Noname manuscript No.
(will be inserted by the editor)

Avoidance of Multicast Incapable Branching Nodes for Multicast Routing in WDM Networks

Fen Zhou · Miklós Molnár · Bernard Cousin

Received: date / Accepted: date

Abstract In this paper, we study the multicast routing problem in all-optical WDM network under the spare light splitting constraint. To implement a multicast session, several light-trees may have to be used due to limited fanouts of network nodes. Although many multicast routing algorithms have been proposed in order to reduce the total number of wavelength channels used (***total cost***) for a multicast session, the maximum number of wavelengths required in one fiber link (***link stress***) and the end-to-end delay are two parameters which are not always taken into consideration. As we know, the shortest path tree results in the optimal delay, but it can not be employed directly for multicast routing in sparse splitting WDM networks. Hence, we propose a novel wavelength routing algorithm, which tries to avoid the multicast incapable branching nodes (MIB, branching nodes without splitting capability) in the shortest path based multicast tree to diminish the ***link stress*** and maintains good parts of the shortest path tree to reduce the end-to-end delay. The given algorithm consists of tree steps: (1)DijkstraPro algorithm with priority assignment and node adoption is introduced to produce a shortest path tree with up to 38% fewer MIB nodes in NSF topology and 46% fewer MIB nodes in USA Longhual Topology. (2)critical articulation and deepest branch heuristics are used to process the MIB nodes, (3)distance based reconnection algorithm is proposed to create the multicast light-trees. Extensive simulations demontrate its efficiency in terms of ***link stress*** and end-to-end delay.

Keywords Multicast Routing · Sparse Splitting · Light-Tree Computation · WDM Networks · DijkstraPro

The preliminary version of this work was presented in IEEE LCN 2008, Canada.

Fen Zhou
IRISA / INSA de Rennes, Campus de Beaulieu, Rennes 35042, France
Tel.: +33-299847194
Fax: +33-299847171
E-mail: fen.zhou@irisa.fr

Miklós Molnár
IRISA / INSA de Rennes, Campus de Beaulieu, Rennes 35042, France

Bernard Cousin
IRISA / University of Rennes 1, Campus de Beaulieu, Rennes 35042, France

1 Introduction

Multicast is a very efficient way for one-to-many or many-to-may communication. A multicast session typically involves a source and a set of destinations. In traditional data networks, usually, a multicast tree rooted at the source is constructed with branches spanning all the destinations to accommodate a multicast session. In order to be able to multicast data in WDM optical networks, optical switch needs to have splitting capability. Note that optical switches with light splitters are always much more expensive to build than those without. Consequently, only a few nodes can support splitting, which is termed as sparse light splitting [1]. Hence, multicast routing in WDM optical networks is greatly different from that in traditional data networks and one must consider the constraint on splitting capability of nodes in a practical optical network. To implement multicast in all-optical WDM network, the light-tree [2] concept was proposed. A light-tree is a set of consecutive lightpaths. Without wavlength conversion, the same wavelength should be kept along all the links in a light-tree. This is called the wavelength continuity constraint [3].

Moreover, two light-trees or lightpaths sharing a common link should be assigned with different wavelengths. This is named as the distinct wavelength constraint [3]. Due to these physical constraints, supporting multicast routing in all optical network is a challenging work.

For multicast routing in WDM optical networks, many multicast light-tree formation algorithms have been proposed to reduce the total number of wavelength channels used(i.e., total cost), but the maximum number of wavlengths required in one fiber link (i.e., link stress) and the end-to-end delay are also two very important factors, which should be taken into account, especially for the time sensitive and bandwidth intensive multicast applications such as HDTV, VoIP and Video Conference. It is known that if a message is transmitted via the shortest path from a source to a destination, then the delay is minimal. Unfortunately, most of the nodes cannot support slitting and the splitting nodes are very rare in optical networks due to its high cost and complex architecture. If most of destinations communicate with the source through their shortest path, there is a big probability that many of the shortest paths will traverse the same node without splitting capability. Then, more wavelengths should be utilized and the link stress will be even high. From the point of view of link stress, shortest paths cannot be used for all the destinations, and some destinations could find longer paths to the source. While, from the point of view of delay, longer paths should not be used for all destinations also. So, a tradeoff should be found between the link stress and the delay in order to obtain the best general performance.

In this paper, a multicast routing algorithm considering sparse light splitting, which tries to avoid the multicast incapable branching nodes in the multicast light-tree, is proposed to resolve the wavelength routing problem in WDM optical networks. It aims to reduce both the link stress and the delay. The significant aspects of this paper lie at: (i) DijkstraPro algorithm with priority assignment and node adoption is introduced to construct a shortest path tree with fewer multicast incapable branching nodes; (ii) critical articulation and deepest branch heuristics are used to process the MIB nodes with the purpose of reducing both the link stress and delay; (iii) distance based reconnection algorithm is proposed to create the multicast forest with smaller delay while keeping the same link stress and cost.

2 Related Work

The difficulty of the multicast routing in WDM networks with sparse light splitting has been addressed in many papers[4–6, 9, 10, 14] and different algorithms has been proposed. There are mainly three main categories according to the routing approach they employ: Source-Based Routing (e.g., Reroute-to-Source, Reroute-to-Any and Member-First [4]), Steiner-Based Routing (e.g., Me-mber-Only [4] and Virtual-Source Capacity-Priority algorithm [5]) and Core-Based Routing (e.g., Virtual Source based algorithm [6, 7]). Essentially, the Source-Based Routing approach constructs the multicast tree by connecting the source to each destination individually using the appropriate shortest path in order to minimize the per source-receiver path cost. The objective of the Steiner-Based Routing schemes, however, is to minimize the overall cost of the multicast tree. The core structure, connects a subset of nodes, called core nodes, which have both light-splitting and wavelength conversion capacities. The multicast session is then established with the help of this core structure [6, 7]. As far as we know from literature [4], in the all optical network with sparse splitting and without wavelength conversion, Member-Only algorithm can get the approximate minimal cost, while Reroute-to-Source algorithm yields the optimal delay.

In Reroute-to-Source, firstly a multicast tree is generated to span all the destinations, by pruning the shortest path tree built by the Dijkstra algorithm. Then, it checks the light splitting capability for each node in the multicast tree. If a node is a branching node with splitting capability, then no modification is needed. But, if it is a multicast incapable branching node (i.e., it has at least two direct children while has no splitting capability), then only one direct child could be kept, which is chosen arbitrarily. And all the other children should be connected to the source through the reverse shortest path each with a different wavelength. It is obvious that the average delay of the Reroute-to-Source is minimal. However, the stress of the link is very high; because different downstream branches of a multicast incapable branching node should be connected to the source using the same shortest path on different wavelengths. In fact, there may find some longer paths to reach the source using the same wavelength.

In Reroute-to-Any, first it also constructs a multicast tree by pruning unnecessary nodes in the shortest path tree for all the nodes in the network. Then, for each multicast incapable branching node, one downstream branch is kept and the others are cut. Finally, the cut destinations can be connected to multicast tree via a multicast capable node or a leaf multicast incapable node in the tree. Although its link stress and total cost are better than the Reroute-to-Source and its average delay is superior to Member-only, it is still not satisfying and should be improved in order to adapt the QoS required traffic. It seems no algorithm has been proposed to decide which branch of the multicast incapable branching node should be kept and what kind of reconnection algorithm can be used to reconnect the cut destinations.

