



**HAL**  
open science

# Reconfiguration du routage multicast dans les réseaux optiques WDM.

Joël Adepo

► **To cite this version:**

Joël Adepo. Reconfiguration du routage multicast dans les réseaux optiques WDM.. Réseaux et télécommunications [cs.NI]. Université Nangui Abrogoua - Abidjan (Côte d'Ivoire), 2016. Français. NNT: . tel-03517032

**HAL Id: tel-03517032**

**<https://theses.hal.science/tel-03517032>**

Submitted on 7 Jan 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

*République de Côte d'Ivoire*  
*Union—Discipline—Travail*

*Ministère de l'Enseignement Supérieur*  
*et de la Recherche Scientifique*



*Unité de Formation et de Recherche des*  
*Sciences Fondamentales et Appliquées*  
*(UFR-SFA)*

N° d'ordre : **316**

## **THESE**

*Pour obtenir le grade de*

*Docteur de l'Université Nangui Abrogoua*

*Mention : Informatique*

*Spécialité : Réseaux*

*Présentée par :*

*ADEPO Joël Christian*

**Sujet :**

# **Reconfiguration du routage multicast dans les réseaux optiques WDM**

**Soutenue le 06 Septembre 2016 devant le Jury composé de :**

|                              |                              |  |                     |
|------------------------------|------------------------------|--|---------------------|
| <i>M. ADOU Kablan Jérôme</i> | <i>Professeur Titulaire</i>  | <i>Université Félix Houphouët Boigny, Abidjan - Côte d'Ivoire</i>            | <i>Président</i>    |
| <i>M. AKA Boko</i>           | <i>Professeur Titulaire</i>  | <i>Université Nangui Abrogoua, Abidjan - Côte d'Ivoire</i>                   | <i>Co-Directeur</i> |
| <i>M. BABRI Michel</i>       | <i>Maître de Conférences</i> | <i>INP-HB, Yamoussoukro - Côte d'Ivoire</i>                                  | <i>Co-Directeur</i> |
| <i>M. BEDJA Koffi-Sa</i>     | <i>Professeur Titulaire</i>  | <i>Université de Lomé, Ecole Nationale Supérieure d'Ingénieurs, Togo</i>     | <i>Rapporteur</i>   |
| <i>M. KORA Ahmed Dooguy</i>  | <i>Maitre de Conférences</i> | <i>Ecole Supérieure Multinationale des Télécommunications, Dakar-Sénégal</i> | <i>Examineur</i>    |

*A mes parents*

# Remerciements

Cette thèse a pu se réaliser grâce à DIEU Tout Puissant.

Nous remercions l'Université Nangui Abrogoua qui a initié la thèse en Informatique en Côte d'Ivoire. Nous remercions l'UFR SFA, de la direction aux enseignants pour leurs conseils et disponibilité. Nous remercions principalement Prof. Aka Boko et Dr. Michel Babri, nos co-Directeurs de thèse, qui contribuent énormément au développement de la recherche en informatique en Côte d'Ivoire. Nous remercions Prof. Souleymane Oumtanaga qui nous a accueillis depuis le DEA dans le laboratoire de recherche LARIT, au sein duquel nous avons fait nos premiers pas de recherche. Nous remercions particulièrement Prof. Bernard Cousin de l'université de Rennes 1 qui nous a conseillés, encadrés et accueillis au sein de son équipe à l'IRISA de Rennes. Il a conçu le sujet de cette thèse et il nous a co-encadrés depuis le début et jusqu'à la fin de cette thèse. Nous remercions le conseil scientifique du LARIT dirigé par Dr Loum Georges qui contribue à l'encadrement des stagiaires du LARIT. Nous remercions nos aînés du LARIT, notamment Dr Tanon Kadjo Lambert, Dr Traoré Issa, Dr Kimou Prospère pour les échanges fructueux que nous avons eus. Nous remercions tous les doctorants et stagiaires du LARIT pour leur sens du travail en équipe. Je remercie toute ma famille, en particulier mon père Adépo Assamoi, ma mère Kimou N'ta Thérèse, mon jeune frère Adépo Serge Martial et ma sœur aînée Adépo Jocéline pour leur soutien durant toute cette thèse.

# Résumé

Les réseaux optiques à multiplexage en longueur d'onde ou réseaux optiques WDM (*Wavelength Division Multiplexing*) sont largement déployés aujourd'hui. Les applications multimédias, la Télévision sur IP (*Internet Protocol*), les visioconférences, destinées à plusieurs utilisateurs sont très gourmandes en termes de bande passante. Le concept de multicast a été introduit dans les réseaux IP pour optimiser l'utilisation des ressources des réseaux pour de telles applications. Ce concept a été étendu aux réseaux optiques WDM. Ces réseaux sont fréquemment perturbés par des phénomènes tels que les surcharges, les pannes, l'ajout ou la suppression des ressources de réseau par des opérations de maintenance. Pour faire face à ces phénomènes perturbateurs, les opérateurs de réseau calculent une nouvelle topologie (un nouveau routage) en fonction de l'état du réseau. Ensuite le routage initial est reconfiguré pour migrer vers la nouvelle topologie. Certaines applications telles que la Télévision sur IP ne tolérant pas les interruptions de connexion, la migration doit se faire sans interruption de flux. De plus, elle doit être rapide pour faire face efficacement à l'évènement l'ayant déclenché. Enfin, elle doit réduire le coût des ressources réseau utilisées.

Le problème étudié dans cette thèse est la reconfiguration rapide du routage multicast dans les réseaux optiques WDM. Cette reconfiguration doit se faire sans interruption de flux, tout en réduisant la durée du processus et les ressources du réseau utilisées.

Les travaux réalisés durant cette thèse ont consisté à proposer différentes approches de résolution de ce problème. Nous avons prouvé qu'il n'existe pas d'algorithme pouvant reconfigurer tout arbre optique initial en tout arbre optique final en utilisant une seule longueur d'onde et sans interruption de flux. Nous avons proposé donc des algorithmes qui utilisent des longueurs d'onde additionnelles. Dans une première approche, nous avons proposé des algorithmes de reconfiguration par branche. Dans une seconde approche, nous avons proposé un algorithme de reconfiguration par sous-arbre. Les performances de ces algorithmes ont été évaluées dans différentes topologies couramment utilisées dans ce domaine de recherche.

**Mots clés :** réseau optique WDM, reconfiguration de réseau, arbre optique, séquence de reconfiguration, interruption de flux optique, durée de reconfiguration.

# Abstract

Wavelength Division Multiplexing (WDM) optical networks are widely deployed today. Multimedia applications, IPTV (i.e. television on Internet network), video conferencing, which have multiple end users, are very greedy in terms of bandwidth. The concept of multicast was introduced in IP networks to optimize the use of network resources because of such applications. This concept was subsequently extended to WDM optical networks. These networks are frequently disrupted by events such as overloads, failures, adding or deleting of network resources by maintenance. To address these disturbing phenomena, network operators compute a new topology (a new routing) depending on network state. Then the initial routing is reconfigured to migrate to the new routing. Since some applications, such as IPTV, do not tolerate interruption of the network connections, the reconfiguration must be done without connection interruption. In addition, it must be quick to respond effectively to the triggering event. Finally, it must reduce the cost of network resources used.

The problem studied in this thesis is the fast reconfiguration of multicast routing in WDM optical networks. This reconfiguration must be done without connection interruption, while reducing the process duration and the network resources used.

The work did during this thesis is to propose different approaches to solve this problem. We prove that there is no algorithm which can reconfigure any initial tree into any final tree using only one wavelength and without connection interruption. We propose algorithms which use additional wavelengths. In a first approach, we proposed a branch-based reconfiguration algorithms. In a second approach, we proposed a sub-tree-based reconfiguration algorithm. The performances of these algorithms were evaluated on different network topologies commonly used in this area of research.

**Keywords:** optical WDM network, network reconfiguration, light-tree, sequence of reconfiguration, multicast tree; lightpath interruption, reconfiguration delay.

# Table des matières

|   |     |
|---|-----|
| Remerciements.....  | ii  |
| <b>Résumé</b> .....   | iii |
| Abstract.....   | iv  |
| Table des matières.....   | v   |
| 1. Introduction.....  | 1   |
| 1.1 Contexte .....  | 1   |
| 1.2 Contributions .....   | 5   |
| 1.3 Plan de la thèse.....   | 6   |
| 2. Généralités sur les réseaux optiques.....  | 7   |
| 2.1 Introduction .....  | 7   |
| 2.2 Transmission optique .....  | 8   |
| 2.2.1 Liaison de transmission optique .....   | 8   |
| 2.2.2 Multiplexage optique.....   | 12  |
| 2.3 Réseaux optiques.....   | 14  |
| 2.3.1 Réseau optique opaque.....  | 14  |
| 2.3.2 Réseau optique transparent .....  | 14  |
| 2.3.3 Caractéristiques des différentes parties d'un réseau de télécommunication ..... | 14  |
| 2.4 Réseaux optiques WDM .....  | 16  |
| 2.4.1 Système de transmission avec multiplexage en longueur d'onde .....              | 16  |
| 2.4.2 Routage optique et allocation de longueurs d'onde .....                         | 17  |
| 2.4.3 Commutateurs optiques.....  | 18  |
| 2.5 Convertisseur optique .....   | 20  |
| 2.6 Conclusion .....  | 20  |
| 3. Routage multicast dans les réseaux optiques WDM.....                               | 22  |
| 3.1 Introduction .....  | 22  |
| 3.2 Avantages du multicast .....  | 23  |
| 3.3 Multicast dans le domaine électronique .....                                      | 23  |
| 3.4 Multicast optique.....  | 24  |
| 3.4.1 Commutateurs optiques supportant le multicast .....                             | 24  |
| 3.4.2 Routage multicast optique .....   | 29  |
| 3.4.3 Allocation de longueur d'onde .....   | 32  |
| 3.5 Contrôle de connexion .....   | 33  |

|       |   |    |
|-------|---|----|
| 3.6   | Conclusion .....  | 33 |
| 4.    | Reconfiguration du routage dans les réseaux optiques WDM.....                 | 35 |
| 4.1   | Introduction .....  | 35 |
| 4.2   | Phases de déroulement de la reconfiguration d'un réseau .....                 | 36 |
| 4.3   | Reconfiguration du routage unicast .....                                      | 37 |
| 4.3.1 | Problèmes de reconfiguration du routage unicast.....                          | 37 |
| 4.3.2 | Algorithme de reconfiguration .....   | 40 |
| 4.4   | Reconfiguration du routage multicast.....                                     | 47 |
| 4.4.1 | Problème de reconfiguration du routage multicast .....                        | 48 |
| 4.4.2 | Revue de la littérature .....   | 50 |
| 4.5   | Conclusion .....  | 53 |
| 5.    | Reconfiguration par branche du routage multicast .....                        | 54 |
| 5.1   | Introduction .....  | 55 |
| 5.2   | Définitions des principales notions .....                                     | 55 |
| 5.2.1 | Arbre courant.....  | 55 |
| 5.2.2 | Paire d'arbres associée à une instance de reconfiguration.....                | 55 |
| 5.2.3 | Hauteur d'un nœud sur un arbre .....  | 55 |
| 5.2.4 | Nœud amont.....   | 56 |
| 5.2.5 | Nœud aval.....  | 56 |
| 5.2.6 | Canal optique.....  | 57 |
| 5.2.7 | Canal optique amont .....   | 57 |
| 5.2.8 | Canal optique aval .....  | 57 |
| 5.3   | Classification des nœuds du réseau pour une instance de reconfiguration ..... | 57 |
| 5.3.1 | Nœud de branchement .....   | 57 |
| 5.3.2 | Nœud isolé.....   | 58 |
| 5.3.3 | Port isolé .....  | 58 |
| 5.3.4 | Nœud divergent.....   | 58 |
| 5.3.5 | Nœud convergent.....  | 58 |
| 5.4   | Opérations de configuration d'un nœud optique .....                           | 59 |
| 5.4.1 | Opération d'ajout de commutation de longueur d'onde.....                      | 59 |
| 5.4.2 | Opération de suppression de commutation de longueur d'onde.....               | 61 |
| 5.4.3 | Opération de conversion de longueur d'onde .....                              | 62 |
| 5.4.4 | Opération de multi-basculement optique.....                                   | 64 |
| 5.4.5 | Opération de convergence de longueur d'onde .....                             | 65 |



|       |   |     |
|-------|---|-----|
| 5.4.6 | Opération de non-convergence.....   | 67  |
| 5.5   | Problème de reconfiguration multicast utilisant une seule longueur d'onde .....   | 67  |
| 5.5.1 | Etude des séquences de reconfigurations d'un exemple de problème .....  | 69  |
| 5.5.2 | Evaluations.....  | 70  |
| 5.6   | Problème de reconfiguration d'une connexion multicast utilisant une longueur d'onde<br>additionnelle (temporairement et localement) ..... | 75  |
| 5.6.1 | Introduction.....   | 75  |
| 5.6.2 | Définitions.....  | 76  |
| 5.6.3 | Processus de reconfiguration par branche <i>BpBAR</i> .....   | 76  |
| 5.6.4 | Algorithme de reconfiguration <i>BpBAR_1</i> .....  | 85  |
| 5.6.5 | Réduction de la durée de reconfiguration du processus <i>BpBAR</i> .....  | 88  |
| 5.6.6 | Evaluations.....  | 95  |
| 5.6.7 | Conclusion .....  | 104 |
| 5.7   | Reconfiguration dans un réseau avec des ressources limitées .....   | 105 |
| 5.7.1 | Introduction.....   | 105 |
| 5.7.2 | Algorithme de reconfiguration <i>TRwRC</i> .....  | 105 |
| 5.7.3 | Evaluations.....  | 107 |
| 5.7.4 | Conclusion .....  | 113 |
| 5.8   | Conclusion .....  | 113 |
| 6.    | Reconfiguration par sous-arbre d'une connexion multicast .....  | 114 |
| 6.1   | Introduction.....   | 114 |
| 6.2   | Contexte et objectifs.....  | 115 |
| 6.3   | Approche de reconfiguration par sous-arbre.....   | 116 |
| 6.3.1 | Définition .....  | 116 |
| 6.3.2 | Challenges de reconfiguration par sous-arbre .....  | 116 |
| 6.4   | Garantie de la continuité du flux vers toutes les destinations du groupe multicast.....   | 119 |
| 6.5   | Garantie de la reconfiguration sans boucle.....   | 123 |
| 6.6   | Réduction des ressources additionnelles .....   | 124 |
| 6.7   | Réduction de la durée de reconfiguration.....   | 125 |
| 6.8   | Algorithme de reconfiguration par sous-arbre <i>STBReconf(P_of_T, n, L)</i> .....   | 126 |
| 6.8.1 | Fonction de construction d'une paire de sous-arbres reconfigurables sans longueur<br>d'onde additionnelle.....                            | 128 |
| 6.8.2 | Description de la fonction de reconfiguration d'une paire de sous-arbres .....  | 131 |
| 6.9   | Evaluations.....  | 132 |

|  |            |
|--|------------|
| 6.9.1 Critères d'évaluation .....        | 132        |
| 6.9.2 Description de la simulation ..... | 132        |
| 6.9.3 Résultats et analyses .....        | 133        |
| 6.10 Conclusion .....                    | 139        |
| 7. Conclusion et perspectives .....      | 140        |
| 7.1 Conclusion .....                     | 140        |
| 7.2 Perspectives .....                   | 141        |
| <b>Bibliographie</b> .....               | <b>142</b> |

# Liste des figures

|  |    |
|--|----|
| Fig. 2.1: Atténuation du signal lumineux dans une fibre optique en fonction de la fréquence..... | 8  |
| Fig. 2.2 : Principe de propagation de la lumière dans une fibre optique.....                     | 9  |
| Fig. 2.3 : Différents types de fibres optiques .....   | 10 |
| Fig. 2.4 : Multiplexage temporel.....  | 13 |
| Fig. 2.5 : Multiplexage en longueur d'onde.....  | 13 |
| Fig. 2.6 : Différentes parties d'un réseau de télécommunication optique.....                     | 15 |
| Fig. 2.7 : Système de transmission avec multiplexage en longueur d'onde.....                     | 16 |
| Fig. 2.8 : Répartiteur F-OXC .....   | 18 |
| Fig. 2.9 : Multiplexeur à insertion/extraction OADM .....  | 19 |
| Fig. 2.10 : Brasseur de longueur d'onde .....  | 20 |
| Fig. 3.1 : Transmission unicast et transmission multicast.....                                   | 23 |
| Fig. 3.2 : Commutateur OXC opaque .....  | 25 |
| Fig. 3.3 : Commutateur OXC opaque supportant le multicast .....                                  | 25 |
| Fig. 3.4 : Commutateur OXC transparent.....  | 26 |
| Fig. 3.5 : Commutateur OXC transparent supportant le multicast.....                              | 26 |
| Fig. 3.6 : Exemple d'arbre optique. ....   | 28 |
| Fig. 4.1 : Exemple de problème de reconfiguration de routage unicast.....                        | 38 |
| Fig. 4.2 : Exemple de problème de reconfiguration de routage unicast.....                        | 39 |
| Fig. 4.3 : Exemple d'instance de reconfiguration de routage unicast.....                         | 44 |
| Fig. 4.4 : Instance de reconfiguration multicast sans solution pour MTV. ....                    | 50 |
| Fig. 4.5 : Instance de reconfiguration multicast non résolue par MBB. ....                       | 52 |
| Fig. 5.1: Exemple d'instance de reconfiguration.....   | 56 |
| Fig. 5.2: Problème de reconfiguration. ....  | 59 |
| Fig. 5.3 : Exemple d'opération d'ajout de commutation d'une longueur d'onde. ....                | 60 |
| Fig. 5.4 : Exemple d'application de l'opération de suppression de commutation optique. ....      | 62 |
| Fig. 5.5 : Exemple d'utilisation de l'opération de conversion de longueur d'onde. ....           | 63 |
| Fig. 5.6 : Exemple d'utilisation de l'opération de multi-basculement.....                        | 65 |
| Fig. 5.7: Exemple d'application de l'opération de convergence.....                               | 66 |
| Fig. 5.8: Arbre initial $T_0$ et arbre final $T_z$ .....   | 68 |
| Fig. 5.9 : Topologie COST 239 .....  | 72 |
| Fig. 5.10 : Topologie NSF Network .....  | 73 |
| Fig. 5.11 : Topologie US Backbone.....   | 73 |

|   |     |
|---|-----|
| Fig. 5.12 : Description d'un tour de reconfiguration avec <i>BpBAR</i> .                                      | 77  |
| Fig. 5.13 : Exemple d'instance de reconfiguration pour <i>BpBAR</i> où le noeud de basculement est la source. | 78  |
| Fig. 5.14 : Séquence de reconfigurations d'une paire de branches par <i>BpBAR</i> .                           | 79  |
| Fig. 5.15 : Exemple de l'instance de reconfiguration où le noeud de basculement n'est pas la source.          | 80  |
| Fig. 5.16 : Etablissement de l'arbre final $Br(Tz, S, C)$   | 80  |
| Fig. 5.17 : a) Etablissement de l'arbre final. b) Basculement du flux au noeud B.                             | 81  |
| Fig. 5.18 : Arbre obtenu à la fin de la phase aller.  | 82  |
| Fig. 5.19 : Arbre optique et les flots associés aux différents canaux optiques.                               | 83  |
| Fig. 5.20 : Les parties d'une phase de reconfiguration avec <i>BpBAR_1</i>                                    | 87  |
| Fig. 5.21 : Etapes de reconfiguration dans une phase avec <i>BpBAR_2</i>                                      | 89  |
| Fig. 5.22 : Séquence de reconfiguration par branche avec <i>BpBAR_2</i>                                       | 91  |
| Fig. 5.23 : Exemple de problème de reconfiguration.   | 93  |
| Fig. 5.24 : Durée d'interruption  | 100 |
| Fig. 5.25 : Durée de reconfiguration  | 100 |
| Fig. 5.26 : Coût de ressources additionnelles.  | 101 |
| Fig. 5.27 : Durée de reconfiguration  | 102 |
| Fig. 5.28 : Coût de ressources additionnelles.  | 103 |
| Fig. 5.29 : Durée de reconfiguration.   | 109 |
| Fig. 5.30 : Coût de ressources additionnelles.  | 110 |
| Fig. 5.31 : Pourcentage d'instances de reconfiguration non résolues dans la topologie COST 239                | 110 |
| Fig. 5.32 : Pourcentage d'instances de reconfiguration non résolues dans la topologie NSF Network.            | 111 |
| Fig. 5.33 : Durée de reconfiguration.   | 112 |
| Fig. 5.34 : Coût de ressources additionnelles.  | 112 |
| Fig. 5.35 : Pourcentage d'instances de reconfiguration non résolues dans le réseau COST 239.                  | 112 |
| Fig. 6.1: Instance d'un problème de reconfiguration.  | 118 |
| Fig. 6.2 : Exemple de reconfiguration par sous-arbre avec boucle  | 118 |
| Fig. 6.3: Problème de reconfiguration.  | 130 |
| Fig. 6.4 : Durée d'interruption de chaque algorithme  | 134 |
| Fig. 6.5 : Durée du processus de reconfiguration de chaque algorithme.  | 135 |
| Fig. 6.6 : Durée du processus de reconfiguration avec chaque algorithme.                                      | 135 |
| Fig. 6.7 : Coût additionnel de chaque algorithme.   | 136 |
| Fig. 6.8 : Coût additionnel de chaque algorithme.   | 136 |

|   |     |
|---|-----|
| Fig. 6.9 : Evolution de la durée de reconfiguration et du coût de ressources additionnelles en fonction du nombre de nœuds destinataires dans la topologie COST 239 .....     | 137 |
| Fig. 6.10 : Evolution de la durée de reconfiguration et du coût de ressources additionnelles en fonction du nombre de nœuds destinataires dans la topologie NSF Network ..... | 138 |
| Fig. 6.11 : Evolution de la durée de reconfiguration et du coût de ressources additionnelles en fonction du nombre de nœuds destinataires dans la topologie US Backbone.....  | 138 |

# Liste des tableaux

|   |     |
|---|-----|
| Tableau 4.1 : Chemins initiaux et finaux du problème de reconfiguration de la figure 4.1.....   | 39  |
| Tableau 5.1 : Caractéristiques des topologies utilisées dans l'évaluation.....  | 72  |
| Tableau 5.2: Les parties d'une phase de reconfiguration avec <i>BpBAR_1</i> .....   | 87  |
| Tableau 5.3 : Opérations exécutées en parallèle dans une phase de reconfiguration de paire de branches avec <i>BpBAR_2</i> .....                  | 89  |
| Tableau 5.4 : Correspondance entre les nœuds de basculement et les différentes branches initiales associées dans l'arbre de la figure 5.23.b..... | 94  |
| Tableau 5.5 : Correspondance entre les nœuds de basculement et les différentes branches initiales associées dans l'arbre de la figure 5.23.d..... | 94  |
| Tableau 5.6 : Durée d'interruption.....   | 99  |
| Tableau 5.7 : Durée de reconfiguration.....   | 99  |
| Tableau 5.8 : Coût des ressources additionnelles.....   | 99  |
| Tableau 5.9 : Durée de reconfiguration.....   | 102 |
| Tableau 5.10 : Coût de ressources additionnelles.....   | 102 |
| Tableau 5.11 : Durée de reconfiguration.....  | 108 |
| Tableau 5.12 : Coût de ressources additionnelles.....   | 108 |
| Tableau 5.13 : Durée d'interruption.....  | 108 |
| Tableau 5.14 : Durée de reconfiguration.....  | 111 |
| Tableau 5.15 : Coût de ressources additionnelles.....   | 111 |
| Tableau 6.1: Durée d'interruption du processus de reconfiguration.....  | 133 |
| Tableau 6.2: Durée du processus de reconfiguration.....   | 133 |
| Tableau 6.3: Coût additionnel du processus de reconfiguration.....  | 134 |
| Tableau 6.4: Durée du processus de reconfiguration.....   | 135 |
| Tableau 6.5: Coût additionnel du processus de reconfiguration.....  | 135 |

# Liste des acronymes

|            |   |
|------------|---|
| AFDE :     | Amplificateur à Fibre Dopée à l'Erbium                  |
| BBM :      | Break-Before-Make                                       |
| CBT :      | Core-based Tree   |
| EDFA :     | Erbium Doped Fiber Amplifier                            |
| F-OXC :    | Fiber Optical Cross-Connect                             |
| FTTx :     | Fiber To The xxx  |
| GMPLS :    | Generalized Multi-Protocol Label Switching              |
| GSMP :     | General Switching Management Protocol                   |
| IP :       | Internet Protocol                                       |
| IPv4 :     | Internet Protocol version 4                             |
| IPv6 :     | Internet Protocol version 6                             |
| LASER :    | Light Amplification by Stimulated Emission of Radiation |
| LED :      | Light Emitting Diode                                    |
| LSP :      | Label Switching Path                                    |
| MC :       | Multicast Capable                                       |
| MC-RWA :   | Multicast Routing and Wavelength Assignment             |
| MBB :      | Make-Before-Break                                       |
| MI :       | Multicast Incapable                                     |
| MIB Node : | Multicast Incapable Branching Node                      |
| MPLS :     | Multi-Protocol Label Switching                          |
| MST :      | Minimum Steiner Tree                                    |
| MTV :      | Move-To-Vacant  |
| OADM :     | Optical Add/Drop Multiplexer                            |

OSW: Optical Switch

OTDM : Optical Time-Division Multiplexing

PCE : Path Computation Element

PIM : Protocol Independant Multicast

QoS : Quality of Service

SDH : Synchronous Digital Hierarchy

SPT : Shortest Path Tree

SONET : Synchronous Optical NETworks

TDM : Time-Division Multiplexing

WR-OXC : Wavelength Routing Optical Cross-Connect

WDM : Wavelength-Division Multiplexing

xDSL : xxx Digital Subscriber Line



# 1. Introduction

---

## Sommaire

---

|                           |   |
|---------------------------|---|
| 1.1 Contexte.....         | 1 |
| 1.2 Contribution.....     | 5 |
| 1.3 Plan de la thèse..... | 6 |

---

## 1.1 Contexte

Les réseaux optiques à multiplexage en longueur d'onde ou réseaux optiques WDM constituent l'architecture sous-jacente principale des réseaux à très haut débit d'aujourd'hui, notamment d'Internet. Un réseau est une topologie constituée de nœuds et des liens entre ces nœuds utilisée pour la transmission de données. La technologie WDM utilisée par les réseaux optiques permet d'augmenter le débit d'une fibre optique, pour atteindre plusieurs centaines de Gb/s. Cette technologie consiste à combiner plusieurs longueurs d'onde sur une même fibre.

De nombreuses applications utilisent ces réseaux. Parmi ces applications, certaines telles que la Télévision sur IP ou la visioconférence sont exigeantes en termes de bande passante et de QoS (qualité de service). Or, ces réseaux sont fréquemment perturbés par des phénomènes tels que les surcharges, les pannes de liens ou des nœuds, les opérations de maintenance, l'ajout de nouvelles ressources ou la suppression de ressources existantes. Ces évènements dégradant (ou améliorant) l'état du réseau, modifient la topologie initiale du réseau. Pour faire face à ces phénomènes perturbateurs, les opérateurs de réseau utilisent des solutions telles que l'ingénierie de trafic, la protection et la restauration de chemins ou la reconfiguration du routage.

L'ingénierie de trafic consiste à placer le trafic sur les ressources du réseau tout en évitant des congestions. Mais cette solution ne change pas au cours du temps le chemin utilisé par le trafic. Les techniques de protection et restauration permettent de répondre efficacement aux pannes de ressources du réseau (nœuds et liens). Une connexion dans les réseaux étant une paire de nœuds (un nœud source et un nœud destinataire) voulant communiquer à travers le réseau, un chemin dans le réseau est une suite de liens entre la source et la destination d'une connexion utilisée pour la transmission du flux.

Les techniques de protection et restauration calculent soit avant la panne soit après la panne un nouveau chemin pour établir la connexion. Le nouveau chemin est calculé en fonction de l'ancien chemin, la topologie du réseau et la panne. Dans notre travail, on suppose cette étape de calcul de nouveau chemin faite, c'est-à-dire, on considère comme hypothèse de départ la connaissance du chemin initial et du chemin final. On s'intéresse à la séquence de reconfiguration afin de basculer de l'un à l'autre. L'objectif de notre travail est de changer la topologie courante du routage en un nouveau routage. Les travaux effectués dans cette thèse concernent la reconfiguration du routage.

La reconfiguration des chemins du réseau permet d'optimiser les performances du réseau en satisfaisant plus de demandes de connexion tout en maximisant l'utilisation du réseau. Elle permet également d'optimiser l'acheminement du trafic. La reconfiguration du routage est constituée de trois phases séquentielles et itératives [Wu11]. La première phase consiste à décider de la politique de reconfiguration du réseau. Il s'agit de déterminer les événements pouvant déclencher une reconfiguration du réseau, la fréquence de reconfiguration et la manière dont va se dérouler le processus de reconfiguration. Les réponses à toutes ces questions vont permettre de passer à la deuxième phase. Celle-ci consiste à calculer la nouvelle topologie de routage qui satisfait les objectifs d'utilisation du réseau. Le calcul du nouveau routage se fait en fonction des réponses de la première phase, de l'état du réseau et des objectifs d'optimisation. La dernière phase de la reconfiguration concerne la migration proprement dite de la topologie initiale à la topologie finale. C'est dans cette phase que les différents nœuds constituant l'ancienne topologie et la nouvelle sont configurés pour passer de l'ancienne topologie à la nouvelle.

La reconfiguration des réseaux optiques WDM [Muk00 ; Ram06] est très bénéfique car ces réseaux sont très largement déployés aujourd'hui. Ce grand intérêt donné aux réseaux optiques WDM est dû à leur large bande passante, leur faible taux d'erreurs et leur coût satisfaisant. Ces réseaux sont constitués de commutateurs optiques appelés *switches* interconnectés par des fibres optiques.

Les erreurs dans les configurations effectuées par les opérateurs sont non négligeables. Elles constituent 50% des pannes des systèmes de communications et réseaux [LWK12]. Les réseaux optiques WDM nécessitent une bonne gestion, un bon contrôle et une reconfiguration bien planifiée pour assurer les exigences en QoS de leurs utilisateurs [Gea02]. La reconfiguration du routage dans les réseaux optiques WDM est un challenge. En effet, certaines ressources du

réseau telles que les émetteurs, les récepteurs et les longueurs d'onde sont temporairement libérées durant le processus de reconfiguration. Cela pourrait causer la perte de données pour un trafic à très haut débit comme celui des réseaux optiques WDM. La perte de données dans de tels réseaux peut causer des dommages à de nombreux utilisateurs, donc dégrader les performances du réseau. De plus, le processus de reconfiguration doit réduire la durée de reconfiguration [GB04, Wei02] et le coût des ressources du réseau utilisées. La réduction de la durée de reconfiguration permet de répondre efficacement à l'événement ayant déclenché la reconfiguration. Par exemple si l'événement déclencheur est une surcharge, la reconfiguration doit s'effectuer rapidement afin d'éviter la surcharge ou de réduire le temps durant lequel la surcharge existe dans le réseau.

Dans le processus de reconfiguration des réseaux optiques WDM, la troisième phase consistant à établir le nouveau chemin et supprimer ou modifier le chemin existant est effectuée en ligne [Wu11]. Les *switches* composant l'ancien routage et le nouveau routage sont configurés pour prendre en compte d'une part, les nouveaux paramètres modifiant leur table d'acheminement (routage et commutation) et d'autre part, les paramètres de gestion des ressources du réseau (mémoire tampon, les files d'attente d'entrée et de sortie des *switches*) [CM07].

Le trafic sur les longueurs d'onde dans les réseaux optiques est à très haut débit. Ces signaux optiques sont le plus souvent sensibles aux interruptions ou au délai de transmission. L'interruption d'un chemin optique peut perturber énormément toutes les communications utilisant cette longueur d'onde. La reconfiguration du routage doit se faire de façon coordonnée pour maintenir la continuité de la transmission du trafic [CM07].

La gestion de la connexion dans les réseaux optiques WDM peut être assurée par des protocoles comme le protocole GSMP [Dor02]. La technique de reconfiguration utilisée par ces protocoles consiste à supprimer l'ancien chemin avant d'établir le nouveau. Cela ne garantit pas la continuité du trafic. De plus, cette technique n'est pas optimale en termes de durée du processus de reconfiguration. La reconfiguration de la topologie doit se faire rapidement afin de répondre spontanément à l'évènement déclencheur avant que d'autres évènements n'adviennent dans le réseau.

