ISSN 2047-3338

Tree Reconfiguration with Network Resources Constraint

Joel Christian Adépo[1], Boko Aka[1] and Michel Babri[2]

[1]Sciences Fondamentales et Appliquées, Université Nangui Abrogoua, Laboratoire de Mathématique et informatique, Abidjan, Côte d'Ivoire
[2]Mathématique et Informatique, INHPB, LARIT, Yamoussoukro, Côte d'Ivoire

***Abstract*– We studied in this paper tree reconfiguration without lightpath interruption in a network wherein resources are limited. In our previous work, we proposed an algorithm called TRwRC. This paper proposes an algorithm which does reconfiguration without ligthpath interruption and which improves the reconfiguration duration in the network wherein resources are limited.**

***Index Terms*– Interruption, Reconfiguration, TRwRC and Constraint**

I. INTRODUCTION

IN a multicast communication [1], [2], one source communicates with several destinations. A multicast connection is represented by the source node and the set of destinations nodes. The path following by the data is a tree rooted at the source node of the connection and spanning all the destinations nodes. In a wavelength division multiplexing (WDM) optical network [3], [4], the tree of a multicast connection is called a light-tree.

Reconfiguration [5], [6] is important to improve network performance. Reconfiguration triggering events can include overload, addition of network resources, deletion of network resources due to a failure or planned network maintenance [5]. In our previous works [7], [8], we studied tree reconfiguration without connection interruption. The formulation of this problem is presented below:

Let $G = (V, E)$ be the network graph; V is the set of nodes and E is the set of edges. A node represents a switch on the physical network and an edge represents a fiber link. Let $T_0 = (V_0, E_0)$ and $T_z = (V_z, E_z)$ be two trees included in G. we assume in this work that the two trees have the same source S and the same set of destinations *Dest_Set*.

We want to move from the tree T_0 into the tree T_z. At each step, we configure simultaneously only one subset of nodes, one subset after the other, as each step must ensure the continuity of communication between the source and each destination. The problem is to determine the subset of nodes to reconfigure at each step such that we produce a sequence of reconfiguration, composed of series of trees included in G: T_0, T_1, …, T_{z-1}, T_z.

Fig. 1 is an example of tree reconfiguration problem. The initial tree (T_0) is red and the final tree (T_z) is blue.

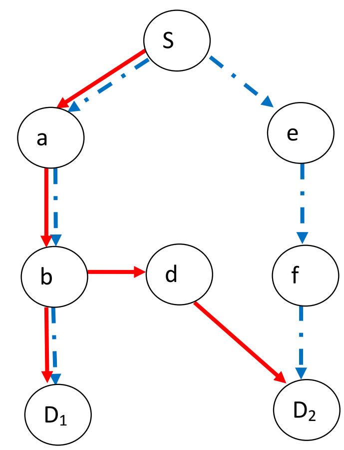

Fig. 1: Tree reconfiguration problem

We proposed good algorithms to reconfigure trees in the previous works. We study in this paper reconfiguration in a network wherein resources are limited. We proposed an algorithm called TRwRC [8] in a previous work. TRwRC reconfigures trees in a network with network resources constraint. In this work, we propose a new algorithm which improve the reconfiguration duration in a network wherein resources are limited. This algorithm called *RCBRwPR*, reconfigure tree in the environment described above and reduce the reconfiguration duration.

II. RELATED WORKS

Path computation and reconfiguration policy are well known problem in literature [9], [12]. [13] studied fast path reconfiguration without data flow interruption. They proposed algorithm for unicast connections reconfiguration.

The existing reconfiguration process in literature is *MBB* (make before break). A path reconfiguration with *MBB* is a well-known lightpath reconfiguration algorithm. Reconfiguration with *MBB* consists to setup the connection on the new path before the old path is torn down. The reconfiguration is done by using the same wavelength during the process. This algorithm will not allow the reconfiguration of a path when the required resources to establish the new path are not available. The extension of *MBB* to tree reconfiguration reconfigure the tree branch by branch. Sine some branches will share some links, the extension of MBB to tree reconfiguration interrupt the connection if some branches share at least one link.

BpBAR_2 and *TRwRC* are tree reconfiguration algorithms proposed in our previous work. They are branch-based algorithms. In this approach, the reconfiguration of a branch is based on the determination of the appropriate switching node. The switching node is the node which maintains the data flow continuity toward all the destinations after its configuration.