In Member-Only algorithm, at each iteration, the nearest destination is added to the multicast tree using the shortest path. But, this shortest path should not include any non-leaf multicast incapable nodes in the tree. It is a modification of Takahashi-Matsuyama heuristic [8,9]. Although its total cost is approached to the optimal one, there is a big possibility that most of the destinations are connected to the source via a node far away from the source. As a result, its delay is big and the diameter of the multicast tree is always very large.

The rest of the paper is organized as follows. Section 3 formulates the wavelength routing problem in sparse splitting WDM network and gives some necessary definitions. The multicast routing algorithm based on avoidance of multicast incapable branching nodes is proposed and simulated respectively in section 4 and 5. Finally, a summary of results is made in section 6.

3 Multicast Routing Under Sparse Light Splitting Constraint

3.1 System Model and Problem Formulation

An all-optical WDM network is considered. The network nodes equipped with light splitters are assumed to be sparse because of their complex architecture and expensive cost, normally below 50% [13]. Furthermore, the costly wavelength converters are not available. Without lack of generality, the splitting capability of a multicast capable node is assumed to be infinite by supposing the proper usage of optical amplifiers [15]. A spare light splitting WDM network can be modeled by an undirected graph G (V, E, c, d). Each node v ∈ V is either a multicast incapable node (without splitting capability) or a mulitcast capable node(equipped with light splitters). Each edge e ∈ E is associated with two weight functions c(e), d(e). c(e) means the cost of fiber link e, and d(e) denotes the propagation delay in fiber link e. Both of them are additive along a lightpath LP(u, v). We consider the arrival of a multicast session ms(s, D), which requires to set up a simultaneous communication from the source s to a group of destinations D. Due to the sparse splitting constraint together with the wavelength continuity constraint, one lingt-tree may not be sufficient to cover all the destinations. Assume k light-trees LT_i(s, D_i) should be built for a multicast session ms(s, D) where i ∈ [1, k] and D_i denotes the set of destinations exclusively served in the i^{th} light-tree. Since these k light-trees are not edge disjoint, different wavelengths should be assigned for each light-tree. Thus the number of wavelengths required for ms(s,D) (i.e., link stress) equals to the number of light-trees built.

$$Stress[ms(s, D)] = k \quad (1)$$

And the total number of wavlength channels used (i.e., total cost) for ms(s,D) can be calculated as

$$c[ms(s, D)] = \sum_{i\in[1,k]} \sum_{e\in LT_i(s,D_i)} c(e) \quad (2)$$

Nowadays multimedia services such as HDTV, VoIP, Video Conference and Video on Demand are largely used in Internet. They are delay sensitive and bandwidth intensive. Consequently, the link stress and the delay are two important parameters for the multicast tree in WDM optical networks. When the link stress is very high, fewer wavelengths can be used for the other multicast sessions. That means the bandwidth for other multicast sessions is limited.

Besides this, power loss and noise are two other challenging problems in all-optical networks. Although power loss can be compensated by appropriate placement of all-optical amplifiers in fibers and cross-connects, noise coming with amplification can be cascaded and is hard to clear without electronic processing. It is practical to limit the length of a path (equals to its delay) in order to decrease the number of amplifiers [10]. In addition, the optical network is more and more used in the Internet Backbone. Although optical messages are transmitted from the source to the destination at a very high speed, the nodes in WDM optical networks are distributed over the world. In this case, the end-to-end delay cannot be negligible especially for the delay sensitive traffic. Let LP(s,d_i) be the lightpath between the source s and the destination d_i in the light-trees built for a multicast session m(s,D), we define the average end-to-end delay and the maximum end-to-end delay as follow:

$$AverDelay[ms(s, D)] = \frac{1}{|D|} \sum_{d_i\in D} \sum_{e\in LP(s,d_i)} d(e) \quad (3)$$

$$MaxDelay[ms(s, D)] = \max_{d_i\in D} \sum_{e\in LP(s,d_i)} d(e) \quad (4)$$

What is more, the end-to-end delay and link stress cannot be minimized simultaneously. If the pruned shortest path tree is used as the multicast light-tree, although its delay is optimal, its link stress is very high. If approximated Sterner tree is employed as the multicast light-tree (using Member-Only algorithm [4]), although its link stress is good, its delay cannot be tolerant. So, a tradeoff should be found between them. In order to minimize the end-to-end delay, shortest path tree could be used to construct a multicast tree with optimal delay. In order to reduce the link stress, the MIB nodes in the shortest path tree could be diminished by making some destinations communicate with the source using longer paths. Based on this main idea, an avoidance of MIB nodes based multicast routing algorithm in WDM network with sparse light splitting is proposed in this paper. To simplify the objective metrics, the cost and delay function of each edge are assumed to be equal and without loss of generality they are given as

$$c(e) = 1 \ unit\ cost \qquad d(e) = 1 \ unit\ delay \tag{5}$$

3.2 Useful Definitions

Before the description of our proposed multicast routing algorithm, some necessary definitions are introduced below.

Definition 1: MI and MC nodes
MI nodes: Multicast incapable nodes are nodes which cannot split, but have TaC [12] capability. That is to say, it can tap a small amount of optical power from the wavelength channel while forward it to only one output link.
MC nodes: Multicast capable nodes are nodes which are equipped with light splitters, which permit to split the incoming message to all the outgoing ports. Without special statement, a MI node is denoted by a rectangle while a MC node is denoted by a circle in the figures of this paper except in the topologies.

Definition 2: Multicast Incapable Branching Node (MIB node)
MIB nodes have no splitting capability, but lead to several downstream branches in a the multicast light-tree. Its out degree in the multicast light-tree is no less than two. Once it forwards the message to one branch, it could not forward it to another branch using the same wavelength.

Definition 3: Set MC_SET, MI_SET and D
For a multicast light-tree under construction,
MC_SET: includes the MC node and the leaf MI nodes in the current multicast light-tree. They may be used to span the current multicast light-tree.
MI_SET: includes only the non-leaf MI nodes in the current multicast light-tree, which are not able to connect a new destination to the current multicast light-tree.
D: includes unvisited multicast members which are not yet joined to current or the previous multicast light-trees.

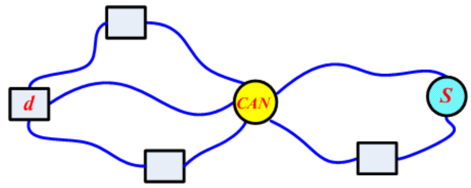


Fig. 1 Critical Articulation Node

Definition 4: Constraint Path (CP) and Shortest Constraint Path (SCP)
The constraint path between a node u and a tree T is a shortest path from node u to a node v in the MC_SET for T. And, this shortest path could not traverse any node in the MI_SET for T. That is:

$$\begin{aligned} CP(u,T) = \{&p(u,v) | v \in MC\_SET, and \\ &\forall x \in p(u,v), x \notin MI\_SET\} \end{aligned} \tag{6}$$

Definition 5: Connection Constraint Node (CC node) and Critical Articulation Node (CAN)
If node u is a CC node, there must be an intermediate node, which is included in all the paths from u to the source. This intermediate node is called the critical articulation node: CAN(u, s). In other words, u could not reach the source without it. For example, in Figure 1, node AN separates the network into 2 parts: node d and source s are in different parts. Without node AN, d is not able to communicate with s. So d is a CC node, and node AN is the CAN(d, s).