La reconfiguration peut être déployée selon deux approches : l'approche centralisée et l'approche distribuée. Dans l'approche distribuée, chaque nœud initie de façon indépendante la reconfiguration. Le déclenchement des opérations de configuration par chaque nœud

nécessite une synchronisation afin d'éviter les interruptions du trafic. La synchronisation de l'exécution de ces opérations de reconfiguration est difficile à réaliser dans le cas distribué. Dans l'approche centralisée, une entité centrale (logicielle et/ou matérielle) a une vue globale de l'ensemble de la topologie du réseau, de l'utilisation des liens et nœuds, et de la demande de trafic des connexions. Cette entité coordonne le processus de reconfiguration. La fonction de reconfiguration dans l'approche centralisée peut être implémentée dans l'architecture PCE (*Path Computation Element*) comme une partie du système de gestion centralisée [FVA06] ou implémenté dans un contrôleur *Openflow* [LTM13]. Dans l'approche centralisée, la synchronisation de l'exécution des opérations est naturellement obtenue. Dans nos travaux, nous supposons que la reconfiguration est contrôlée de façon logiquement centralisée. Donc, lorsque la reconfiguration de la topologie d'un réseau est demandée, le contrôleur envoie des messages de configuration à tous les nœuds du réseau devant être configurés ou reconfigurés. Le problème qui se pose alors est l'ordonnancement de l'envoi des différents messages de configuration aux différents nœuds afin de garantir la continuité du trafic. Le contrôleur doit transmettre les messages d'une façon coordonnée [CM07]. Cela revient à résoudre le problème de détermination de sous-ensembles de nœuds à configurer et l'ordre de configuration.

Les applications telles que la télévision sur IP ou la vidéoconférence sont destinées à plusieurs destinataires. Elles nécessitent le *multicasting* [HCT02; SA98]. Dans une communication point-à-point, le *multicasting* est obtenu en demandant à la source de transmettre à chaque destinataire une copie de la donnée à transmettre. Cette technique est très coûteuse en bande passante, car la source émet plusieurs fois la même donnée. Une technique efficace consiste à transmettre la donnée aux différents destinataires en utilisant un arbre comme chemin de diffusion. La source, située à la racine de l'arbre, n'émet alors qu'une seule fois chaque donnée. Cette technique appelée le multicast a donc été introduite dans les réseaux IP pour réduire la consommation en bande passante des applications nécessitant le *multicasting*. Le multicast a ensuite été étendu aux réseaux optiques WDM pour améliorer ces performances [DP03 ; MZQ98]. Dans ces réseaux, l'arbre utilisé pour acheminer le signal optique est appelé arbre optique ou *light-tree*. L'extension du multicast aux réseaux optiques nécessite d'équiper les nœuds optiques de *splitter* permettant de dupliquer le signal optique reçu sur un port d'entrée vers un ou plusieurs ports de sortie.

Dans cette thèse, nos travaux s'intéressent à la reconfiguration d'un arbre optique sans interruption du flux. Les algorithmes proposés doivent non seulement garantir la continuité du

flux vers tous les destinataires du groupe multicast, mais aussi réduire la durée du processus de reconfiguration et le coût des ressources utilisées.

Pour garantir la continuité du flux vers tous les destinataires, les algorithmes proposés garantissent à chaque exécution d'une opération de configuration sur un nœud, l'existence d'un arbre enraciné à la source de la connexion et atteignant les différents nœuds destinataires du groupe multicast.

Pour réduire la durée du processus, les algorithmes de reconfiguration déterminent un sous-ensemble de nœuds à configurer en parallèle. Ces sous-ensembles de nœuds doivent être déterminés de telle sorte que leur configuration garantit l'existence d'un arbre entre la source de la connexion multicast et les nœuds destinataires. La durée de configuration en parallèle d'un sous-ensemble de nœuds devient alors majorée par la durée maximale de configuration d'un nœud. Cela permet de réduire la durée de reconfiguration du processus.

Pour réduire les ressources utilisées durant le processus de reconfiguration, nous avons étudié dans une première partie la reconfiguration sans utilisation de longueurs d'onde additionnelles. Nous avons montré que la reconfiguration de tout arbre optique n'est pas toujours possible avec cette approche. Les algorithmes proposés utilisent alors des ressources additionnelles pour reconfigurer les arbres optiques. Ces algorithmes proposent de réduire le nombre de longueurs d'onde utilisées simultanément par lien pour réduire le coût des ressources utilisées.

### 1.2 Contributions

Différents algorithmes de reconfiguration d'arbre optique sont proposés dans cette thèse. La reconfiguration nécessitant l'exécution d'opérations sur les nœuds à configurer, nous définissons les différentes opérations de configuration utiles à la reconfiguration du routage multicast sans interruption de flux. Ensuite deux types d'algorithmes ont été proposés.

Dans une première approche, nous proposons des algorithmes de reconfiguration par branche. Ces algorithmes déterminent une paire de branches appartenant à l'arbre initial et l'arbre final dont la reconfiguration peut se faire sans interruption de flux vers toutes les destinations du groupe multicast. Avec cette approche, on reconfigure l'arbre optique branche par branche jusqu'à l'obtention de l'arbre final.

Dans la seconde approche, nous proposons des algorithmes de reconfiguration par sous-arbre. Ces algorithmes déterminent dans l'arbre à reconfigurer, une paire de sous-arbres appartenant à l'arbre initial et l'arbre final à configurer. La paire de sous-arbres est déterminée de telle sorte que sa reconfiguration s'effectue en parallèle, et sans interruption de flux vers les destinataires du groupe multicast. De plus, elle cherche à réduire le coût de ressources additionnelles utilisées pour la reconfiguration.

Les publications [CAOB12] et [CABO14] sont issues de nos travaux. Dans [CAOB12], nous avons montré que la reconfiguration d'un routage multicast peut se faire sans interruption de flux, en utilisant différentes longueurs d'onde. Nous avons proposé dans cet article, l'algorithme *BpBAR\_2* qui reconfigure tout arbre sans interruption de flux. Dans [CABO14], nous avons étudié le problème de reconfiguration dans un réseau ayant des ressources limitées. Nous avons proposé l'algorithme *TRwRC* qui reconfigure tout arbre sans interruption de flux dans un réseau ayant un nombre limité de longueurs d'onde par lien. Nos travaux ont également fait l'objet d'un article lors de la conférence CARI 2012 [AO12].

### 1.3 Plan de la thèse

Le chapitre 2 de la thèse présente les détails techniques et le fonctionnement du réseau optique WDM qui est le réseau support de notre travail.

Dans le chapitre 3, nous présentons le routage multicast dans les réseaux optiques WDM. Dans ce chapitre, nous introduisons la notion de multicast dans les réseaux optiques WDM, et présentons le problème de routage et d'affectation de longueur d'onde pour les connexions multicast.

Le chapitre 4 s'intéresse au problème de reconfiguration du routage dans les réseaux optiques WDM. Ce chapitre fait un état de l'art de la reconfiguration du routage dans les réseaux optiques WDM.

Les chapitres 5 et 6 présentent les différents algorithmes de reconfiguration que nous avons proposés. La reconfiguration par branche est présentée dans le chapitre 5. La reconfiguration par sous-arbre est présentée dans le chapitre 6. Le chapitre 7 conclut la thèse et donne les perspectives.

## 2. Généralités sur les réseaux optiques

---

### Sommaire

---

|   |    |
|---|----|
| 2.1 Introduction.....   | 7  |
| 2.2 Transmission optique.....   | 8  |
| 2.2.1 Liaison de transmission optique.....                                      | 8  |
| 2.2.2 Multiplexage optique.....   | 12 |
| 2.3 Réseaux optiques.....   | 14 |
| 2.3.1 Réseau optique opaque.....  | 14 |
| 2.3.2 Réseau optique transparent.....   | 14 |
| 2.3.3 Caractéristiques des différentes parties d'un réseau de télécommunication |    |
| 2.4 Réseaux optiques WDM.....   | 16 |
| 2.4.1 Système de transmission avec multiplexage en longueur d'onde.....         | 16 |
| 2.4.2 Routage optique et allocation de longueurs d'onde.....                    | 17 |
| 2.4.3 Commutateurs optiques.....  | 18 |
| 2.5 Convertisseur optique.....  | 20 |
| 2.6 Conclusion.....   | 20 |

---

### 2.1 Introduction

Le problème de reconfiguration de routage que nous étudions se pose dans tous les réseaux à connexions, c'est-à-dire les réseaux qui nécessitent l'établissement d'une connexion avant tout transfert de données. Nous étudions dans cette thèse le problème de reconfiguration dans les réseaux optiques WDM car ils présentent aujourd'hui de nombreux avantages et sont largement déployés. Dans ce chapitre, nous présentons dans la section 2.2 les principaux détails techniques de la transmission optique et le fonctionnement des réseaux optiques. La section 2.3 décrit le réseau optique. Nous décrivons dans la section 2.4 les réseaux optiques WDM et présentons quelques problèmes qui y sont rencontrés notamment celui du routage et de l'allocation de longueurs d'onde. La section 2.5 présente le convertisseur optique, composant utilisé pour augmenter le nombre de connexions pouvant être établies dans le réseau optique WDM.

## 2.2 Transmission optique

De nombreux livres, articles et thèses expliquent la transmission sur la fibre optique [Bor97, Per05, RSS10, Vac07]. [He14] est un état de l'art sur la communication optique. Nous présentons ci-dessous les caractéristiques de la fibre optique et le système de transmission de l'information à partir d'un signal optique.

### 2.2.1 Liaison de transmission optique

#### 2.2.1.1 Fibre optique

La fibre optique constitue aujourd'hui un support de transmission très largement utilisé grâce à des caractéristiques faisant d'elle un excellent support de transmission : faible atténuation, large bande passante, petit encombrement, flexible, insensible aux parasites, peu chère... Une fibre optique se présente sous forme d'un cylindre ayant un diamètre de quelques centaines de micromètres, utilisée pour la propagation de la lumière. La fibre optique est conçue pour propager la lumière le long de la fibre sur une longue distance avec un très faible taux de perte. Cela est possible grâce au phénomène physique de la réflexion totale.

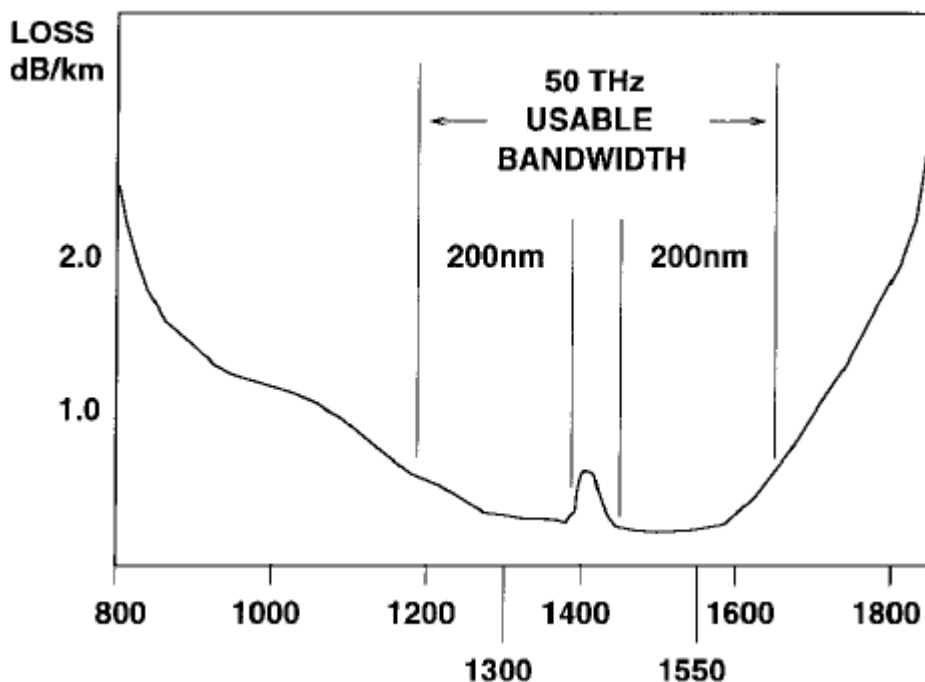


Fig. 2.1: Atténuation du signal lumineux dans une fibre optique en fonction de la fréquence.



La figure 2.1 présente les deux régions de faible atténuation du signal sur une fibre [Bor97]. Dans la région centrée à 1300 nm, de largeur 200 nm, l'atténuation est inférieure à 0.5 dB par kilomètre. Dans la région centrée à 1550 nm, de même largeur que la première, l'atténuation est inférieure à 0.2 dB par kilomètre.

La fibre est constituée d'un cœur et d'une gaine, d'indices de réfraction différents. Les indices de réfraction et l'angle d'incidence du rayon lumineux sont déterminés de sorte que la lumière soit totalement réfléchie dans le cœur de la fibre. La figure 2.2 décrit un exemple de propagation de la lumière dans une fibre.

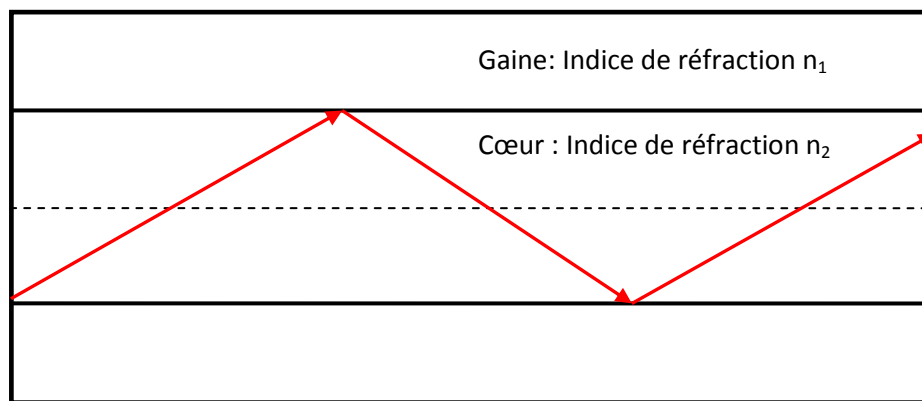


Fig. 2.2 : Principe de propagation de la lumière dans une fibre optique

Le matériau le plus utilisé dans la fabrication de la fibre est la silice. La silice existe en quantité très suffisante et est peu chère.

Il existe deux types de fibres optiques : la fibre multimode et la fibre monomode. La principale différence entre ces deux types de fibre est le mode de propagation de ou des ondes lumineuses au sein de la fibre et la source lumineuse permettant d'émettre le signal lumineux sur la fibre.

La fibre multimode est la plus utilisée. La source lumineuse utilisée pour la propagation du signal est de manière usuelle la LED. Cette fibre permet la propagation du signal lumineux sous différents angles de réfraction. Ainsi, on a plusieurs chemins (ou modes) différents dans la fibre multimode. On distingue la fibre multimode à saut d'indice et la fibre multimode à gradient d'indice. Le diamètre de la fibre multimode à saut d'indice est d'environ 200 micromètres. Ce type de fibre entraîne une déformation importante du signal. La fibre multimode à gradient d'indice a un diamètre de cœur variant de 50 à 100 micromètres. Ce type de fibre réduit la déformation du signal.

La fibre monomode utilise le laser comme source d'émission lumineuse. Elle est très adaptée pour les transmissions sur de longues distances. Le diamètre de son cœur est très fin, de l'ordre de 8,3 à 10 micromètres, évitant ainsi des réflexions car le signal ne rencontre pas la gaine. Ce type de fibre assure la propagation d'un seul signal sur un seul chemin (mode). La figure 2.3 présente les différences et les caractéristiques de ces différents types de fibre.

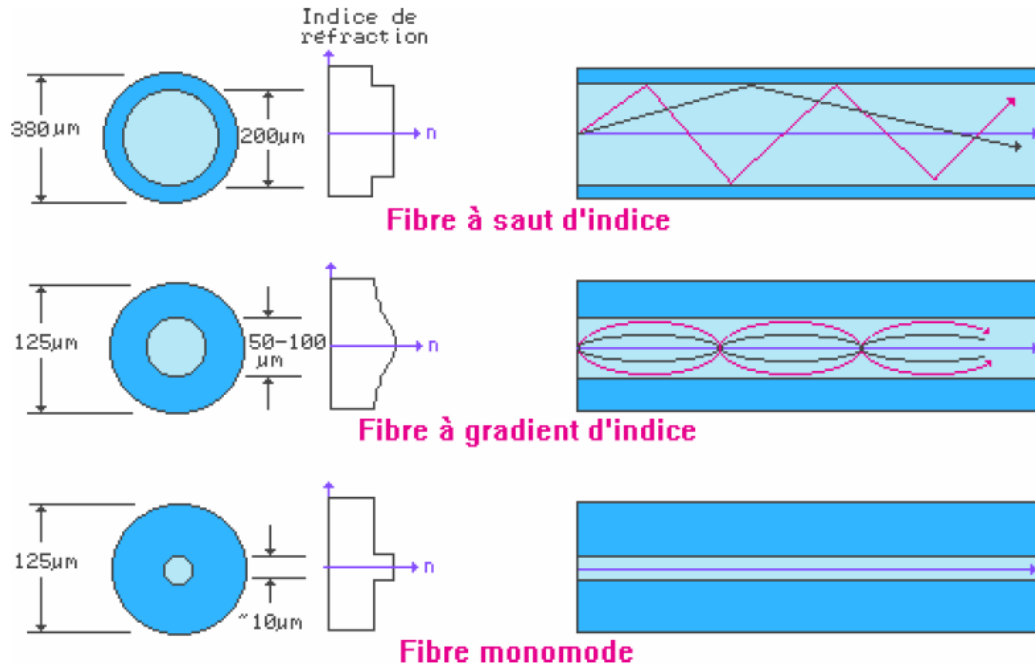


Fig. 2.3 : Différents types de fibres optiques

### 2.2.1.2 Emetteur optique

Un émetteur dans une transmission optique a pour fonction de faire la conversion électronique-optique du signal et de moduler ce signal par l'information à transmettre. Il transforme un signal électrique (Ethernet, SDH, SONET...) en un signal optique à transporter sur une longueur d'onde. Un émetteur est constitué d'une source de lumière et d'un modulateur.

La diode électroluminescente ou LED (*Light Emitting Diode*) et le laser (*Light Amplification by Stimulated Emission of Radiation*) signifiant amplification de lumière par émission stimulée de rayonnement constituent les sources de lumière utilisées pour la communication optique. Le principe de l'émission stimulée permet au laser de produire un faisceau intense de lumière monochromatique. Il existe différents types de lasers utilisés. Les lasers à semi-conducteurs compacts sont les plus utilisés. Les lasers sont par contre très chers et ne sont pas adaptés aux applications qui ne nécessitent pas de très grands débits et aux réseaux de transmission sur de

courtes distances. Les LED constituent une alternative aux lasers dans ce type de communication. La source de lumière la plus utilisée est la LED.

### **2.2.1.3 Récepteur optique**

Un récepteur dans un système de transmission optique a pour fonction de convertir le signal optique en signal électrique. Cette fonction est accomplie grâce à des techniques de détection du signal optique. Il existe deux techniques de détection du signal optique : la détection directe et la détection cohérente. La détection cohérente utilise un laser auxiliaire comme oscillateur. Le signal résultant de la combinaison des signaux des deux lasers accordés sur une fréquence proche est transmis à une photodiode. Cette technique permet une détection plus facile. Mais elle est difficile à mettre en œuvre et plus coûteuse. La détection directe est composée de plusieurs blocs fonctionnels regroupés en trois parties. La première partie est constituée d'un photo-détecteur sensible (diode photosensible) qui convertit le flot de photons en flot d'électrons. Le courant électrique obtenu en sortie de la première partie est amplifié par la deuxième partie constituée d'amplificateurs. La dernière partie est le bloc de récupération de données. Le courant résultant de la deuxième partie est soumis à un test de seuil pour déterminer l'information équivalente (0 ou 1).

### **2.2.1.4 Amplificateur**

Le signal optique transmis sur une fibre optique s'atténue au bout d'une certaine distance. Lorsque l'atténuation atteint 20 dB (après un parcours d'environ 100 km), le signal devient trop faible ou contient des bruits. Il ne peut plus être détecté ou est difficile à détecter. Pour atteindre des portées supérieures, des dispositifs appelés amplificateurs sont placés à intervalles réguliers (50 à 100 km) sur le chemin pour amplifier le signal optique afin de compenser les pertes ou restaurer la qualité du signal bruité.

La qualité d'un amplificateur est déterminée par certains paramètres comme le gain, la largeur de bande de fréquences amplifiables, le gain efficace [Bor97]. Le gain est le rapport entre la puissance du signal entrant et celle du signal sortant. Il est très souvent exprimé en décibels. Le gain efficace est le rapport entre le gain et la puissance d'amplification. Il est exprimé en dB/mW.

Plusieurs familles d'amplificateurs optiques existent. Les deux principales sont les amplificateurs à semi-conducteurs et les amplificateurs à fibre dopée. L'élément dopant le plus

utilisé dans l'amplificateur à fibre dopée est l'erbium. On parle alors d'Amplificateur à Fibre Dopée à l'Erbium (AFDE) ou *Erbium Doped Fiber Amplifier* (EDFA). L'amplificateur constitue une véritable révolution dans les télécommunications car permettant une transmission sur de longues distances sans conversion du signal optique en signal électrique. Cependant, elle présente quelques limites. L'amplification d'un signal introduit un bruit dans le signal qui est amplifié chaque fois qu'il rencontre un autre amplificateur. Alors il faut régénérer (convertir opto-électriquement, reformater puis réémettre) le signal optique au bout d'un certain temps de transmission pour éliminer le bruit.

### 2.2.2 Multiplexage optique

Le multiplexage optique permet de mieux exploiter la large bande passante de la fibre optique qui est de l'ordre de 50 THz. Il consiste à transmettre sur un même support des signaux provenant de plusieurs sources. Dans le multiplexage optique, la bande passante de la fibre optique est divisée en plusieurs canaux soit dans la dimension temporelle, soit dans la dimension fréquentielle. Lorsque le multiplexage est effectué dans la dimension temporelle, on parle de multiplexage temporel ou *Time-Division Multiplexing* (TDM). Lorsque le multiplexage est effectué dans la dimension fréquentielle, on parle de multiplexage en longueur d'onde ou *Wavelength-Division Multiplexing* (WDM).

#### 2.2.2.1 Multiplexage temporel (TDM)

Le multiplexage temporel consiste à allouer la totalité de la bande passante d'une fibre optique à chaque utilisateur (source de données) durant une courte durée pour la transmission du signal optique. L'allocation de la bande passante se fait en divisant l'axe de temps en intervalles de temps fixe. Un intervalle de temps est accordé successivement à chaque source pour la transmission.

Les fonctions de multiplexage et démultiplexage sont réalisées à l'aide de circuits électroniques ultra-rapides. Lorsqu'il est réalisé dans le domaine optique, on parle de multiplexage temporel optique ou *Optical Time-Division Multiplexing* (OTDM). La figure 2.4 présente le principe de multiplexage temporel (on notera que le multiplexage de la figure est réalisé dans le domaine électrique).

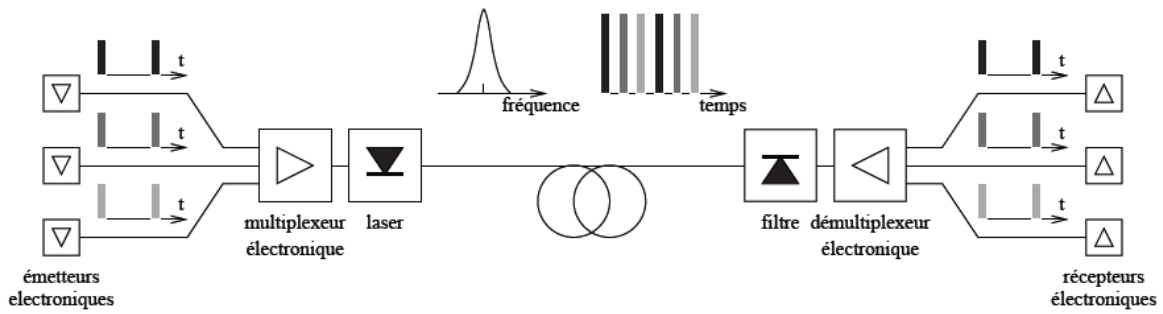


Fig. 2.4 : Multiplexage temporel

### 2.2.2.2 Multiplexage en longueur d'onde (WDM)

Le multiplexage en longueur d'onde se fait dans la dimension fréquentielle. Ce n'est plus l'axe de temps qui est divisé et alloué à chaque utilisateur, mais c'est la bande passante de la fibre qui est divisée. Plusieurs signaux sont générés simultanément sur des fréquences optiques (longueurs d'onde) différentes et sont transmises simultanément sur une même fibre optique. La bande passante de la fibre optique est subdivisée en un certain nombre de sous-bandes ou intervalles de longueurs d'onde distinctes. Chaque sous-bande supporte la transmission d'un signal. Le nombre de longueurs d'onde utilisables d'une fibre est limité par la technologie des composants optiques et de l'espacement entre les longueurs d'onde. Ce nombre est de 2 longueurs d'onde pour le multiplexage à deux voies *wide wavelength division multiplexing* (WDM), 4 à 8 longueurs d'onde pour le multiplexage *coarse wavelength multiplexing* (CWDM) et plus de 100 longueurs d'onde pour le multiplexage dense *dense wavelength multiplexing* (DWDM). La figure 2.5 présente le principe de multiplexage en longueur d'onde.

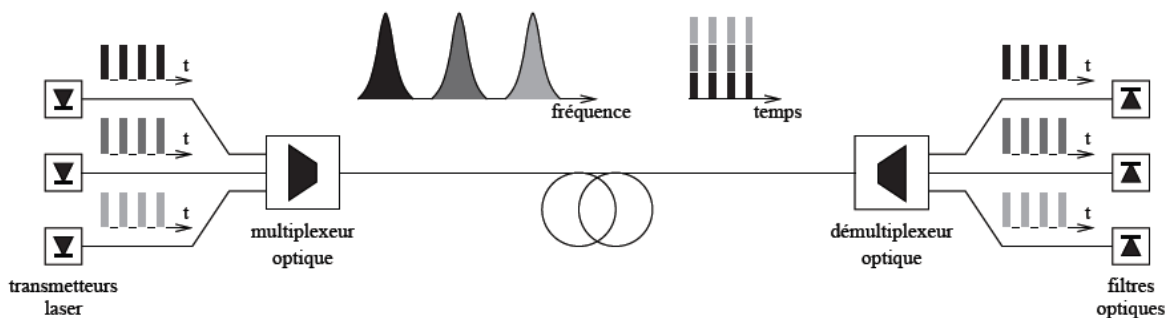


Fig. 2.5 : Multiplexage en longueur d'onde

### 2.3 Réseaux optiques

Cette section présente les réseaux optiques. Les réseaux optiques sont subdivisés en deux types: les réseaux optiques opaques et les réseaux optiques transparents. Nous décrivons d'abord ces deux types de réseaux. Ensuite nous présentons les trois principales parties constituant un réseau de télécommunication.

#### 2.3.1 Réseau optique opaque

Un réseau optique opaque est un réseau optique nécessitant des conversions opto-électroniques du signal entrant sur chaque port d'entrée d'un nœud du réseau. Le signal optique est converti en signal électrique lors de chaque passage dans un nœud. Dans ce type de réseau, la couche optique sert uniquement à transporter un signal optique entre deux nœuds du réseau. Ce réseau ne permet pas d'établir une connexion optique de bout à bout.

#### 2.3.2 Réseau optique transparent

Le terme transparence peut être lié à plusieurs critères. En général, il concerne le débit binaire, le format de modulation ou le format de transmission (SDH, PDH...). Dans le cas des réseaux optiques, la transparence désigne l'absence de conversion opto-électronique dans le réseau. Dans un réseau où la transparence est liée à la conversion opto-électronique, le signal optique n'est pas converti en un signal électrique dans chaque nœud, donc le réseau sera plus efficace. En effet, grâce à l'absence de conversion opto-électronique, le signal est transmis plus rapidement donc globalement le débit du réseau sera plus élevé, et la consommation électrique est réduite. Il permet l'établissement d'une connexion optique de bout à bout.

#### 2.3.3 Caractéristiques des différentes parties d'un réseau de télécommunication

Les réseaux de télécommunication peuvent être subdivisés en trois parties : le réseau cœur, le réseau métropolitain et le réseau d'accès [Zha10]. La figure 2.6 illustre ces différentes parties. Ces différentes parties se caractérisent par la dimension de leurs zones de couverture, les technologies mises en œuvres et les fonctions à remplir. Nous présentons ci-dessous les caractéristiques principales des différentes parties d'un réseau de télécommunication.

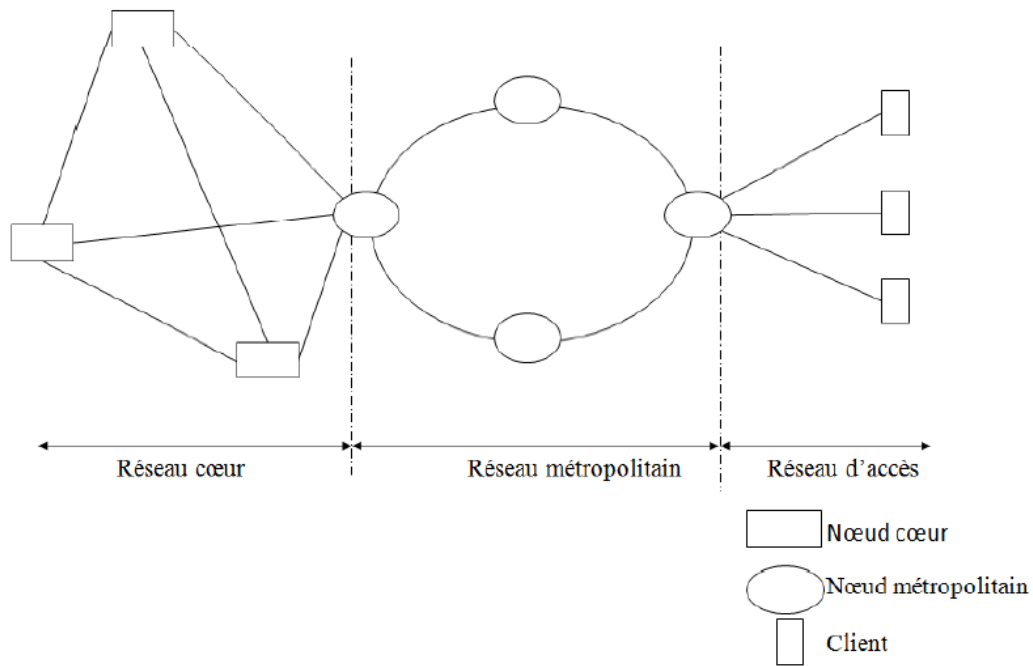


Fig. 2.6 : Différentes parties d'un réseau de télécommunication optique

### 2.3.3.1 Réseau d'accès

Le réseau d'accès permet de raccorder les clients aux réseaux des opérateurs. Différentes technologies sont utilisées dans la conception des réseaux d'accès, notamment xDSL, wifi, Ethernet ou FTTx. Le réseau d'accès a une portée de moins de 10 km.

### 2.3.3.2 Réseau métropolitain

Le réseau métropolitain permet de collecter le flux d'information venant des réseaux d'accès et de le transmettre au réseau cœur ou de distribuer le flux reçu aux différents réseaux d'accès. Le réseau métropolitain a une portée comprise entre 10 et 100 km.

### 2.3.3.3 Réseau cœur

Le réseau cœur permet d'interconnecter les réseaux métropolitains entre eux, à l'Internet et aux différents réseaux des autres opérateurs. Le réseau cœur a typiquement une topologie maillée et s'étend sur plusieurs kilomètres. Il utilise les technologies les plus récentes pour accroître le débit de transmission car il supporte une grande quantité de données du réseau. Il a une portée de plus de 100 km.

Nous retenons de la description des réseaux effectuées ci-dessus que les réseaux optiques transparents permettent d'accroître de façon générale le débit de la transmission comparé aux réseaux optiques opaques. Nos travaux se déploient dans la partie optique des réseaux où est établie une connexion optique de bout à bout notamment les réseaux optiques transparents.

### 2.4 Réseaux optiques WDM

Avec l'explosion des données de l'Internet, les fournisseurs doivent augmenter les débits de transmission entre différents sites. Les solutions qui s'offraient étaient soit l'augmentation du nombre de fibres optiques déployées entre les différents nœuds, soit transmettre plusieurs signaux sur la même fibre. L'augmentation du nombre de fibres revenait très chère notamment avec le génie civil. Le multiplexage en longueur d'onde, WDM, se présente alors comme la solution la plus prometteuse pour augmenter le débit sans augmenter le nombre de fibres déployées. De nombreux livres, articles et thèses sur les réseaux optiques WDM indiquent clairement cela [Bea00, Blo13, Cou01, IM03, Muk06, Per05, RSVA10, Web1, Web2].

#### 2.4.1 Système de transmission avec multiplexage en longueur d'onde

A l'origine, dans les réseaux optiques, le signal optique est transmis dans un seul canal sur chaque fibre optique. Mais cette technique limitait le débit de transmission maximal pouvant être atteint sur la fibre optique. Alors le nombre de fibres optiques entre deux nœuds était multiplié pour atteindre les débits souhaités. Avec l'émergence du multiplexage en longueur d'onde (*Wavelength Division Multiplexing*, WDM), plusieurs canaux sont utilisés sur la même fibre optique. Chaque canal de la fibre optique utilise une longueur d'onde différente. Cela permet d'augmenter le débit de transmission d'une fibre sans avoir à la changer.

