An Improved Multicast Routing Algorithm in Sparse Splitting WDM Networks

Miklós Molnár, Dinh Danh Le, Jérôme Palaysi

HAL Id: lirmm-00807299

https://hal-lirmm.ccsd.cnrs.fr/lirmm-00807299

Submitted on 3 Apr 2013

An Improved Multicast Routing Algorithm in Sparse Splitting WDM Networks

Dinh Danh Le
LIRMM
Université Montpellier 2, France
Email: dinhdanh.le@lirmm.fr

Miklós Molnár
LIRMM
Université Montpellier 2, France
Email: miklos.molnar@lirmm.fr

Jérôme Palaysi
LIRMM
Université Montpellier 2, France
Email: jerome.palaysi@univ-montp2.fr

Abstract—In this paper we study the multicast routing problem in all-optical WDM networks with sparse splitting capacity. This problem have been attracted a lot of attentions by the researchers worldwide due to its challenges and interest. Most of the work makes use of light-trees (or light-forests) to solve the problem. The objective focuses mainly on minimizing the network resources, e.g. the maximum number of wavelengths (the link stress), the number of wavelength channels used (the total cost), or the end-to-end delay from the source to the destinations (the delay). However, archiving multiple objectives is not trivial. For this reason, we propose a comparative study of the most known algorithms and introduce a new one which can provide a good trade-off among those three criteria. Simulation results and comparison point out that our proposal produces multicast light-forests with the lowest link stress, low total cost and a low end-to-end delay among considered algorithms. Especially, our proposal is more advantageous in dense networks, and/or with a large multicast group size in comparison to the classical algorithms.

Keywords: all-optical WDM networks, sparse splitting, light-tree, light-forest, multicast routing.

I. INTRODUCTION

Wavelength Division Multiplexing (WDM) network can provide huge bandwidth (about Tb/s) and low communication latency. Under WDM, each fiber consists of several communication channels at different wavelengths. Different channels can only use the same wavelength if they do not share any common fibers [7].

Multicasting in WDM networks relates to transmitting optical signal from a single source to multiple destinations concurrently. It is getting more important in today core Internet for the increasing number of high performance applications concerning several destinations, such as video and phone conferencing, distance e-learning, online television, etc. Multicasting at WDM layer can be realized by creating a *light-tree* or a *light-forest* with the support of splitters, i.e. Multicast Capable Optical Cross-Connects (MC OXCs, or MC in short) [9], the optical devices that are capable of splitting incoming light signal to several copies for outgoing ports. Because several common links can be used, WDM muticasting is bandwidth effective.

However, the implementation of WDM multicasting is challenging for several optical constraints. The first constraint comes from sparse light splitting capacity, i.e. not all nor none of the nodes but a subset of them can be MC nodes. Many studies showed that, about below 50% of the nodes are multicast capable (the remanings are multicast incapable (MI) nodes), is enough to support effeciently multicasting [1], [2]. This constraint makes WDM multicasting challenge, because the heterogeneous networks (with the presence of both types of MC and MI nodes) require more sotisphicated designs.

Moreover, although the number of commercially available wavelengths/fiber is in the hundreds, the exponential growth of Internet traffic every year (about 40% according to [5]) poses the requirement of utilizing economically wavelengths in order to serve as many communication requests as possible.

In addition, wavelength conversion (WC), the device that is capable of switching one wavelength to another, is expensive and not mature enough to deploy widely in reality [3]. Without WC's support, edge-joint light-paths or light-trees can not be assigned with the same wavelength, imposing the *distinct wavelength constraint* [7]. In our study, we make the assumption of no WC presented in the networks. It means that the distinct wavelength constraint must be respected.