4 Avoidance of MIB Nodes for Multicast Routing

The avoidance of MIB nodes based multicast routing algorithm can be viewed as a post-processing of the shortest path tree (SPT). Just because of the MIB nodes in a shorest path tree, one wavelength may not be sufficient to cover all the destinatons and thus several wavelengths may be required to accommodate the multicast group. Based on this observation, they should be avoided in order to decrease the link stress. If there is no MIB node in the shortest path tree, then the shortest path tree is an optimal multicast light-tree with both minimum delay and minimum link stress. Otherwise, some process should be done on the MIB nodes. This algorithm mainly consists of three steps: the shortest path tree construction step, MIB nodes process step and the multicast tree or forest reconstruction step. In the first step, an enhanced DijkstraPro algorithm is introduced to construct a multicast tree with fewer MIB nodes and smaller link stress, which makes use of the priority method and node adoption. In the second step, the MIB nodes in the shortest path tree are processed, where the deepest branch and the critical articulation heuristics are proposed to keep only one downstream branch of the MIB nodes aiming to reduce both the link stress and delay. In the last step, distance based reconnection algorithm is presented to create the multicast forest, which can also reduce the delay.

Algorithm 1 Avoidance of MIB Nodes for Multicast Routing

Require: A multicast session ms(s,D)
1: Using DijkstrasPro to constuct the shortest path tree SPT rooted at the source s. Prune all the non-destination leaf nodes and the nodes which do not lead to any destinations.
2: Using the Deepest Branch and the Critical Articulation Heuristics to process the MIB nodes in SPT.
3: Using distance based light-tree reconnection algorithm to create the required set of light-trees for ms(s,D).

4.1 Construction of SPT and DijkstraPro Algorithm

First of all, a shortest path tree rooted at the source is constructed for all nodes in the network. Then, according to the multicast session, non-destination nodes and the nodes that don't lead to any destination should be pruned from this shortest path tree.

Generally, Dijkstra's algorithm is employed to build the shortest path tree. In Dijkstra algorithm, a node is said to be labeled permanently [11] if its shortest path to the source is found. Otherwise it is said to be tentatively labeled [11]. Initially, only the source s is permanently labeled and all the other nodes are tentatively labeled. In each iteration, the node with the shortest distance to the source among all the tentatively labeled nodes is chosen and labeled permanently. What is worth noting is that, in one iteration, there may be several nodes that have the same shortest distance to the source, here we call them as *candidate nodes* and the distance is named as their *level*. However, according to the Dijkstra algorithm, we should label only one of the candidate nodes permanently in order to update the distances of the other nodes. But, how to choose it? In traditional Dijkstra algorithm, it is chosen arbitrarily. Think about this situation: there are two candidate nodes at the same level; one is a MI node and another is a MC node; they share the same two adjacent nodes. If the MI *candidate* node is firstly chosen to be permanently labeled, then the two adjacent nodes will update their distances to the source, and thus will be connected to the source via this MI *candidate* node. The problem is that the MI *candidate* node cannot split the incoming signal into more than one outgoing port. As a result, it will become a MIB node in the shortest path tree. In contrary, if the MC *candidate* node is firstly chosen to be permanently labeled, then those two adjacent nodes update their distances to the source also, and thus they will be connected to the source via this MC *candidate* node. Subsequently, the MI candidate node is chosen to be permanently labeled. At this moment, no adjacent node needs to update its distance and no adjacent node is left to be connected to the source via this MI *candidate* node. So, the risk for a MI *candidate* node to become a MIB node is reduced or even avoided.

Due to the constraint on splitting capability, the traditional Dijkstra algorithm may not yield a favorable result. It could be improved and some modifications are required. Hence, DijkstraPro algorithm with priority and node adoption is presented. When building the shortest path tree using Dijkstra, in case of several *candidate* nodes at the same *level*:

- *Giving Higher Priority to MC Candidate Node*
  The *candidate* node with multicast splitting capability (MC *candidate* node) should be given higher priority than the MI *candidate* nodes. Since, they can connect as many destination nodes as possible to the tree without producing any MIB node. In other words, the possibility for a MI *candidate* node in latter iterations to connect more than one destination to the tree is greatly decreased.
  Look at the NSF network in Figure 2, node 1, 8 and 10 are assumed to be MC nodes. A multicast session comes: m_1= {source: 10 | members: 1 ∼ 14}. If Dijkstra is used, then we can get the shortest path tree in Figure 3. There are 2 MIB nodes in this shortest path tree. But, we can see that, node 1, 6, 7, 9 and 13 have the same shortest distance to the source node 10. So, they can be viewed as *candidate* nodes at the same *level.* And, if node 1 (MC node) is offered higher priority and firstly chosen to be permanently labeled, followed by 7, 9, 13 and 6, then we can get a new shortest path tree in Figure 4, which has only one MIB node.

- *Giving High Priority to MI Candidate Node with Smaller Degree*
  Moreover, if there is no MC *candidate* node, then the *candidate* node with smaller degree is given a higher priority. That is because, the possibility for a MI *candidate* node with smaller degree (especially for the *candidate* nodes which has a degree of two) to be a MIB node is very low. That is to say, the number of nodes left, which should be connected to the source through other MI *candidate* nodes

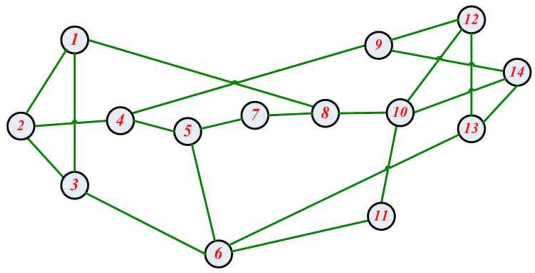


Fig. 2 NSFNET Topology

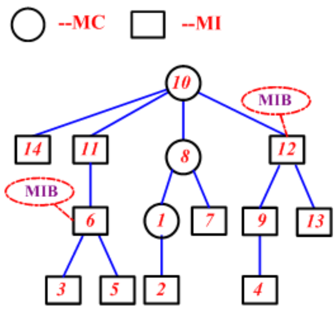


Fig. 3 The SPT for m_1 constructed by Dijkstra

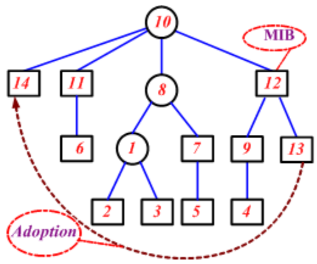


Fig. 4 The SPT for m_1 constructed by offering higher priority to MC *candidate* nodes

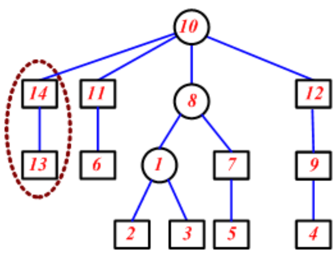


Fig. 5 The SPT after Node Adoption from Figure 4

with higher degree, is very small. Consequently, the possibility for a *candidate* node with higher degree to become a MIB node is reduced. So, the average probability for a node to be a MIB node is slightly decreased.