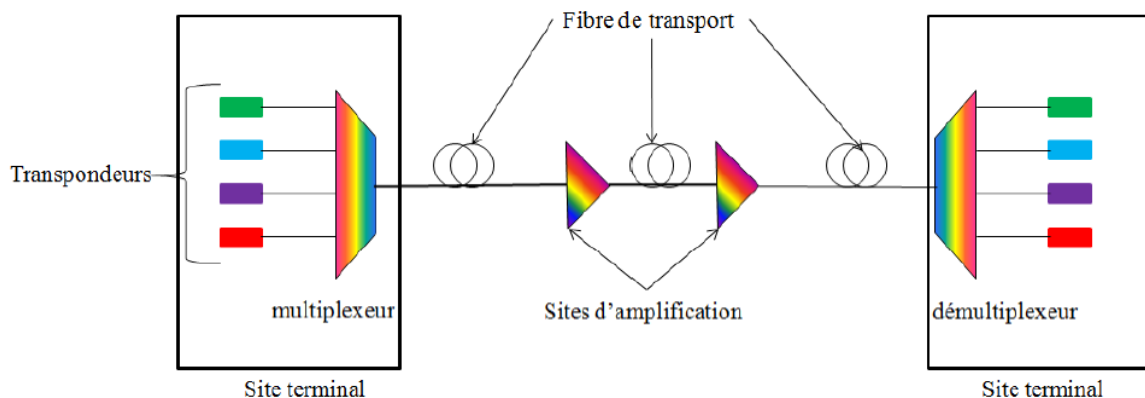


Fig. 2.7 : Système de transmission avec multiplexage en longueur d'onde



La figure 2.7 présente un système de transmission avec multiplexage en longueur d'onde. Ce système est constitué de sites terminaux qui sont composés de transpondeurs, multiplexeurs/démultiplexeurs et de site intermédiaires qui sont composés d'amplificateurs.

### 2.4.1.1 *Transpondeur*

Un transpondeur est un composant remplissant les fonctions d'émission et de réception du signal optique sur une longueur d'onde particulière, dans un système de transmission avec multiplexage en longueur d'onde. Il est bidirectionnel, équipé d'émetteurs optiques dans un sens et de récepteurs optiques dans l'autre pour remplir ces fonctions.

### 2.4.1.2 *Amplificateur*

L'amplificateur à fibre dopée à l'erbium ou EDFA (*Erbium Doped Fiber Amplifier*) est le plus utilisé dans un système de transmission avec multiplexage en longueur d'onde. Ce mode d'amplification se fait sans conversion du signal optique en signal électrique. On dit que l'amplification est tout-optique. Cela constitue un premier avantage pour ce type d'amplification. Un autre avantage est qu'il est compatible avec le multiplexage WDM. Il amplifie toutes les longueurs d'onde sur une bande passante d'environ 30 nm (bande C) centrée autour de 1550 nm [YT02].

### 2.4.1.3 *Multiplexeur/Démultiplexeur*

Le multiplexeur optique permet de remplir la fonction de multiplexage et le démultiplexeur permet de faire l'opération inverse. Le multiplexeur optique permet de combiner et de transmettre les signaux optiques provenant des différents transpondeurs (avec des longueurs d'onde distinctes) sur une même fibre. Il utilise la technique de multiplexage en longueur d'onde décrite dans la section 2.2.2. Le démultiplexeur sépare les différents signaux optiques reçus d'une fibre optique et les envoie sur un transpondeur de destination.

## 2.4.2 **Routage optique et allocation de longueurs d'onde**

La communication entre deux nœuds dans un réseau optique nécessite la détermination du chemin que doit emprunter le signal optique (appelé problème de routage) et l'affectation d'une longueur d'onde sur chaque lien du chemin. Ces deux problèmes constituent le problème de routage et d'affectation de longueur d'onde [Muk00, ZJM00]. Le chemin suivi par le flux d'une connexion optique est appelé chemin optique ou *lightpath*. Un *lightpath* est une structure

(canal) optique entre la source et la destination d'une connexion dans le réseau optique. Il peut être constitué de plusieurs liens (fibres optiques). La résolution de ce problème doit respecter deux contraintes:

- Continuité de longueur d'onde : Un *lightpath* doit utiliser la même longueur d'onde sur chaque lien du chemin le composant en l'absence de convertisseur de longueur d'onde.
- Unicité de longueur d'onde : Deux *lightpaths* distincts ne doivent pas utiliser la même longueur d'onde sur un lien qu'ils partagent.

### 2.4.3 Commutateurs optiques

La fonction de commutation dans les réseaux optiques est réalisée par différents types de nœuds. Nous présentons trois types de commutateurs : le répartiteur ou brasseur de fibres, le multiplexeur à insertion ou extraction et le brasseur de longueurs d'onde.

#### 2.4.3.1 Répartiteur : F-OXC

Le répartiteur ou brasseur de fibres (*Fiber Optical Cross-Connect*, F-OXC) est un équipement qui réalise la fonction de commutation entre les fibres entrant à un nœud et les fibres à la sortie de ce nœud. Le répartiteur travaille sur l'ensemble des longueurs d'onde, sans faire de distinction entre elles. Il ne permet donc pas de faire les opérations de multiplexage ou de démultiplexage. Il peut extraire ou insérer l'ensemble des longueurs d'onde de la fibre, dans leur totalité et sans distinction. La figure 2.8 présente le fonctionnement du répartiteur. Les trois longueurs d'onde (rouge, bleu et verte) de l'entrée 3 sont dirigées ensemble vers la sortie 4.

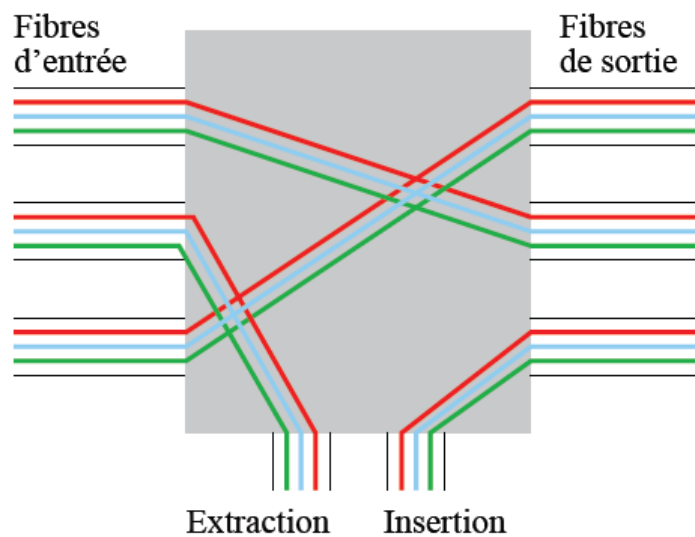


Fig. 2.8 : Répartiteur F-OXC

### 2.4.3.2 Multiplexeur à insertion/extraction : OADM

Certains signaux multiplexés sur une fibre optique qui arrivent à un nœud ont besoin d'être extraits. Certains nouveaux signaux ont besoin d'être insérés à partir d'un nœud ou des signaux extraits ont besoin d'être multiplexés avec d'autres longueurs d'ondes sur une autre fibre de sortie. Un multiplexeur à insertion/extraction (*Optical Add/Drop Multiplexer, OADM*) permet d'effectuer ces opérations d'extraction et d'insertion de longueurs d'onde. La figure 2.9 présente les composants d'un tel équipement. Il est composé d'un démultiplexeur de longueurs d'onde en entrée, d'un multiplexeur de longueurs d'onde à la sortie et de commutateurs élémentaires  $2 \times 2$  constituant une matrice de commutation à 4 groupes d'accès. Ces matrices de commutation sont les canaux entrants, les canaux sortants, les canaux extraits et les canaux insérés [Pie15].

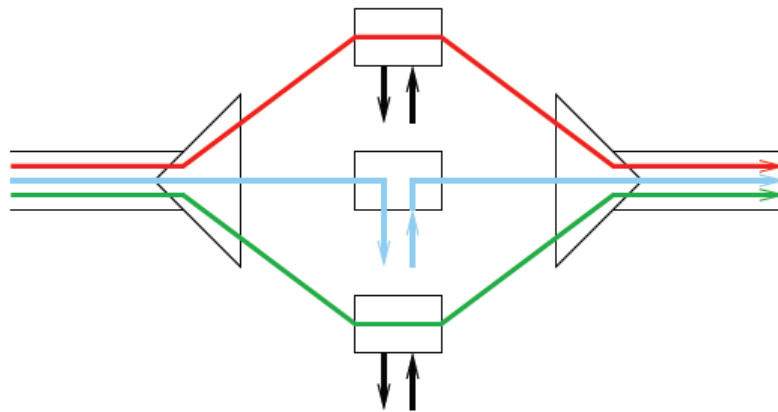


Fig. 2.9 : Multiplexeur à insertion/extraction OADM

### 2.4.3.3 Brasseur : WR-OXC

Un brasseur de longueurs d'onde (*Wavelength Routing Optical Cross-Connect WR-OXC*) est un équipement qui réalise la commutation de longueur d'onde de façon sélective. Il permet de commuter une longueur d'onde entrante sur un port d'entrée d'un nœud vers un des ports de sortie du nœud, indépendamment des autres longueurs d'onde. La figure 2.10 présente la composition d'un brasseur de longueurs d'onde.

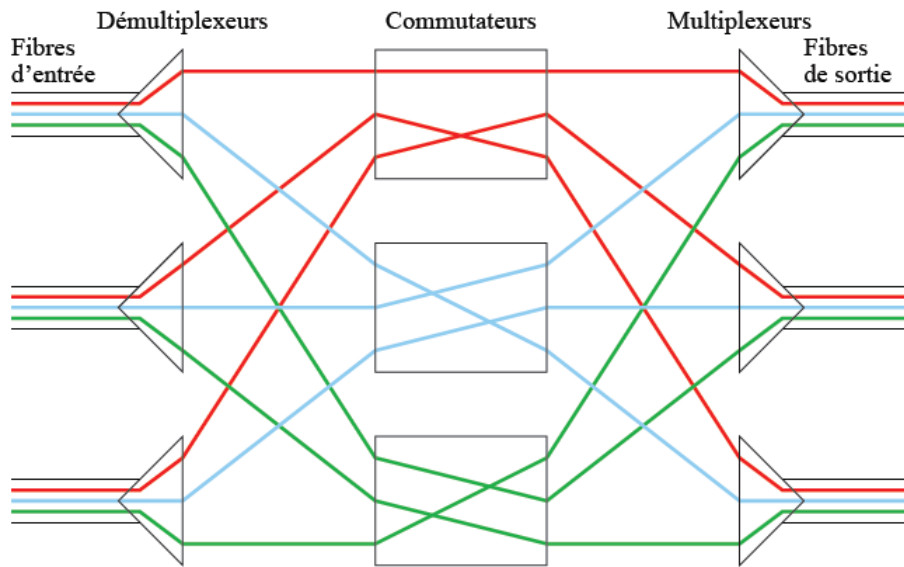


Fig. 2.10 : Brasseur de longueurs d'onde

## 2.5 Convertisseur optique

Un convertisseur optique est un équipement qui permet de s'affranchir de la contrainte de continuité de longueur d'onde. Considérons qu'une longueur d'onde est identifiée par une couleur sur la fibre. La contrainte de continuité de longueur d'onde impose qu'une longueur d'onde commutée à la sortie d'un nœud conserve la même couleur que la longueur d'onde à l'entrée du nœud. Deux principales technologies sont utilisées dans les convertisseurs :

- La conversion opto-électronique : le signal optique est converti en signal électrique avant le changement de longueur d'onde à la sortie ;
- La conversion tout-optique : le signal demeure dans le domaine optique durant la conversion.

## 2.6 Conclusion

Ce chapitre a présenté dans la première partie le système de transmission par fibre optique et l'utilisation de ce système dans un réseau de télécommunication. Dans la seconde partie, le réseau optique WDM qui est largement déployé car présentant de nombreux avantages est décrit. La description de ce réseau est importante car la suite de ce mémoire est basée essentiellement sur ce réseau.

Une intense activité de recherche se déroule autour des problématiques que posent les réseaux optiques WDM afin d'en améliorer les performances. Les problèmes tels que l'utilisation optimale de la bande passante de la fibre optique conduisent à des recherches sur des réseaux optiques dits élastiques. Ce type de réseau introduit une certaine flexibilité dans la gestion de la bande passante, à la différence des réseaux WDM qui attribuent de façon fixe les longueurs d'onde sur la fibre. La protection de connexion, le groupage de trafic, le dimensionnement du réseau, le routage et l'affectation de longueur d'onde sont quelques problèmes posés par ces réseaux. Le problème qui nous intéresse dans ce mémoire est la reconfiguration du routage multicast. Le chapitre suivant s'intéresse au processus de communication entre les utilisateurs d'un réseau optique. Nous allons présenter le routage dans les réseaux optiques WDM, en particulier le routage multicast.

## 3. Routage multicast dans les réseaux optiques WDM

---

### Sommaire

---

|  |    |
|--|----|
| 3.1 Introduction.....                                    | 22 |
| 3.2 Avantages du multicast.....                          | 23 |
| 3.3 Multicast dans le domaine électronique.....          | 23 |
| 3.4 Multicast optique.....                               | 24 |
| 3.4.1 Commutateurs optiques supportant le multicast..... | 24 |
| 3.4.2 Routage multicast optique.....                     | 29 |
| 3.4.3 Allocation de longueur d'onde.....                 | 32 |
| 3.5 Contrôle de connexion.....                           | 33 |
| 3.6 Conclusion.....                                      | 33 |

---

### 3.1 Introduction

Les applications telles que la télévision sur IP (IPTV), la télévision à haute définition sur Internet (HDTV), la vidéoconférence, les jeux en réseau ou les applications à la demande qui sont des applications multi-utilisateurs connaissent aujourd'hui une croissance importante. Ces applications nécessitant le *multicasting* sont très gourmandes en bande passante. Le *multicasting* consiste à transmettre un même message à plusieurs destinataires. Dans les réseaux ne supportant que les communications point-à-point, le *multicasting* est réalisé en envoyant séparément une copie de la donnée de la source à chaque destinataire. Dans cette technique, la duplication de la donnée est effectuée par la source autant de fois qu'il y a de destinataires. Cette technique consomme une grande bande passante dès que le nombre de destinataires augmente et encombre la source de la requête. L'introduction du multicast dans les réseaux permet de résoudre le problème de consommation de bande passante posé par ces applications nécessitant le *multicasting*. Dans ce chapitre, nous présentons d'abord les avantages du multicast dans la section 3.2. Ensuite nous décrivons le multicast dans le domaine électronique et présentons les principaux protocoles nécessaires à sa mise en œuvre dans la section 3.3. La section 3.4 présente le multicast dans le domaine optique. Dans cette section, les composants utilisés pour la mise œuvre du multicast dans les réseaux optiques sont présentés. Le problème de routage et d'affectation de longueurs d'onde pour une

communication multicast y est décrit et les solutions existantes sont présentées. Enfin le contrôle d'une connexion multicast dans un réseau optique est présenté dans la section 3.5.

### 3.2 Avantages du multicast

Le multicast est une technique très efficace pour réduire la consommation en bande passante des applications multi-utilisateurs. Il consiste à confier la charge de duplication de la donnée aux nœuds du réseau lorsque cela est nécessaire. Dans ce cas, le chemin suivi par les données est un arbre multicast. L'arbre multicast permet d'économiser de la bande passante sur une partie des chemins parcourus et évite à la source de dupliquer la donnée. La figure 3.1 présente un exemple de transmission unicast et de transmission multicast. Elle est une illustration de l'économie de bande passante réalisée avec l'utilisation du multicast pour la transmission de la même donnée d'une source vers plusieurs destinataires. Dans la figure 3.1.a, trois connexions unicast sont utilisées pour atteindre les destinations  $D_1$ ,  $D_2$  et  $D_3$  à partir de la source  $S$ . Dans la figure 3.1.b, une connexion multicast est utilisée pour atteindre les mêmes destinations à partir de la même source  $S$ . On remarque une économie de ressources sur les liens entre les nœuds  $S$  et  $RM$  en utilisant le multicast.

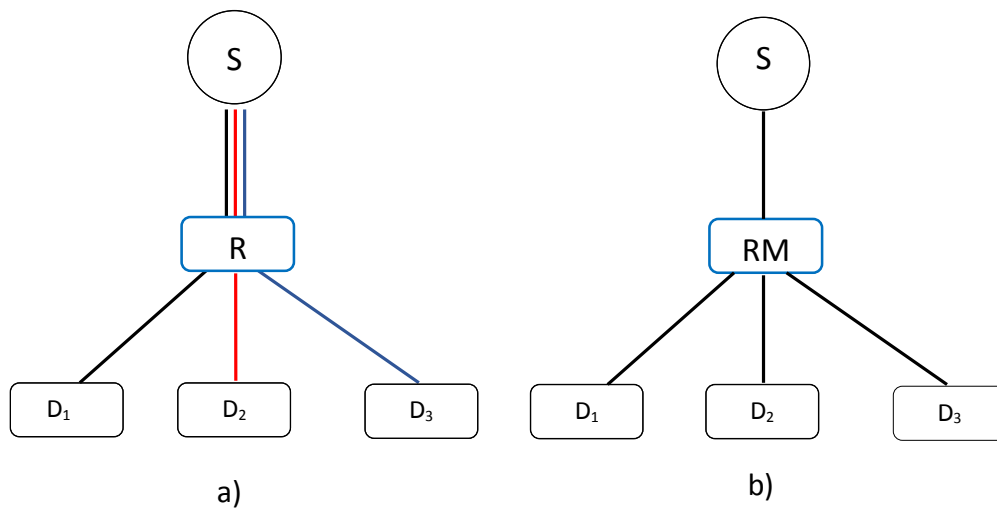


Fig. 3.1 : Transmission unicast et transmission multicast.

### 3.3 Multicast dans le domaine électronique

Dans le domaine électronique tel que le routage multicast IP, ce sont les routeurs IP qui se chargent de dupliquer les paquets à transmettre aux différents récepteurs. La duplication des paquets par le réseau est menée de telle sorte que le chemin suivi par les paquets soit un arbre.

La mise en œuvre du routage multicast IP nécessite certaines fonctions. Le routage multicast IP nécessite un certain type d'adresse de destination particulier, notamment une adresse IPv4 de classe D ou une adresse IPv6 de préfixe FF00::/8 [HD06]. Le protocole de routage multicast permet de construire l'arbre de diffusion des paquets, c'est-à-dire le chemin que vont suivre les paquets de la source aux différents destinataires. Les principaux protocoles de routage multicast sont le protocole PIM (*Protocol Independant Multicast*) [FHH06] et le protocole CBT (*Core-based Tree*) [Bal97]. On peut utiliser un protocole spécifique pour la gestion des destinataires du groupe multicast. Ce protocole gère les échanges entre les destinataires et les routeurs IP auxquels ils sont reliés. Le protocole utilisé est IGMP (*Internet Group Management Protocol*) [Fen97]. Dans l'Internet, les fonctions de duplication des paquets multicast, et de routage multicast sont réalisées dans le domaine électronique.

## 3.4 Multicast optique

Le concept de multicast peut être étendu aux réseaux optiques WDM pour en améliorer les performances [CLT08, DP03, HCT02, HSZ04, MZQ98, Rou03, SM99].

### 3.4.1 Commutateurs optiques supportant le multicast

Il existe deux types de modèles de commutateurs optiques supportant le multicast dans les réseaux optiques. On distingue les commutateurs opaques et les commutateurs transparents.

#### 3.4.1.1 Commutateur optique opaque

Un commutateur optique opaque nécessite la conversion du signal optique reçu sur un port en signal électrique avant toute opération de commutation. Il nécessite une conversion du signal optique en signal électrique et ensuite du signal électrique en signal optique (conversion O-E-O). La figure 3.2 présente un commutateur opaque et la figure 3.3 présente un commutateur opaque supportant le multicast.

Dans le cas opaque, les commutateurs sont de même taille et cette taille est donnée par :

$$(D + l). W \times (D + l). W$$

Les notations utilisées sont :

$D$  : Nombre de fibres

$W$  : Nombre de longueurs d'onde par fibre



$l$  : Nombre de ports d'insertion et d'extraction.

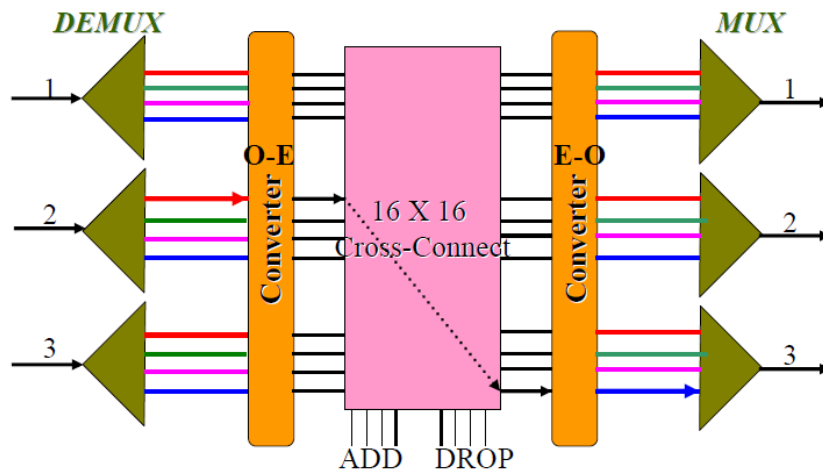


Fig. 3.2 : Commutateur OXC opaque

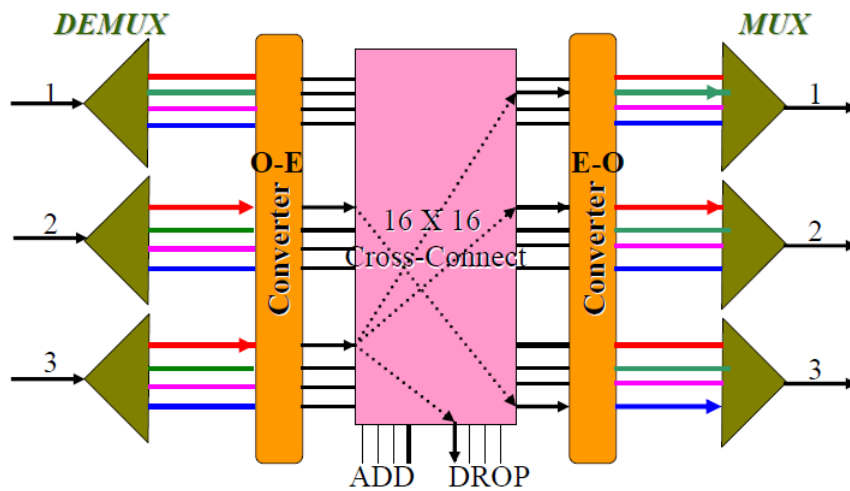


Fig. 3.3 : Commutateur OXC opaque supportant le multicast

### 3.4.1.2 Commutateur optique transparent

La figure 3.4 représente un exemple de commutateur optique capable de commuter le signal entièrement dans le domaine optique. Ce commutateur ne nécessite pas de conversion O-E-O pour effectuer la commutation. Un tel commutateur est dit commutateur transparent. Pour qu'il soit capable de supporter le multicast tout-optique, il doit être équipé de splitters optiques. Un signal entrant sur un port d'entrée du splitter avec une longueur d'onde sera dupliqué à la sortie du splitter sur plusieurs ports de sortie comme l'illustre la figure 3.5. Dans les figures 3.4 et 3.5, X et Y représentent des splitters et OSW représente un commutateur optique. Un signal ne nécessitant pas d'être dupliqué (par exemple un signal d'une connexion unicast) ne passe pas

par les splitters comme le montre le signal émis sur la longueur d'onde en rouge de la fibre 1 de la figure 3.5.

Ce nœud optique nécessite un amplificateur pour compenser l'atténuation du signal due à la duplication du signal par le splitter. On peut également y intégrer un convertisseur de longueur d'onde pour augmenter la probabilité d'établir les connexions multicast.

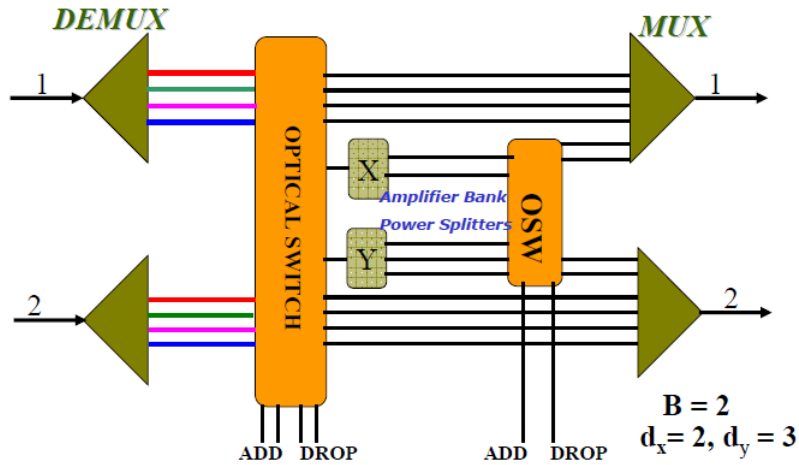


Fig. 3.4 : Commutateur OXC transparent

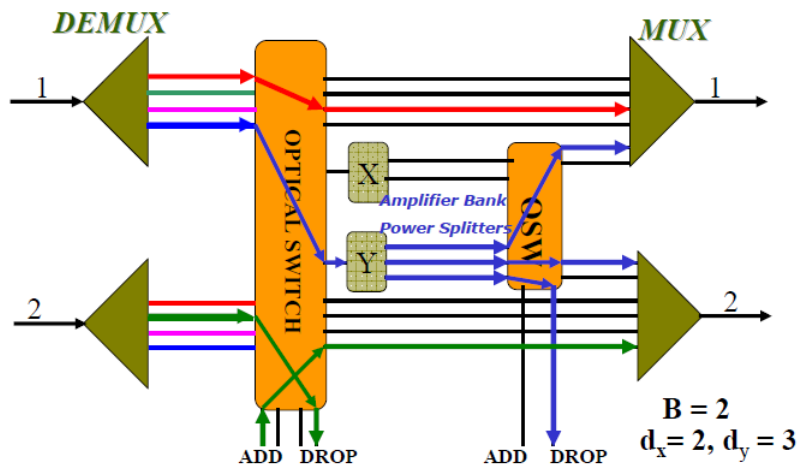


Fig. 3.5 : Commutateur OXC transparent supportant le multicast

Dans le commutateur OXC transparent supportant le multicast, les commutateurs utilisés n'ont pas la même taille [Muk06]. Dans la figure 3.5, deux commutateurs sont utilisés. La taille du plus grand commutateur est donnée par :

$$(D.W + l) \times (D.W + l + B)$$

La taille du petit commutateur est donnée par :

$$(B \cdot d + L - l) \times (B \cdot d + L - l)$$

Les notations utilisées sont :

$d_i$ : Degré de duplication du splitter  $i$

$B$  : Nombre de splitters

$L$  : Nombre total de ports d'insertion ou d'extraction

$l$  : Nombre de ports d'insertion ou extraction du grand commutateur

$D$  : Nombre de fibres

$W$  : Nombre de longueurs d'onde par fibre

#### 3.4.1.3 *Arbre multicast optique*

Dans les réseaux optiques, le chemin suivi par le trafic multicast est un arbre optique appelé *light-tree*. C'est la généralisation de la notion de chemin optique ou *lightpath* utilisée dans le cas unicast (communication point-à-point). Un *light-tree* est une structure tout-optique ayant une source et plusieurs destinataires et qui utilise la même longueur d'onde sur chacun de ces liens. Le signal optique est envoyé sur une longueur d'onde et dupliqué par les nœuds du réseau là où c'est nécessaire. Sur la figure 3.6, la connexion multicast  $C_m(A\{C,E,F\})$  a pour source le nœud A et pour destinations les nœuds C, E et F. L'arbre (en rouge) sur la figure représente l'arbre optique de la connexion  $C_m$ . Cet arbre utilise la même longueur d'onde sur chacun de ses liens. Les contraintes de continuité de longueur d'onde et d'unicité de longueur d'onde sont également étendues aux arbres optiques.

L'établissement d'une connexion multicast dans un réseau optique nécessite d'abord la résolution de la détermination du chemin entre la source et les différentes destinations ; c'est le problème de routage. Ensuite, allouer aux différents liens du chemin déterminé une longueur d'onde en respectant les contraintes de continuité et d'unicité de longueur d'onde ; c'est le problème d'affectation de longueur d'onde. Pour obtenir une solution optimale, ces deux problèmes doivent être résolus ensemble et sont connus sous le nom de problème de routage et d'affectation de longueur d'onde.

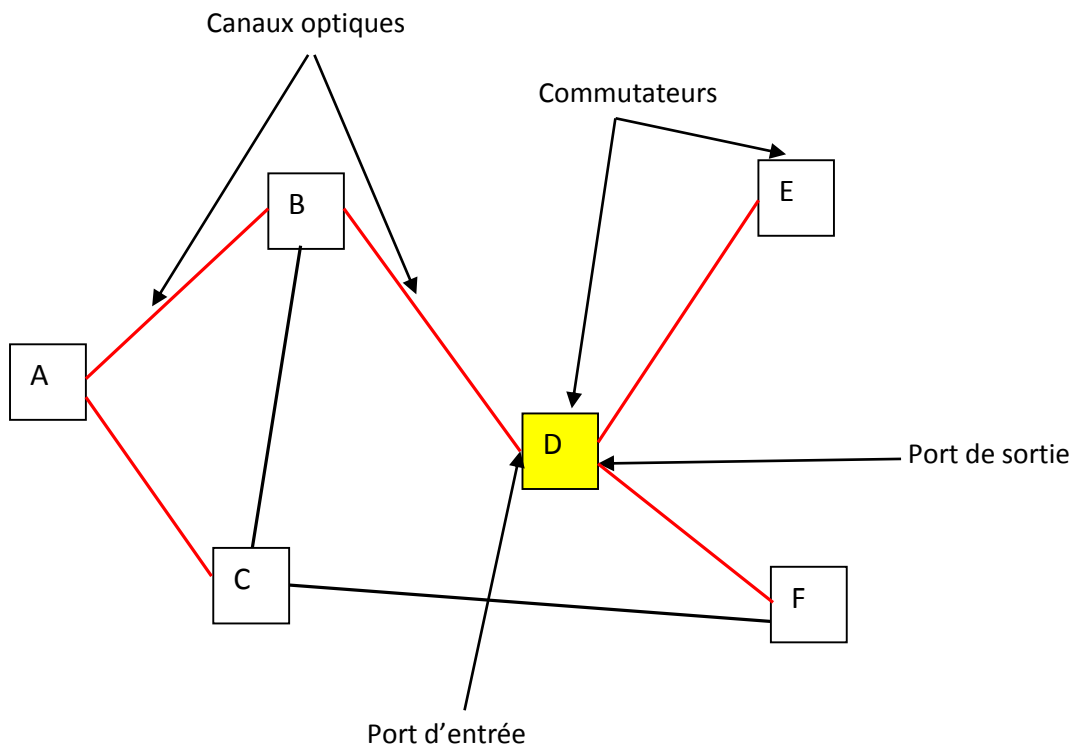


Fig. 3.6 : Exemple d'arbre optique.

Un canal optique sur un arbre optique est une longueur d'onde utilisée sur un lien pour la transmission du signal optique sur cet arbre. On peut dire qu'un canal optique alimente un autre si le port d'entrée de l'un et le port de sortie de l'autre appartiennent au même commutateur et que l'élément de commutation est configuré pour propager le signal venant du port d'entrée vers le port de sortie. Sur l'arbre optique de la figure 3.6, le canal optique (B – D) alimente les canaux optiques sur les liens (D – E) et (D – F).

Un canal optique peut être partagé par plusieurs chemins optiques pour alimenter plusieurs ports de sortie d'un nœud. Un tel canal optique permet d'atteindre plusieurs destinataires d'une connexion multicast. Le canal optique sur le lien (B – D) est partagé par le chemin entre A et la destination E et le chemin entre A et la destination F pour atteindre les destinations E et F.

Pour qu'un nœud optique ait la capacité de dupliquer le signal optique entrant sur un port d'entrée vers un ou plusieurs ports de sortie, il doit être équipé d'un splitter.

#### 3.4.2 Routage multicast optique

Le splitter est le composant de base d'un réseau optique WDM supportant le multicast. Le splitter est un composant optique ayant une fabrication compliquée et un coût élevé. Donc les nœuds des réseaux supportant le multicast ne sont pas tous équipés de splitters ou le nombre de splitters n'est pas très élevé dans la grande majorité des nœuds d'un réseau. On distingue deux types de nœuds dans le réseau selon qu'ils ont la propriété de dupliquer le signal ou non.