In this paper, we investigate the well-known algorithms for constructing multicast light-forests proposed in [8] in the consideration of optical constraints mentioned above. Previous studies show that these algorithms are not always effective in term of balancing multiple criteria (i.e. cost, delay and link stress). We therefore recommend a new algorithm that can provide a better trade-off solution. The rest of this paper is organized as follows. We first review the studies related to WDM multicast routing in sparse splitting networks in Section II. In Section III, the problem is modelled and performance metrics for the evaluations are also introduced. In Section IV, we focus on our proposed algorithm and describe it in detail. The simulation results with evaluations are showed in Section V before the conclusion.

II. RELATED WORK

Our research is carried out in sparse splitting WDM networks without WC as supposed in many studies [1], [2], [4], [8], [9]. In addition, for simplicity, we also assume that any MC nodes in the networks are *complete* splitting capapble, i.e they can split incoming light signal to any of their outgoing ports. In this section, we review the four well-

known multicast routing algorithms proposed in [8], including Reroute-to-Source (Re2S), Reroute-to-Any (Re2A), Member-First (MF) and Member-Only (MO).

The first two algorithms (i.e. R2S and R2A) are based on *Shortest Path Tree* (SPT). First, a spanning tree is created from a source to all destinations by employing a shortest path algorithm (e.g. Dijkstra's algorithm). Then, the algorithm checks the light splitting capability of each branching node in the SPT. If the number of its children is greater than its splitting capacity, only some children are kept, and the other must be re-routed. For instance, if the branching node is a multicast incapable branching node (MIB node), then only one child can be kept, and all the other children (and sub-trees rooted at them) must be re-routed to the current tree either at an MC node along the shortest paths to the source (R2S), or to any other node on the tree if possible (R2A). Obviously, the end-to-end delay of R2S is minimized. However, the number of wavelengths (and hence, the total cost) can be very high, because downstream branches of an MI node have to connect to the source on different wavelengths. In contrast, the link stress resulted from R2A is less, but the end-to-end delay is higher compared to R2S.

On the Steiner Tree Approximation Based approach, Member Only begins to build a multicast light-tree by connecting the destinations to the tree one by one just through the shortest paths (the closest, the first). At each step, it tries to find the shortest path from the destinations (yet to be included) to the current multicast light-tree so that the shortest paths do not traverse any MIB nodes. If it is found, the corresponding destination and the path are added to the light-tree. Otherwise, the current tree is terminated and a new one on another wavelength is started until all the destinations have been covered. According to [8], the light-forest computed by MO have the minimum total cost. However, because the distance from the destinations to the source is not taken into account, many destinations might be connected to the light-tree via a node far away from the source. Consequently, the diameter (and hence, the end-to-end delay) of the multicast tree is often very high.

Member-First algorithm is also based on SPT while taking membership information into consideration. The algorithm manages adjacent fringe links and constructs the light-tree iteratively by adding the highest priority link at each step. We will describe MF in more details in Section IV. According to [8], MF achieves a better link stress and cost in comparison to R2A, and produces a good trade-off among performance metrics compared to the other algorithms.

In short, R2S and MO algorithms focus on optimizing one metric (delay or cost) without considering others, so the improvement of them is not trivial. Besides, MF outperforms R2A, and especially, its framework (i.e. the incremental construction of light-trees using fringe links) is promising to design efficient algorithms. Thus, we select MF to design a more efficient routing algorithm. In the next sections, our proposed algorithm is described in detail, analysis and comparison between the aforementioned algorithms are also given.

III. PROBLEM MODELLING AND PERFORMANCE METRICS

The network is modelled by a pair (G, S) where G is a connected undirected graph and S is a subset of vertices of G representing the MC nodes (V_G presents network nodes and E_G presents network links[1]). Any link $e \in E$ is associated with a cost $c(e)$ and a propagation delay $d(e)$. We consider (s, D) as the multicast session triggered from the source node s to the set of destinations $D = \{d_1, d_2, ..., d_n\} \subset V_G$, $s \notin D$. Let $S \subset V_G$ be the set of MC nodes in the network. Because of the sparse splitting capacity of the network, it is likely that a single light-tree may not be sufficient to span all the destinations. Therefore, we assume that k light-trees will be built composing a light-forest $F = \{LT_i, i = 1, \ldots, k\}$. Since these k light-trees are not edge-disjoint, they must be assigned with different wavelengths (due to the *distinct wavelength constraint*). Hence, the number of wavelengths required for the multicast session (s, D) is equal to the number of light-trees in F, and it is called *link stress*. (We also suppose that there are enough wavelengths for any multicast session.) Accordingly, the first performance metric is defined as $LinkStress(F) = k$.

