	CRB
CRB	Steiner-FAB	Steiner-FAB	CRB	CRB	CRB	Steiner-FAB	ILP-MinWL
Steiner-FAB	ILP-MinWR	ILP-MinWR	ILP-MinWR	ILP-MinWR	SMT	SMT	ILP-MinWR

The ordering of some approaches may change slightly based upon the topology or number of requests in a set, but this order generally holds. It is important to note that all approaches except SMT perform equally well in the *Num. Protected* metric

in the Manhattan network compared to the NSFNet topology. The number of resource failures required to disconnect an established session does experience a slight increase as the network grows larger, which can be tied to the greater path length in the network. Resource nodes, which are chosen uniformly, may end up more "spread out" in a larger network, increasing the number of failures required to disconnect a solution completely. For the SMT approach, which is the only non-survivable algorithm, the average number of requests in a request set which are protected against the failure of any single resource node increases slightly as the size of the network increases. The running times for ILP-MinWR, by far the worst due to the difficulty of minimizing the maximum number of wavelengths consumed on *any* link in the network, and ILP-MinWL, both increase as the topology size increases, and almost converge as the number of resources in a set increases. SMT, meanwhile, always completes in the shortest amount of time, taking slightly longer in a larger topology. Steiner-FAB follows a similar pattern of growth, but CRB scales at a much higher rate in terms of running time compared to other heuristics. This is related to the greater computational complexity in comparison with SMT and Steiner-FAB, and CRB appropriately greatly increases in running time as a larger number of resource nodes are required per request.

While SMT consumes fewer wavelength-links on average, it is important to keep in mind that the SMT solutions are often not protected against the failure of a single resource node. The ILPs and heuristics, on the other hand, always require at least 2 resources to fail before the established request is considered disconnected. In addition, Steiner-FAB, which builds backup paths alongside a SMT to provide survivability, provisions requests in such a way as to provide the average lowest diameter and jitter among the examined approaches. This is a by-product of the backup paths: resources further from the source in a SMT are more likely to have another resource present as an intermediate node in the SMT, so they often require a backup path directly from the source. The SMT, while reducing the number of hops to connect the source to all resources with one tree, does not necessarily use the shortest path from the source to any particular resource. The additional backup paths, while not necessarily minimal, can be shorter than the established path in the SMT to the furthest resources. In several ways, the heuristics appear to be satisfactory substitutes for the ILP, as they outperform the ILP in several metrics, and can run much more efficiently at scale.

Overall, it can be seen that CRB allows wavelengths to be used more efficiently than Steiner-FAB for the same level of protection, but the greater end-to-end delay could be a detrimental factor for time-sensitive multicast scenarios. SMT outperforms both survivable heuristics on nearly all fronts, being beaten only by Steiner-FAB in terms of diameter and jitter, as previously mentioned, and by both CRB and Steiner-FAB in terms of the number of resource failures the provisioned requests can withstand, on average. The relationship between SMT and the two survivable heuristics perfectly demonstrates the trade-off a user can expect when survivability is a requirement: the survivable methods are likely to be far more inefficient in terms of cost. This trade-off between survivability and cost must be weighed when deciding which methods to use for provisioning network requests. When comparing survivable heuristics, CRB tends to be more cost-effective in terms of wavelengths, but is more costly in terms of delay and running time. If a large number of requests, or a greater number of resources, must be protected, Steiner-FAB may be chosen over CRB if time is a concern. On the other hand, if protection against *multiple* resource node failure is a priority, CRB, on average, is more resilient against multiple failures than every other approach.

6 Conclusion

The point-to-multipoint nature of the multicast communication paradigm plays an important role in supporting a wide variety of networking applications, including cloud-based services, streaming media, and distributed storage or retrieval. The high-bandwidth available in optical WDM networks makes them an excellent candidate for supporting the paradigm. However, the most efficient solution for provisioning multicast requests may create points of vulnerability that can lead to loss of data or service. We have proposed two optimal solutions through ILP to solve this issue in the static case of provisioning an entire set of multicast requests in networks that do not have optical-level multicast splitters available, and we compared their performance to two survivable heuristics, finding that they solve the same problem with a slightly higher wavelength consumption, but in much faster time. The built-in redundancy provided by these solutions is guaranteed to protect any single request against the failure of one of its resources in well-connected networks. When a demand requires guaranteed survivability, due to either its importance, its size, or both, these methods can secure network transmissions against a potentially devastating type of failure, should the trade-off in terms of cost be acceptable.