- Nœud MC : Les nœuds MC sont les nœuds équipés de splitters. Ces nœuds ont la propriété de dupliquer le signal optique entrant sur les ports de sortie.
- Nœud MI : les nœuds MI ne sont pas équipés de splitters. Ces nœuds ne peuvent pas dupliquer le signal optique reçu.

La conversion de longueur d'onde favorise plus de flexibilité dans la construction d'arbre optique mais a un coût élevé. Donc de nombreux travaux se sont intéressés à la construction de l'arbre optique dans un réseau où certains nœuds seulement sont équipés de splitters et sans conversion de longueur d'onde.

##### 3.4.2.1 Routage multicast optique

Le problème de routage multicast est un sous-problème du problème de routage multicast et d'affectation de longueur d'onde (*Multicast Routing and Wavelength Assignment, MC-RWA*). De nombreuses approches dans la littérature résolvent ce problème en deux parties : la détermination du routage multicast et l'affectation des longueurs d'onde.

La détermination du routage multicast consiste à déterminer un arbre de diffusion entre les membres du groupe optimisant la diffusion des messages. Cela revient à déterminer dans une première approche, un arbre de diffusion de coût minimal qui couvre les membres du groupe. Il est appelé arbre couvrant partiel minimum du groupe. Le problème qui consiste à déterminer un arbre couvrant partiel minimum est appelé problème de Steiner. Le routage multicast consistant à déterminer l'arbre minimum de Steiner (*Minimum Steiner Tree, MST*) est un problème NP-Complet dans le cas général [Kar72]. Lorsque l'arbre est couvrant total alors la complexité du problème est polynomiale. Pour le cas général, on a recours à des heuristiques pour approcher la solution de ce problème qui n'est donc pas optimale.

La seconde approche de détermination de routage multicast consiste à calculer l'arbre des plus courts chemins entre la source et chaque destination du groupe multicast. La complexité du

problème est polynomiale dans le cas général. L'algorithme de Dijkstra (*Shortest Path Tree*, SPT) permet d'obtenir la solution optimale.

Dans les réseaux optiques WDM, la détermination du routage pour une connexion multicast consiste à déterminer un arbre optique enraciné à la source de la connexion et couvrant les membres du groupe. Celle-ci se réalise à partir de la connaissance de la topologie du réseau et des membres du groupe. Les heuristiques basées sur les approches d'arbre couvrant minimum cherchent à minimiser le coût total de l'arbre. Les algorithmes basés sur l'algorithme du plus court chemin d'abord cherchent à déterminer l'arbre des plus courts chemins entre la source et chaque destination du groupe multicast.

Deux types d'algorithme de routage existent: les algorithmes de routage statique et les algorithmes de routage dynamique. Dans le routage multicast statique, l'arbre optique d'une connexion multicast est déterminé en avance, sans tenir compte de l'utilisation des liens et de la charge du réseau. Cette approche s'avère souvent inefficace car certains liens peuvent être sous-utilisés tandis que d'autres sont surchargés à un instant donné. De plus, la distribution de l'utilisation des liens change de façon dynamique. Ainsi un algorithme de routage ne tenant pas compte de l'état courant du réseau ne peut pas garantir une bonne performance du réseau. Une demande de connexion est bloquée si aucun chemin parmi ceux déjà calculés ne permet de l'établir.

Les algorithmes de routage dynamique déterminent les chemins en tenant compte de l'utilisation courante des ressources du réseau. Dans ce cas, une demande de connexion est bloquée si les ressources disponibles lors de la demande ne permettent pas de l'établir.

Le problème de routage multicast et affectation de longueurs d'onde (*Multicast Routing and Wavelength Assignment*, MC-RWA) est étudié selon différents contextes.

- MC-RWA dans un réseau où tous les nœuds sont équipés de splitters

Dans ce problème, la construction de l'arbre MST se fait en utilisant tout nœud du réseau. Le routage multicast statique (*static MC-RWA*) se subdivise en deux classes : routage fixe et routage alternatif [HRW92]. Si un seul arbre MST est calculé pour une connexion multicast, on parle de l'approche fixe. Si deux arbres MST ou plus sont calculés pour une connexion multicast, on parle de l'approche alternative.

Dans le routage multicast dynamique, l'utilisation des ressources (des longueurs d'onde sur chaque lien du réseau) est déterminée. L'algorithme détermine par la suite, dans l'ensemble des topologies de longueurs d'onde, celle qui optimise le coût de l'arbre MST construit à l'instant de la demande de connexion.

Le routage multicast statique est plus simple que le routage multicast dynamique. Mais le routage dynamique produit de meilleures performances (réduction du taux de blocages de demandes de connexion) que le routage statique.

- *MC-RWA dans un réseau où certains nœuds sont équipés de splitters*

Lorsque certains nœuds du réseau ne sont pas équipés de splitters, la construction de l'arbre MST se fait en utilisant au mieux les (rares) nœuds ayant la capacité de dupliquer le signal (nœuds MC) et en évitant ceux n'ayant pas cette capacité (nœuds MI). Les heuristiques existant peuvent être regroupées en deux catégories [DP03].

La première catégorie construit l'arbre optique à partir de l'arbre multicast existant, il est construit à partir de la couche IP. Les protocoles appartenant à cette catégorie ont besoin d'échanger avec les protocoles de routage de la couche IP. Les nœuds peuvent échanger avec le protocole GSMP.

La seconde catégorie construit l'arbre multicast directement dans la couche WDM. Certains protocoles appartenant à cette couche nécessitent une connaissance globale de la topologie du réseau. Un seul arbre peut être insuffisant pour couvrir l'ensemble des nœuds destinataires du groupe multicast. Dans ce cas, un ensemble d'arbres enracinés à la source de la connexion est utilisé. Cet ensemble d'arbres est appelé forêt multicast.

Les auteurs de [ZWQ00] présentent quatre algorithmes de calcul de forêt multicast appelés, respectivement, *Reroute-to-Source*, *Reroute-to-Any*, *Member-First* et *Member-Only*.

Les algorithmes *Reroute-to-Source*, *Reroute-to-Any*, *Member-First* sont basés sur le calcul de l'arbre des plus courts chemins entre la source et chaque destinataire du groupe multicast (*Shortest Path Tree*, SPT). L'algorithme *Member-Only* est basé sur le calcul d'une approximation de l'arbre minimum de Steiner à partir de l'arbre couvrant minimum. Dans [Zho10], les auteurs présentent un algorithme appelé *Avoidance of MIB Nodes for multicast routing* qui est une amélioration de l'algorithme *Member-Only*.

#### 3.4.3 Allocation de longueur d'onde

L'allocation de longueur d'onde consiste à affecter une longueur d'onde à chaque lien (fibre optique) composant l'arbre multicast (ou la forêt multicast) déterminé en résolvant le problème de routage multicast. Lorsque le nombre de connexions multicast devient important, le problème d'allocation de longueur d'onde devient difficile à résoudre. Ce problème est NP-Complet [SA98]. La difficulté de ce problème est fonction du nombre de longueurs d'onde disponibles sur les liens du réseau et de la capacité de conversion des nœuds du réseau. Si les liens du réseau disposent de suffisamment de longueurs d'onde, il n'y a pas de conflit de longueurs d'onde.

Si tous les nœuds du réseau ont la capacité de conversion d'une longueur d'onde en toute autre longueur d'onde (conversion totale de longueurs d'onde), alors toute longueur d'onde peut être affectée à tout lien d'un arbre multicast, en respectant la contrainte d'unicité des longueurs d'onde.

Si la conversion de longueur d'onde n'est pas autorisée dans le réseau, alors la même longueur d'onde doit être utilisée sur tous les liens de l'arbre. En considérant séparément les longueurs d'onde sur chaque lien du réseau, on a un ensemble de topologies disjointes. Chaque topologie constituée par la même longueur d'onde et les nœuds du réseau, représente la disponibilité des longueurs d'onde sur chaque lien. Le problème ici va consister à exhiber l'arbre optique dans l'un de ces réseaux. [SA98] présente un algorithme d'affectation de longueur d'onde lorsque la conversion de longueur d'onde n'est pas autorisée.

Dans la majorité des problèmes d'affectation de longueur d'onde, on suppose que les nœuds du réseau ont la propriété de conversion partielle de longueur d'onde. Cela signifie que le nombre de nœuds ayant la capacité de conversion est limité. Les travaux présentés dans [JDHLG01, ZWQ00] se sont intéressés à ce problème. [ZWQ00] propose un algorithme basé sur la subdivision de l'arbre en plusieurs segments. Un segment est dans cette approche une suite de liens. Ces segments sont déterminés en supprimant du graphe tous les nœuds intermédiaires qui n'ont pas la propriété de conversion de longueur d'onde et en conservant les liens entre les nœuds. La suite de liens entre deux nœuds du nouveau réseau obtenu est un segment sur lequel sera assignée la même longueur d'onde. Enfin l'algorithme First-Fit est utilisé pour assigner les longueurs d'onde sur chacun de ces segments.



[JDHLG01] propose un algorithme d'affectation de longueur d'onde en transformant le problème d'affectation de longueur d'onde sur un arbre en problème de coloration de graphe.

## 3.5 Contrôle de connexion

Le contrôle de connexion est utile pour l'établissement ou la suppression de connexion surtout pour la reconfiguration d'un routage qui consiste à établir et supprimer des routes. Pour l'établissement d'une connexion, le mécanisme de contrôle doit d'abord déterminer le chemin, ensuite assigner les longueurs d'onde et enfin envoyer les messages de configuration aux différents nœuds. Le mécanisme de contrôle doit connaître les informations sur l'utilisation courante des ressources du réseau notamment les longueurs d'onde. Le contrôle des connexions se fait selon deux approches: l'approche centralisée ou l'approche distribuée. L'approche centralisée est plus facile à mettre en œuvre mais l'approche distribuée est plus robuste. [Wei02] présente les principales architectures, le contrôle et la gestion des réseaux IP sur les réseaux optiques WDM. [PPJ11] présente les principaux résultats du contrôle et la gestion des réseaux internet de nouvelle génération. [He14] présente un état de l'art du contrôle et la gestion des communications optiques.

## 3.6 Conclusion

Nous avons présenté dans ce chapitre les avantages du multicast dans les réseaux. L'apport principal du multicast dans les réseaux est qu'il permet de réduire la bande passante utilisée par les applications dont les mêmes données sont destinées à plusieurs utilisateurs simultanément. La mise en œuvre du multicast dans les réseaux optiques WDM nécessite d'équiper les nœuds d'un splitter, composant utile pour la duplication des signaux optiques par les nœuds du réseau. Nous avons présenté les problèmes les plus importants liés à la communication multicast dans les réseaux optiques WDM, notamment le routage et l'affectation de longueur d'onde, ainsi que les algorithmes permettant de les résoudre et le contrôle de connexion dans ces réseaux.

Les connexions, une fois établies dans le réseau, peuvent être utilisées par les applications désireuses de communiquer. Mais certains phénomènes tels que l'ajout ou la suppression de nouvelles ressources du réseau, les pannes de liens ou de nœuds, les opérations de maintenance et les surcharges peuvent dégrader les performances de ces réseaux. Alors la reconfiguration d'un tel réseau permet d'améliorer ces performances. Le chapitre suivant aborde la

### **3. Routage multicast dans les réseaux optiques WDM**

reconfiguration du routage d'un réseau optique en général, et du routage multicast en particulier.

## 4. Reconfiguration du routage dans les réseaux optiques WDM

---

### Sommaire

---

|  |    |
|--|----|
| 4.1 Introduction.....  | 35 |
| 4.2 Phases de déroulement de la reconfiguration d'un réseau..... | 36 |
| 4.3 Reconfiguration du routage unicast.....                      | 37 |
| 4.3.1 Problèmes de reconfiguration du routage unicast.....       | 37 |
| 4.3.2 Algorithme de reconfiguration.....                         | 40 |
| 4.4 Reconfiguration du routage multicast.....                    | 47 |
| 4.4.1 Problème de reconfiguration du routage multicast.....      | 48 |
| 4.4.2 Revue de la littérature.....                               | 50 |
| 4.5 Conclusion.....  | 53 |

---

### 4.1 Introduction

Dans un réseau optique à multiplexage en longueur d'onde, un chemin du réseau ainsi qu'une longueur d'onde sur chaque lien sont attribués à chaque demande de connexion. Cela constitue le routage optique d'une connexion. La détermination du routage optique est faite sous les contraintes de continuité de longueurs d'onde et d'unicité de longueur d'onde sur chaque fibre. Dans ce chapitre, nous considérons le réseau et les connexions optiques qui y sont établies.

La reconfiguration du réseau est utilisée par les opérateurs de réseau pour optimiser les performances du réseau et l'acheminement du trafic. C'est un mécanisme très important ajoutant une flexibilité dans la gestion du réseau. Elle répond à des phénomènes perturbateurs tels que la surcharge, une panne du réseau, un ajout ou une suppression de composants du réseau, ou encore une solution d'économie d'énergie dans le réseau. La reconfiguration du réseau est dénommée ingénierie de topologie [LM04].

Pour répondre aux phénomènes perturbateurs, la reconfiguration détermine un nouveau routage. La détermination de ce routage dépend non seulement de l'état courant du réseau, des objectifs de la reconfiguration mais aussi de la nature des connexions qui sont établies dans le réseau. On distingue les connexions unicast et les connexions multicast qui sont fréquemment

utilisées par les applications de l'Internet. Chaque type de connexion ayant ses caractéristiques, la reconfiguration du routage dans un réseau devra tenir compte du type de connexion établie. Nous allons présenter dans la suite la reconfiguration du routage unicast et la reconfiguration du routage multicast. Nous nous intéressons dans cette thèse à la reconfiguration du routage multicast que nous allons détailler. La section 4.2 de ce chapitre présente les différentes phases de déroulement de la reconfiguration d'un réseau. La section 4.3 présente les problèmes liés à la reconfiguration du routage unicast et des algorithmes de résolution. La section 4.4 présente les problèmes liés à la reconfiguration du routage multicast et les techniques de résolution existantes dans la littérature.

### 4.2 Phases de déroulement de la reconfiguration d'un routage

La reconfiguration du routage dans un réseau se déroule en trois phases séquentielles et itératives.

La première phase consiste à décider de la politique de gestion du réseau. Elle détermine quand et comment se tiendra la reconfiguration du réseau. Dans cette phase, l'événement déclencheur de la reconfiguration, la fréquence de reconfiguration et les contraintes de reconfiguration sont déterminés. Cette phase permet de choisir entre la reconfiguration sans interruption de flux, la reconfiguration avec minimisation du nombre de connexions interrompues, et la reconfiguration avec minimisation du nombre de connexions interrompues de façon simultanée. Un résultat positif à cette phase conduit à la deuxième phase.

La deuxième phase consiste à déterminer le nouveau routage optimisant les performances du réseau et l'acheminement du trafic. Le calcul du nouveau routage est étudié dans [BR99, BR01, Kar06, KP05].

La troisième phase intervient après la détermination du nouveau routage. Elle consiste à migrer du routage initial vers le nouveau routage calculé (appelé routage final) dans la deuxième phase. Cette phase consiste à déterminer comment configurer les routes des différentes connexions constituant le routage initial et final afin de respecter les contraintes de reconfiguration définies dans la première phase. Dans cette thèse, nous nous intéressons à cette troisième phase. Nous étudions comment mener la reconfiguration du routage multicast dans les réseaux optiques WDM. Les problèmes auxquels nous nous intéressons dans cette thèse sont les suivants :

#### 4. Reconfiguration du routage dans les réseaux optiques WDM

- Est-il possible de reconfigurer tout routage initial vers un routage final en utilisant une seule longueur d'onde ? Autrement dit, est-il possible de reconfigurer le routage multicast dans un réseau dont les nœuds ne sont pas équipés de convertisseurs de longueurs d'onde ?
- Quelles sont les opérations de configuration utiles et nécessaires sur les différents nœuds pour reconfigurer le routage multicast?
- Dans quel ordre configurer les nœuds afin de garantir, si cela est possible, la continuité du flux vers l'ensemble des destinataires du groupe multicast ?
- Quel sous-ensemble de nœuds doit être reconfiguré avant quel autre afin de réduire la durée de reconfiguration tout en garantissant la continuité du flux vers l'ensemble des destinataires du groupe multicast ?
- Comment reconfigurer les nœuds afin de réduire le coût des ressources additionnelles utilisées si une seule longueur d'onde ne permet pas de faire la reconfiguration ?

### 4.3 Reconfiguration du routage unicast

Considérons un réseau optique WDM modélisé par un graphe  $G = (V, E)$  où  $V$  et  $E$  représentent respectivement l'ensemble des nœuds du réseau et l'ensemble des liens entre les différents nœuds. L'ensemble des  $W$  longueurs d'onde est  $L$  et représenté par  $L = \{\lambda_1, \lambda_2, \dots, \lambda_w\}$ .

#### 4.3.1 Problèmes de reconfiguration du routage unicast

Différents problèmes de reconfiguration du routage unicast existent dans la littérature [Maz11, Wu11]. Nous allons présenter ci-dessous quelques-uns de ces problèmes.

##### Problème 1 :

Etant donné un ensemble  $C$  représentant les connexions établies dans un réseau, le routage initial  $R_0$  des connexions de l'ensemble  $C$  et le routage final  $R_z$  des mêmes connexions, le problème de reconfiguration consiste à déterminer un ordre  $O = \{c_1, c_2, \dots, c_{|C|}\}$  de reconfiguration des connexions pour basculer du routage initial  $R_0$  au routage final  $R_z$ , où  $c_i$  représente la  $i^{eme}$  connexion de  $C$  reconfigurée.

Des questions importantes se posent alors :

- Quel ordre choisir ?

#### 4. Reconfiguration du routage dans les réseaux optiques WDM

- Quels sont les paramètres importants liés aux interruptions?

La figure 4.1 décrit un exemple de ce type de problème de reconfiguration.

Notation :  $(G, H, D)$  désigne le chemin entre les nœuds G et D, passant par H.

$C_0^i$  désigne le chemin initial de la connexion  $c_i$ .

$C_z^i$  désigne le chemin final de la connexion  $c_i$ .

$c_i$  désigne la connexion  $i$  du réseau

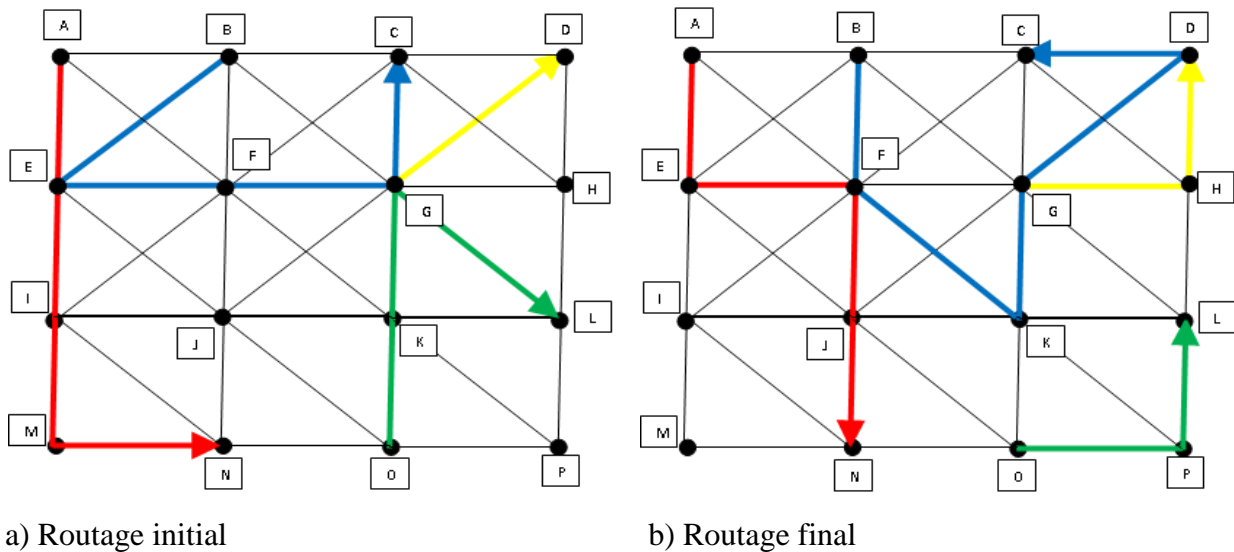


Fig. 4.1 : Exemple de problème de reconfiguration de routage unicast

Dans l'exemple de problème de reconfiguration de routage décrit à la figure 4.1, le réseau est composé de 16 nœuds ; et 4 connexions unicast y sont établies. La Fig. 4.1.a représente le routage initial de l'ensemble des connexions unicast et la Fig. 4.1.b représente le routage final de l'ensemble des mêmes connexions unicast. Toutes les connexions (et donc tous leurs chemins initiaux ou finaux) utilisent la même longueur d'onde (par exemple  $\lambda_1$ ). On suppose que la longueur d'onde  $\lambda_1$  est la seule disponible sur tous les liens entre les différents nœuds. On veut reconfigurer des chemins initiaux  $C_0^i$  de la Fig. 4.1.a vers respectivement les chemins finaux  $C_z^i$ , de la Fig. 4.1.b, pour les 4 connexions  $i = 1, \dots, 4$ . Le tableau 4.1 représente les chemins (initiaux et finaux) pour les 4 connexions.

Le problème à résoudre dans l'exemple de la figure 4.1 est la détermination d'un ordre de reconfiguration des chemins initiaux et finaux des quatre connexions afin de respecter la politique d'interruption.

Tableau 4.1 : Chemins initiaux et finaux du problème de reconfiguration de la figure 4.1

| Chemins initiaux          | Chemins finaux            |
|---------------------------|---------------------------|
| $C_0^1 = (A, E, I, M, N)$ | $C_z^1 = (A, E, F, J, N)$ |
| $C_0^2 = (B, E, F, G, C)$ | $C_z^2 = (B, F, K, G, C)$ |
| $C_0^3 = (G, D)$          | $C_z^3 = (G, H, D)$       |
| $C_0^4 = (O, K, G, L)$    | $C_z^4 = (O, P, L)$       |

**Problème 2 :**

Etant donné une topologie de réseau modélisé par un graphe  $G = (V, E)$ , une connexion  $c_1$ , le routage initial  $R_0$  de la connexion et le routage final  $R_z$  de la même connexion, le problème de reconfiguration consiste à déterminer un ordre de configuration  $Seq = \{n_1, n_2, \dots, n_{|Seq|}\}$  sur les nœuds des routages initial et final à configurer, où  $n_i$  représente le nœud configuré à la  $i^{\text{ème}}$  étape et  $Seq$  représente l'ensemble des étapes. Une étape de reconfiguration est une action de configuration sur un nœud des routages à configurer. Cette action est obtenue grâce à l'exécution d'une opération de configuration sur un nœud.

Les questions qui se posent pour ce problème sont :

- Quel ordre choisir ?
- Quels sont les paramètres importants liés aux interruptions?

La figure 4.2 décrit un exemple de problème de reconfiguration de routage unicast du type décrit ci-dessus. Le chemin initial utilisé pour la transmission du signal optique est en rouge dans cet exemple. Le chemin final sur lequel on veut basculer le flux est en bleu. Les deux chemins ont la même source B et la même destination E.

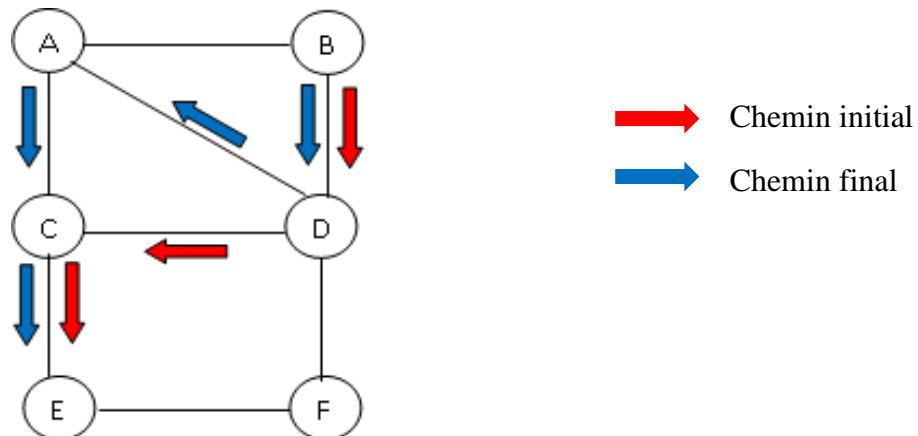


Fig. 4.2 : Exemple de problème de reconfiguration de routage unicast.

## 4. Reconfiguration du routage dans les réseaux optiques WDM

Le problème à résoudre dans cet exemple est la détermination d'un ordre de reconfiguration des différents ports des nœuds pour que le nouveau chemin utilisé pour la transmission du signal optique soit celui en bleu en respectant la politique d'interruption.

Dans les deux problèmes, l'ordre de configuration doit être déterminé de telle sorte que la reconfiguration du routage respecte la politique d'interruption. Dans le problème 1, l'ordre de configuration des connexions doit respecter la politique d'interruption et dans le problème 2, c'est l'ordre de configuration des nœuds qui doit satisfaire à cette politique. Dans le problème 1, les connexions doivent être reconfigurées une par une. Dans le problème 2, la configuration d'un nœud consiste à configurer un port de sortie de ce nœud.

Avec l'émergence des applications sensibles aux interruptions telle qu'IPTV, la politique liée aux interruptions qui est considérée dans les problèmes de reconfiguration de routage dans les réseaux à connexion est la non interruption des connexions établies dans le réseau. La reconfiguration doit être menée tout en garantissant la continuité du flux vers les destinations.

### 4.3.2 Algorithme de reconfiguration

#### 4.3.2.1 Approche Break-Before-Make (BBM)

Le *Break-Before-Make* [CCMNN10, CHS09, Cou01, Maz11] consiste à interrompre d'abord le chemin courant (chemin initial) avant d'établir le nouveau chemin optique de la connexion. Cette technique interrompt temporairement le trafic de la connexion durant la reconfiguration. Ces concepts sont standardisés pour les réseaux MPLS.

Le protocole GSMP (*General Switching Management Protocol*) [Dor02] implémente l'approche BBM. Ce protocole se présente comme candidat pour la gestion des connexions dans les réseaux optiques WDM. Une implémentation de l'approche BBM avec GSMP est de configurer tous les nœuds du chemin initial pour le supprimer, ensuite configurer tous les nœuds du nouveau chemin pour l'établir. Cette technique a pour inconvénient d'interrompre le flux de la connexion durant la reconfiguration. De plus, elle n'est pas optimale en terme de durée de reconfiguration car elle reconfigure d'une part nœud après nœud l'ensemble des nœuds à reconfigurer. D'autre part, elle reconfigure certains nœuds appartenant aux deux chemins (initial et final) ne nécessitant pas d'être reconfigurés. Cela augmente le nombre d'étapes donc ne permet pas de faire un gain sur la durée de reconfiguration.



### 4.3.2.2 *Approche Move-To-Vacant*

La technique *Move-To-Vacant* [CBL07, LL96, MM99] consiste à changer les chemins des connexions jusqu'à obtenir un routage satisfaisant. *Move-To-Vacant* (MTV) consiste d'abord à chercher des séquences de nouveaux chemins libres, à établir la connexion sur ce nouveau chemin et ensuite à libérer les ressources de l'ancien chemin. Cette manière de reconfigurer le routage n'interrompt pas le flux. Mais l'existence d'une séquence d'opérations *Move-To-Vacant* n'est pas toujours garantie dans ce cas. De plus, des questions sur la durée du processus se posent.

Une autre approche consiste à déterminer un nouveau routage satisfaisant, en minimisant le nombre de connexions ayant des chemins initial et final différents.

Certains travaux déterminent un nouveau routage satisfaisant et la séquence d'opérations *Move-To-Vacant* pour la reconfiguration. Ces approches sont basées soit sur la programmation linéaire en nombres entiers qui est difficile à résoudre lorsque la taille du réseau augmente, soit sur des heuristiques pour approcher une solution satisfaisante [BM00, GM03, SZ05, ZYWS07].

Les travaux présentés ci-dessus ne décrivent pas la manière dont la migration du routage initial au nouveau routage est concrètement effectuée. Un aspect limitant que présentent toutes ces approches est qu'il n'existe pas toujours (pour toutes les instances de problèmes donnés) une séquence d'opérations *Move-To-Vacant* pour reconfigurer le routage initial sans interrompre le trafic.

### 4.3.2.3 *Approche Make-Before-Break (MBB)*

L'approche *Make-Before-Break* [CCMNN10, Cou01, CHS09, Maz11] consiste à établir d'abord le nouveau chemin optique de la connexion avant d'interrompre le flux sur l'ancien chemin. L'utilisation de la technique *MBB* suppose la connaissance du routage initial et du routage final de la connexion.

L'algorithme *MBB* (Algorithme 4.1) consiste d'abord à préétablir le nouveau chemin entre la source et la destination de la connexion. Dans cette étape, tous les nœuds du nouveau chemin sont configurés en parallèle sauf la source. Ensuite on configure la source pour interrompre le

flux sur l'ancien chemin et alimenter le nouveau chemin. Enfin on supprime l'ancien chemin en configurant en parallèle tous les nœuds du chemin et la destination de la connexion.

#### Algorithme 4.1: Algorithme de reconfiguration *MBB*

---

*MBB* ( $P_0, P_z$ ) /\* Chemin initial  $P_0$  et chemin final  $P_z$ \*/

1. {
2. Préétablir le nouveau chemin  $P_z$  entre la source et la destination. Tous les nœuds du nouveau chemin sont configurés en parallèle sauf la source.
3. Configurer la source pour interrompre le flux sur l'ancien chemin et alimenter le nouveau chemin.
4. Supprimer l'ancien chemin  $P_0$ . Tous les nœuds de l'ancien chemin sont (dé-)configurés en parallèle.
5. }

---

Cet algorithme de reconfiguration *MBB* ne produit pas d'interruption si et seulement si les deux chemins ont des liens disjoints ou sont identiques. En effet, si les deux chemins ne sont pas identiques et s'ils partagent un lien, par définition une seule longueur d'onde peut être utilisée pour établir un canal optique. Alors les deux chemins doivent être à liens disjoints sinon la reconfiguration avec *MBB* produira une interruption.

L'approche *MBB* est associée à l'approche *BBM* décrite à la section 4.3.2.1 pour la reconfiguration de problèmes qui imposent l'interruption de flux. Cela permet de réduire le nombre d'interruption produits durant le processus.

#### 4.3.2.4 Exemple de résolution de problèmes

##### - Résolution du problème 1

La résolution de ce problème utilise la notion de graphe de dépendances entre connexions. Un graphe de dépendances entre un ensemble de connexions est un graphe dans lequel chaque nœud représente une connexion à reconfigurer. Dans ce graphe, il existe un arc d'une connexion  $i$  à une connexion  $j$  si le chemin initial de la connexion  $j$  utilise des ressources nécessaires pour la configuration du nouveau chemin de la connexion  $i$ .

On montre qu'il existe une reconfiguration sans interruption pour une instance de problème de reconfiguration d'un ensemble de connexions unicast si et seulement si le graphe de dépendances associé est sans circuit. En effet, lorsque le graphe de dépendances est sans circuit, on peut faire la configuration des nouvelles connexions en commençant par les nœuds feuilles du graphe (représentant une nouvelle connexion dont les ressources nécessaires à

l'établissement sont disponibles) et poursuivre de manière récursive la configuration des autres connexions. En d'autres termes, lorsqu'une connexion est établie, elle peut disparaître du graphe de dépendances, permettant l'apparition de nouveaux nœuds feuilles à l'étape suivante.

##### - *Résolution du problème 2*

La reconfiguration du routage initial de ce problème avec les techniques de reconfiguration présentées dans les chapitres 4.3.2.2 et 4.3.2.3 (*MTV* ou *MBB*) va consister à établir le nouveau chemin de la connexion entre la source et la destination, et ensuite à supprimer l'ancien chemin. Considérons l'exemple de problème de reconfiguration de la figure 4.2. Le chemin initial de la connexion de source B et de destination E est (B – D – C – E) et le chemin final est (B – D – A – C – E). En supposant que chaque lien du réseau dispose d'une seule longueur d'onde, le nouveau chemin ne peut être entièrement établi sans interruption. En effet, la longueur d'onde sur les liens (B – D), et (C – E) n'est plus disponible car utilisée par l'ancien chemin. Donc les ports de sortie sur ces nœuds ne peuvent être configurés pour établir le nouveau chemin car utilisés par l'arbre initial. Dans cet exemple, on montre que ce problème ne peut pas être résolu par *MBB* sans interruption.

[CM07] propose une optimisation de la technique *MBB* appliquée à la reconfiguration du routage en se limitant aux segments strictement différents des chemins initial et final. Ce sera aussi l'idée de notre approche de reconfiguration par segment, mais notre proposition est plus générale. Elle s'applique sur n'importe quel graphe et n'importe quel routage, alors que les auteurs de [CM07] imposent la contrainte que les segments soient disjoints pour procéder à la configuration.