Another metric, the *total cost* is the sum of cost of all the links on the light-trees of the forest F:
$$TotalCost(F) = \sum_{i \in [1,k]} \sum_{e \in LT_i} c(e).$$
Besides, although optical fibers can support a very high speed in optical networks, there is a propagation delay on each link. When the network size becomes large, the additive end-to-end delay will be considerably along the long distance. Thus, end-to-end delay is a metric that should be considered. Let LP_{s,d_i} be the light-path from the source s to destination d_i. Two metrics relating to end-to-end delay (i.e. *maximum delay* and *average delay*) can be calculated as follow:
$$MaxDelay(F) = \max_{d_i \in D} \sum_{e \in LP_{s,d_i}} d(e)$$
$$AvgDelay(F) = \frac{1}{|D|} \sum_{d_i \in D} \sum_{e \in LP_{s,d_i}} d(e)$$
With all the notations mentioned above, the problem can be stated as follows:

- **Input:** A connected undirected graph G, a set of MC nodes S, a multicast session (s, D)
- **Output:** A light-forest F for the multicast session (s, D) satisfying the constraints: the leaves of the light-trees are destinations and the branching nodes are in S.
- **Objective:** A trade-off solution among following criteria: *link tress*, *total cost*, and *end-to-end delay*.

As many studies, for simplicity in calculating performance metrics, we assume that all the wavelengths have the same quality, the bandwidth consumed using a wavelength on different links is the same as well ([8], [9]). In addition, the propagation delay is also the same on each link. With this assumption, for every link e, $c(e) = 1$ *unit cost* and $d(e) = 1$ *unit delay*.

---

[1]For any graph G, we denote V_G the set of its vertices and E_G the set of its edges.

IV. MEMBER-SPLITTER FIRST ALGORITHM

Before describing our proposed algorithm, we first give some more details of Member First algorithm, then analyse it for possible improvements.

A. Member First algorithm

This heuristic constructs light-trees incrementally by examining the possible fringe links of the trees (a *fringe link* is a link adjacent to the tree without forming a cycle with its own edges). A priority is associated with each fringe link. The tree construction begins at the source and can be briefly described as follows:

1: Initialize the fringe link set L with the adjacent links of the source
2: Select the fringe link with the highest priority from L
3: Add the selected link (and its end-node n) to the tree and remove it from L
4: Remove impossible multiple children of MI nodes from n back to the source
5: Update L with the adjacent links of n
6: If there are fringe links in L, go to step 2
7: Prune branches that do not lead to any destinations
8: If there are destinations not yet covered, then go to step 1 to construct another tree

Keeping the basic sequence of MF as a framework, the algorithm can be improved to construct more favourable light-trees by changing the priority of fringe links and the procedure to update them as described in the following.

B. Priority Definition

One of the primary factors that affect a lot on algorithm's performance is the priority of links in the fringe link set. In MF, the priority of a link (v, u) can be briefly represented as the order of (*h, member*), in which h is the number of hops from the source to u, and *member* stands for the multicast membership of u. Accordingly, the link (v_i, u_i) have higher priority than (v_j, u_j) if $h(u_i) < h(u_j)$, or when $h(u_i) = h(u_j)$, u_i is a member, but u_j is not [8].