*BpBAR_*2 algorithm contains two phase. In the first each branch is reconfigure per stage; a new tree, a copy of the final tree which uses wavelengths different from the initial, is obtained. In the second phase, each branch is reconfigure per stage; a tree, a copy of the final tree which uses the initial wavelength, is obtained.

TRwRC (tree reconfiguration with resources constraint) is an algorithm proposed for reconfiguration problem when the availability of network resources is considered. This algorithm is efficient to solve reconfiguration problems in a network wherein the number of available wavelengths is limited.

TRwRC produces a series of light-forests during the tree configuration process. Each light-forest obtained at each reconfiguration step contains at most two trees: one tree spans the source and the destinations nodes whose branches are reconfigured and the second tree spans the destinations nodes whose branches are not configured. TRwRC configure one branch in one stage. A branch reconfiguration with TRwRC consists of two rounds. In the first round, the branch is established with an additional wavelength (for instance wavelength l_2). In the second round, the branch is established with other additional wavelength (for instance l_1). The function used by TRwRC to reconfigure branch in a stage is called *ReconfBr*. This algorithm need 3 wavelengths per link to reconfigure any problem without interruption.

III. PROPOSED ALGORITHM

In this paper, we propose to improve the reconfiguration duration produced by *TRwRC* in the networks wherein the resources are limited. The proposed algorithm configured in one stage, a set of branches. The branches configured in one stage are selected such as their parallel reconfiguration produced at must two trees in the light-forest obtained at each step. Since the number of branches are known, this process will reduced the reconfiguration duration.

The branches selection function (algorithm 1) called *BrSelection(LT, T_z, SW_nodes)* is described below:

1. Generate a graph G' with the current light-forest (light-forest used to transmit data) and the new path between each candidate switching node and its correspondent destination.

2. If a link of this graph has more than 3 wavelengths (for instance nb_wl, where $nb\_wl > 3$), remove from the graph, $(nb\_wl - 3)$ new paths whose use this link.

3. The new paths which retains in the graph will be reconfigured in one stage. The set of their destination nodes is noted *Dest* and the set of their corresponding switching nodes is noted *Selcted_Sw_Nodes*.

BrSelection(*LT*, T_z, *SW_nodes)*

1. {

2. Generate the graph G'

3. For each link of the graph G' {

4. **If** ($nb\_wl > 3$) {

5. Remove from the graph, $(nb\_wl - 3)$ new paths whose use this link

6. **Goto** 2

7. }

8. }

9. Return *Dest*

10.}

Algorithm 1: Branches selection function

The function used to reconfigure the set of branches in each stage is described in following (algorithm 2). In the function *ReconfBr* used by the algorithm TRwRC, only one branch is configured in a stage. In the proposed function called *BranchesReconfig*, the branches of the selected destination nodes will be configured in one stage. The nodes between each destination node of *Dest* and its switching node are configured in parallel to pre-establish the new branches. Configure in parallel the set of switching nodes to feed the pre-established branches and interrupt the flow on the old. Then remove in parallel the old branches whose flow has been interrupted.

The proposed tree reconfiguration algorithm is called $RCBRwPR$ (algorithm 3). It produced a series of light-forests during the reconfiguration process. Each light-forest contains at most two light-trees. It configures a set of branches in one reconfiguration stage. Firstly it determines the set of candidate switching nodes in the current light-forest. Then it determines the set of branches which can be reconfigured according to the network resources constraint. Then it configures those branches in parallel. In the first round, all the branches are established with an additional wavelength (for instance wavelength l_2). In the second round, the branches are established with other additional wavelength (for instance l_1).

This algorithm reduces the duration of reconfiguration by reconfiguring a set of branches simultaneously. It maintains continuity because it computed the appropriate switching node for each branch.