##### **Définition** : Segment

Un segment dans cette approche est une suite de liens appartenant à un chemin (initial ou final), ayant une extrémité initiale et une extrémité finale.

Lorsque deux segments des chemins initial et final sont identiques, cela signifie que les deux segments ont la même extrémité initiale, la même extrémité finale et sont constitués des mêmes liens. Cela signifie que le nouveau chemin requiert les mêmes ressources que le chemin initial sur le segment identique. Un tel segment ne nécessite donc pas d'être reconfiguré.

#### 4. Reconfiguration du routage dans les réseaux optiques WDM

La technique *MBB* peut s'appliquer pour la reconfiguration de chemin optique en se limitant aux segments différents des deux chemins de la connexion (l'ancien chemin et le nouveau chemin).

Reprenons l'exemple de la figure 4.2 dans lequel on veut reconfigurer le chemin de la connexion établie entre les nœuds B et E en se limitant aux segments différents. Les segments identiques sur ces deux chemins optiques sont (B – D) et (C – E). Le nœud B ne nécessite donc pas d'être configuré. Le nœud E, qui est la destination de la connexion ne nécessite pas d'être configuré. Les nœuds devant être reconfigurés sont {D, A}. Une séquence de reconfiguration, ici, va consister à configurer le nœud A pour ajouter une commutation de longueur d'onde sur le lien (A – C), à configurer le nœud D pour simultanément interrompre le flux sur l'ancien chemin et à alimenter le nouveau chemin. On constate alors qu'en limitant les segments à reconfigurer à ceux strictement différents sur les deux chemins, le problème de reconfiguration de la figure 4.2 a une solution sans interruption et avec une seule longueur d'onde par lien.

La reconfiguration d'un routage sans interruption de flux avec la technique *MBB* limitée aux segments strictement différents des chemins initial et final n'est pas toujours possible. [CM07] montre que la reconfiguration du routage sans interruption de flux n'est pas toujours possible quel que soit le chemin initial et final. C'est le cas lorsque les deux chemins contiennent une dépendance cyclique ; c'est-à-dire qu'il existe un nœud appartenant aux deux chemins dont le successeur sur le chemin initial est son prédécesseur sur le chemin final ou vice versa.

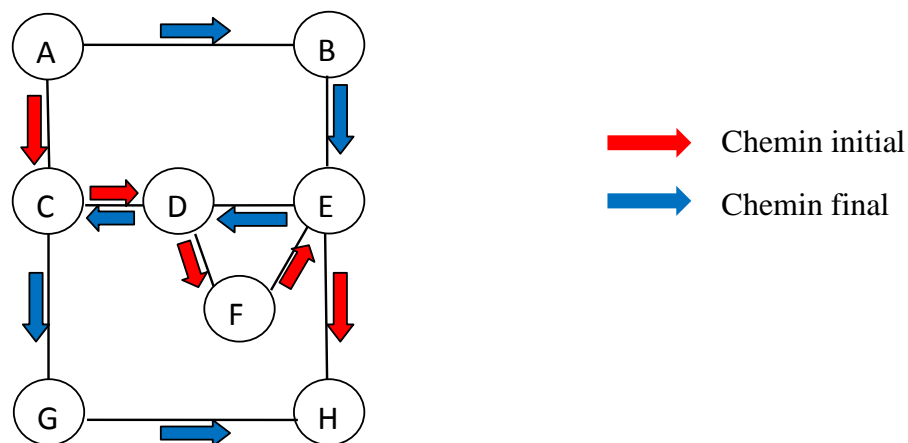


Fig. 4.3 : Exemple d'instance de reconfiguration de routage unicast.

Par exemple, dans l'instance de reconfiguration de la figure 4.3, le nœud A est la source de la connexion et le nœud H est la destination. Dans cet exemple, Le nœud D est le successeur du nœud C sur le chemin initial (en rouge) et est le prédécesseur de C sur l'arbre final (en bleu).

#### 4. Reconfiguration du routage dans les réseaux optiques WDM

Les auteurs de [CM07] proposent une méthode efficace de résolution de ce problème que nous décrivons ci-dessous.

[CM07] présente une méthode de reconfiguration rapide des connexions unicast et sans interruption dans un réseau orienté connexion. Cette méthode s'applique à toute sorte de réseau orienté connexion, notamment WDM (qui utilisent des chemins optiques – "Lightpath"), MPLS ou GMPLS utilisant les LSP (connexion MPLS – "Label Switching Path"). Un chemin dans le réseau est une succession de routeurs interconnectés par des liens entre une source et une destination. Les objectifs de cette méthode de reconfiguration sont donc doubles à savoir ne pas interrompre la connexion lors du processus de reconfiguration et réduire le temps d'exécution du processus.

On considère deux chemins : un chemin initial utilisé pour la transmission qui est l'ancien chemin et un autre final sur lequel on veut transférer le trafic qui est le nouveau chemin.

Un même nœud peut appartenir à la fois au nouveau et à l'ancien chemin mais ayant un routeur suivant différent en aval sur les chemins. Ce nœud doit être configuré avant certains nœuds et après d'autres. Cela nécessite donc un ordonnancement pour la reconfiguration des nœuds.

L'approche proposée par ces auteurs se base sur une méthode de reconfiguration qui consiste à reconfigurer les nœuds du chemin final en le remontant de la destination vers la source. Le principe de cette technique est de configurer les nœuds du nouveau chemin depuis la destination H jusqu'à la source A. Chaque étape consiste à configurer un seul nœud et à la dernière étape configurer simultanément les autres nœuds de l'ancien chemin n'appartenant pas au nouveau chemin.

Le temps de reconfiguration peut être réduit à travers le nombre de nœuds à reconfigurer, le délai de reconfiguration de chaque nœud ou le nombre d'étapes dans le processus de reconfiguration. Le nombre minimal de nœuds ne pouvant pas être changé, et le délai de configuration de chaque routeur dépendant du réseau, le seul point que l'on peut améliorer est le nombre d'étapes dans le processus de reconfiguration. Ainsi une méthode optimale de reconfiguration doit produire un nombre minimal d'étapes. Chaque étape est constituée de nœuds à reconfigurer simultanément et la liste de ces étapes définit l'ordre d'exécution dans le processus de reconfiguration.

#### 4. Reconfiguration du routage dans les réseaux optiques WDM

Une autre méthode est la reconfiguration remontante optimisée qui est une optimisation du nombre d'étapes de la première méthode proposée. Ici, on introduit la notion de *latch* (verrou).

Un verrou ici est un nœud qui appartient à la fois à l'ancien et au nouveau chemin dont le nœud suivant diffère entre le chemin initial et le chemin final.

La première étape consiste à configurer en une seule étape simultanément tous les nœuds appartenant exclusivement au nouveau chemin. Ensuite configurer tous les verrous sur le nouveau chemin à partir de la destination D. A chaque étape, on configure au plus un verrou. La dernière étape va consister à configurer simultanément tous les autres nœuds restants sauf ceux appartenant aux deux chemins et ayant le même prochain saut quel que soit le chemin (ancien ou nouveau). La destination ne nécessite pas de reconfiguration.

Un autre algorithme optimisant les deux précédents est proposé [CM07]. Il utilise les nœuds isolés. Un nœud est isolé pendant une étape du processus de reconfiguration s'il n'est pas sur le chemin de la source à la destination créée par les nœuds de cette étape.

Tous les nœuds isolés d'une étape n'étant pas utilisés pour la transmission peuvent être reconfigurés dans cette étape. La notion de nœud isolé est fonction de l'étape de configuration; Un nœud peut être isolé pour une étape du processus et non isolé pour d'autres étapes. Un nœud isolé peut être aussi un verrou. La difficulté est qu'il y a une interdépendance des verrous qui rend difficile l'ordonnancement des étapes de reconfiguration et le choix des verrous à configurer simultanément en une étape. L'algorithme consiste à une itération de deux étapes, où à chaque itération, au moins un nœud est configuré. Il se déroule jusqu'à ce que tous les nœuds soient configurés. Chaque itération comporte deux étapes : la première étape pour la recherche de nœuds isolés et la seconde pour la recherche du verrou approprié pour faire évoluer l'algorithme sinon il s'arrête. L'itération est décrite par l'algorithme 4.2.

#### **Algorithme 4.2 : Algorithme de reconfiguration de connexion unicast**

---

**Tant qu'il** existe des nœuds à configurer

1. Configurer simultanément tous les nœuds isolés en une étape
2. Configurer tout verrou approprié dans une autre étape

**Fin**

---

Un verrou  $j$  est approprié lorsque  $Old(j) < Old(Suc(j))$ .

$Old$  est le chemin formé uniquement des nœuds de l'ancienne route à configurer.  $New$  est la route formée uniquement des nœuds de la nouvelle route à configurer.  $Old(j)$  est la position du nœud  $j$  dans la liste des nœuds pas encore configurés de l'ancien chemin à partir de la source.

Par définition,  $Old(S) = 0$ .  $Suc(j)$  est le premier verrou successeur du verrou  $j$  non configuré sur  $New$ . Par hypothèse,  $Suc(j) = D$  si  $j$  n'a pas de verrou non configuré dans ses successeurs sur  $New$ .

Après la deuxième étape de chaque itération, un nœud situé sur l'ancien chemin strictement entre le verrou  $j$  et son successeur  $suc(j)$  devient un nœud isolé.

Cet algorithme utilise les ensembles et variables suivants :

$A$  est la liste de tous les nœuds non isolés à configurer. On initialise  $A$  à  $Old$ .

$B$  est l'ensemble de tous les nœuds isolés à configurer. On initialise  $B$  à  $New - Old$ .

Après chaque itération,  $B$  est constitué de tous les nœuds de l'ancien chemin situés strictement entre le verrou déterminé et  $suc(j)$ . C'est la liste des nœuds configurés. On l'initialise à  $\{ \}$ . A la fin du processus,  $C$  contient toutes les étapes de reconfiguration.  $n$  est le dernier verrou configuré. On initialise  $n$  avec la destination de la connexion.  $n$  est utilisé comme indicateur d'emplacement sur le nouveau chemin pour remonter jusqu'à la source tant que la liste de tous les nœuds non isolés à configurer,  $A$ , n'est pas vide.

Nous avons décrit au chapitre 4.3 des techniques de reconfiguration de routage existantes dans la littérature. Ces techniques ne s'appliquent qu'aux connexions unicast. Nous avons présenté des algorithmes de configuration entre deux chemins d'une connexion, et aucun ne garantit qu'il n'y aura pas d'interruption dans le cas général. Le dernier algorithme présenté dans [CM07] est le plus performant en terme de nombre d'étapes de reconfiguration.

Nous allons présenter dans la section suivante, le problème de reconfiguration du routage multicast que nous avons étudié durant cette thèse.

#### 4.4 Reconfiguration du routage multicast

Aujourd'hui avec l'extension du multicast aux réseaux optiques, la reconfiguration du routage qui se posait pour les connexions unicast se pose également pour les connexions multicast. Le

chemin suivi par le flux optique d'une connexion multicast étant un arbre optique, le problème va consister à reconfigurer un arbre optique.

De nombreuses applications utilisant les connexions multicast sont sensibles aux interruptions. De plus, les opérateurs doivent respecter les contrats signés avec les clients, contrats qui tiennent compte de la fréquence et la durée de ces interruptions à travers la fiabilité (MTTF : "*Mean Time To Failure*"), le délai ou le débit de la connexion. En effet, une interruption de flux intervient directement ou indirectement sur au moins ces trois facteurs. La reconfiguration de l'arbre optique doit donc se faire si possible sans interruption de flux.

La reconfiguration d'un nœud va consister à ajouter ou supprimer une commutation de longueur d'onde entre des liens du réseau. Dans le cas unicast, la commutation a toujours lieu entre deux liens. Dans le cas multicast, la commutation peut mettre en jeu plus que deux liens. Grâce au splitter, un même flux optique issu d'un lien optique arrivant sur un port d'entrée du nœud peut être transmis vers plusieurs ports de sortie du nœud, chacun étant associé à un lien optique. Toutes ces configurations doivent garantir, lors de toute étape de reconfiguration, l'existence d'un arbre qui couvre la source et les destinations du groupe multicast dans le réseau, d'une part. D'autre part, aucune boucle ne doit exister dans le réseau.

##### 4.4.1 Problème de reconfiguration du routage multicast

**Problème (reconfiguration du routage multicast) :**

Soit une topologie de réseau modélisé par un graphe  $G = (V, E)$ , où  $V$  représente l'ensemble des nœuds et  $E$  l'ensemble des liens. Un nœud du graphe représente un commutateur optique dans le réseau physique et un lien du graphe représente une fibre dans le réseau physique.  $L = \{\lambda_1, \lambda_2, \dots, \lambda_w\}$  représente l'ensemble de  $w$  longueurs d'onde disponibles sur les liens. Supposons que  $T_0 = (V_0, E_0)$  et  $T_z = (V_z, E_z)$  représentent deux arbres de même racine et de même ensemble de destinations inclus dans le graphe  $G$ . Un nœud du réseau pouvant avoir plusieurs ports, le problème de reconfiguration du routage multicast consiste à déterminer un ordre  $Seq = \{o(n_1), o(n_2), \dots, o(n_{|Seq|})\}$  sur les ports des nœuds constituant les arbres initial et final.  $n_i$  représente le port configuré à la  $i^{ème}$  étape avec l'opération de configuration  $o$ .

La reconfiguration du routage devant se faire sans interruption de flux, l'ordre  $Seq = \{o(n_1), o(n_2), \dots, o(n_{|Seq|})\}$  doit être déterminé de telle sorte qu'on produise une série



d'arbres couvrant la source et les destinations du groupe multicast à chaque étape de configuration.

**Définition :** Séquence de reconfiguration

Une séquence de reconfiguration est la série constituée des chemins suivis par le flux optique à chaque étape de reconfiguration.

Le problème que nous étudions consiste alors à déterminer un ordre  $Seq = \{o(n_1), o(n_2), \dots, o(n_{|Seq|})\}$  sur les ports des nœuds de telle sorte qu'on produise une séquence de reconfiguration constituée d'une série d'arbres  $T_0; T_1; T_2; \dots; T_{z-1}; T_z$ , avec  $T_0$  l'arbre initial de la connexion et  $T_z$  l'arbre final.

Nous supposons que les nœuds sont équipés de convertisseurs. Toutefois, les convertisseurs optiques sont rares, chers et induisent un affaiblissement du signal. Donc la connexion multicast doit utiliser aussi peu de longueur d'onde que possible (éventuellement une longueur d'onde) durant les opérations normales (routage et affectation de longueurs d'onde) et les opérations de reconfiguration. Cela nous permet d'étudier le problème de reconfiguration dans deux contextes différents selon que les nœuds du réseau sont équipés de convertisseurs de longueurs d'onde ou non.

Lorsque les nœuds du réseau ne sont pas équipés de convertisseurs de longueurs d'onde, la reconfiguration de l'arbre optique doit se faire en utilisant la même longueur d'onde sur tous les arbres produits.

Lorsque les nœuds sont équipés de convertisseurs de longueur d'onde, différentes longueurs d'onde peuvent être utilisées, mais de façon économe durant le processus de reconfiguration.

Un autre paramètre important à considérer est la propriété de duplication du signal (*splitting*) (présentée dans la section 3.4.2) qu'ont les nœuds du réseau ou pas. Nous supposons que les nœuds sont équipés de splitters et ont une capacité de duplication appropriée. Cela signifie que la capacité de duplication d'un nœud est égale (ou supérieure) au nombre maximum de nœuds voisins de ce nœud moins un, et que le nombre de splitters dans un nœud est supérieur au nombre de sessions multicast passant par ce nœud. Cela nous permet de construire l'arbre optique initial et l'arbre optique final quel que soit le nombre de nœuds voisins d'un nœud donné et le nombre de sessions multicast à établir dans le réseau.

#### 4.4.2 Revue de la littérature

Peu de travaux se sont intéressés à la reconfiguration d'un arbre optique dans la littérature. [HFZJ04, PCL08] ont traité la reconfiguration du routage multicast. [PCL08] étudie la dégradation de l'arbre optique causée par le comportement dynamique de ses nœuds feuilles. En effet, de nouveaux nœuds peuvent rejoindre le groupe multicast, ou des nœuds de ce groupe peuvent le quitter. Les auteurs présentent le bénéfice de la reconfiguration régulière de l'arbre optique pour faire face à sa dégradation. Le principal bénéfice présenté est le gain en ressources utilisées par l'arbre optique. [HFZJ04] présente un algorithme de groupage de trafic permettant la détermination du routage de nouvelles connexions minimisant le taux de blocage de celles-ci. Dans ces deux articles les auteurs ne traitent pas de la meilleure méthode pour passer sans interruption de flux de l'arbre optique initial à l'arbre optique final. C'est ce à quoi nous nous attachons dans notre travail.

Une approche de solution consiste à appliquer la technique *Move-To-Vacant* au problème de reconfiguration du routage. Cette méthode ne permet pas de résoudre toutes les instances de ce problème. En effet, pour certaines instances, aucune séquence d'opérations *Move-To-Vacant* ne permet de trouver un ordre de reconfiguration des nœuds sans interruption du flux optique.

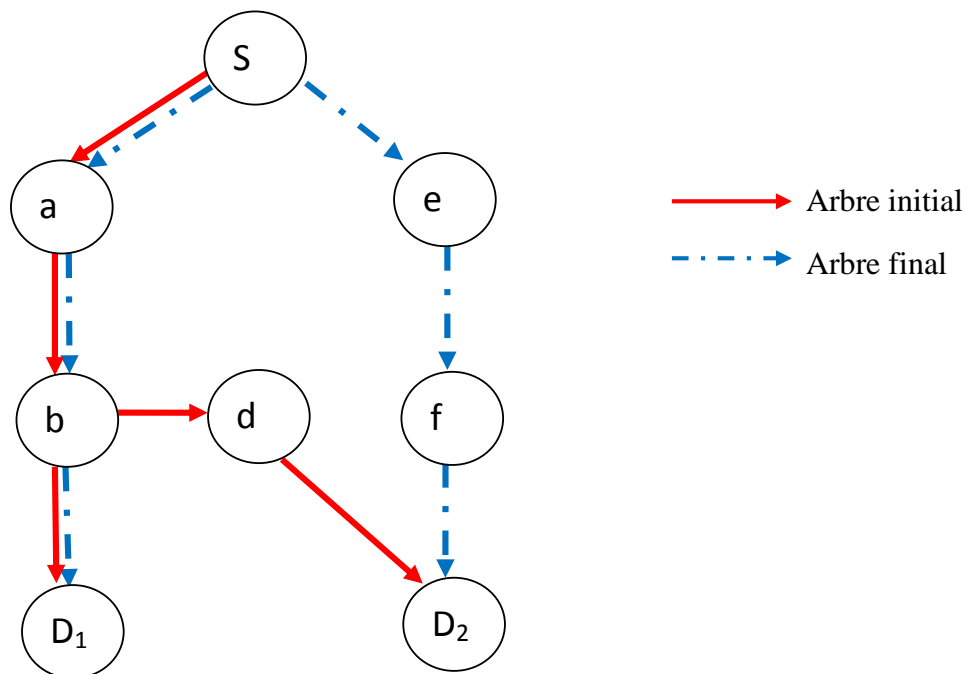


Fig. 4.4 : Instance de reconfiguration multicast sans solution pour MTV.

#### 4. Reconfiguration du routage dans les réseaux optiques WDM

La figure 4.4 décrit un exemple d'instance de reconfiguration multicast pour laquelle aucune séquence d'opérations *Move-to-Vacant* n'existe. Dans cette figure, l'arbre initial  $T_0$  est en rouge et l'arbre final  $T_z$  est en bleu. L'arbre initial et l'arbre final ont la même racine  $S$  qui est la source de la connexion multicast et le même ensemble de destinations  $\{D_1 ; D_2\}$ . Dans le graphe représenté par cette figure, aucune séquence *Move-To-Vacant* ne permet d'établir l'arbre final car il n'y a pas de ressources disponibles sur certains liens de la branche entre les nœuds  $S$  et  $D_2$ , notamment entre les nœuds  $S$  et  $a$ .

Une autre approche de solution va consister à étendre la technique de reconfiguration *Make-Before-Break* utilisée pour les connexions unicast aux connexions multicast. L'algorithme *MBB* peut être étendu à la reconfiguration d'un arbre optique. L'extension de *MBB* pour la reconfiguration d'un arbre est appelée *MBBI*. Un arbre étant constitué de plusieurs branches, chaque branche de l'arbre initial (respectivement de l'arbre final) peut être considérée comme un chemin initial (respectivement un chemin final).

Alors l'extension de l'algorithme *MBB* pour la reconfiguration d'un arbre va consister à configurer branche par branche tout l'arbre, en utilisant *MBB* appliqué à une branche de l'arbre initial et la branche correspondante sur l'arbre final. L'algorithme *MBB* est présenté dans Algorithme 4.3.

#### Algorithme 4.3: Algorithme *MBBI*

---

```
MBBI ( $T_0, T_z$ )          /* Arbre initial  $T_0$  et arbre final  $T_z$ */
1. {
2. Pour chaque destination  $d$  de la paire d'arbres /*Ancienne branche  $branch(T_0, S, d)$ 
                                                et nouvelle branche  $branch(T_z, S, d)$ */
3. {
4.     MBB(( $branch(T_0, S, d), branch(T_z, S, d)$ ))
5. }
6. }
```

---

Pour chaque destination  $d$ , la nouvelle branche entre la source de la connexion et la destination  $d$  est préétablie. Ensuite on configure la source pour interrompre le flux sur l'ancienne branche et alimenter la nouvelle branche. Enfin on supprime l'ancienne branche en configurant en parallèle tous les nœuds de cette ancienne branche. La reconfiguration s'achève lorsque toutes les branches sont reconfigurées.

*MBBI* a pour avantage d'être rapide mais provoque souvent des interruptions. Par exemple, lorsque certaines branches de l'arbre partagent des ressources, la reconfiguration d'une branche

#### 4. Reconfiguration du routage dans les réseaux optiques WDM

avec *MBBI* avant les autres branches va provoquer une interruption du flux optique sur les autres branches. Un autre (contre-)exemple apparait lorsque une destination (par exemple  $d$ ) a des nœuds aval (par exemple  $x$ ), la reconfiguration de la branche de destination  $d$  va provoquer l'interruption du flux optique vers les destinations alimentées par le canal ( $d - x$ )

L'algorithme *MBBI* ne permet pas d'obtenir de séquence d'opérations de reconfiguration sans interruption de flux de l'instance de la figure 4.4. La branche de l'arbre final entre la source  $S$  et la destination  $D_1$  ne peut pas être établie car les ressources dont elle a besoin ne sont pas disponibles.

Soit l'instance de reconfiguration de routage multicast présentée dans la figure 4.5. Cette instance ne peut pas être résolue avec la méthode *Make-Before-break* car la longueur d'onde sur le lien ( $b-D_1$ ) n'est pas disponible.

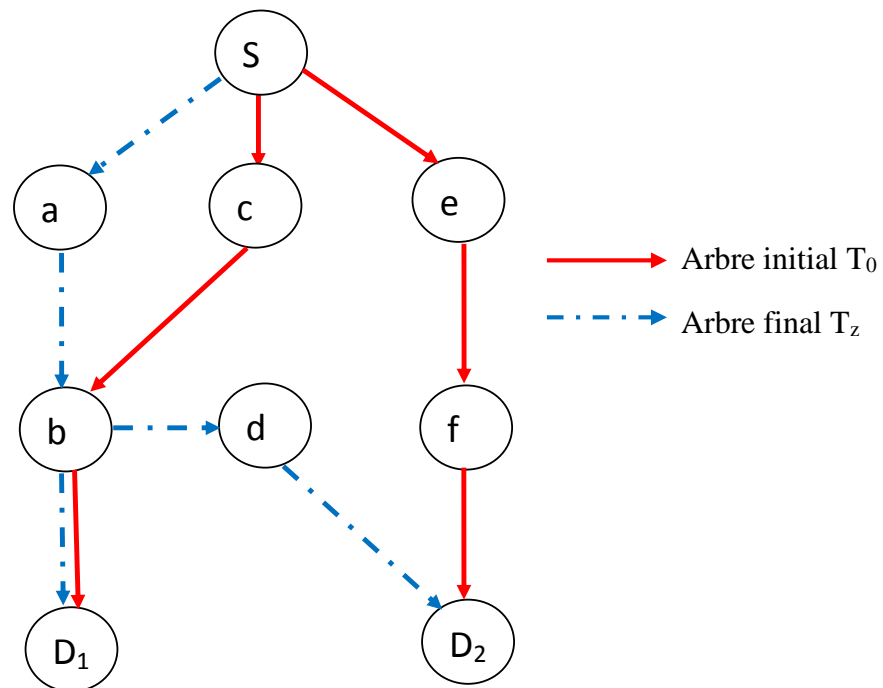


Fig. 4.5 : Instance de reconfiguration multicast non résolue par MBB.

Les techniques *Move-To-Vacant* et *Make-Before-Break* ne permettent pas, comme le montrent les exemples des figures 4.4 et 4.5, de résoudre certaines instances de reconfiguration du routage multicast.

L'utilisation de la technique *BBM* oblige à interrompre le flux vers les destinations de la connexion. Cela interrompt de façon naturelle la connexion donc ne garantit pas la continuité du flux, quel que soit l'instance de reconfiguration.

A l'issue des exemples présentés ci-dessus, il se pose alors les questions suivantes :

1. Est-il possible de reconfigurer un routage multicast sans interruption de flux sous la contrainte de continuité de longueur d'onde? En d'autres termes, existe-t-il une séquence de reconfiguration constituée d'arbres optiques couvrant la source et les destinations du groupe permettant de passer de l'arbre initial à l'arbre final, les arbres étant constitués de la même longueur d'onde ? A un instant donné du processus de reconfiguration, il doit exister un seul arbre couvrant la source et les destinations de la connexion multicast. Une destination du groupe multicast ne doit pas recevoir le flux optique en double. En d'autres termes, un nœud du réseau ne doit pas être alimenté par plus d'un nœud.
2. Que faire si la reconfiguration du routage multicast sans interruption de flux sous la contrainte de continuité de longueur d'onde n'est pas possible?
3. Quelles sont les opérations de configuration à exécuter sur les différents nœuds à configurer durant le processus de reconfiguration ?

La configuration d'un arbre optique consiste concrètement à exécuter des opérations sur les nœuds appartenant aux arbres initial et final afin de configurer leurs différents ports. Cette question permet de définir les fonctions utiles lors d'une reconfiguration de routage multicast et les opérations de configuration associées à chaque fonction.

## 4.5 Conclusion

Nous avons présenté, dans ce chapitre le problème de reconfiguration du routage dans un réseau optique WDM. Nous avons présenté dans une première partie, les problèmes de reconfiguration du routage unicast et les approches de solutions existantes. Dans la seconde partie, nous avons présenté le problème de reconfiguration du routage multicast qui est l'objet de cette thèse. La revue de la littérature a montré que les algorithmes de reconfiguration existants, la plupart pour le routage unicast, ne permettent pas de résoudre toutes les instances de reconfiguration de routage multicast sans interruption de flux.

Les chapitres suivants s'intéressent aux différentes propositions que nous avons faites. Nous définirons dans le chapitre suivant des opérations de configuration utiles à la reconfiguration d'un routage et proposer des algorithmes de reconfiguration.

# 5. Reconfiguration par branche du routage multicast

---

## Sommaire

---

|  |     |
|--|-----|
| 5.1 Introduction.....  | 55  |
| 5.2 Définitions des principales notions.....   | 55  |
| 5.3 Classification des nœuds du réseau pour une instance de reconfiguration.....       | 57  |
| 5.3.1 Nœud de branchement.....   | 57  |
| 5.3.2 Nœud isolé.....  | 58  |
| 5.3.3 Port isolé.....  | 58  |
| 5.3.4 Nœud divergent.....  | 58  |
| 5.3.5 Nœud convergent.....   | 58  |
| 5.4. Opérations de reconfiguration d'un nœud optique.....                              | 59  |
| 5.4.1 Opération d'ajout de commutation de longueur d'onde.....                         | 59  |
| 5.4.2 Opération de suppression de commutation de longueur d'onde.....                  | 61  |
| 5.4.3 Opération de conversion de longueur d'onde.....                                  | 62  |
| 5.4.4 Opération de multi-basculement.....  | 64  |
| 5.4.5 Opération de convergence de longueur d'onde.....                                 | 65  |
| 5.4.6 Opération de non-convergence de longueur d'onde.....                             | 67  |
| 5.5. Problème de reconfiguration multicast utilisant une seule longueur d'onde.....    | 67  |
| 5.5.1. Etude des séquences de reconfiguration d'un exemple de problème.....            | 69  |
| 5.5.2 Evaluations.....   | 70  |
| 5.6. Problème de reconfiguration multicast utilisant une longueur d'onde additionnelle |     |
| 5.6.1 Introduction.....  | 75  |
| 5.6.2 Définitions.....   | 76  |
| 5.6.3 Procédé de reconfiguration par branche <i>BpBAR</i> .....                        | 76  |
| 5.6.4 Algorithme de reconfiguration <i>BpBAR_1</i> .....                               | 85  |
| 5.6.5 Réduction de la durée de reconfiguration du processus <i>BpBAR</i> .....         | 88  |
| 5.6.6 Evaluations.....   | 95  |
| 5.6.7 Conclusion.....  | 104 |
| 5.7 Reconfiguration dans un réseau avec des ressources limitées.....                   | 105 |
| 5.7.1 Introduction.....  | 105 |
| 5.7.2 Algorithme de reconfiguration <i>TRwRC</i> .....                                 | 105 |
| 5.7.3 Evaluations.....   | 107 |
| 5.7.4 Conclusion.....  | 113 |
| 5.8 Conclusion.....  | 113 |

---

### 5.1 Introduction

Dans ce chapitre, nous étudions la reconfiguration d'un arbre optique sans interruption de flux dans un réseau, selon deux contextes. Dans un premier contexte, nous supposons que les liens du réseau ne disposent que d'une seule longueur d'onde. Dans un second contexte, nous supposons que les liens disposent de plusieurs longueurs d'onde et que les nœuds sont équipés de convertisseurs de longueur d'onde. Nous donnons dans la section 5.2 les définitions des principales notions utilisées dans la suite du travail. Ensuite, nous classifions dans la section 5.3 les nœuds appartenant aux différents arbres (initial et final) d'une instance de reconfiguration. La section 5.4 présente les opérations de configuration utiles à la reconfiguration du routage multicast. Les sections 5.5 à 5.7 s'intéressent à la proposition d'algorithmes de reconfiguration sans interruption de flux dans différents contextes. La section 5.5 s'intéresse à la résolution du problème de reconfiguration multicast en utilisant une seule longueur d'onde. La section 5.6 s'intéresse à la résolution du problème de reconfiguration multicast en utilisant localement et temporairement une longueur d'onde supplémentaire. La section 5.7 s'intéresse à la résolution du problème de reconfiguration multicast dans un réseau ayant des ressources limitées.

### 5.2 Définitions des principales notions

Dans cette section, nous donnons les définitions de notions utilisées dans la suite du mémoire.

#### 5.2.1 Arbre courant

Un arbre courant à une étape de reconfiguration est l'arbre utilisé pour la transmission du flux à cette étape.

#### 5.2.2 Paire d'arbres associée à une instance de reconfiguration

La paire d'arbres associée à une instance de reconfiguration est constituée des nœuds de l'arbre initial et l'arbre final et des liens de ces deux arbres dans le graphe du réseau étudié.

#### 5.2.3 Hauteur d'un nœud sur un arbre

La hauteur d'un nœud sur un arbre est le nombre de liens de cet arbre entre la racine de l'arbre (représentant la source de la connexion) et ce nœud. La hauteur de la source est 0. La fonction

$hauteur(T, i)$  donne la hauteur du nœud  $i$  sur l'arbre  $T$ . La hauteur du nœud  $i$  est égale à «  $h+1$  » si le nœud  $i$  est un nœud aval d'un nœud de hauteur  $h$ .

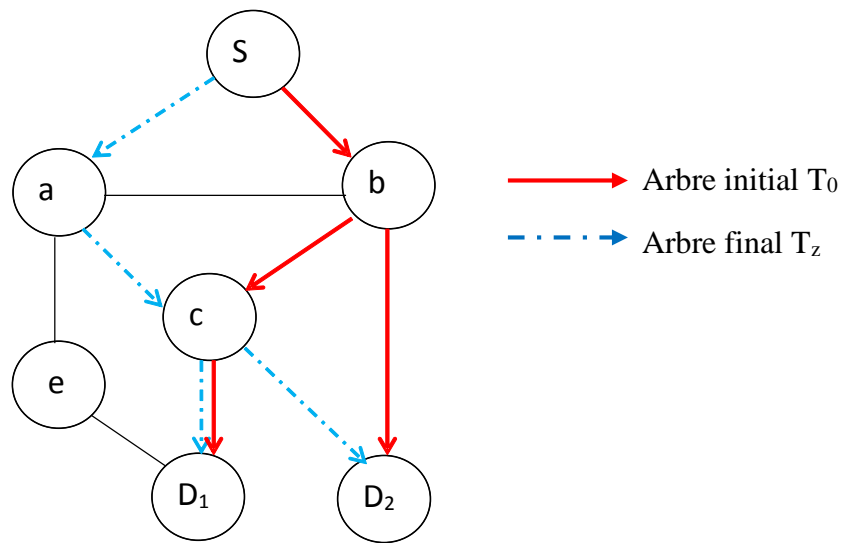


Fig. 5.1: Exemple d'instance de reconfiguration.