However, MF does not regard to MC nodes available in the network. Thus, it leaves a question that when all the factors *(h, member)* are the same for two fringe links, which link should be given higher priority. Obviously, in such a case, the link leading to an MC node should be the first choice because MC nodes can connect to many of its children (probably including destinations) with only a single wavelength. From these children, the connection to other destinations can be more likely. Thus, we propose to give higher priority to links leading to MC nodes. In fact, the idea to give priority to MC nodes have already been proposed in [4], where spanning trees composed from MC nodes that are computed first, then the destinations are connected to the MC trees. The solution uses few wavelengths but can contain long paths (e.g. when most MC nodes are not multicast members) and the end-to-end delay can be high because the distance form the source to a destination is not taken into account.

Moreover, the degree of the connected nodes also has another effect. By simulation we find out that, giving higher priority to the nodes with higher degree can lead to lower end-to-end delay [6]. Indeed, the probability that a high degree node can lead to other destinations not yet spanned is higher. However, high degree MI nodes are likely to produce more MIB nodes in the resultant light-tree, resulting in more wavelengths needed to cover all the destinations. That means the link stress can be high. In contrast, smaller degree MI nodes probably induce higher delay but lower link stress [6]. From this fact, we give higher priority to links leading to the higher degree MC nodes, and then smaller degree MI nodes (when they have the same h and *member*). By that way, the end-to-end delay and the link stress can be balanced.

Combination of these aforementioned elements, the new priority of fringe links can be defined in the order of: *h, member, MC, degree*. In particular, a link (v_i, u_i) have higher priority than (v_j, u_j) if: $h(v_i) < h(v_j)$; or when $h(v_i) = h(v_j)$, u_i is a *member*, but u_j is not; or when $h(v_i) = h(v_j)$ and if their memberships are the same, u_i is MC and u_j is MI; or when all the above criteria are the same, if both are MC nodes, then u_i has highest degree; otherwise, if both are MI nodes, then u_i has smallest degree. The effectiveness of this combination is verified the by simulations in our research paper [6].

For demonstration, let us consider the network in Figure 1 where the MC nodes are drawn in circles (including 4, 6, 9), and MI nodes in squares (the remaining nodes). Suppose that the multicast session is triggered at node 6 spanning the destination set {1, 2, 3, 5, 7, 8, 10, 11} that are shaded in grey.

We can compare the two light-forests constructed by original MF (Figure 2) and by MF with the new priority definition (Figure 3). After pruning branch (6, 4) (step (6)), the light-forest created by the original MF consists of 2 light-trees (*link stress* = 2), with the total number of branches is 10 (*total cost* = 10), the maximum number of hops from the source to destinations is 3 (*max delay* = 3) and *average delay* = 2.0. In Figure 3, the light-forest created by MF with the new priority definition (*h, member, MC, degree*) is better with one light-tree (*link stress* = 1), *total cost* = 9, *max delay* = 4, and *average delay* = 2.25.
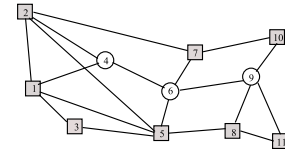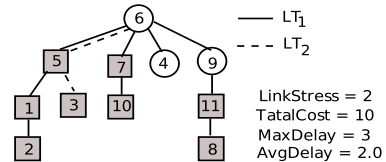

Fig. 1. A network to consider
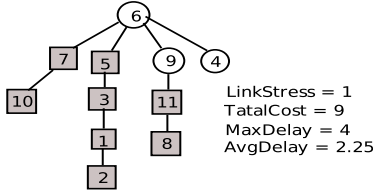

Fig. 2. Member First light-forest

Fig. 3. Member First with New Priority Definition light-forest
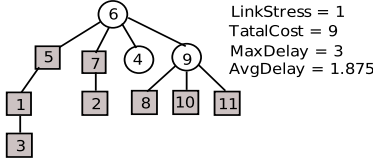

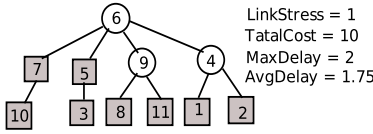Fig. 4. Member First with New UpdatingFL light-forest