BranchesReconfig (T0, Tz , Dest,T')

1. {

2. Configure in parallel the nodes between each destination node of *Dest* and its switching node to pre-establish the new branches

3. Configure in parallel the set of switching nodes to feed the pre-established branches and interrupt the flow on the old

4. Configure in parallel the nodes between the switching nodes and the destination node of Dest to remove the old branches remove in parallel the old branches whose flow has been interrupted

5. If (*T_0 and T_z have different topologies*)

6. **{**

7. Configure the source S to interrupt the segment between S and each switching node which is different to S

8. Configure in parallel the nodes between S and each switching node to remove those segments

9. **}**

10. }

Algorithm 2: Function which reconfigure a set of branches in a stage. T_0 is the initial tree and T_z is the final tree

---

RCBRwPR **(T_0, T_z)**

1. {

2. While (*the current tree T' is different to T_z*)

3. **{**

4. Determine all the candidate switching nodes *SW_nodes*

5. $Dest = BrSelction(LT, T_z, SW\_nodes)$

6. *BranchesReconfig (T0, Tz, Dest, T)*

7. *BranchesReconfig (T, Tz, Dest, T')*

8. **}**

9. Configure simultaneously the ports of the nodes of the tree *T'* to convert the wavelength used into the initial wavelength.

10. }

Algorithm 3: Proposed tree reconfiguration algorithm

IV. CONCLUSION

We studied in this paper tree reconfiguration without lightpath interruption in a network wherein resources are limited. This topic was studied in our previous work. The previous algorithm is called TRwRC. This paper propose an algorithm which improve the reconfiguration duration in the network wherein resources are limited while maintains the continuity of connection. It configures in parallel a set of branches. We will do in the future work, the evaluations of the proposed algorithm. The evaluation criteria are the reconfiguration duration, the interruption duration, the additional resources cost and the number of connections interrupted.

REFERENCES

[1]. A. Ding and G-S. Poo, "A survey of optical multicast over WDM networks", Computer Communications, Vol. 26 No. 2, 2003, pp. 193–200.

[2]. R. Malli, X. Zhang, and C. Qiao, "Benefit of Multicasting in All-Optical Networks", Photonics East (ISAM, VVDC, IEMB) in International Society for Optics and Photonics, 1998.

[3]. B. Mukherjee, "WDM Optical Communication Network: Progress and Challenges", IEEE Journal on Selected Areas in Communication, Vol. 18 No. 10, 2000, pp. 1810–1824.

[4]. R. Ramaswami, "Optical networking technologies: what worked and what didn't," *IEEE Communications Magazine*, vol. 44, no. 9, pp. 132–139, 2006.

[5]. J. Wu, "A survey of WDM network reconfiguration: strategies and triggering methods," *Computer Networks*, vol. 55, no. 11, pp. 2622–2645, 2011.

[6]. N. Geary, N. Parnis, A. Antonopoulos, E. Drakopoulos, and J. O'Reilly, "The benefits of reconfiguration in optical networks," in: *Proceedings of the 10th International Telecommunication Network Strategy and Planning Symposium*, 2002.

[7]. B. Cousin, J. C. Adépo, S. Oumtanaga, and M. Babri, "Tree Reconfiguration Without Lightpath Interruption in WDM Optical Networks", International Journal of Internet Protocol Technology, Vol. 7 No. 2, 2012, pp. 85–95.

[8]. B. Cousin, J.C. Adépo, M. Babri, and S. Oumtanaga, "Tree Reconfiguration without Lightpath Interruption in Wavelength Division Multiplexing Optical Networks with Limited Resources" International Journal of Computer Science Issues, vol. 11, issue 2, no. 2, pp. 7–17, 2014.

[9]. T. Kárász and Z. Pándi, "Optimal reconfiguration of provisioning oriented optical networks," in: *3rd International Working Conference on Performance Modeling and Evaluation of Heterogeneous Networks (HET-NETs)*, 2005.

[10]. I. Baldine and G. N. Rouskas, "Dynamic reconfiguration policies for WDM networks," in: *18th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 1, pp. 313–320, 1999.

[11]. I. Baldine and G. N. Rouskas, "Traffic adaptive WDM networks: a study of reconfiguration issues," *IEEE/OSA Journal of Light Wave Technology*, vol. 19, no. 4, pp. 433–455, 2001.

[12]. T. Kárász, "Consolidation strategies of provisioning oriented optical networks," *Journal of Optical Networking*, vol. 5, no. 6, pp. 445–462, 2006.

[13]. B. Cousin and M. Molnar, "Fast Reconfiguration of Dynamic Networks", International Workshop on Dynamic Networks, 2007.