Dans le graphe du réseau de l'exemple de la figure 5.1, l'arbre initial est en rouge et l'arbre final est en bleu. Nous obtenons pour cet arbre  $hauteur(T_0, S) = 0$ ,  $hauteur(T_z, S) = 0$ ,  $hauteur(T_0, D_2) = 2$  et  $hauteur(T_z, D_2) = 3$ .

### 5.2.4 Nœud amont

Le nœud amont d'un nœud sur un arbre donné est le nœud père de ce nœud sur cet arbre. Si le nœud  $X$  a pour hauteur  $h$ , le nœud amont de  $X$  sur l'arbre a pour hauteur  $h-1$ . Un nœud (différent du nœud racine) a un unique nœud amont sur un arbre.

Sur la figure 5.1, le nœud amont du nœud  $c$  sur l'arbre initial est le nœud  $b$ . Le nœud amont du nœud  $c$  sur l'arbre final est le nœud  $a$ .

### 5.2.5 Nœud aval

Un nœud aval d'un nœud sur un arbre donné est un nœud fils de ce nœud sur cet arbre. Si le nœud  $X$  a pour hauteur  $h$ , les nœuds avals du nœud  $X$  ont pour hauteur  $h+1$ . Un nœud peut avoir plusieurs nœuds avals sur un arbre.

Sur la figure 5.1, le nœud aval du nœud  $c$  sur l'arbre initial est  $D_1$ . Les nœuds aval du nœud  $c$  sur l'arbre final sont  $D_1$  et  $D_2$ .



### 5.2.6 Canal optique

Un canal optique est un lien du réseau muni d'une longueur d'onde utilisée pour la transmission du flux optique entre un nœud et un de ses nœuds avals.

### 5.2.7 Canal optique amont

Soit  $X$  un nœud appartenant à un arbre et  $Y$  son nœud amont sur l'arbre. Le canal optique entre les nœuds  $Y$  et  $X$  est un canal optique amont du nœud  $X$  sur cet arbre. Un nœud (hormis le nœud racine) a un unique canal optique amont sur un arbre.

Sur la figure 5.1, la longueur d'onde utilisée entre les nœuds  $S$  et  $b$  est le canal optique amont du nœud  $b$  sur l'arbre initial.

### 5.2.8 Canal optique aval

Soit  $X$  un nœud appartenant à un arbre et  $Z$  un nœud aval de  $X$  sur l'arbre. Le canal optique entre  $X$  et  $Z$  est un canal optique aval de  $X$  sur cet arbre. Un nœud peut avoir plusieurs canaux optiques aval sur un arbre.

Sur la figure 5.1, la longueur d'onde utilisée entre les nœuds  $b$  et  $c$  est un canal optique aval du nœud  $b$  sur l'arbre initial.

## 5.3 Classification des nœuds du réseau pour une instance de reconfiguration

Nous distinguons différents types de nœuds dans une instance de reconfiguration du routage multicast. Cette distinction entre les types de nœuds est faite en fonction de leur appartenance à l'arbre initial ou final et de leur position sur ces deux arbres.

### 5.3.1 Nœud de branchement

Un nœud de branchement est un nœud appartenant à un arbre et qui a au moins deux nœuds aval sur cet arbre.

Dans l'exemple de la figure 5.1, le nœud  $b$  est un nœud de branchement sur l'arbre initial.

### 5.3.2 Nœud isolé

Un nœud isolé est un nœud appartenant à l'arbre initial ou l'arbre final d'une instance de reconfiguration de routage multicast qui n'est pas utilisé par le flux optique à une étape donnée.

Dans l'exemple de la figure 5.1, le nœud  $a$  est le seul nœud isolé du graphe de l'instance de reconfiguration.

Un nœud peut être isolé à une étape  $i$ , et ne plus l'être à l'étape  $i+1$ .

### 5.3.3 Port isolé

Un port isolé est un port d'un nœud appartenant à l'arbre initial ou l'arbre final d'une instance de reconfiguration de routage multicast qui n'est pas utilisé pour la transmission du signal optique.

Dans l'exemple de la figure 5.1, le port du nœud  $c$  vers le nœud  $D_2$  sur l'arbre initial que nous notons ( $c \rightarrow D_2$ ) est un port isolé.

On remarque qu'un port isolé peut appartenir à un nœud qui n'est pas isolé et que tous les ports d'un nœud isolé sont des ports isolés.

### 5.3.4 Nœud divergent

Un nœud divergent est un nœud appartenant à l'arbre initial et l'arbre final d'une instance de reconfiguration, ayant au moins un nœud aval sur l'arbre final et un nœud aval sur l'arbre initial différent des nœuds aval sur l'arbre final.

Dans l'exemple de la figure 5.1, le nœud  $a$  est un nœud aval du nœud  $S$  sur l'arbre final et le nœud  $b$  est un nœud aval du nœud  $S$  sur l'arbre initial. Les deux nœuds aval sur l'arbre initial et l'arbre final étant différents, alors le nœud  $S$  est un nœud divergent. De même, le nœud  $c$  est un nœud divergent. En effet, le nœud  $D_2$  est un nœud aval du nœud  $c$  sur l'arbre final et le nœud  $D_1$  est un nœud aval du nœud  $c$  sur l'arbre initial.

### 5.3.5 Nœud convergent

Un nœud convergent est un nœud appartenant à l'arbre initial et l'arbre final et qui est tel que son nœud amont sur l'arbre final est différent du nœud amont sur l'arbre initial.

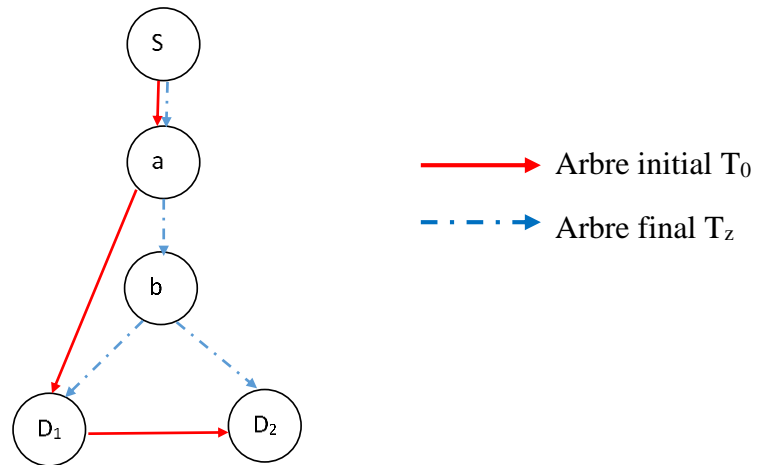


Fig. 5.2: Problème de reconfiguration.

Dans l'exemple de la figure 5.2 l'arbre initial est en rouge et l'arbre final en bleu. Le nœud  $D_1$  est un nœud convergent. En effet, le nœud  $a$  est le nœud amont du nœud  $D_1$  sur l'arbre initial et le nœud  $b$  est le nœud amont du nœud  $D_1$  sur l'arbre final. Les deux nœuds amonts du nœud  $c$  sur chaque arbre sont différents. Donc le nœud  $D_1$  est un nœud convergent pour ces deux arbres. De même, le nœud  $D_2$  est un nœud convergent.

## 5.4 Opérations de configuration d'un nœud optique

Les opérations de configuration d'un nœud du routage multicast sont définies en fonction des besoins de configuration. Nous définissons dans cette partie six types d'opérations que nous présentons ci-dessous. Les notations suivantes sont utilisées :

$X$  : Nœud représentant un commutateur

$X_i^O$  : Port de sortie  $i$  du nœud  $X$

$X_j^I$  : Port d'entrée  $j$  du nœud  $X$

$l$  : Longueur d'onde

### 5.4.1 Opération d'ajout de commutation de longueur d'onde

La première opération utile dans un processus de reconfiguration multicast est l'ajout d'une commutation de longueur d'onde d'un port d'entrée vers un port de sortie sur un commutateur. Cette opération est notée  $(+ X_i^I - X - l - X_j^O)$ . Le port d'entrée et le port de sortie peuvent

## 5. Reconfiguration par branche du routage multicast

être respectivement remplacés par le nom du commutateur amont et commutateur aval de  $X$  sur l'arbre, si cela n'introduit pas d'ambiguïté (c.-à-d. lorsque il n'existe au plus qu'une seule fibre optique entre tout couple de nœuds du réseau). Nous ferons cette hypothèse qui simplifie la notation, sans perte de généralité. Si  $T$  est un arbre, on notera  $Add(T)$  pour traduire l'exécution d'une opération d'ajout de commutation de longueur d'onde sur un nœud (ou ensemble de nœuds) de l'arbre  $T$ .

- Pré-condition : La condition d'exécution de cette opération est que la longueur d'onde  $l$  soit libre sur le port de sortie  $j$  (port  $X_j^O$ ) du commutateur  $X$ . Le port  $X_j^O$  est un port isolé.
- Post-condition : Après l'exécution de l'opération d'ajout de commutation sur le nœud  $X$ , la longueur d'onde  $l$  est occupée sur le port d'entrée  $i$  ( $X_i^I$ ) et le port de sortie  $X_j^O$  du commutateur  $X$ .
- Action : La modification apportée par cette opération à la table de commutation du nœud  $X$  est que tout signal reçu sur le port d'entrée  $X_i^I$  du commutateur est propagé au port de sortie  $X_j^O$  du même commutateur.

Si le commutateur  $X$  commutait déjà la longueur d'onde  $l$  du port d'entrée  $X_i^I$  vers un port de sortie  $X_k^O$ , le nœud  $X$  devient un nœud de branchement.

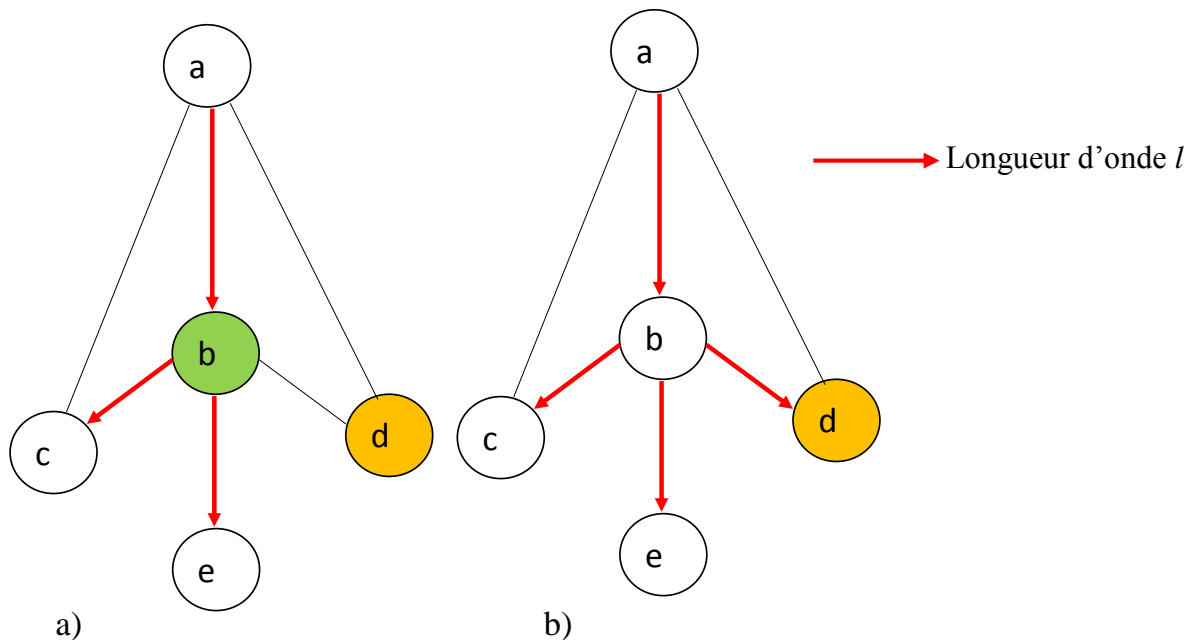


Fig. 5.3 : Exemple d'opération d'ajout de commutation d'une longueur d'onde.

La figure 5.3 décrit un exemple d'utilisation de l'opération d'ajout de commutation. Dans la figure 5.3.a, l'arbre de source  $a$  et de destinations les nœuds  $c$  et  $e$  représente l'arbre optique initial d'une connexion multicast de source  $a$  et de destinations  $\{c, e\}$ . On veut que le flux de cette connexion multicast atteigne le nœud  $d$ . Pour y arriver, on peut configurer le nœud  $b$ . Comme le montre la figure 5.3.a, la longueur d'onde sur le lien entre  $b$  et  $d$  est libre. Le port de  $b$  permettant d'atteindre  $d$  est donc isolé. On exécute alors l'opération d'ajout de commutation  $(+ a - b - l - d)$  sur le nœud  $b$ . Après l'exécution de l'opération, on obtient la figure 5.3.b. Le signal venant du nœud  $a$  atteint le nœud  $d$  sur la longueur d'onde  $l$  en passant par  $b$ . La longueur d'onde entre  $b$  et  $d$  est maintenant occupée. Le nœud  $b$  devient un nœud de branchement.

### 5.4.2 Opération de suppression de commutation de longueur d'onde

La deuxième opération utile dans un processus de reconfiguration multicast est la suppression d'une commutation de longueur d'onde d'un port d'entrée à un port de sortie sur un commutateur. Cette opération est notée  $(-X_i^l - X - l - X_j^o)$ . Le port d'entrée et le port de sortie peuvent être respectivement remplacés par le nœud amont et nœud aval de  $X$  sur l'arbre. Si  $T$  est un arbre, on notera  $Del(T)$  pour traduire l'exécution d'une opération de suppression de commutation de longueur d'onde sur un nœud (ou ensemble de nœuds) de l'arbre  $T$ .

- Pré-condition : La condition d'exécution de cette opération est que la longueur d'onde  $l$  soit occupée sur les ports d'entrée  $X_i^l$  et de sortie  $X_j^o$  du commutateur  $X$ .
- Post-condition : Si le commutateur  $X$  commutait la longueur d'onde  $l$  du port d'entrée  $X_i^l$  vers plusieurs ports de sortie (au moins  $X_j^o$  et par exemple  $X_k^o$ ), la longueur d'onde  $l$  est libérée seulement sur le port de sortie  $X_j^o$  du commutateur  $X$ . Si le commutateur  $X$  commutait la longueur d'onde  $l$  du port d'entrée  $X_i^l$  vers le seul port de sortie  $X_j^o$ , la longueur d'onde  $l$  est libérée sur le port d'entrée  $X_i^l$  et sur le port de sortie  $X_j^o$  du commutateur  $X$ .
- Action : La modification apportée par cette opération à la table de commutation du nœud  $X$  est qu'aucun signal de longueur d'onde  $l$  venant du port d'entrée  $X_i^l$  ne sort du port de sortie  $X_j^o$  du commutateur  $X$ .

Si le commutateur  $X$  ne commutait pas la longueur d'onde  $l$  du port d'entrée  $X_i^l$  vers le port de sortie  $X_j^o$ , l'action est vide. C'est-à-dire qu'on s'interdit la commutation optique de type incast.

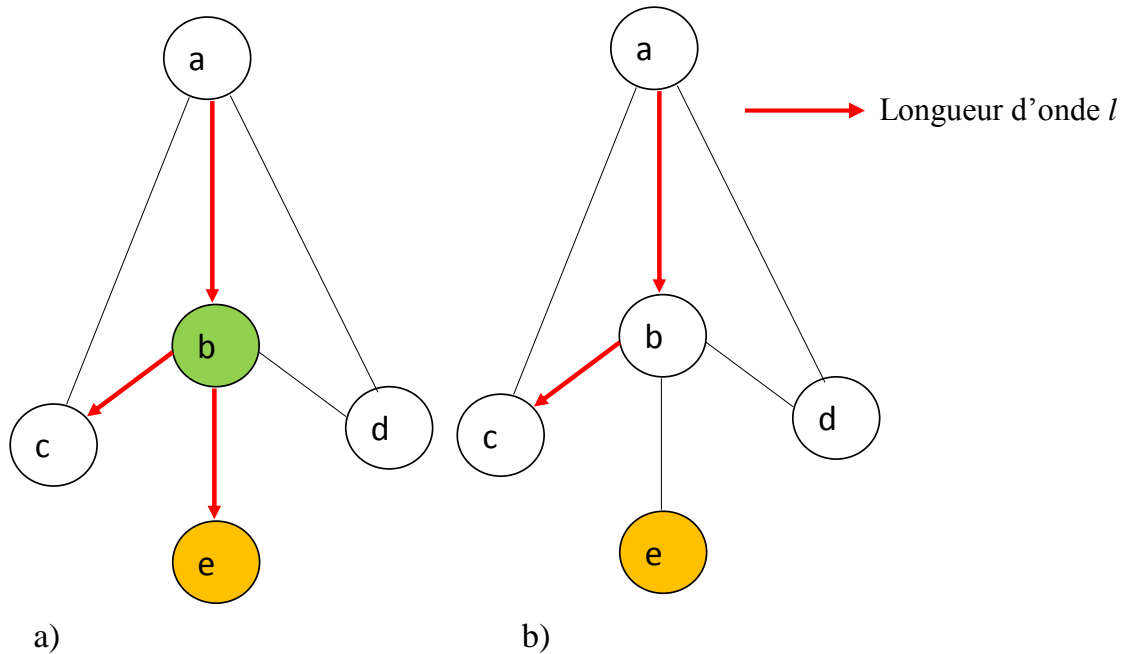


Fig. 5.4 : Exemple d'application de l'opération de suppression de commutation optique.

La figure 5.4 décrit un exemple d'utilisation de l'opération de suppression de commutation de longueur d'onde. Dans la figure 5.4.a, l'arbre de source  $a$  et de destinations les noeuds  $c$  et  $e$  représente l'arbre optique d'une connexion multicast de source  $a$  et destinations  $\{c ; e\}$ . On veut que le flux de la connexion dont l'arbre est représenté sur la figure 5.4.a n'atteigne plus le noeud  $e$ . Pour y arriver, on exécute alors l'opération de suppression de commutation ( $- a - b - l - e$ ) sur le noeud  $b$ . Après l'exécution de l'opération, on obtient la figure 5.4.b. Le signal venant du noeud  $a$  sur la longueur d'onde  $l$  n'est plus commuté vers le noeud  $e$ . La longueur d'onde  $l$  est libérée exclusivement sur le port de sortie de  $b$  vers le noeud  $e$ .

### 5.4.3 Opération de conversion de longueur d'onde

L'opération de conversion de longueur d'onde est utile pour changer la longueur d'onde du port d'entrée  $X_i^l$  en une autre longueur d'onde sur un port de sortie  $X_j^o$  du commutateur  $X$ . Cette opération est notée  $(X_i^l - X - l \Rightarrow m - X_j^o)$ . Le port d'entrée et le port de sortie peuvent être respectivement remplacés par le noeud amont et aval de  $X$  sur l'arbre. Si  $T$  est un arbre, on notera  $Conv(T)$  pour traduire l'exécution d'une opération de conversion de longueur d'onde sur un noeud (ou ensemble de noeuds) de l'arbre  $T$ .

## 5. Reconfiguration par branche du routage multicast

- Pré-condition : La condition d'exécution de cette opération est que la longueur d'onde  $m$  soit libre sur le port de sortie  $X_j^O$  du commutateur  $X$ .
- Post-condition : La longueur d'onde  $m$  est occupée sur le port de sortie  $X_j^O$  du commutateur  $X$ .
- Action : La modification apportée par cette opération à la table de commutation du nœud  $X$  est que tout flux optique reçu sur le port d'entrée  $X_i^I$  du commutateur sur la longueur d'onde  $l$  est propagé au port de sortie  $X_j^O$  du même commutateur sur la longueur d'onde  $m$ .

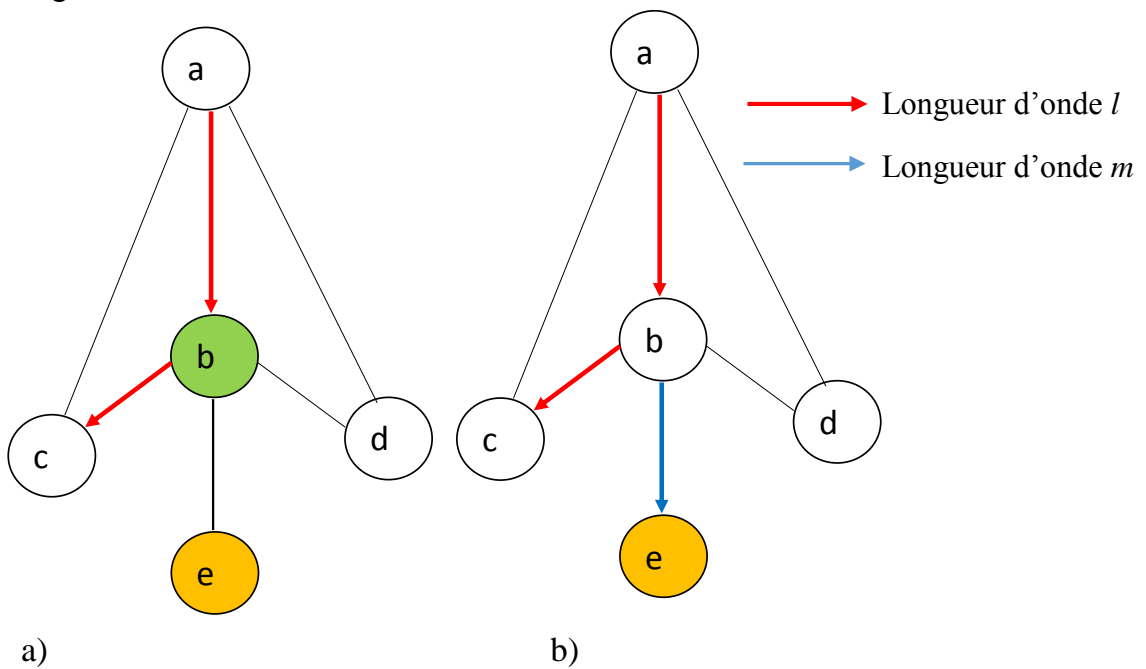


Fig. 5.5 : Exemple d'utilisation de l'opération de conversion de longueur d'onde.

La figure 5.5 décrit un exemple d'utilisation de l'opération de conversion de longueur d'onde. Dans la figure 5.5.a, l'arbre de source  $a$  et de destination le nœud  $c$  représente le chemin optique initial d'une connexion multicast de source  $a$  et destinations  $\{c\}$ . On suppose que la longueur d'onde  $l$  est occupée sur le lien  $(b - e)$  par une autre connexion. On veut que le flux de la connexion dont l'arbre est représenté sur la figure 5.5.a atteigne le nœud  $e$ . On exécute alors l'opération de conversion de longueur d'onde  $(a - b - l \Rightarrow m - e)$  sur le nœud  $b$ . Cette opération va diffuser le signal venant du nœud  $a$  sur la longueur d'onde  $l$  vers le nœud  $e$  sur la longueur d'onde  $m$ , avec  $l$  différente de  $m$ . Après l'exécution de l'opération, on obtient la figure 5.5.b. La longueur d'onde  $m$  devient occupée sur le lien  $(b - e)$ .

#### 5.4.4 Opération de multi-basculement optique

L'opération de multi-basculement de longueur d'onde est utile pour simultanément interrompre le flux sur un port de sortie  $X_j^O$  et diffuser le même flux sur un ensemble de ports de sortie  $\{X_k^O\}$  du même commutateur  $X$ . Cette opération est notée  $(X_i^I - X - l - X_j^O \Rightarrow \{X_k^O\})$ . Le port d'entrée et les ports de sortie peuvent être respectivement remplacés par les noms des commutateurs amont et aval de  $X$  sur l'arbre. Si  $T$  est un arbre, on notera  $Comb(T)$  pour traduire l'exécution d'une opération de multi-basculement de longueur d'onde sur un nœud (ou ensemble de nœuds) de l'arbre  $T$ .

- Pré-condition : La condition d'exécution de cette opération est que la longueur d'onde  $l$  soit libre sur les ports de sortie  $\{X_k^O\}$  du commutateur  $X$  et occupée sur le port de sortie  $X_j^O$ .
- Post-condition : La longueur d'onde  $l$  est occupée sur les ports de sortie  $\{X_k^O\}$  et libre sur le port de sortie  $X_j^O$  du commutateur  $X$ .
- Action : La modification apportée par cette opération à la table de commutation du nœud  $X$  est que tout signal reçu sur le port d'entrée  $X_i^I$  du commutateur est propagé aux ports de sortie  $\{X_k^O\}$  du même commutateur sur la longueur d'onde  $l$  et qu'aucun signal n'est diffusé sur le port de sortie  $X_j^O$ .

La figure 5.6 décrit un exemple d'utilisation de l'opération de multi-basculement de longueur d'onde. Dans la figure 5.6.a, l'arbre (en rouge) représente le chemin optique initial d'une connexion de source  $a$  et de destinations  $c$  et  $d$ . On veut changer le chemin de cette connexion.

Le nouveau chemin qu'on veut utiliser est l'arbre (en bleu) de racine le nœud  $a$ . La reconfiguration ici consistera à établir le nouveau chemin comme le montre la figure 5.6.a. Le nouveau chemin préétabli mais non alimenté par le flux optique sur cette figure est représenté en bleu avec la base barrée (pour dire qu'il n'est pas encore alimenté par la source  $a$ ). Ensuite on configure le nœud racine  $a$  pour simultanément interrompre le flux sur le chemin initial et alimenter le nouveau chemin.



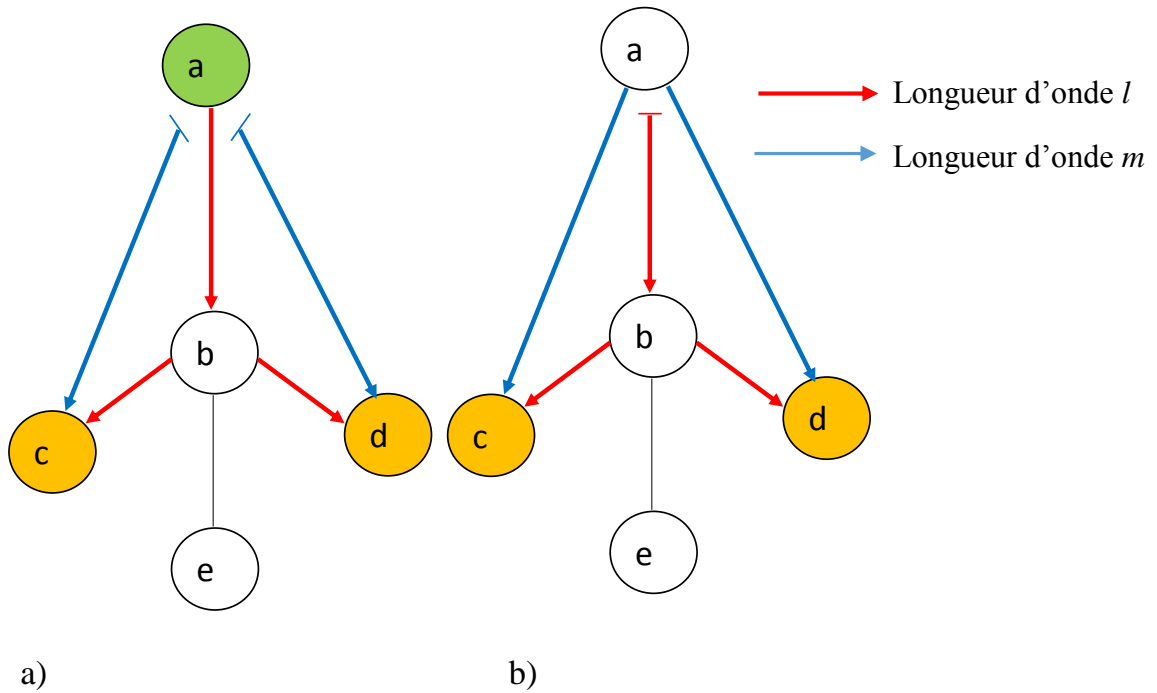


Fig. 5.6 : Exemple d'utilisation de l'opération de multi-basculément.

L'opération utilisée pour configurer le nœud  $a$  afin de mener cette action est l'opération de multi-basculément ( $- - a - l - b \Rightarrow \{c, d\}$ ). Dans cette opération, le nœud  $a$  étant le nœud source, il n'y a pas de port d'entrée.

Un cas particulier d'opération de multi-basculément est l'opération de basculément. Dans ce cas, un seul port de sortie est alimenté après le basculément.

### 5.4.5 Opération de convergence de longueur d'onde

L'opération de convergence est utile pour changer le port d'entrée  $X_i^l$  (canal optique amont du nœud  $X$ ) d'un commutateur  $X$  dont le signal est commuté sur un port de sortie  $X_j^o$  sans interrompre le flux vers ce port de sortie. Cette opération est notée  $(X_i^l \Rightarrow X_k^l - X - l - X_j^o)$ . Pour simplifier les notations, et sans perte de généralité, les ports d'entrée et le port de sortie peuvent être respectivement remplacés par les noms des commutateurs amont et aval de  $X$  sur l'arbre, si il existe au plus un lien optique entre tout couple de nœuds.

- Pré-condition : La condition d'exécution de cette opération est que la longueur d'onde  $l$  est libre sur le port d'entrée  $X_k^l$  ; et occupée sur le port d'entrée  $X_i^l$  et sur le port de sortie  $X_j^o$  du commutateur  $X$ .
- Post-condition : La longueur d'onde  $l$  est occupée sur le port d'entrée  $X_k^l$  du commutateur  $X$ .

## 5. Reconfiguration par branche du routage multicast

- Action : La modification apportée par cette opération à la table de commutation du nœud  $X$  est que tout signal reçu sur le port d'entrée  $X_k^I$  du commutateur  $X$  avec la longueur d'onde  $l$  est propagé au port de sortie  $X_j^O$  du même commutateur sur la même longueur d'onde  $l$ .

Pendant la reconfiguration d'une connexion, le nœud destinataire nécessite d'être reconfiguré si le port d'entrée dans la destination n'est pas le même entre les deux chemins et si on est dans un réseau optique. En effet, si on route des paquets indépendamment de l'interface d'entrée (comme sur IP) alors il n'est pas nécessaire de reconfigurer la destination. Mais dans les réseaux optiques, il faut préparer le nœud de destination avant de configurer la source de la connexion. L'opération pouvant être utilisée pour la préparation de la destination de la connexion est l'opération de convergence.

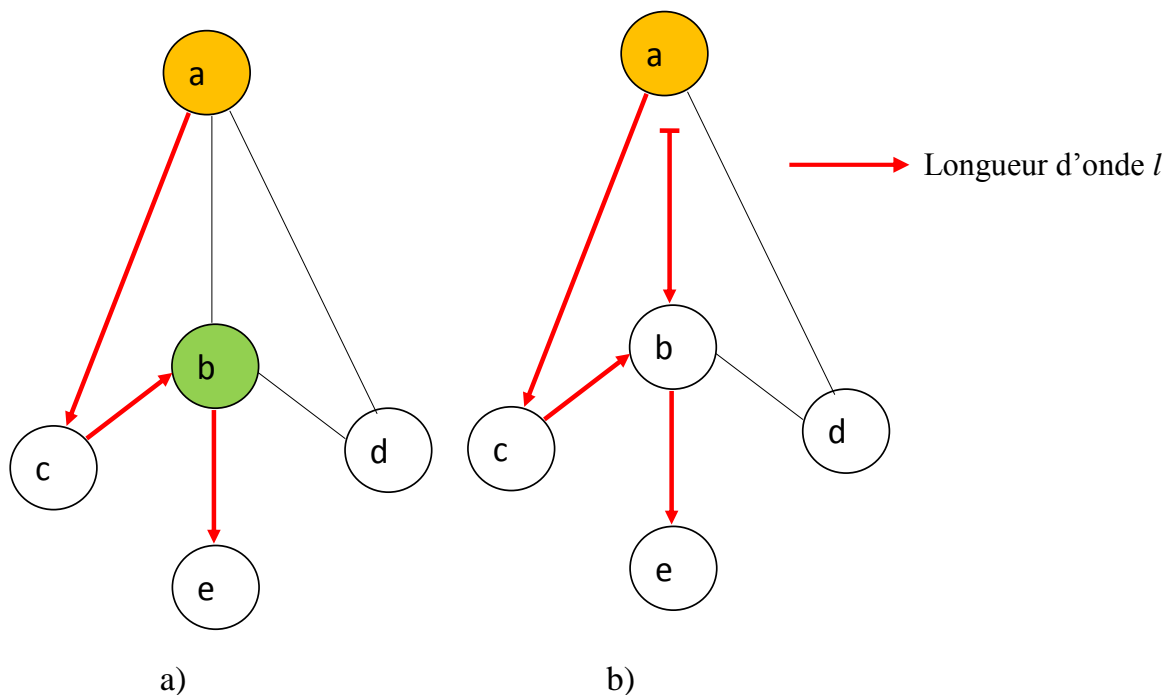


Fig. 5.7: Exemple d'application de l'opération de convergence

La figure 5.7 décrit un exemple d'utilisation de l'opération de convergence de longueur d'onde. Dans la figure 5.7.a, le chemin (en rouge) entre les nœuds  $a$  et  $e$  représente le chemin optique initial d'une connexion de source  $a$  et de destination  $e$ . On veut changer le chemin de cette connexion. Le nouveau chemin qu'on veut utiliser est  $(a - b - e)$ . La longueur d'onde  $l$  n'est pas libre sur le lien  $(b - e)$ , donc ce chemin ne peut être établi pour la connexion. Mais on peut changer le routage en utilisant l'opération de convergence sur le nœud  $b$ . On exécute l'opération de convergence ( $c \Rightarrow a - b - l - e$ ) sur le nœud  $b$ . La longueur d'onde  $l$  sur le

lien  $(a - b)$  devient occupée par cette connexion. Le nœud  $b$  est donc configuré pour commuter le flux venant du nœud  $a$  sur la longueur d'onde  $l$  vers le nœud  $e$  sur la même longueur d'onde. La figure 5.7.b est la configuration des nœuds obtenus après cette action. Ensuite, on peut configurer le nœud  $a$  avec l'opération de multi-basculement pour basculer le flux de l'ancien chemin au nouveau chemin.