Fig. 5. Member-Splitter First light-forest

C. Updating Bud-Links

When updating fringe links (step 5) from node v, MF algorithm adds all the possible adjacent links of v to the fringe link set L regardless to its splitting capacity. It is worth noting that, if v is an MC node, there is no problem. However, if it is an MI node, it can support only one child, if all the possible adjacent links are added, there is no chance for other nodes (if any) connecting to these links. This is one of the key reasons making Member-First inefficient. Consequently, it is more reasonable to add only one link from an MI node, and the selected link must have the highest priority. By this way, when selecting the highest priority link from L in order to add to the multicast tree, we just go straight-forward without any branch-cutting or link-removing. Moreover, there are chances for other adjacent nodes of v to be end-nodes of other fringe links.

To realize this idea, we define a new concept corresponding to fringe links and call them *bud-links*. A bud-link is similar to a fringe link except that, from an MI node there is only one possible bud-link (whereas there may be more than one in term of fringe links). Formally, given a network (G, S), a session multicast (s, D) and a tree T, a *bud-link* is an edge $\{x, y\}$ such that $x \in V_T$ and $y \notin V_T$ and if $x \notin S$ (i.e. x is MI) then the out-degree of x in T is equal to or less than 1. Besides, at any time, a bud-link $\{x, y\}$ is the link with highest priority among possible links in the form $\{x', y\}$. In addition, from an MI node, the only selected bud-link is the link with highest priority among possible adjacent links of it.

Applying the concept of bud-links and the way they are updated to the network in Figure 1, the resultant light-forest including a single light-tree with the total cost of 9, the

maximum delay of 3 and the average delay of 1.875 is created as shown in Figure 4.

D. Member-Splitter First algorithm

Combination of the above improvements results in the new algorithm, and we call it Member-Splitter First, or MSF in short. The resultant forest is shown in Figure 5. As it is shown, this light-forest includes a single light-tree with the total cost of 10, the maximum delay of 2 and the average delay of 1.75. We can easily verify that it is the best trade-off light-forest in comparison to the previous ones.

Our proposal can be described as below:

MEMBER-SPLITTER FIRST ALGORITHM

Input: A network (G, S) and a multicast session (s, D)
Output: A light forest F satisfying (s, D) in (G, S)

1: $F \leftarrow \emptyset$; $G' \leftarrow G$; $D' \leftarrow D$; $\{D'$ is set of destinations yet to be included$\}$
2: **while** $D' \neq \emptyset$ **do**
3:     $T \leftarrow (\{s\}, \emptyset)$
4:     **while** a bud-link exists **do**
5:         Choose a bud-link $e = \{u, v\}$ with highest priority
6:         Add $\{u, v\}$ and v to T
7:         **if** $u \in D'$ **then** $D' \leftarrow D' - \{u\}$
8:         **while** there exists a *dead-vertex*[2] v_d in T **do**
9:             remove v_d and its adjacent edges from T and G
10:         **end while**
11:     **end while**
12:     add T to F
13:     $T' \leftarrow T$
14:     Remove vertices that are leaves of T' and their adjacent edges from T' and G'
15:     **while** there exists a leaf l of T' such that l is also a pendant vertex of G' **do**
16:         remove l and its adjacent edges from T' and G'
17:     **end while**
18: **end while**

In the loop of line 4 the tree T grows as much as possible in the graph G' by choosing a highest priority bud-link. (The bud-link set can be computed incrementally by using a specific procedure for each addition or removal of a vertex of the tree [6].) During this time, the loop of line 8 prunes unnecessary branches of the tree that cannot grow more. Finally, when the tree is constructed and added to the forest, instructions in lines 14-17 remove vertices from G' that are no longer necessary to continue the computation of the forest. An example network and explanation of the execution of the algorithm is shown in Figure 6.