– *Node Adoption*
Just at the moment that all *candidate* nodes at the same *level* have been permanently labeled, the following situation may occur. Some MI *candidate* nodes connect only two direct children to the tree (i.e., MIB *candidate* nodes) while some *candidate* nodes are leaf nodes in the created tree. So, why doesnot the leaf *candidate* node adopt one child from the MIB *candidate* node at the same *level*, if this child can reach the source through the leaf *candidate* node also? By doing so, this MIB node could be avoided. Through node adoption between the *candidate* nodes at the same *level*, the number of MIB nodes in the shortest path tree can be greatly reduced or the load of a MIB node can be balanced. Normally, a destination node should be given a higher priority to be adopted.
Still see the example in Figure 4. It is obvious that node 11, 12 and 14 have the same least distance to the source node 10. Hence, they can be viewed as candidate nodes. After all of they have been permanently labeled, we can see node 12 is a MIB node and node 14 is a leaf node. Note that, node 13 or 9 can reach the source node 10 by the shortest path through both of node 12 and 14. Thus, one of them could be adopted by node 14, and a new shortest path tree without MIB node is obtained in Figure 5.

4.2 Processing of the MIB nodes

Due to the fact that the MIB nodes in the shortest path tree can forward the incoming message to one and only one outgoing branch, the existence of the MIB nodes is the most important cause of high link stress. So, they should be processed and avoided. In Reroute-to-Source algorithm [4], all downstream links of MIB nodes are connected to the source through the reverse shortest path on different wavelengths, which result in high link stress. Although Reroute-to-Any algorithmis also proposed in literature [4], there is no detail about how to keep one branch when processing the MIB nodes. So, in this paper, the deepest branch and critical articulation heuristics are employed to decide which branch should be kept in order to decrease the link stress and the delay.

4.2.1 MIBPro

– *Critical Articulation Heuristic*
A CC node u can only communicate with the source through its CAN(u, s). In a multicast tree, if CAN(u, s) is unfortunately a MIB node, then the branch containing u should be assigned a higher priority and kept, when processing this MIB node. Since, there is no alternative path for u to reach the source without traversing its CAN(u, s). However, destinations

Algorithm 2 Processing of MIB nodes Using Critical Articulation and Deepest Branch Heuristics

```
Search all the MIB nodes in the shortest path tree
for each MIB node do
    if No downstream branch contains a CC node then
        Keep the deepest branch
    else if Only one downstream branch contains a CC node
    & MIB node = CAN(CC, s) then
        Keep the branch with the CC node
    else if Several downstream branches contain CC nodes
    & MIB node = CAN(CC_i, s), i =1, 2, then
        Keep the deepest branch with a CC node
    end if
end for
Delete the downstream branches of MIB nodes which are not
kept
```

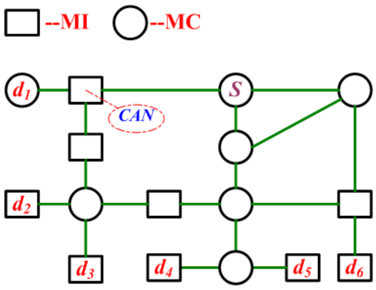


Fig. 6 An example network with a CAN node

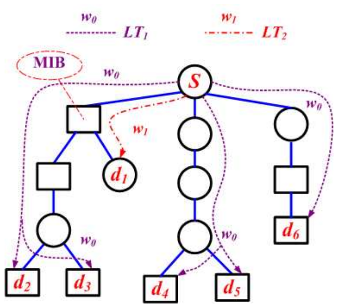


Fig. 7 A Shortest Path Tree for m_2

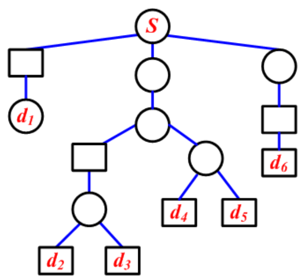


Fig. 8 Process of MIB nodes using the Critical Articulation Heuristic

in the other branches may find another path to the source, which will not traverse this MIB node. In fact, CC and CAN(CC, s) nodes are very rare in a real optical network. However, in case that some nodes in the network have failed, they may exist, and this heuristic will be very practical. In the network of Figure 6, node d_1 is a CC node. The shortest path tree for multicast session $m_2 = \{$source: s | destinations: $d_1 \sim d_6\}$ is given by Figure 6. We can see CAN(d_1, s) is a MIB node in the shortst path tree built for m_2 as plotted in Figure 7. Hence it should be processed. If disconnect node d_1 from CAN(d_1, s) and keep the branch leading to node d_2 and d_3, then two light-trees on two different wavelengths w_0 and w_1 are required as shown in Figure 7. But, if the CC node d_1 is kept and cut the other one, then only one light-tree (*or one wavelength*) is needed as shown in Figure 8.

– *Deepest Branch Heuristic*
The deepest branch should also be assigned a higher priority. This is because it seems difficult for a destination far away from the source to find another path to the source without traversing any non-leaf MI node in the tree. On the other hand, the delay for a destination node far away from the source will be limited by the length of its shortest path to the source, which is very useful to reduce the average delay. To implement this step, breadth-first traversal algorithm can be employed. So, the worst case time complexity is O(N), where N is the number of nodes in the network.

4.2.2 MIBPro2

In addition, another method is also proposed to process the MIB nodes in the shortest path tree. For all the MIB nodes, it suggests delete all the downstream branches without employing any heuristic. These two methods will be compared in section 5.

4.3 Reconnection of Multicast Light-trees

After the MIB nodes processing step, the shortest path tree is divided into a subtree plus several separated destinations, which is a disconnected forest and should be reconnected in order to accommodate all the multicast members. The Member-Only [4] like algorithm is a very good method to reconnect the multicast forest. It always adds the nearest destination to the multicast light-tree using the shortest path, but this shortest path should not use any non-leaf MI node in this light-tree. That is to say, in each iteration, only the destination with the shortest SCP is connected to the tree through

Algorithm 3 Distance Based Light-trees Reconnection Algorithm

```
T ← subtree obtained after MIB process
MC_SET ← {MC nodes and leaf MI nodes in T}
MI_SET ← {non-leaf MI nodes in T}
D ← {destinations not in T}
while D ≠ Φ do
    repeat
        Find the closest destination d ∈ D to T, and its shortest path to T should not traverse any node in MI_SET
        if there are several destinations satisfying equation 7 then
            Select the destination nearest to the source in T as d
        end if
        if there are several connector nodes for d in MC_SET satisfing equation 8 then
            Select the connector node nearest to the source in T as c and Choose the corresponding SCP
        end if
        T ← T ⋃ SCP(d, c)
        MC_SET ← MC_SET ⋃ {d and MC-nodes on SP(d, c)}
        MI_SET ← MI_SET ⋃ {non-leaf MI-nodes on SP(d, c)}
        D ← D \ d
        if c= MI then
            MC_SET ← MC_SET \ c
            MI_SET ← MI_SET ⋃ {c}
        end if
    until no destination can be added to T
    return T
    Begin a new tree T ← {s}
    MC_SET ← {s}
    MI_SET ← φ
end while
```

$$dist\{SCP(d,T)\} = \min_{d_i \in D} dist[SCP(d_i,T)] \quad (7)$$

$$dist\{SCP(\mathbf{d},T)\} = dist\{SP(d, connector_i)\}, i = 1, 2, \ldots \quad (8)$$

its SCP. To the best of our knowledge Member-Only algorithm can get the best link stress and the minimum cost. However, its delay is very large. What is worth noting that some improvements can be done to reduce the end-to-end delay in some extent while obtaining the same cost and the same link stress. The example below will explain how to improve the end-to-end delay.