Future areas of work include further simulation and evaluation of performance for these approaches on larger topologies and for more sizable request sets. It is possible that the relationship between the heuristics in terms of resource consumption could vary based on topology, and the optimal solution may outperform the other approaches to an even greater degree, although the running time is likely to increase significantly on more complex topologies. Additional objectives, such as minimizing the diameter of

solutions, can also be considered and formulated. How the proposed methods perform in other types of networks, such as the increasingly researched Elastic Optical Networks, where the spectrum allocated per request can be flexibly tailored to meet a demand, will be examined. The approaches presented in this paper did not consider blocking, so future work can include updated versions of the heuristics which are able to prioritize blocking reduction. Going beyond just the static problem of provisioning a known set of requests, the dynamic problem can be considered, where requests are satisfied as they become known, and failure events and recovery times can be simulated following a probabilistic model.

Acknowledgments This work has been supported by the National Science Foundation CARGONET project under Grant CNS-1406370 and by the Department of Energy (DOE) PROPER project under Grant DE-SC0012115TDD.

References

[1] CERN. Worldwide Large Hadron Collider Computing Grid. http://lcg.web.cern.ch/lcg/public/default.htm
[2] Rouskas, G.: Optical layer multicast: rationale, building blocks, and challenges. IEEE Netw. **17**, 60–65 (2003)
[3] Sahasrabuddhe, L., Mukherjee, B.: Light trees: optical multicasting for improved performance in wavelength routed networks. IEEE Commun. Mag. **37**(2), 67–73 (1999)
[4] Karp, R.M.: Reducibility among combinatorial problems. In: Raymond E. Miller, James W. Thatcher, Jean D. Bohlinger, (eds.) Complexity of Computer Computations, pp 85–103. Springer US (1972)
[5] Zhou, D., Subramaniam, S.: Survivability in optical networks. IEEE Netw. **14**(6), 16–23 (2000)
[6] Modiano, E., Narula, A.: Survivable lightpath routing: A new approach to the design of WDM-based networks. IEEE J. Sel. Areas Commun. **20**(4), 800–809 (2002)
[7] Habib, M.F., Tornatore, M., Dikbiyik, F., Mukherjee, B.: Disaster survivability in optical communication networks. Comput. Commun. **36**(6), 630–644 (2013)
[8] Neumayer, S., Zussman, G., Cohen, R., Modiano, E.: Assessing the vulnerability of the fiber infrastructure to disasters. IEEE/ACM Trans. Netw. **19**(6), 1610–1623 (2011)
[9] Kwasinski, A.: Effects of hurricanes Isaac and Sandy on data and communications power infrastructure. In: Proceedings of 2013 35th International Telecommunications Energy Conference 'Smart Power and Efficiency' (INTELEC), pp. 1–6, Oct 2013
[10] Adachi, T., Ishiyama, Y., Asakura, Y., Nakamura, K.: The restoration of telecom power damages by the great East Japan earthquake. In: IEEE 33rd International Telecommunications Energy Conference (INTELEC), pp. 1–5, Oct 2011
[11] Msongaleli, D.L., Dikbiyik, F., Zukerman, M., Mukherjee, B.: Disaster-aware submarine fiber-optic cable deployment. In: 2015 International Conference on Optical Network Design and Modeling (ONDM), pp. 245–250, May 2015
[12] Sterbenz, J.P., Hutchison, D., Çetinkaya, E.K., Jabbar, A., Rohrer, J.P.: Resilience and survivability in communication networks: strategies, principles, and survey of disciplines. Comput. Netw. **54**(8), 1245–1265 (2010)
[13] Heegaard, P.E., Trivedi, K.S.: Network survivability modeling. Comput. Netw. **53**(8), 1215–1234 (2009)
[14] Amar, D., Le Rouzic, E., Brochier, N., Lepers, C.: Multilayer restoration in elastic optical networks. In: 2015 International Conference on Optical Network Design and Modeling (ONDM), pp. 239–244, May (2015)
[15] Singhal, N., Sahasrabuddhe, L., Mukherjee, B.: Provisioning of survivable multicast sessions against single link failures in optical WDM mesh networks. J. Lightwave Technol. **21**(11), 2587–2594 (2003)
[16] Constantinou, C.K., Ellinas, G., Manousakis, K.: Survivability of multicast requests in mesh optical networks. In: 2014 International Conference on Optical Network Design and Modeling, pp. 7–12, May 2014
[17] Kmiecik, W., Walkowiak, K.: Survivable overlay multicasting in WDM optical networks with dual homing architecture. In: 2014 International Conference on Optical Network Design and Modeling, pp. 19–24, May 2014
[18] Luekijna, K., Saivichit, C.: Multicast traffic reconfiguration in WDM network for single node failure design. In: The 9th International Conference on Advanced Communication Technology, vol. 3, pp. 1833–1838, Feb 2007
[19] Guo, L., Wu, J., Hou, W., Li, Y.: Multicast protection algorithms based on aggregated logical topology in survivable multi-domain optical networks. Optik **123**, 521–526 (2012)
[20] Zhong, W., Zhang, F.: An overview of p-Cycle based optical multicast protection approaches in mesh WDM networks. Opt. Switch. Netw. **8**(4), 259–274 (2011)
[21] Jaumard, B., Li, H.: Design of p-Cycles for full node protection in WDM mesh networks. In: 2011 IEEE International Conference on Communications (ICC), pp. 1–5, Jun 2011
[22] Zhang, F., Zhong, W.: P-Cycle based tree protection of optical multicast traffic for combined link and node failure recovery in WDM mesh networks. IEEE Commun. Lett. **13**(1), 40–42 (2009)
[23] Liao, D., Sun, G., Anand, V., Yu, H.: Survivable provisioning for multicast service oriented virtual network requests in cloud-based data centers. Opt. Switch. Netw. **14**(Part 3), 260–273 (2014)
[24] Guo, B., Qiao, C., Wang, J., Yu, H., Zuo, Y., Li, J., Chen, Z., He, Y.: Survivable virtual network design and embedding to survive a facility node failure. J. Lightwave Technol. **32**(3), 483–493 (2014)
[25] Davis, D.A.P., Plante, J.M., Vokkarane, V.M.: Critical resource multicast protection in data center networks. In: IEEE ICC 2015—Next Generation Networking Symposium, London, United Kingdom, June 2015
[26] Davis, D.A.P., Vokkarane, V.M.: Static protection against single multicast resource failure. In: 2015 International Conference on Optical Network Design and Modeling (ONDM), pp. 92–97, May 2015
[27] Gadkar, A., Plante, J.M., Vokkarane, V.M.: Multicast overlay for high-bandwidth applications over optical WDM networks. J. Opt. Commun. Netw. **4**(8), 571–585 (2012)
[28] Takahashi, H., Matsuyama, A.: An approximate solution for the Steiner problem in graphs. Math. Jpn. **24**, 573–577 (1980)