During the execution of the algorithm, the graph G' remains connected, since each removed vertex is either a leaf of a maximal branches of G' or a pendant vertex of G'. Besides, every tree built has at least two vertices (s and a destination). So at least one vertex of G' is removed by instructions in lines

---

[2]*dead-vertex* is a leaf of T that is not a destination nor end-node of any possible bud-links.
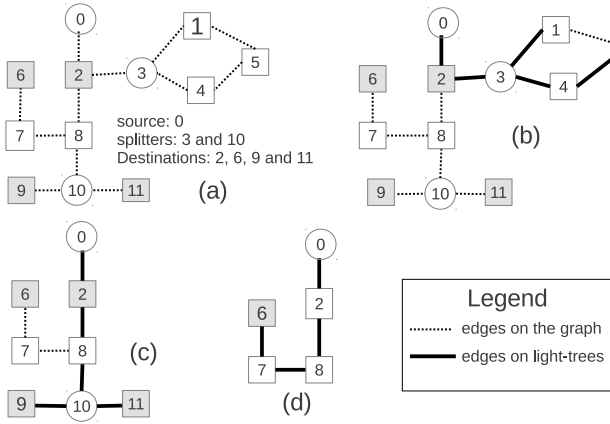
Fig. 6. Figure (a) represents a network and a multicast session. Figure (b) shows the first tree in which vertices 3, 1, 4 and 5 are not destinations. These vertices have been removed by the loop of line 8. Then the bud-link set is re-updated from MI node 2. The actual first light-tree continue to cover 8, 9, 10 and 11 as shown in (c). After that, vertices 9, 11 and then 10 are removed by instructions in lines 14-17. Finally, a second tree including 0, 2, 8, 7 and 6 is created as shown in (d).

14-17 at each turn of the main loop. Since $D' \subset V'_G$ and $|V'_G|$ strictly decreases, it is eventually empty and the algorithm stops.

V. SIMULATION RESULTS

To evaluate the proposed algorithm in comparison with the other classical ones, we carry out simulations with three well-known networks: USA NSF network, USA Longhaul network and European Cost-239 network. Due to the limited space of the paper, we only present the results for NSF network (Figure 7), but the results are quantitatively the same for the other two (the full results can be found in our research paper [6]).

In our simulation, each node of the network is in turn selected as the source for a multicast session. For a given source, a given number of MC nodes and a given number of destinations, 100 random multicast sessions are generated. (The destinations and MC nodes are distributed randomly through the network.) Hence, the result of each point in the simulation figures is the average of $100 \times |V|$ computations on the four performance metrics: *link stress*, *total cost*, *maximum delay* and *average delay*.
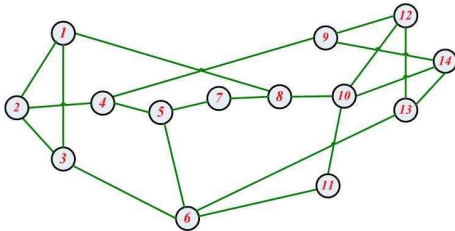

Fig. 7. NSF network topology

Effect of Group Size (the number of destinations)

Firstly, we study the performance of the proposed algorithm versus multicast group size: the number of MC nodes is set

fixed while the group size varies. For the sparse splitting capacity of network, only a few MC nodes can be set.

For 14-node NSF network, the number of MC nodes is set at 3 nodes (23%) and the group size is varied from 1 to 13 nodes. MC nodes and destinations are distributed randomly in the network. The performance metrics are calculated and plotted in Figure 8. When the group size varies, the link stress (Figure 8(a)) achieved by MSF remains flat and close to the value of 1. It is much better compared to MF, especially when the group size becomes large. As it is shown, when the group size is 100%, the difference between the two is maximal at about 50%. In Figure 8(b), the total cost produced by all the algorithms increases when the group size increases. Accordingly, MSF provides a better total cost than MF. When group size gets larger, MSF gets close to the optimal resulted from MO. In Figure 8(c), the maximum delay of MSF (and so is its average delay whose figure is not shown due to the limited space) is lower than MF (as well as Re2A and MO).