For instance multicast session m_3 = {source: 10 | destinations: 6, 11, 13, 14} is required in NSF network in Figure 2. We assume the first tree only contains the source node 10. According to the member-only like method, the destination with the shortest SCP should be added to this tree firstly. The shortest path for node 11 to source and that for node 14 to the source have the same length 1. Without loss of generality, node 14 is chosen to be connected. Then, on the new tree, we can see that both SCP for node 11 and 13 have the same length. Also without loss of generality, node 13 is chosen to be connected. After that node 6 is chosen, finally

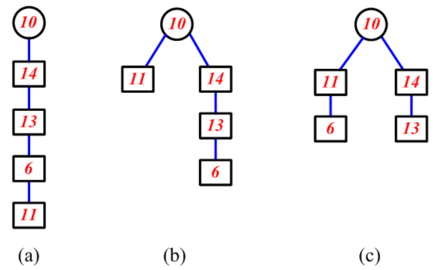


Fig. 9 Two hints for the reconnection of light-trees

node 11 is chosen. Following these steps, the multicast tree can be gotten in Figure 9(a). But it is not difficult to notice that node 11 can be connected to the tree via node 10 or node 6. Why don't we connect it through node 10 like in Figure 9(b)? The difference is that the connector node has different distances to the source (for node 10 the distance is 0 while that for node 6 is 3). In addition, it is even more interesting to see the Figure 9(c). All of these three multicast trees have the same cost of 4 while have different average delays: 10/4, 7/4 and 6/4. It is also easy to find that, after node 14 is added to the tree, if node 11 is added to tree earlier than node 13 then, we can get the result in Figure 9(c).

So, from this simple example, we can get two hints in order to reduce the average delay while maintaining the same cost and the same link stress. From this point of view, distance based reconnection algorithm is developed. If there are several nodes, whose SCPs to the multicast tree have the same length, and then these nodes should be added in the order of their distance to the source (the distance in the network): the nearer, the earlier. What is more, when the destination with the shortest SCP has at least two connector nodes in the subtree, it is better to use the connector node nearest to the source (the distance in the multicast light-tree under construction). Otherwise its delay will be too long.

5 Performance Evaluation and Simulation

Optical fibers, which can provide high bandwidth, are always used in the backbone of Internet. To ensure the effectiveness of our proposed multicast routing algorithm, two different network topology are employed as test beds for the simulation:the 14 nodes NSF network Figure 2 as well as the 28 nodes USA Longhaul network in Figure 10. These networks has been used as a reference topology in many papers [5,7,10,14,16,17], that is why we select them.

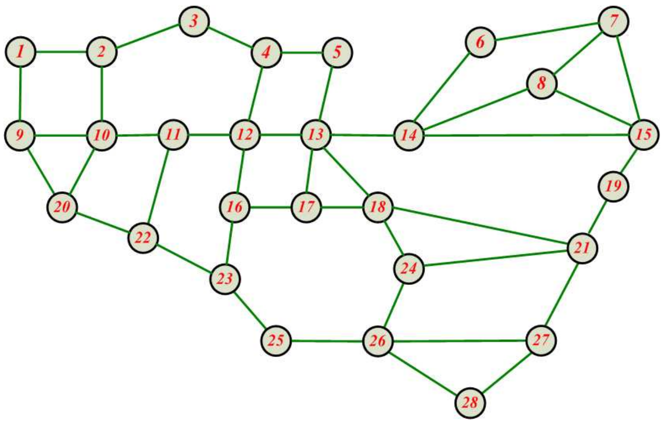


Fig. 10 USA Longhaul Network

5.1 Performance of DijkstraPro Algorithm

To shown the improvement of the DijkstraPro algorithm, it is compared with the traditional Dijkstra algorithm using the following two parameters:

- N: the number of the MIB nodes in the shortest path tree.

- S: the maximum number of wavlength required in one fibe link to conver all the destinations in the shortest path tree (i.e., link stress of the SPT).

In each comparison, two conditions are considered. Condition 1 only regards the source as a MC node, and Condition 2 regards the nodes with high connectivity as MC nodes. The reason why we choose these two conditions can be explained as follow. In Condition 1, as only the source is a MC node , MC *candidate* node priority is not fonctional. Thus, the result in Condition 1 can check the merit of the node adoption operation in DijkstraPro algorithm. As stated in [18,19], it could a method to place the splitters at the nodes with high connectivity. Hence, the nodes with high connectivity are treated as MC nodes in Condition 2. In this condition, the MC *candidate* node priority is operational, and the overall performance can be verified.

In Table 1, we evaluate the performance of 14 shotest path trees rooted at each node of the NSF network. Source ID denotes the root of the shortest path tree built. Two conditions are considered:

Condition 1 (only the source is a MC node)
The average number of MIB nodes in the shortest path tree constructed by DijkstraPro algorithm is 0.85 less (23%) than the traditional Dijkstra algorithm and the link stress is 0.36 smaller. The result in this condition verifies that the node adoption op-

Table 1 Comparision of Dijkstra and DijkstraPro in NSFNET in Figure 2

SPT in NSF	Condition1 MC: source Members: 1 ∼ 14				Condition2 MC: 6,10, and source Members: 1 ∼ 14			
Source ID	Dijkstra N	Dijkstra S	DijkstraPro N	DijkstraPro S	Dijkstra N	Dijkstra S	DijkstraPro N	DijkstraPro S
1	3	4	3	3	1	2	1	2
2	4	3	3	3	3	3	2	3
3	3	4	2	4	2	2	1	2
4	4	3	3	2	4	3	3	2
5	4	4	2	3	3	2	1	2
6	3	2	3	2	3	2	3	2
7	5	5	4	4	3	3	2	2
8	4	4	2	3	3	2	1	2
9	4	3	3	3	3	3	2	3
10	3	2	1	2	2	2	1	2
11	3	4	3	4	1	2	1	2
12	4	3	4	3	2	2	1	2
13	3	4	2	4	2	2	1	2
14	4	3	4	3	2	2	1	2
Average	3.64	3.43	2.79	3.07	2.43	2.29	1.5	2.14

Table 2 Comparision of Dijkstra and DijkstraPro in Longhual Network of Figure 10

SPT in Longhaul	Condition1 MC: source Members: 1 ∼ 28				Condition2 MC: 10,12 ∼ 15, 18, 21, 26 and source Members: 1 ∼ 28			
Source ID	Dijkstra N	Dijkstra S	DijkstraPro N	DijkstraPro S	Dijkstra N	Dijkstra S	DijkstraPro N	DijkstraPro S
1	6	8	5	6	2	3	1	2
2	6	7	5	6	1	2	0	1
3	8	9	6	7	2	2	2	2
4	8	9	5	6	2	2	1	2
5	9	8	5	6	2	3	1	2
6	6	8	3	5	2	2	1	2
7	5	6	3	5	2	2	1	2
8	4	7	2	5	1	2	1	2
9	5	9	5	6	0	1	0	1
10	7	10	4	6	1	2	0	1
11	6	9	5	7	0	1	0	1
12	7	6	5	6	3	2	1	2
13	6	5	3	3	1	2	1	2
14	3	7	2	5	1	2	1	2
15	6	6	3	5	2	2	1	2
16	6	6	6	6	1	2	1	2
17	6	6	5	5	1	2	1	2
18	4	6	3	4	0	1	0	1
19	8	8	3	4	2	2	0	1
20	6	9	4	4	2	3	1	2
21	7	7	3	4	2	2	0	1
22	5	5	5	5	2	2	2	2
23	7	6	6	6	4	3	2	2
24	4	5	5	5	0	1	0	1
25	6	5	6	6	4	5	3	4
26	7	6	5	4	4	4	2	3
27	6	8	4	7	1	2	0	1
28	7	5	6	6	3	4	2	3
Average	6.11	7.0	4.36	5.36	1.71	2.25	0.93	1.82

eration in DijkstraPro algorithm can really work.