Dylan A. P. Davis earned his B.S. in computer science in 2014 at the University of Massachusetts Dartmouth. He is currently pursuing a Ph.D. in Electrical and Computer Engineering at the University of Massachusetts Lowell. His research interests include renewable energy in networks, multicast and manycast communication, survivable network design, and software defined networking.

Vinod M. Vokkarane is an Associate Professor in the department of Electrical and Computer Engineering at the University of Massachusetts Lowell. Prior to this, he was an Associate Professor of Computer and Information Science at the University of Massachusetts Dartmouth. He was a Visiting Scientist at the Claude E. Shannon Communication and Network Group, Research Laboratory of Electronics at MIT from 2011 to 2014. He received the B.E. degree with Honors in Computer Science and Engineering from the University of Mysore, India, and the M.S. and the Ph.D. degree in Computer Science from the University of Texas at Dallas. His primary research areas include design and analysis of architectures and protocols for ultra-high-speed networks, grid and cloud networks, and green networking. He has published more than 120 peer-reviewed journal and conference papers. Dr. Vokkarane is the co-author of a book, "Optical Burst Switched Networks," Springer, 2005. He is currently on the Editorial Board of IEEE/OSA Journal of Optical Communications and Networking and Springer Photonic Network Communications Journal and has also served as an Editor of IEEE Communications Letters and Elsevier Journal of Optical Switching and Networking. He has co-authored several Best Paper Awards, including the IEEE GLOBECOM 2005 and IEEE ANTS 2010. He has served as the Technical Program Committee Chair for the Optical Networks and Systems symposia at ICCCN 2007 and 2010, GLOBECOM 2011, ICC 2012, INFOCOM High-Speed Networks workshop 2011, and IEEE ANTS 2013 and 2014. He is currently serving as the General Vice-Chair for IEEE ANTS 2015.