### 5.4.6 Opération de non-convergence

L'opération de non-convergence est utile pour annuler l'action de l'opération de convergence sur un nœud. Cette opération est exécutée sur un nœud convergent qui est déjà configuré avec une opération de convergence. Cela permet au nœud de commuter le flux venant uniquement d'un seul port d'entrée défini.

- Pré-condition : l'opération de convergence est exécutée sur le nœud  $X$ .
- Post-condition : l'opération de convergence est annulée sur le nœud  $X$ .
- Action : le nœud  $X$  a un seul port d'entrée précisé dans la configuration.

## 5.5 Problème de reconfiguration multicast utilisant une seule longueur d'onde

Considérons le premier contexte d'étude qui consiste à utiliser une seule longueur d'onde et la même pour tous les arbres lors du processus de reconfiguration. Ce problème se présente comme suit :

Soit un réseau modélisé par un graphe  $G = (V, E)$  où  $V$  représente l'ensemble des nœuds et  $E$  l'ensemble des liens. Un nœud du graphe représente un commutateur optique dans le réseau physique et un lien du graphe représente une fibre dans le réseau physique. Soit  $W$  représentant l'ensemble des longueurs d'onde de chaque lien du réseau. Soit  $\lambda_0$  une longueur d'onde appartenant à  $W$ . Soit une connexion multicast de source  $S$  et de destinations l'ensemble  $D$ . Supposons que cette connexion est supportée initialement par un arbre initial  $T_0 = (V_0, E_0)$ . Supposons que l'arbre initial de la connexion multicast doit être modifié (reconfiguré) pour former un autre arbre (arbre final)  $T_z = (V_z, E_z)$ . L'arbre initial et l'arbre final ont la même source (celle de la connexion multicast) et le même ensemble de nœuds destinataires (celui de la connexion multicast) et sont inclus dans le graphe  $G$ . Un nœud du réseau pouvant avoir plusieurs ports, le problème de reconfiguration du routage multicast consiste à déterminer un ordre  $Seq = \{o(n_1), o(n_2), \dots, o(n_{|Seq|})\}$  sur les ports des nœuds constituant les arbres initial

## 5. Reconfiguration par branche du routage multicast

et final de telle sorte qu'on produise une séquence de reconfiguration constituée d'une série d'arbres optiques  $T_0; T_1; T_{..}; T_{z-1}; T_z$  couvrant la source et les destinations du groupe multicast associé à la connexion multicast. Chaque arbre doit utiliser la même longueur d'onde  $\lambda_0$  sur chacun de ses liens.  $n_i$  représente le port configuré à la  $i^{\text{ème}}$  étape avec l'opération de configuration  $o$ .

Supposons que la solution de ce problème existe, c'est-à-dire qu'il existe un algorithme qui permet de faire le changement d'arbre sous les contraintes d'existence d'arbres optiques utilisant la même longueur d'onde  $\lambda_0$  couvrant la source et les destinations du groupe multicast à chaque étape du processus. Alors pour toute instance du problème constituée des arbres  $T_i$  et  $T_{i+1}$  ayant la même racine et le même ensemble de destinations, on peut passer de  $T_i$  à  $T_{i+1}$  en produisant une séquence de reconfiguration constituée d'une série d'arbres optiques utilisant la même longueur d'onde  $\lambda_0$  sur chaque lien. Considérons le problème de reconfiguration formé par la paire d'arbres de la Fig. 5.8):

$T_0: \{ S \{B\{D ; A\{C\}\}\}\}$  et  $T_z: \{S \{A\{C\} ; B\{D\}\}\}$ .

$T_0$  est l'arbre initial (en rouge) et  $T_z$  est l'arbre final (en bleu). Supposons que l'arbre optique initial utilise la longueur d'onde  $\lambda_0$ .

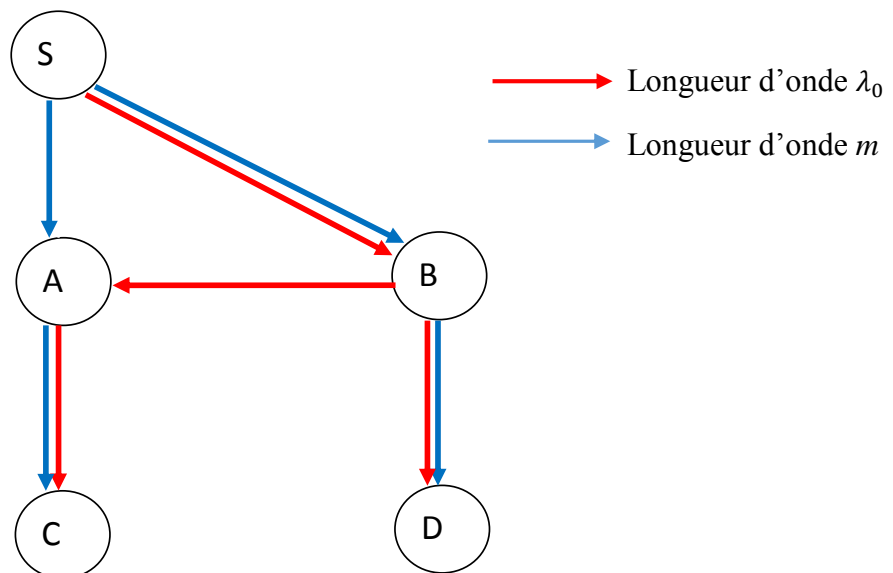


Fig. 5.8: Arbre initial  $T_0$  et arbre final  $T_z$

Nous allons étudier le passage de l'arbre initial à l'arbre final en explorant toutes les séquences de reconfiguration possibles des ports des nœuds. Nous allons utiliser pour cela les opérations de commutation optiques définies. Il n'est donc pas nécessaire de configurer les destinations

lorsqu'elles sont des nœuds feuilles car les opérations ne le permettent pas. Donc dans notre cas, seuls peuvent être configurés les trois nœuds : S, A et B.

### 5.5.1 Etude des séquences de reconfigurations d'un exemple de problème

Dans cette section, nous allons étudier les différentes séquences de reconfiguration de l'exemple de la figure 5.8. Les différents ports à configurer sont :

- Le port de sortie du nœud A vers le nœud C noté  $A \rightarrow C$
- Le port de sortie du nœud B vers le nœud D noté  $B \rightarrow D$
- Le port de sortie du nœud B vers le nœud A noté  $B \rightarrow A$
- Le port de sortie du nœud S vers le nœud B noté  $S \rightarrow B$
- Le port de sortie du nœud S vers le nœud A noté  $S \rightarrow A$ .

Le port de sortie  $B \rightarrow D$  appartient à l'arbre initial et est alimenté par le port d'entrée  $S \rightarrow B$ . Ce même port sera utilisé par l'arbre final et sera alimenté également par le port d'entrée  $S \rightarrow B$ . Donc le port  $B \rightarrow D$  ne nécessite plus de configuration. Le port de sortie  $S \rightarrow B$  appartient à l'arbre initial et est alimenté par la source de la connexion S. Ce même port sera utilisé par l'arbre final et sera alimenté également par le nœud source S. Donc le port  $S \rightarrow B$  ne nécessite plus de configuration. Les ports nécessitant une configuration sont :  $A \rightarrow C$ ,  $B \rightarrow A$  et  $S \rightarrow A$ . Les trois ports appartiennent à des nœuds différents. Nous allons regrouper les séquences de reconfigurations selon le nœud de début afin de les étudier.

#### a) Cas 1 : Ordre de reconfiguration commençant par le nœud A

Les ordres de configuration possibles dans ce cas sont : (A-B-S) et (A-S-B).

- Ordre (A-B-S) : Sur le nœud B, le port à configurer est  $B \rightarrow A$ . Ce port n'est pas isolé. Seule l'opération de suppression de commutation peut être exécutée pour ce port. Mais cette configuration n'est pas envisageable car elle interrompt le flux vers le nœud destinataire C. Donc l'ordre (A-B-S) n'est pas envisageable.
- Ordre (A-S-B) : Sur le nœud S, les opérations possibles pour le port  $S \rightarrow A$  sont l'ajout de commutation et le basculement de flux. L'ajout de commutation ne peut être effectué sinon le port de sortie sera alimenté par deux ports d'entrée  $S \rightarrow A$  et  $B \rightarrow A$ . Cela ne respecte pas les contraintes liées à notre problème de reconfiguration notamment un canal optique doit être alimenté par au plus un canal optique amont. L'opération de

basculement n'est pas envisageable car elle interrompt le flux vers la destination D. Donc la séquence (A-S-B) n'est pas envisageable.

Aucun ordre de reconfiguration commençant par le nœud A n'est donc envisageable pour une reconfiguration sans interruption de flux de l'exemple de la figure 5.8.

### b) Cas 2 : Ordre de reconfigurations commençant par le nœud B

Le port à configurer sur le nœud B est  $B \rightarrow A$ . Ce port n'est pas isolé. L'opération possible à exécuter sur ce port est la suppression de commutation. Mais cette opération n'est pas envisageable car elle interrompt le flux vers le nœud destinataire C. Donc aucun ordre de reconfiguration commençant par le nœud B n'est envisageable.

### c) Cas 3 : Ordre de reconfigurations commençant par le nœud S

Le port à configurer sur le nœud S est  $S \rightarrow A$ . Les opérations de configuration possibles sont l'ajout de commutation et le basculement de flux. L'ajout de commutation ne peut être effectué sinon le port de sortie sera alimenté par deux ports d'entrée  $S \rightarrow A$  et  $B \rightarrow A$ . Cela ne respecte pas les contraintes liées à notre problème de reconfiguration notamment un canal optique doit être alimenté par au plus un seul canal amont. Le basculement de flux n'est pas envisageable car il interrompt le flux vers la destination D. Donc aucun ordre de reconfiguration commençant par le nœud S n'est envisageable.

L'exemple de la figure 5.8 présente un exemple d'instance du problème de reconfiguration pour lequel aucun ordre des séquences de reconfiguration ne permet de faire la reconfiguration sans interruption de flux, en utilisant une seule longueur d'onde (par conséquent la même sur chaque fibre). On conclut qu'il existe des instances du problème de reconfiguration ne pouvant être résolus sans interruption de flux, avec une seule longueur d'onde. Cela arrive lorsque certaines ressources optiques ne sont pas disponibles sur les liens partagés par l'arbre initial et l'arbre final. Il n'existe donc pas de solution pour les problèmes de reconfiguration appartenant à ce premier contexte. En appliquant la technique de résolution *Make Before Break*, la section suivante donne une idée du nombre d'étapes pendant lesquelles la connexion est interrompue.

### 5.5.2 Evaluations

Nous présentons les critères d'évaluation, les paramètres de la simulation et les résultats.

### 5.5.2.1 Critères d'évaluation

Nous définissons un premier critère d'évaluation qui est la durée d'interruption dans une séquence de reconfiguration. Le deuxième critère d'évaluation est la durée d'une séquence de reconfiguration. Une séquence de reconfiguration, commençant à l'arbre initial  $T_0$  et prenant fin à l'arbre final  $T_z$  sera notée  $Seq(T_0, T_z)$ .

- Durée d'interruption du flux optique

La durée d'interruption d'une étape  $k$  à une étape  $k + 1$  est supposée égale à la durée d'une étape de reconfiguration. La durée d'interruption mesurée est la durée totale des étapes de la séquence de reconfiguration ne contenant pas d'arbre couvrant la source et tous les nœuds destinataires.

Si  $duration(O(T_k))$  désigne la durée d'une étape de reconfiguration, elle est égale à la durée d'interruption d'une étape  $k$  à une étape  $k + 1$ . Si  $interrupt\_duration(Seq(T_0, T_z))$  désigne la durée d'interruption d'une séquence de reconfiguration  $Seq(T_0, T_z)$ , on a:

$$interrupt\_duration(Seq(T_0, T_z)) = \sum_{k=0}^{|Seq(T_0, T_z)|-1} b_k \times duration(O(T_k))$$

$$\text{où } b_k = \begin{cases} 1 & \text{s'il n'existe pas d'arbre couvrant la source} \\ & \text{et les destinations à l'étape } k. \\ 0 & \text{s'il existe un arbre couvrant la source} \\ & \text{et les destinations à l'étape } k. \end{cases}$$

- Durée d'une séquence de reconfiguration

La durée d'une séquence de reconfiguration est la durée totale de toutes les étapes de reconfiguration constituant la séquence. On désigne par  $duration(Seq(T_0, T_z))$  la durée d'une séquence de reconfiguration.

Elle est déterminée par l'expression ci-dessous:

$$duration(Seq(T_0, T_z)) = \sum_{k=0}^{|Seq(T_0, T_z)|-1} duration(O(T_k))$$

### 5.5.2.2 Description de la simulation

Les topologies de réseau utilisées pour évaluer les algorithmes sont COST 239 (Fig. 5.9), NSF Network (Fig. 5.10) et US Backbone Network (Fig. 5.11). Ces topologies sont bien connues, et souvent utilisées dans la littérature. Elles présentent différents caractéristiques. COST 239 est formé de 11 nœuds et 26 liens, NSF Network est formé de 14 nœuds et 21 liens et US Backbone Network est formé de 24 nœuds et 43 liens. Le tableau 5.1 présente quelques caractéristiques de ces trois topologies.

Tableau 5.1 : Caractéristiques des topologies utilisées dans l'évaluation

|             | Min<br>nodal degree | Average<br>nodal degree | Max<br>nodal degree |
|-------------|---------------------|-------------------------|---------------------|
| COST 239    | 4                   | 4.7                     | 6                   |
| NSF Network | 2                   | 3                       | 4                   |
| US Backbone | 2                   | 3.58                    | 5                   |

Pour une topologie donnée, nous sélectionnons de façon aléatoire une connexion multicast. La source de la connexion est choisie de façon aléatoire parmi les nœuds du réseau. Le nombre  $k$  de destinations du groupe multicast est déterminé de façon aléatoire parmi les nœuds restants du réseau,  $k \in [1; n - 1]$  où  $n$  est le nombre de nœuds du réseau. Pour chaque connexion, deux arbres sont choisis comme arbre initial et arbre final du problème de reconfiguration : L'arbre des plus courts chemins entre la source et chaque destination (SPT) et l'arbre minimum de Steiner (l'arbre partiel couvrant la source et les destinations de poids minimum, MST).

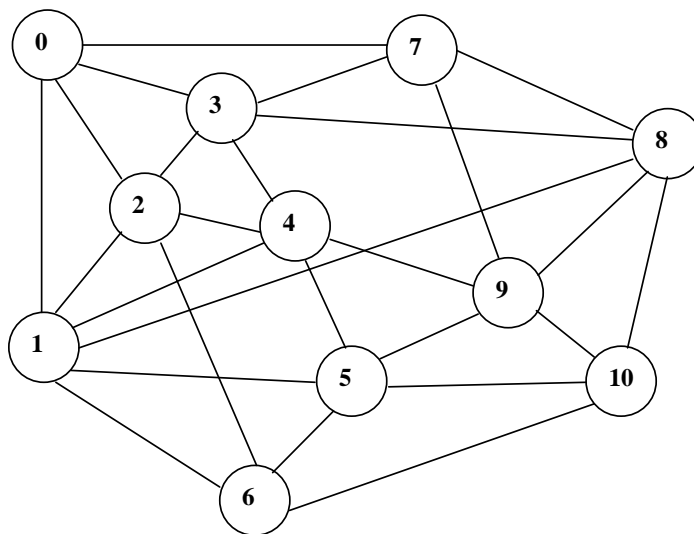


Fig. 5.9 : Topologie COST 239



## 5. Reconfiguration par branche du routage multicast

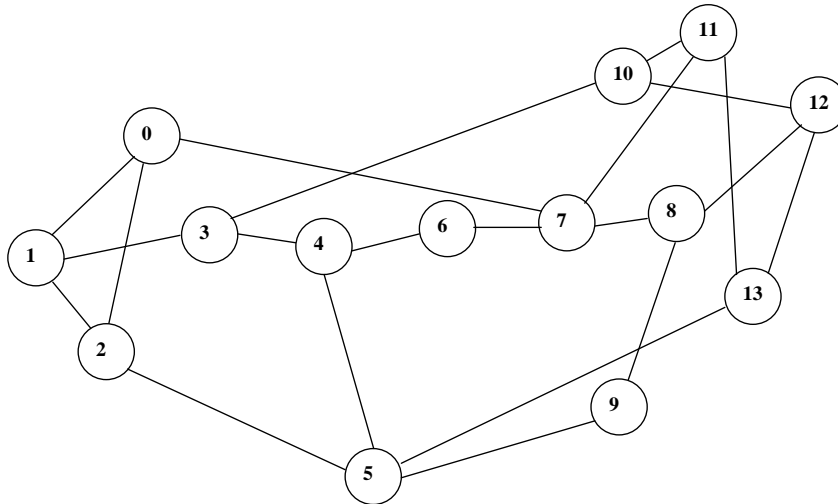


Fig. 5.10 : Topologie NSF Network

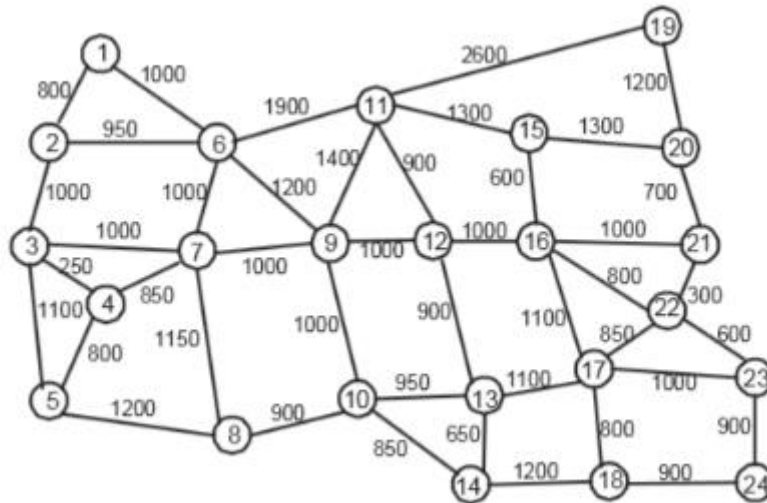


Fig. 5.11 : Topologie US Backbone

L'algorithme de Dijkstra [Dij57] est appliqué entre le nœud source et chaque destination pour calculer l'arbre des plus courts chemins. La détermination de l'arbre minimum de Steiner est connue comme un problème difficile à résoudre. Nous utilisons l'algorithme de Prim [Pri57] pour déterminer une approximation de l'arbre minimum de Steiner couvrant la source et les nœuds destinataires. Nous supposons par défaut que l'arbre initial est l'arbre SPT et l'arbre final est l'approximation de l'arbre MST. Nous étudierons le choix inverse et nous verrons qu'il ne change pas les résultats.

Nous supposons que les nœuds ont une capacité de *splitting* (duplication) appropriée. Cela signifie que la capacité de *splitting* de chaque nœud est supérieure ou égale au nombre maximum de nœuds voisins à un nœud du réseau moins un et que le nombre de *splitters* dans

le nœud est supérieur au nombre de connexions multicast utilisant ce nœud comme nœud de branchement. Cela permet de construire l'arbre initial et l'arbre final de notre problème de reconfiguration sans contrainte de nombre de nœuds voisins à chaque nœud et du nombre de sessions multicast établies dans le réseau.

Pour chaque paire d'arbres, on vérifie si les deux arbres sont différents sinon on détermine une nouvelle paire d'arbres (car sinon la reconfiguration est triviale). Pour chaque détermination d'arbres, différentes tailles du groupe multicast sont utilisées. La taille du groupe multicast est déterminée de façon aléatoire. Le nombre de paires d'arbre initial et d'arbre final représentant les problèmes de reconfiguration par simulation est fixé à 5000. Ce nombre a été fixé de façon expérimentale. Après 5000 tirages de paires d'arbres, la variation de la moyenne est supposée petite (inférieure à 3 %). Nous utilisons la méthode de Monte-Carlo pour évaluer les performances des différents critères retenus (durée d'interruption, coût de ressources additionnelles et durée d'interruption dans chaque topologie)

### 5.5.2.3 Résultats et analyses des résultats

La reconfiguration se fait sous la contrainte d'utilisation d'une seule longueur d'onde par lien. Donc aucune ressource additionnelle n'est utilisée durant le processus de reconfiguration. Pour les réseaux utilisés le nombre minimum d'étapes d'une séquence de reconfiguration est 3 et le nombre maximum d'étapes est 4. La durée moyenne d'une séquence de reconfiguration est 3,68 étapes avec une variance de 0,20 étapes dans la topologie COST 239, 3,80 étapes avec une variance de 0,16 étapes dans la topologie NSF Network, et 3,73 étapes avec une variance de 0,18 étapes dans la topologie US Backbone. Ces durées s'expliquent par le fait qu'en absence de longueurs d'onde disponibles pour la reconfiguration du nouvel arbre, on interrompt le flux sur l'ancien arbre avant l'établissement du nouveau. Cela nécessite 3 étapes (lorsque les ressources sont disponibles) à savoir le pré-établissement du nouveau chemin, l'alimentation du chemin préétabli à la source et la suppression du nouveau chemin. En cas d'indisponibilité des ressources, 4 étapes sont nécessaires. Ce sont l'interruption du flux sur l'ancien chemin, la suppression de l'ancien chemin, le pré-établissement du nouveau chemin et l'alimentation du chemin préétabli à la source.

L'objectif principal de nos travaux est de reconfigurer une paire d'arbres optiques sans interrompre le flux. La reconfiguration d'un arbre sous la contrainte d'utilisation d'une seule et même longueur d'onde produit des interruptions dans certains problèmes. La durée d'interruption moyenne est 2,13 étapes avec une variance de 2,23 étapes dans la topologie

COST 239 ; 2,36 étapes dans la topologie NSF Network avec une variance de 2,65, et 2,21 étapes avec une variance de 2,37 étapes dans la topologie US Backbone. Cela montre que la continuité du flux n'est pas garantie lorsque la reconfiguration se fait sous la contrainte d'utilisation d'une seule longueur d'onde par lien. Ces interruptions sont dues à l'indisponibilité de ressources sur les liens que l'arbre initial et l'arbre final ont en commun.

Notre objectif est de résoudre le problème de reconfiguration d'une connexion multicast dans un réseau optique quel que soient l'arbre initial et l'arbre final la constituant. Nous avons étudié que certaines instances de reconfiguration ne peuvent être résolues sans interruption de flux. Nous avons évalué le nombre moyen d'interruptions par séquence de reconfiguration. Ce nombre est élevé, il produit un délai d'interruption qui ne peut être toléré dans de nombreuses situations. Nous allons étudier ce problème dans le contexte où on s'autorise l'utilisation d'une longueur d'onde additionnelle, lorsque cela s'avère absolument nécessaire, temporairement et localement. Dans ce cas, on suppose que les nœuds sont équipés de (rares) convertisseurs de longueurs d'onde.

### **5.6 Problème de reconfiguration d'une connexion multicast utilisant une longueur d'onde additionnelle (temporairement et localement)**

#### **5.6.1 Introduction**

Cette section présente nos différentes contributions à la résolution du problème de reconfiguration d'une connexion multicast en utilisant une longueur d'onde additionnelle. Considérons le second contexte dans lequel on s'autorise à utiliser une longueur d'onde additionnelle pour passer de l'arbre initial à l'arbre final ; on note toutefois que l'arbre initial et l'arbre final n'utilisent qu'une longueur d'onde (la même longueur d'onde). L'hypothèse que nous faisons, à savoir « l'arbre initial et l'arbre final » doivent utiliser la même longueur peut être abandonnée a priori. Mais en la conservant, cela permettra par exemple d'éviter les conversions de longueur d'onde durant toute la durée de la connexion. Cette durée pouvant être longue (plusieurs mois), l'utilisation de la même longueur d'onde peut permettre des économies d'énergie liées aux opérations de conversion.

La continuité du flux vers toutes les destinations de la connexion multicast à reconfigurer est garantie par l'existence d'un arbre couvrant la source et toutes les destinations du groupe multicast après l'exécution de toute opération sur un nœud.

Nous donnons quelques définitions utilisées dans la suite du travail.

### 5.6.2 Définitions

#### - Nœud de basculement

Un nœud de basculement est un nœud sur lequel est exécutée une opération de basculement de flux optique d'un port de sortie vers un autre port de sortie.

#### - Branche

Une branche est un chemin optique entre la source de la connexion multicast et une destination du groupe multicast sur l'arbre optique de la connexion. La branche de l'arbre  $T_0$  de source  $S$  et de destination  $D$  sera notée  $Br(T_0, S, D)$ .

Une branche appartenant à l'arbre optique initial sera appelée ancienne branche. Une branche appartenant à l'arbre optique final sera appelée nouvelle branche. Une branche appartenant à l'arbre optique courant sera appelée ancienne branche. Sur l'arbre produit après configuration d'une branche sur l'arbre courant, une branche sera qualifiée de nouvelle branche.

#### - Paire de branches reconfigurable

Une paire de branches constituée d'une ancienne branche et une nouvelle branche est reconfigurable si on peut préétablir la nouvelle branche et basculer le flux de l'ancienne branche à la nouvelle sans interrompre le flux vers les autres destinations.

Nous présentons ci-dessous un procédé de reconfiguration systématique branche par branche (que nous appellerons "*BpBAR*" : Branche par branche aller-retour) permettant de reconfigurer un arbre sans interruption de flux.

### 5.6.3 Processus de reconfiguration par branche *BpBAR*

Le processus de reconfiguration d'une paire d'arbres optiques appartenant à la même connexion multicast par branche *BpBAR* permet de reconfigurer sans interruption de flux. Ce processus

permet de reconfigurer une ancienne branche en une nouvelle branche en utilisant une longueur d'onde disponible sur les liens du réseau.

Supposons que  $Br(T_0, S, d_j)$  et  $Br(T_z, S, d_j)$  constituent une paire de branches à reconfigurer sans interruption de flux.  $Br(T_0, S, d_j)$  est l'ancienne branche et  $Br(T_z, S, d_j)$  la nouvelle branche de même destination. Supposons que la longueur d'onde utilisée par l'arbre initial soit  $\lambda_0$ . Le processus de reconfiguration *BpBAR* proposé se décrit comme suit :

- Déterminer le nœud de basculement correspondant à la paire de branches ;
- Préétablir la nouvelle branche  $Br(T_z, S, d_j)$  entre le nœud de basculement et la destination  $d_j$  en utilisant une autre longueur d'onde (par exemple  $\lambda_1$ ). On note que par définition le segment de la branche entre la source et le nœud de basculement est identique sur les deux branches. L'opération d'ajout de commutation est utilisée pour tous les nœuds sauf la destination. L'opération de convergence est utilisée pour la destination ;
- Configurer le nœud de basculement pour basculer le flux de l'ancienne branche  $Br(T_0, S, d_j)$  vers la nouvelle branche  $Br(T_z, S, d_j)$  ;
- Supprimer le segment de l'ancienne branche dont le flux est interrompu. C'est le segment entre le nœud de basculement et la destination  $d_j$  appartenant à l'ancienne branche. ; l'opération de suppression de commutation est utilisée pour tous les nœuds sauf la destination. L'opération de non-convergence est utilisée pour la destination.

On répète ce processus pour toutes les branches menant à chaque destination. On obtient un nouvel arbre qui est l'arbre final et qui utilise différentes longueurs.

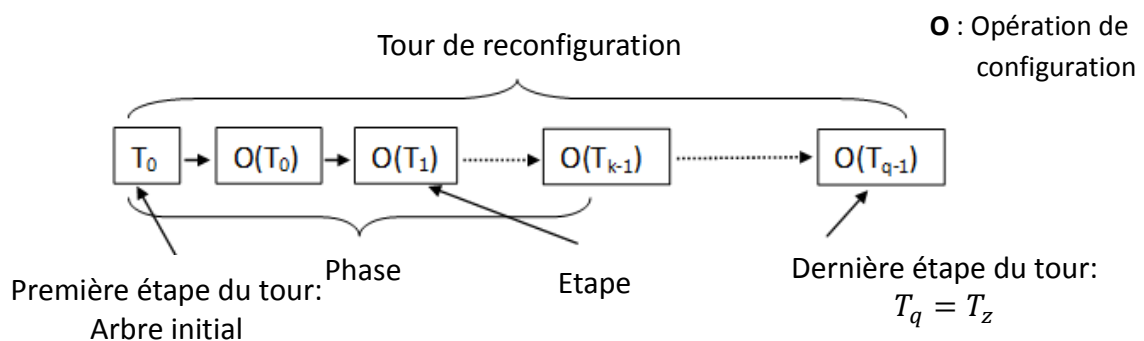


Fig. 5.12 : Description d'un tour de reconfiguration avec *BpBAR*.

Le processus décrit ci-dessus constitue le tour ALLER du processus de reconfiguration *BpBAR*. Il faut reprendre le même processus dans le tour RETOUR en reconfigurant les paires de branches avec une même longueur d'onde (longueur d'onde de l'arbre initial). On passe de l'arbre initial  $T_0$  à l'arbre final  $T_z$  en produisant une succession d'arbres  $T_k$ . La Fig. 5.12 décrit un tour de reconfiguration avec *BpBAR*. Dans cette description, O représente une opération de configuration.

La reconfiguration de toute paire de branches avec le processus *BpBAR* nécessite la détermination du nœud sur lequel exécuté l'opération de basculement, c'est-à-dire le nœud de basculement. Nous allons présenter ci-dessous l'importance du nœud de basculement de ce procédé et comment le déterminer.

### 5.6.3.1 Détermination du nœud de basculement

La détermination du nœud de basculement pour la reconfiguration d'une paire de branches est d'une importance capitale dans le processus de reconfiguration par branche. Ce nœud est déterminé de telle sorte que sa configuration avec l'opération de basculement optique interrompt le flux sur l'ancienne branche et alimente la nouvelle branche tout en assurant la continuité du flux vers toutes les destinations du groupe multicast. Le nœud de basculement par défaut est le nœud source de la connexion multicast.

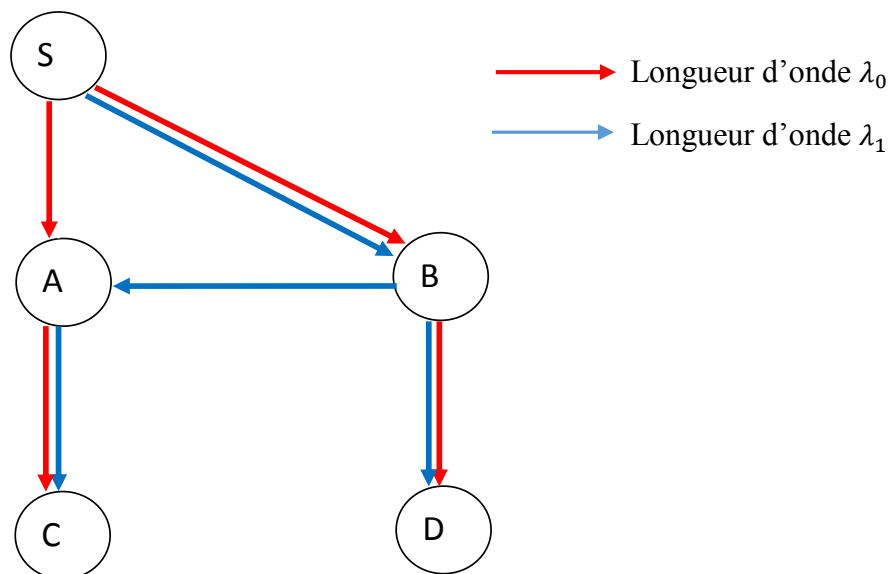


Fig. 5.13 : Exemple d'instance de reconfiguration pour BpBAR où le nœud de basculement est la source.