Condition 2 (node 6, 10 and the source are MC nodes) In NSF network, both node 6 and node 10 have a high degree of 4), so they can be assumed to be MC nodes which are very useful for multicast sessions. The DijkstraPro algorithm can also produce a shortest path tree with fewer MIB nodes and smaller link stress. The average number of MIB nodes is 0.93 less (38%) and the link stress is 0.15 smaller.

In Table 2, we also give the performance of 28 shotest path trees rooted at each node in the USA Longhaul network.

Condition 1 (only the source is a MC node) DijkstraPro algorithm results in 1.75 (29%) fewer MIB nodes in average than the traditional Dijkstra algorithm. And the link stress of the shortest path tree built by DijkstraPro is 1.64 smaller. This signifies that the function of node adoption operation is independent of the network topology.

Condition 2 (node 10, 12 15, 18, 21, 26 and the source are MC nodes) In the USA Longhaul network, node 10, 12 15, 18, 21 and 26 have a degree equal to or above 4, so they regared as the MC nodes in this condition. the DijkstraPro algorithm can also produce a shortest path tree with fewer MIB nodes and smaller link stress. The average number of MIB nodes is 0.78 less (46%) and the average link stress is 0.43 smaller.

Moreover, it is easy to think out that when all the nodes in a WDM network are MC nodes, all the shortest path tree constructed by the Dijkstra or the DijkstraPro algorithm will not have any MIB nodes and their link stress are always 1. So, it is obvious that, when the ratio of MC nodes in the network is very high the improvement of DijkstraPro algorithm is not significant. But when the MC nodes are very sparse its performance is much better than the traditional Dijkstra algorithm not only in term of the number of MIB nodes but also in terms of the link stress. This is why we introduce DijkstraPro algorithm in the construction of shortest path tree step for the implementation of our proposed multicast routing algorithm.

5.2 Performance of Avoidance of MIB Nodes based Multicast Routing Algorithm

There is no literature which describes Reroute-to-Any [4] algorithm keeps which branch of MIB nodes and which algorithm is used to reconnect the cut destinations. In our simulation, arbitrary branch is assumed to be kept and Member-Only [4]like reconnection method is employed in Reroute-to-Any algorithm.

To evaluate the performance of the proposed avoidance of MIB nodes based multicast routing algorithm (MIBPro/MIBPro2), the following four metrics are used to measure the quality of the set of the multicast light-trees built for a multicast session.

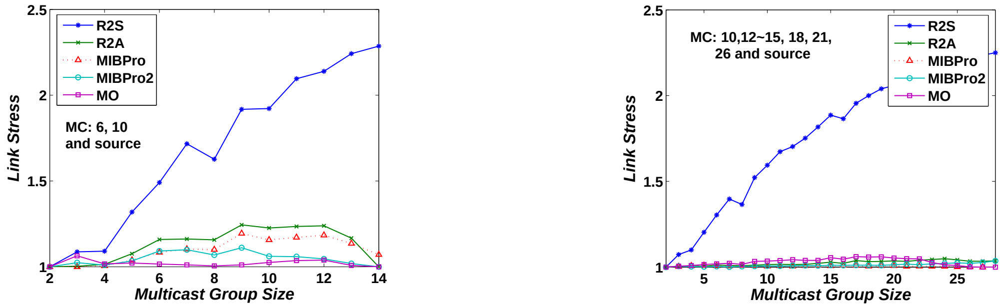
- link stress
- average end-to-end delay
- maximum end-to-end delay
- total cost

In addition, each multicast session has only one single source. Each node in the network is selected as the source of a multicast session once in sequence. The destinations of a multicast group is distributed independently and uniformly through the network. For a given source and a given multicast group size, 100 random multicast sessions are generated. Hence, the result of each point in the simulation figures is the average of 100×|V| computations. In addition, Reroute-to-Source (R2S), Reroute-to-Any (R2A) and Member-Only (MO) are also implemented for the comparison.

5.2.1 Effect of Group Size (Number of Multicast Members)

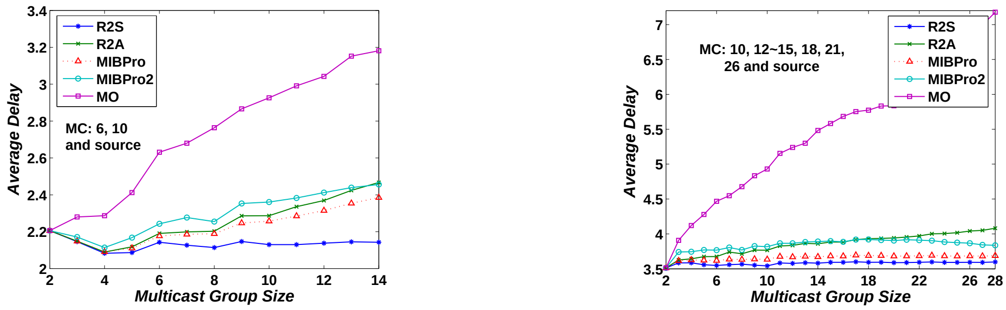
Here, we study the performance of the proposed algorithm against the multicast group size. As mentioned in subsection 5.1, the nodes with high connectivity have a big probability to be a MC nodes. To simplify the simulation in this part, we regard these nodes as MC nodes and change the group size to evaluate the quality of light-trees build by MIBPro multicast routing algorithm.

In NSF network, nodes 6, 10, and the source are set as MC nodes. The simulation results in NSF network are ploted in Figures 11-14(b). As shown in Figure 11(a), we can see when the group size is above 4, MIBPro achieves better link stress than R2A. And the link stress of MIBPro2 is much smaller than MIBPro. But the performance of R2S in term of link stress is always the worst one. Figures 12(a) and 14(a) show that the average end-to-end delay and maximum end-to-end delay of MIBPro is only second to the optimal one R2S. As the multicast group size grows, the improvement of end-to-end delay by MIBPro compared to R2A become more and more significant. Moreover, the total costs of R2A, MIBPro and MIBPro2 are almost the same but R2S results in the highest one while MO results in the lowest one.