In short, with the sparse splitting capacity, when the group size varies, MSF algorithm outperforms MF. Among all the algorithms, it produces the lowest link stress, a low total cost and a low end-to-end delay. Especially, its performance is better when the group size is large.

Effect of Splitting Capacity (the number of MC nodes)

We also study the performance of the proposed algorithm versus the number of MC nodes. The group size is set at 10 nodes (77%) while the number of MC nodes varies from 1 to 13 nodes. The simulation results with NSF network are shown in Figure 9.

According to Figure 9(a), while the link stress of the other algorithms converge to 1 in skew lines from left to right, MSF and MO keep the link stress flat at the value very close to 1, in which MSF is better than MO when the number of MC nodes is very sparse. For the total cost and the delay, (Figure 9(b), 9(c)), while MO achieves the lowest total cost but the highest delay and Re2S, in contrary, produces the lowest delay but the highest total cost, MSF provides a good trade-off between the two metrics when always ranking second in those metrics. Compared to MF, it is always better, and the difference between the two is more obvious when the number of MC nodes is sparse.

VI. CONCLUSION

In this paper the multicast routing problem in sparse splitting WDM networks are investigated and well-known algorithms for light-tree construction and their performance are also reviewed. The comparison results show that the algorithms offer various performance on the important metrics as cost, delay and generated link stress. To ensure a best trade-off among these metrics, we also proposed a new algorithm called Member-Splitter First (MSF). It is based on the framework of the known algorithm - Member First. Our proposal exploited the two very important elements in Member-First: the priority definition of network links and the way of constructing light-tree. MSF associates higher priority to MC nodes and avoids MI nodes with high degree. To compare and evaluate the

(a) Link Stress vs. Group size

(b) Total Cost vs. Group size
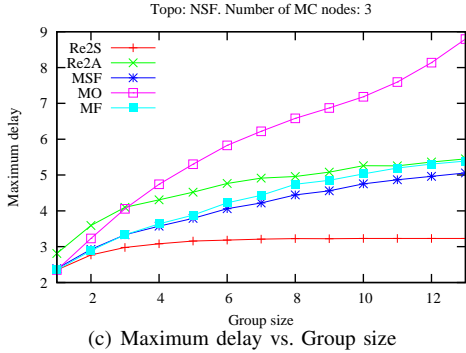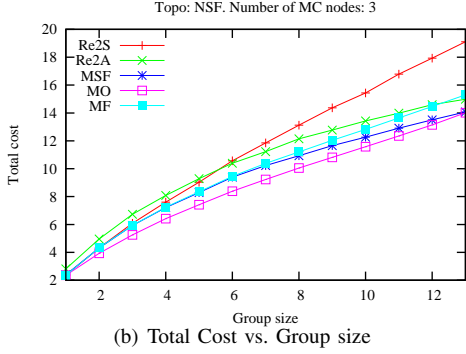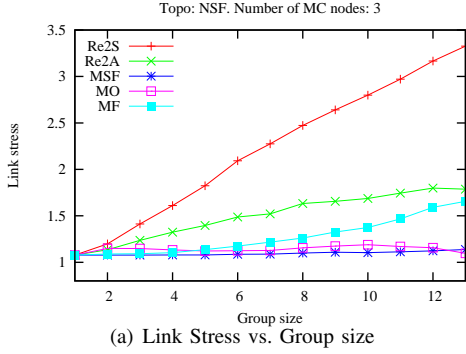
(c) Maximum delay vs. Group size

Fig. 8. Performance of algorithms vs. Group size in NSF network with 3 random MC nodes