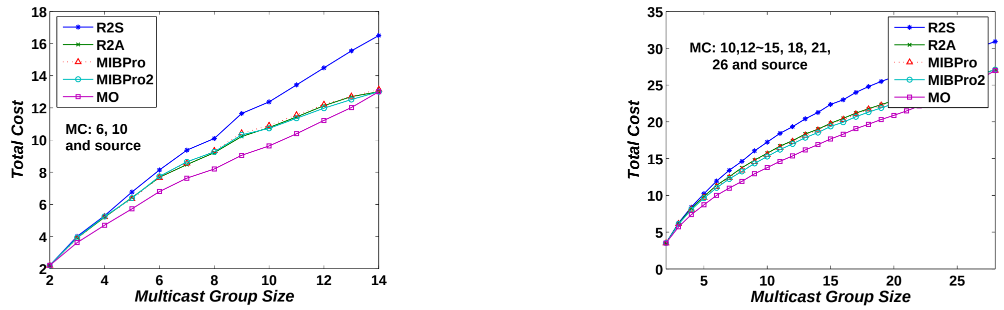
(a) (b)

Fig. 11 Link Stress against the Multicast Group Size in (a)NSF Network(b)Longhaul Network



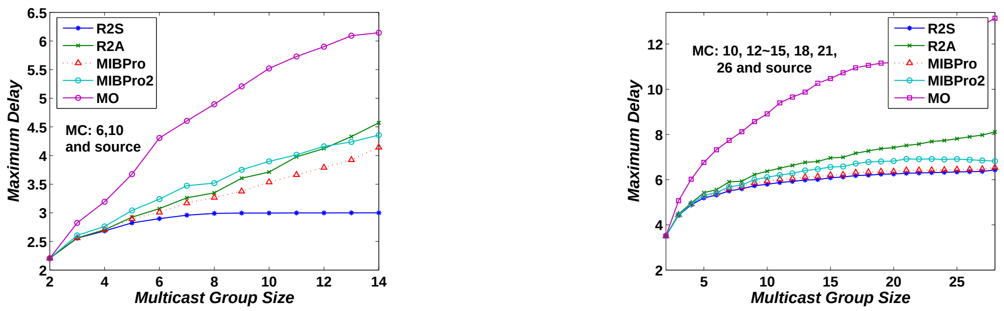
(a) (b)

Fig. 12 Average Delay against the Multicast Group Size in (a)NSF Network(b)Longhaul Network



(a) (b)

Fig. 13 Total Cost against the Multicast Group Size in (a)NSF Network(b)Longhaul Network



(a) (b)

Fig. 14 Maximum Delay against the Multicast Group Size in (a)NSF Network(b)Longhaul Network

In USA Longhaul network, nodes 10, 12 15, 18, 21, 26, and the source are set as MC nodes. Figure 11-14(b) have compared the performances of those five algorithms in USA Longhaul network. The link stress of five algorithms are almost the same and very near to 1 regarding Figure 11(b). This is because, the ratio of MC nodes is very high (32%) in this configuration. The end-to-end delay for MIBPro algorithm is very close to the optimal one R2S. And to our surprise, MIBPro obtains almost the same maximum end-to-end delay as R2S. From the point of view of the total cost, R2A, MIBPro and MIBPro2 get the same value, which is same case in NSF network.

From the simulation results above, we can see MIBPro algorithm can get nearly the same or a litter better link stress than R2A. Its reduction in average and maximum end-to-end delay compared to R2A becomes more obvious when the group size is large. This is because the MC nodes priority mechanism, node adoption and distance based reconnection are not able to operate when the group size is too small. Only when there are enough destinations, these strategies work well.

5.2.2 Effect of Splitting Capability (Number of MC nodes)

The performances when the number of MC nodes varies are also studied. According to the results gotten in the previous part, MIBPro is more advantageous when the multicast group size is large. This is why we set the multicast group size to big value and change the number of MC nodes in the simulation of this part. And the MC nodes are assumed to be independently and uniformly distributed in the topology. The multicast group size is set to 12(14 nodes in total) in NSF network and set to 21(28 nodes in total) in USA Longhaul network. The numeric results are plotted in Figures 15-18. According to these figures, as the MC nodes are saprse, (1) MIBPro achieves mumber better performance in terms of link stress, average end-to-end delay and maximum end-to-end delay than R2A while get the same cost as R2S. (2)MIBPro2 results in both lower link stress and smaller cost than R2A. Its link stress is even better than MO in Longhaul network. However, its end-to-end delay is sometimes or sometimes worse than R2A.

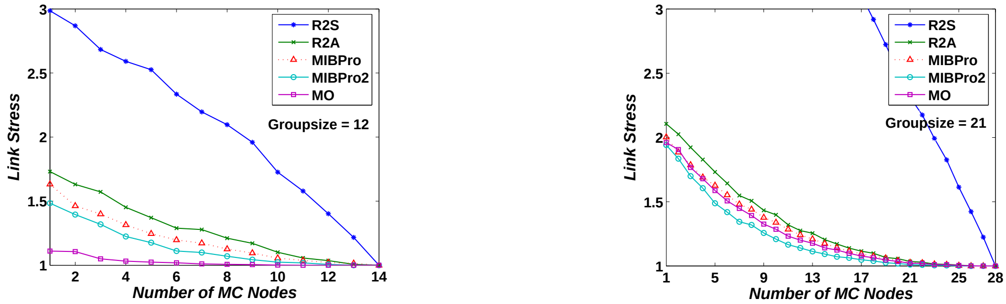
These results indicate our proposed MIBPro algorithm works well in case of spare light splitting. And when the ratio of MC nodes becomes big, MIB nodes in the shortest path tree is fewer. As a result, the advantage of MIBPro will be less significant.

6 Conclusion

Due to the physical constraints, supporting multicast routing in optical network with spare light splitting is not easy. Many multicast routing algorithms have been proposed to attempt to construct the Steiner based light-tree in all-optical networks to get the minimum cost, but this problem is proved to be NP-hard. In fact, QoS required applications become more and more popular in Internet nowadays. The bandwidth (or number of wavelengths supported per fiber link in WDM networks) and then end-to-end delay are two important parameters for QoS. Hence, a multicast routing algorithm based on avoidance of MIB nodes is presented for QoS required traffic in WDM networks in order to decrease the link stress and delay. It keeps the good parts of the shortest path tree which results in the optimal delay for if not may at least some multicast members. In order to reduce the number of MIB nodes and link stress in the construction of the shortest path tree step, DijkstraPro algorithm is presented, where higher priority is assigned to MC *candidate* node and node adoption are conducted between the *candidate* nodes at the same *level*. To keep one branch of MIB nodes in the shortest path tree, critical articulation and deepest branch heuristics are introduced. And the distance based reconnection algorithm is also developed to rejoin the multicast light-forest. The first part of the simulation in section 5 shows that DijkstraPro algorithm is a better tool for the shortest path tree construction in all-optical network than the traditional Dijkstras algorithm. It can really reduce the MIB nodes and link stress of the shortest path tree. Moreover, the second part of the simulation proves that, the proposed MIBPro algorithm yield a good performance in term of link stress when MC nodes are very sparse. In addition, when the group size is large it is able to improve the average end-to-end delay and the maximum end-to-end delay a lot, which are very close to the optimal solution Reroute-to-Source algorithm [4].