(a) Link Stress vs. Number of MC nodes

(b) Total Cost vs. Number of MC nodes
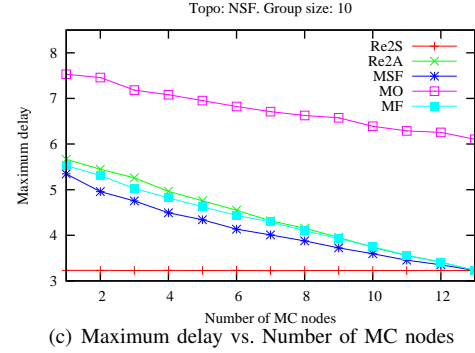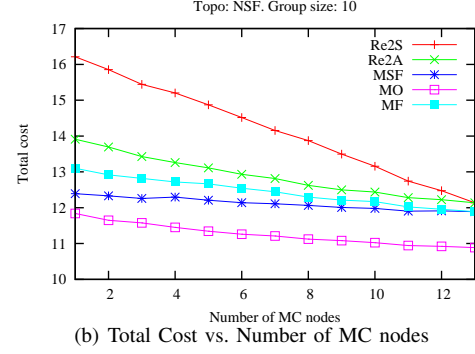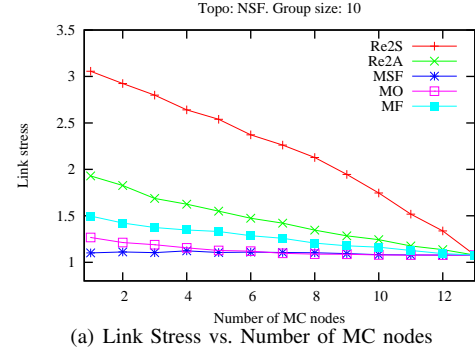
(c) Maximum delay vs. Number of MC nodes

Fig. 9. Performance of algorithms vs. Splitting capacity in NSF network with group size of 10

performance of our proposed algorithm, number of simulations have been carried out with all the algorithms. Simulation results show that MSF outperforms Member-First in all the metrics. Particularly, MSF achieves the lowest link stress (among all the algorithms), a low total cost (ranks second below the optimal produced by Member-Only) and a low end-to-end delay (also ranks second below the optimal resulted from Reroute-to-Source). In general, MSF provides a good trade-off among performance metrics in sparse splitting WDM networks. Especially, it works better with the large number of destinations and in dense networks (e.g. European Cost-239 network [6]).

REFERENCES

[1] Maher Ali and Jitender Deogun. Allocation of Splitting Nodes in All-Optical Wavelength-Routed Networks. *Photonic Network Communications*, 2:247–265, 2000.

[2] Maher Ali, Jitender S. Deogun, and Associate Member. Cost-effective implementation of multicasting in wavelength-routed networks. *EEE/OSA Journal of Lightwave Technology*, 18:1628–1638, 2000.

[3] J. M.H. Elmirghani and H. T. Mouftah. All-optical wavelength conversion: technologies and applications in DWDM networks. *IEEE Communications Magazine*, 38(3):86–92, March 2000.

[4] Cheng-Yu Hsieh and Wanjiun Liao. All-Optical Multicast Routing in Sparse Splitting WDM Networks. *IEEE Journal on Selected Areas in Communications*, 25:51–62, August 2007.

[5] Infinera. Coherent DWDM Technologies. Technical report, http://www.infinera.com, 2012.

[6] Dinh Danh Le, Miklós Molnár, and Jérôme Palaysi. Performance Analysis of All-Optical Multicast Routing Algorithms with Sparse Splitting. Technical report, http://hal-lirmm.ccsd.cnrs.fr/lirmm-00737121, October 2012.

[7] Biswanath Mukherjee. *Optical WDM Networks*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[8] Xijun Zhang, John Y. Wei, and Chunming Qiao. Constrained Multicast Routing in WDM Networks with Sparse Light Splitting. *IEEE/OSA Journal of Lightwave Technology*, 18(12):1917–1927, 2000.

[9] Fen Zhou. *Routage multicast tout optique dans les réseaux WDM*. PhD thesis, Université de Rennes 1, September 2010.