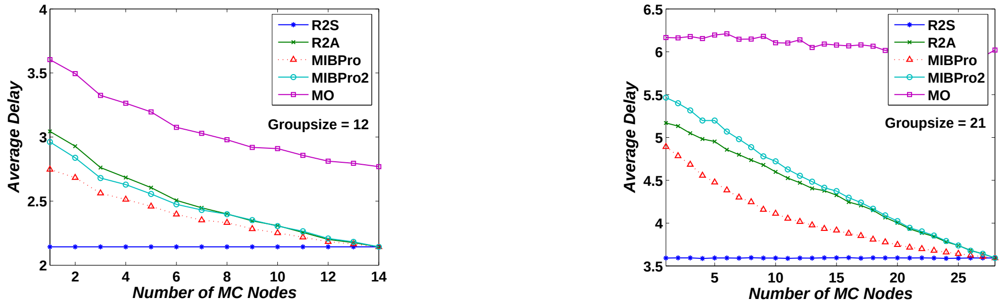
References

1. R. Malli, Xijun Zhang, Chunming Qiao. Benefit of Multicasting in all-optical networks. SPIE Proceeding on All-Optical Networking, 2531:209-220, 1998.
2. L. H. Sahasrabuddhe, B. Mukherjee. Light-tree: Optical multicasting for improved performance in wavelength-routed networks. IEEE Communication. Magazine, 37: 67-73, 1999.
3. Biswanath Mukherjee. WDM optical communication networks: Progress and challenges. IEEE Journal on Selected Areas in Communications, 18(10):1810-1824, 2000.
4. Xijun Zhang, John Wei, Chunming Qiao. Constrained Multicast Routing in WDM Networks with Sparse Light Splitting. Journal of Lightware Technology, 18(12):1917-1927, 2000.



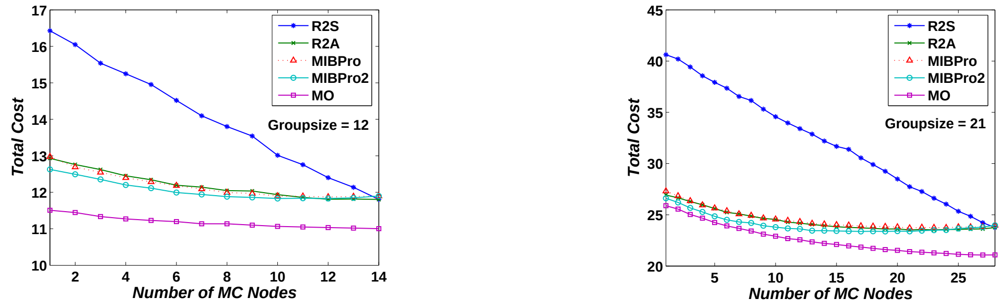
(a) (b)

Fig. 15 Link Stress against the Number of MC Nodes in (a)NSF Network(b)Longhaul Network



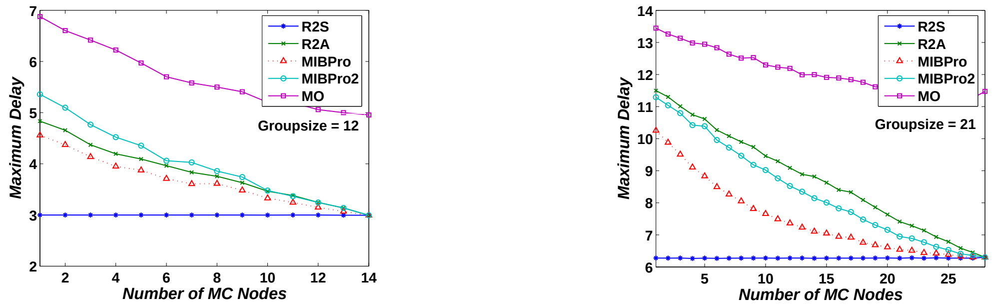
(a) (b)

Fig. 16 Average Delay against the Number of MC Nodes in (a)NSF Network(b)Longhaul Network



(a) (b)

Fig. 17 Total Cost against the Number of MC Nodes in (a)NSF Network(b)Longhaul Network



(a) (b)

Fig. 18 Maximum Delay against the Number of MC Nodes in (a)NSF Network(b)Longhaul Network

5. N. Sreenath, K. Satheesh, G. Mohan, C. Siva Ram Murthy. Virtual Source Based Multicast Routing in WDM Optical Networks. Photonic Network Communications, 3(3): 213-226, 2001.
6. N. Sreenath, K. Satheesh, G. Mohan, C. Siva Ram Murthy. Virtual Source Based Multicast Routing in WDM Networks with Sparse Light Splitting. IEEE Workshop on High Performance Switching and Routing'01, p141-145, 2001.
7. Ashraf Hamad, Tao Wu, Ahmed E. Kamal, Arun K. Somani. On multicasting in wavelength-routing mesh networks. Computer Networks, 50: 3105-3164, 2006.
8. H. Takahashi, A. Matsuyama. An approximate solution for the Steiner problem in graphs. Math. Japonica, 24(6): 573-577, 1980.
9. Aniko Zsigri, Alexandre Guitton, Miklos Molnar. Construction of light-trees for WDM multicasting under splitting capability constraints. International Conference on Telecomunication'03, p171-175, 2003.
10. Shuguang Yan, Jitender S. Deogun, Maher Ali. Network Routing in sparse splitting optical networks with multicast traffic. Computer Network, 41: 89-113, 2003.
11. C. Siva Ram Murthy, Mohan Gurusamy. WDM Optical Networks: concepts, design and algorithms. Prentice Hall, 2002.
12. Maher Ali, Jitender S. Deogun. Cost-Effective Implementation of Multicasting in Wavelength-Routed Networks. IEEE Journal of Lightwave Technology, 18(12): 1628-1638, 2000.
13. Maher Ali, Jitender S. Deogun. Allocation of Splitting Nodes in All-Optical Wavelength-Routed Networks. Photonic Network Communication, 2(3): 247-265, 2000.
14. Fen Zhou, Miklós Molnár, Bernard Cousin. Avoidance of Multicast Incapable Branching Nodes for Multicast Routing in WDM Networks. The 33^{rd} IEEE International Conference on Local Computer Network, pp336-344, Montréal, Canada, October, 2008.
15. E. Desurvire. Erbium-Doped Fiber Amplifiers: Principles and Applications. Wiley, New York, 1991.
16. Fen Zhou, Miklós Molnár, Bernard Cousin. Distance Priority Based Multicast Routing in WDM Networks Considering Sparse Light Splitting. The 11^{th} IEEE International Conference on Communication System, pp709-714, Guangzhou, China, November, 2008.
17. Zhenghao Zhang, Yuanyuan Yang. On-Line Optimal Wavelength Assignment in WDM Networks with Shared Wavelength Converter Pool. IEEE Transaction On Networking, 15(1):234-245, 2007.
18. Hwachun Lin, Shengwei Wang. Splitter Placement in All-Optical WDM Networks. In proceeding of IEEE GlobalCom, p306-310, 2005.
19. Abdur Billah, Bin Wang, Abdul A. S. Awwal. Topology based Placement of Multicast Capable Nodes for Supporting Efficient Multicast Communication in WDM Optical Networks. Photonic Network Communication, 14:35-47, 2007.

Fen ZHOU was born in HeFei, P. R. China in 1982. He received the B.S. and M.S. degree in Communication and Information System respectively in 2004 and 2007, from Northwestern Polytechnical University in P. R. China. Currently, he is preparing for his PhD degree in the lab IRISA Rennes (Institut de Recherche en Informatique et Systèmes aléatoires), France. His research interest includes multicast routing in optical networks and constrained routing.