## 5. Reconfiguration par branche du routage multicast

Considérons le problème de reconfiguration de routage multicast formé par les arbres suivants (Fig. 5.13):  $T_0 : \{S \{A \{C\}; B \{D\}\}\}$  qui est l'arbre initial (en rouge) utilisant une longueur d'onde  $\lambda_0$  et  $T_z : \{S \{B \{A\{C\}; B \{D\}\}\}$  qui est l'arbre final (en bleu).

Dans l'application du processus de reconfiguration branche par branche, on choisit d'abord une paire de branches à reconfigurer et ensuite on applique le procédé *BpBAR*.

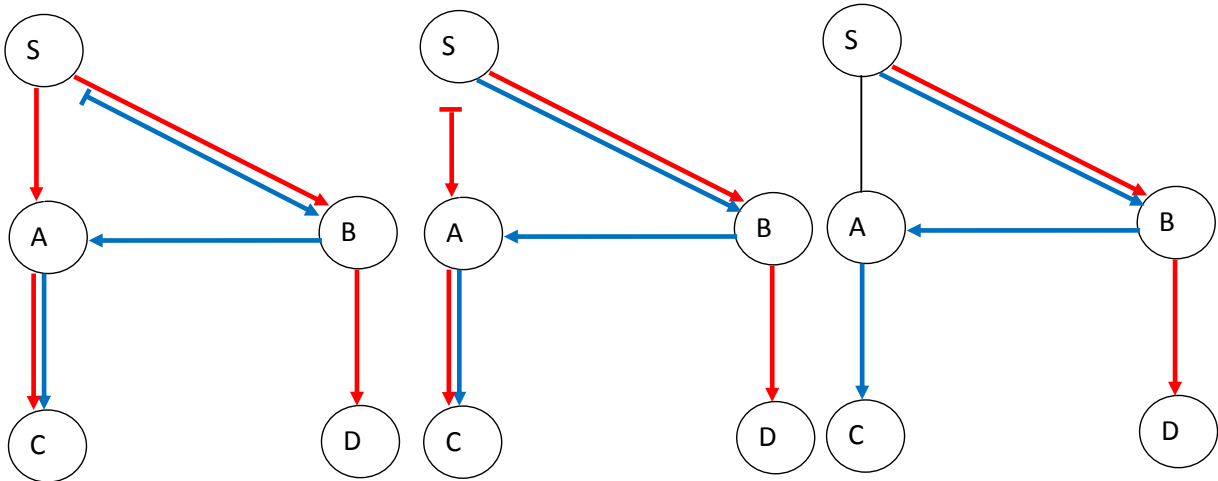


Fig. 5.14 : Séquence de reconfigurations d'une paire de branches par *BpBAR*.

Dans l'exemple 5.13, la paire de branches constituée de  $Br(T_0, S, D)$  et  $Br(T_z, S, D)$  ne nécessite pas d'être reconfigurée car ces deux branches sont identiques. On remarque que l'autre paire de branches constituée de  $Br(T_0, S, C)$  et  $Br(T_z, S, C)$  nécessite d'être reconfigurée. Le nœud de basculement pour cette paire de branches est la source S (nœud de basculement par défaut). La figure 5.14 décrit la reconfiguration dans le tour ALLER de cette paire de branches.

Considérons maintenant l'instance de reconfiguration de routage multicast formée par les arbres suivants (Fig. 5.15):  $T_0 : \{S \{B \{A\{C\}; B \{D\}\}\}\}$  qui est l'arbre initial (en rouge) utilisant une longueur d'onde  $\lambda_0$  et  $T_z : \{S \{A \{C\}; B \{D\}\}\}$  qui est l'arbre final (en bleu).

Dans cet exemple, la paire de branches  $Br(T_0, S, D)$  et  $Br(T_z, S, D)$  ne nécessite pas d'être reconfigurée car les branches sont identiques. On remarque que l'autre paire de branches constituée de  $Br(T_0, S, C)$  et  $Br(T_z, S, C)$  doit être reconfigurée. Mais le nœud de basculement pour cette paire de branches n'est pas la source S (ce n'est pas le nœud de basculement par défaut).

## 5. Reconfiguration par branche du routage multicast

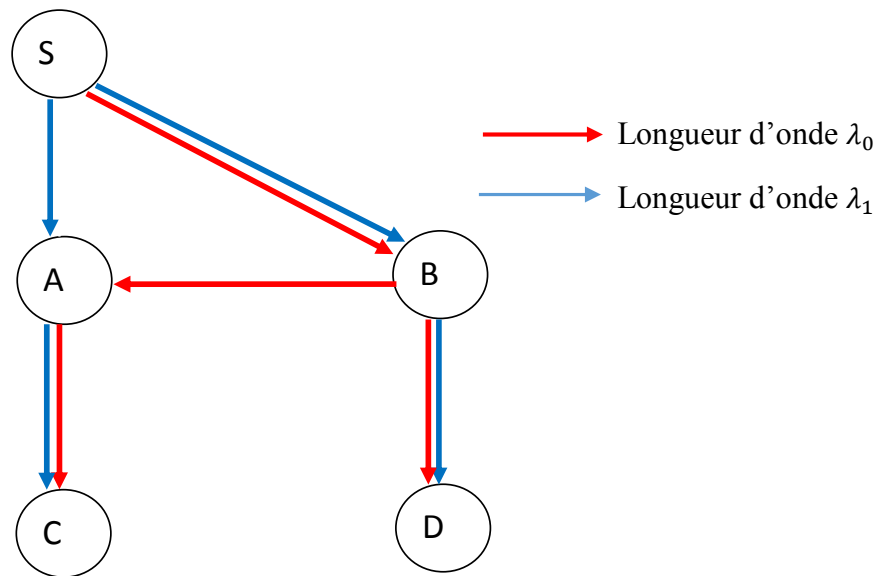


Fig. 5.15 : Exemple de l'instance de reconfiguration où le nœud de basculement n'est pas la source.

En effet, supposons que le nœud S soit le nœud de basculement de cette paire de branches. On établit la nouvelle branche entre la destination C et le nœud de basculement S (Fig. 5.16).

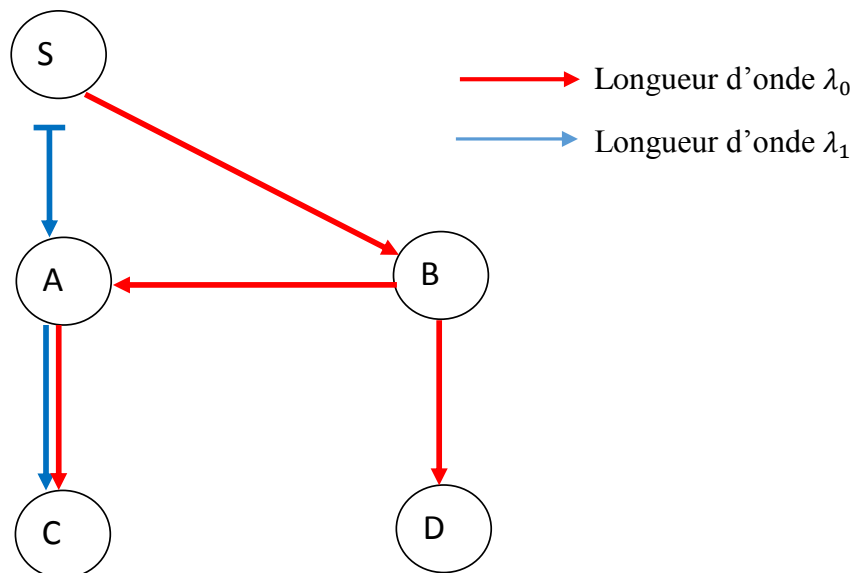


Fig. 5.16 : Etablissement de l'arbre final  $\mathbf{Br}(T_z, S, C)$

L'étape qui va suivre est l'alimentation de la nouvelle branche et l'interruption du flux sur l'ancienne branche. Mais sur la figure 5.16, configurer la source S avec l'opération de basculement de flux va certes alimenter la nouvelle branche (la branche  $Br(T_z, S, C)$ ), mais elle va couper le flux vers la destination D. Dans ce cas, la source S n'est pas le bon nœud pour



faire le basculement. Le basculement doit se faire en un nœud où l'on peut couper le flux de la partie exclusive de l'ancienne branche vers la destination C.

Pour cette paire de branches, c'est le nœud B. Alors il faut alimenter la nouvelle branche  $Br(T_z, S, C)$  à partir du nœud B. Cela nécessite un convertisseur de longueur d'onde en B. Après la détermination du nœud de basculement pour la paire de branches à reconfigurer (ici les branches  $Br(T_0, S, C)$  et  $Br(T_z, S, C)$ ), on préétablit le chemin entre la destination (ici le nœud C) et le nœud de basculement (Fig. 5.17.a). Ensuite on exécute l'opération de basculement sur le nœud de basculement, ici le nœud B (Fig. 5.17.b) et on supprime le segment de l'ancienne branche entre le nœud de basculement et la destination car le flux y est interrompu (Fig. 5.18).

Le nœud de basculement étant différent du nœud source S, on configure par la suite la source S pour interrompre le flux sur le segment du chemin établi entre la source de la connexion et le nœud de basculement (ici les extrémités sont les nœuds S et B). Enfin on supprime ce segment car le flux y est interrompu. On répète le même processus dans le tour RETOUR.

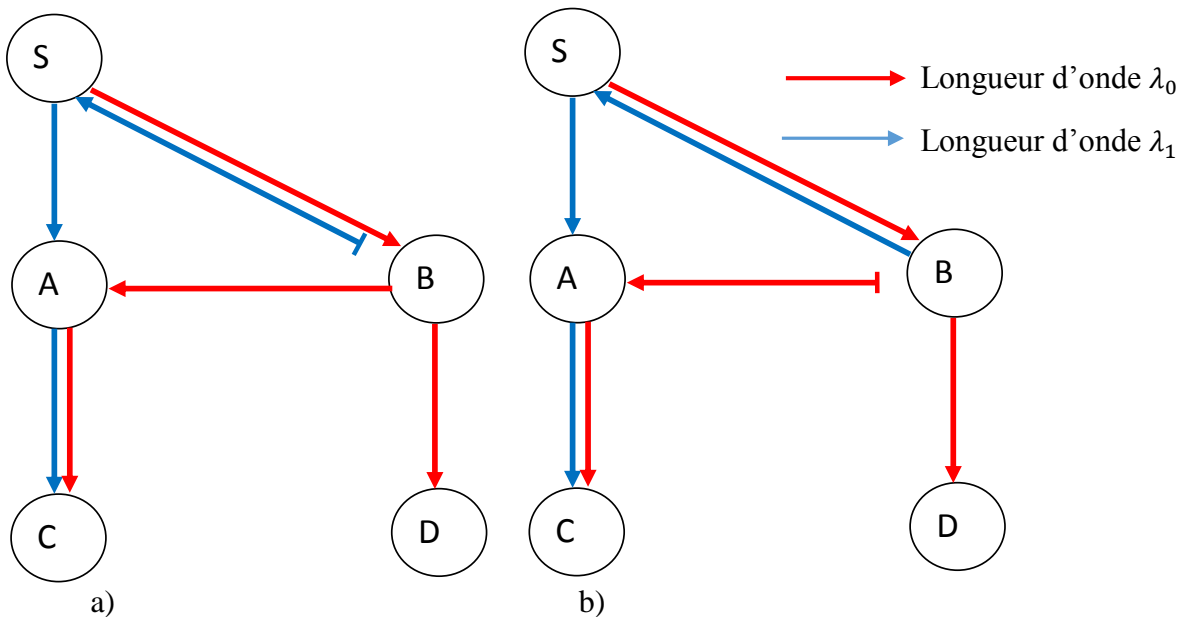


Fig. 5.17 : a) Etablissement de l'arbre final. b) Basculement du flux au nœud B.

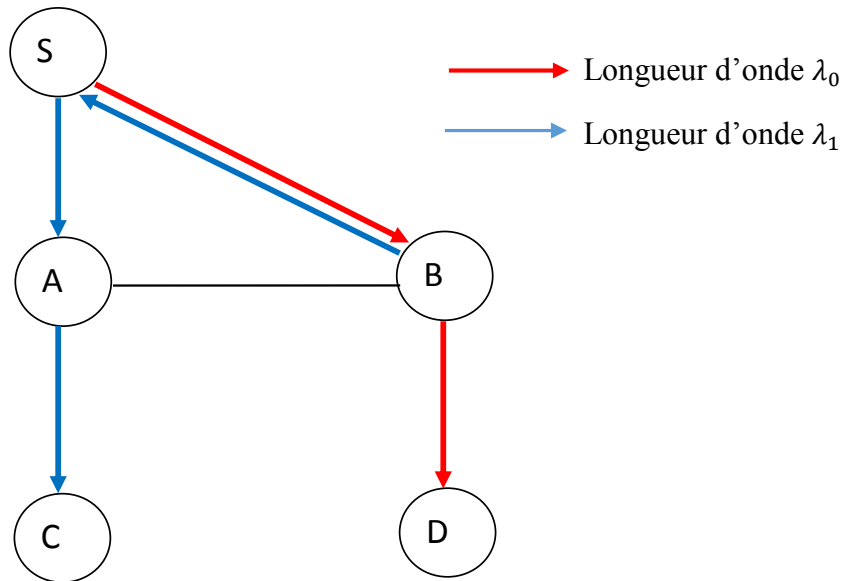


Fig. 5.18 : Arbre obtenu à la fin de la phase aller.

La question qui se pose maintenant est comment déterminer le nœud de basculement d'une paire de branches donnée. Nous présentons dans la section suivante un algorithme de détermination du nœud de basculement d'une paire de branches à reconfigurer.

### 5.6.3.2 Algorithme de détermination d'un nœud de basculement

Dans un arbre optique, nous pouvons affecter à chaque canal optique un flot. Il va nous aider à déterminer le nœud de basculement.

#### Définition :

##### - Flot sur un canal optique

Un flot sur un canal optique d'un arbre est le nombre de nœuds destinataires atteints à partir de ce canal optique.

La figure 5.19 représente un arbre optique et les flots sur chacun de ces canaux. Le canal optique entre les nœuds S et B permet d'atteindre les destinations C et D. Donc le flot sur ce lien est 2. Les canaux optiques entre les nœuds B et A et entre les nœuds A et C permettent d'atteindre une seule destination qui est le nœud C. Donc le flot sur chacun de ces lien est 1. Le canal optique entre B et D permet d'atteindre la seule destination D. Donc le flot sur ce lien est 1. Il n'y a pas de canal optique pour l'arbre entre les nœuds S et A, implicitement le flot est nul.

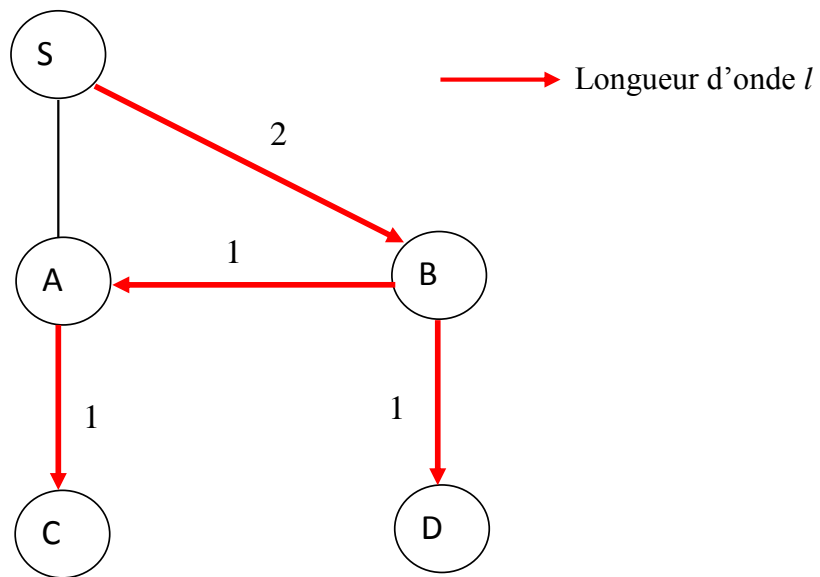


Fig. 5.19 : Arbre optique et flots associés aux différents canaux optiques.

**Propriété :**

Un segment constitué de canaux optiques ayant pour flot 1 atteint exactement une seule destination.

En effet, le nombre de destinations atteintes par un canal représente le flot sur ce canal. Si le flot sur chaque canal d'un segment est 1, alors chaque canal atteint exactement une destination. D'où le segment lui-même atteint exactement une destination.

**Propriété :**

Une paire de branches est reconfigurable sans interruption de flux si l'ancienne branche de la paire est constituée de canaux optiques de flot 1.

En effet en appliquant le processus de reconfiguration par branche *BpBAR*, on établit la nouvelle branche entre la destination et le nœud de basculement avec une longueur d'onde différente de celle utilisée par l'arbre courant (arbre utilisé pour la transmission du flux). Ensuite on bascule le flux de l'ancienne branche à la nouvelle branche en configurant le nœud de basculement (avec l'opération de basculement). Cela interrompt le flux sur les canaux optiques aval du nœud de basculement appartenant à l'ancienne branche. Les canaux optiques aval du nœud de branchement sur l'ancienne branche atteignant seulement une seule destination (la destination alimentée par la nouvelle branche), le flux vers les autres

destinations n'est pas interrompu. Il existe alors un arbre couvrant toutes les destinations et la source de la connexion.

**Notation :**

Le nœud de basculement permettant de configurer la paire de branches  $Br(T_0, S, d_j)$  et  $Br(T_z, S, d_j)$  à l'étape  $i$  de reconfiguration sera noté  $Sw\_node(T_i, d_j)$  où  $T_i$  représente l'arbre couvrant la source et toutes les destinations à cette étape.

L'algorithme 5.1 présente le processus de détermination du nœud de basculement  $Sw\_node(T_i, d_j)$  pour une paire de branches  $Br(T_i, S, d_j)$  et  $Br(T_{i+1}, S, d_j)$  à une étape  $i$  de reconfiguration.

Le nœud de basculement  $Sw\_node(T_i, d_j)$  pour cette paire de branches est tel que:

- les canaux optiques sur le segment de l'ancienne branche  $Br(T_i, S, d)$  entre  $Sw\_node(T_i, d_j)$  et  $d_j$  ont pour flot 1,
- si  $y$  est le nœud amont de  $Sw\_node(T_i, d_j)$  sur l'ancienne branche  $Br(T_i, S, d)$ , alors le flot sur le canal optique entre  $y$  et  $Sw\_node(T_i, d_j)$  est strictement supérieur à 1.

**Algorithme 5.1: Algorithme de détermination de nœud de basculement**

---

```

Sw_Node ( $T_i, T_z, d$ )
1. {
2.    $Nd\_de\_Basculement = d$  /*initialisation*/
3.   Si ( $flot(Amont(T_i, Nd\_de\_Basculement), Nd\_de\_Basculement)=1$ )
4.   {
5.      $Nd\_de\_Basculement = Amont(T_i, Nd\_de\_Basculement)$ 
6.   Sw_Node ( $T_i, T_z, Nd\_de\_basculement$ )
7.   }
8. }
```

---

Dans l'exemple de la figure 5.19, le nœud de basculement pour la paire de branches  $Br(T_0, S, C)$  et  $Br(T_z, S, C)$  est le nœud B. En effet, le nœud amont de B sur l'ancienne branche est S. Les canaux optiques du segment entre les nœuds B et C sur l'ancienne branche  $Br(T_0, S, C)$  ont pour flot 1. Le flot du canal entre les nœuds S et B sur la même branche est 2.

Nous présentons dans les sections qui suivent des algorithmes de reconfiguration d'arbre optique basés sur l'approche de reconfiguration par branche.

Nous définissons un premier algorithme de reconfiguration du routage multicast utilisant le procédé *BpBAR*. Cet algorithme est un processus systématique de reconfiguration par branche, exécutant une seule opération par étape. Cet algorithme est appelé *BpBAR\_1*. Une amélioration de cet algorithme qui exécute certaines opérations en parallèle est ensuite présentée. Cette approche permet de réduire la durée de reconfiguration, mais au plus une seule paire de branches est reconfigurée à la fois. Cet algorithme est appelé *BpBAR\_2*. La durée de reconfiguration est encore améliorée par un autre algorithme de reconfiguration par branche appelé *PRoTB*. Cet algorithme permet la reconfiguration de plusieurs paires de branches en parallèle. Le dernier algorithme de reconfiguration par branche est appelé *TRwRC*, il permet la reconfiguration d'un arbre dans un réseau ayant des ressources (longueurs d'onde) limitées.

#### 5.6.4 Algorithme de reconfiguration *BpBAR\_1*

L'algorithme 5.2 présente l'algorithme *BpBAR\_1*. C'est un algorithme de reconfiguration par branche. Il se déroule donc en deux tours : un tour ALLER et un tour RETOUR.

##### Algorithme 5.2: Algorithme de reconfiguration par branche *BpBAR\_1*

---

```

BpBAR_1( $T_0, T_z$ )
1. {
2.  $T = T_0$ 
3. Conf_Arbre ( $T, T_z$ )           /*tour ALLER*/
4. Conf_Arbre ( $T, T_z$ )           /*tour RETOUR*/
5. }
```

---

*BpBAR\_1* utilise la fonction *Conf\_Arbre* ( $T_0, T_z$ ) présentée par l'algorithme 5.3 dans chaque tour du processus de reconfiguration. *Conf\_Arbre* ( $T_0, T_z$ ) prend en entrée l'arbre initial et l'arbre final. Dans le premier tour, la fonction reconfigure la paire d'arbres en utilisant la technique *BpBAR* et produit une copie de l'arbre final utilisant différentes longueurs d'onde (appelé  $T$ ). Dans le second tour, la fonction prend en entrée l'arbre  $T$  obtenu à la fin du premier tour et l'arbre final  $T_z$ . Elle reconfigure cette paire d'arbres en utilisant le même processus, mais cette fois avec une seule longueur d'onde et la même. Elle produit à la fin, l'arbre final utilisant une seule longueur d'onde.

**Algorithme 5.3: Fonction de reconfiguration par branche**


---

```

Conf_Arbre ( $T_0, T_z$ )
1. {
2.  $T = T_0$           /* $T$  est l'arbre courant de la connexion*/
3. Pour chaque nœud destinataire  $d$  de  $T$  qui est nœud feuille
4. {
5.   Reconf_Paire_Br ( $T, T_z, d$ )
6. }
7. }

```

---

La fonction *Conf\_Arbre* ( $T_0, T_z$ ) utilise la fonction de reconfiguration de paires de branches *Reconf\_Paire\_Br* ( $T_i, T_z, d$ ) présentée dans l'algorithme 5.4. *Reconf\_Paire\_Br* ( $T_i, T_z, d$ ) prend en entrée l'arbre courant à l'étape  $i$ , l'arbre final et un nœud destinataire  $d$  représentant la destination de la paire de branches à reconfigurer. Le nœud destinataire  $d$  donné en argument de la fonction est de préférence un nœud destinataire qui est nœud feuille de l'arbre initial. En effet, si une destination  $d$  n'est pas nœud feuille de l'arbre initial, alors les canaux optiques avals de  $d$  alimentent au moins une autre destination (différente de  $d$ ). Donc le nœud de branchement de  $d$  sera tel que sa hauteur dans l'arbre initial est plus grande que celle de  $d$ . cela a pour conséquence direct l'augmentation des ressources additionnelles utilisées entre la source et le nœud de basculement.

**Algorithme 5.4 : Fonction de reconfiguration d'une paire de branche**


---

```

Reconf_Paire_Br ( $T_i, T_z, d$ )
1. {
2. Sw_Node ( $T_i, T_z, d$ )      /*détermination de nœud de basculement*/
3. Etablir la nouvelle branche  $Br(T_z, S, d)$  en configurant un à un les ports de sortie des nœuds entre
   la destination  $d_j$  et la source  $S$  pour commuter la longueur d'onde (par exemple  $\lambda_1$ ). L'opération
   d'ajout de commutation est utilisée pour tous les nœuds sauf la destination. L'opération de
   convergence est utilisée pour la destination.
4. Si (le nœud de basculement différent du nœud source)
5. {
6.   Etablir avec l'opération d'ajout de commutation le segment entre la source  $S$  et le nœud de
   basculement sur l'ancienne branche  $Br(T_i, S, d)$  en configurant un à un les ports de sortie
   des nœuds pour commuter la longueur d'onde (par exemple  $\lambda_1$ ).
7. }
8. Configurer le nœud de basculement pour basculer le flux de l'ancienne branche  $Br(T_i, S, d)$  vers
   la nouvelle branche  $Br(T_z, S, d)$ .
9. Configurer un à un les ports de sortie des nœuds du segment de la branche reconfigurable initiale
   entre le nœud de destination et le nœud de basculement pour le supprimer. L'opération de
   suppression de commutation est utilisée pour tous les nœuds sauf la destination. L'opération de
   non-convergence est utilisée pour la destination.
10. }

```

---

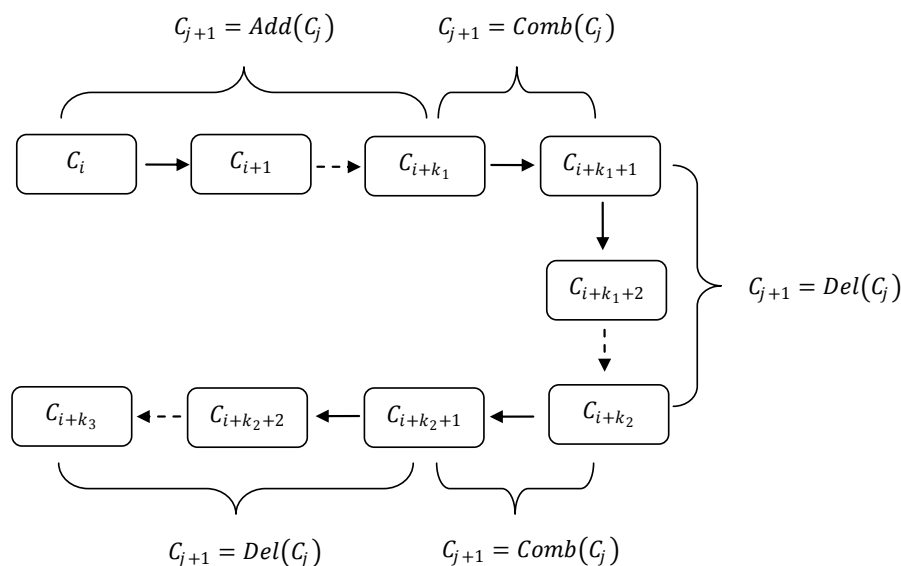

 Fig. 5.20 : Les parties d'une phase de reconfiguration avec *BpBAR\_I*

 Tableau 5.2: Les parties d'une phase de reconfiguration avec *BpBAR\_I*

| Parties | Description  |
|---------|--|
| 1       | De $C_i$ à $C_{i+k_1}$ : Dans cette partie, les opérations établissent le segment entre la destination $d$ et le nœud de basculement de la paire de branches à reconfigurer. L'opération utilisée est <i>Add</i> : $C_{j+1} = Add(C_j)$ . La longueur de cette partie est égale au nombre de ports de sortie configurés.                         |
| 2       | De $C_{i+k_1}$ à $C_{i+k_1+1}$ : Dans cette partie, les opérations alimentent le segment établi avec l'exécution de l'opération de basculement <i>Comb</i> sur le nœud de basculement de la paire de branches à reconfigurer: $C_{j+1} = Comb(C_j)$ . Cette partie est constituée d'une seule étape.   |
| 3       | De $C_{i+k_1+1}$ à $C_{i+k_2}$ : Dans cette partie, les opérations suppriment le segment entre la destination $d$ et le nœud de basculement de la paire de branches à reconfigurer. L'opération utilisée est <i>Del</i> : $C_{j+1} = Del(C_j)$ . La longueur de cette partie est égale au nombre de ports de sortie configurés.                  |
| 4       | De $C_{i+k_2}$ à $C_{i+k_2+1}$ : Dans cette partie, les opérations alimentent le segment établi avec l'exécution de l'opération de basculement <i>Comb</i> sur le nœud S et interrompt le flux sur le segment entre le nœud de basculement de la paire de branches à reconfigurer et la source S. Cette partie est constituée d'une seule étape. |
| 5       | De $C_{i+k_2+1}$ à $C_{i+k_3}$ : $C_{j+1} = Del(C_j)$ . Dans cette partie, les opérations suppriment le segment établi entre le nœud de basculement de la paire de branches à reconfigurer et le nœud source S. La longueur de cette partie est égale au nombre de ports de sortie nconfigurés.  |

La figure 5.20 résume les différentes opérations réalisées et leur ordre d'exécution dans une phase de reconfiguration d'une paire de branches avec *BpBAR\_1*. Les différentes parties constituant les opérations de reconfiguration d'une paire de branches sont détaillées dans le tableau 5.2.

L'algorithme de reconfiguration  $BpBAR_1(T_0, T_z)$ , basé sur l'approche de reconfiguration par branche assure l'existence d'un arbre couvrant les destinations et la source à chaque étape du processus. Toutefois il produit une très longue séquence d'étapes car un seul port est configuré par étape. Les configurations de ports isolés n'ont pas de conséquence sur la continuité du flux car ils ne sont pas utilisés pour la transmission du flux. Leur ordre de configuration non plus n'a pas de conséquence sur la continuité du flux. Tous les ports isolés des nœuds peuvent alors être reconfigurés dans une seule étape. Les sections suivantes proposent la reconfiguration de certains ports en parallèle afin de réduire la durée de reconfiguration.

### 5.6.5 Réduction de la durée de reconfiguration du processus *BpBAR*

Les algorithmes présentés dans cette section déterminent des opérations à exécuter en parallèle tout en garantissant la continuité du flux.

#### 5.6.5.1 Algorithme de reconfiguration *BpBAR\_2*

Le procédé de reconfiguration *BpBAR\_2* propose une nouvelle fonction de reconfiguration de paires de branches. Cette fonction reconfigure en parallèle certains ports tout en garantissant la continuité du flux. Les actions menées en parallèle sont :

- Pré-établissement de la nouvelle branche de la paire de branches à reconfigurer : les ports constituant la nouvelle branche à établir ne sont pas utilisés par le flux à cette étape. Ce sont des ports isolés. De plus, l'ordre de configuration de ces ports isolés n'a pas d'influence sur la continuité du flux car ils ne sont pas utilisés par le flux à cette étape. Ainsi les opérations de configuration de ces ports peuvent être effectuées en parallèle.
- Suppression de la partie exclusive de l'ancienne branche dont le flux est interrompu après le basculement : après l'opération de basculement, le segment de l'ancienne branche de la paire à reconfigurer entre le nœud de basculement et la destination de la branche n'est plus utilisé par le flux. Les ports de ce segment deviennent isolés. Ils peuvent être configurés en parallèle pour supprimer la branche.



## 5. Reconfiguration par branche du routage multicast

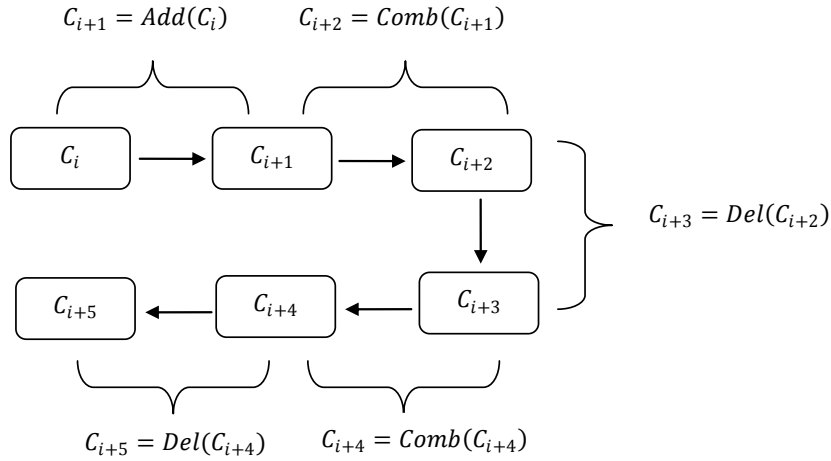


Fig. 5.21 : Etapes de reconfiguration dans une phase avec *BpBAR\_2*

La figure 5.21 résume les différentes opérations réalisées et leur ordre d'exécution dans une phase de reconfiguration d'une paire de branches avec *BpBAR\_2*. Le tableau 5.3 présente les différentes opérations exécutées en parallèle dans une phase de reconfiguration avec *BpBAR\_2*.

Tableau 5.3 : Opérations exécutées en parallèle dans une phase de reconfiguration de paire de branches avec *BpBAR\_2*

| Parties d'une phase | Séries de configurations par parties de la Fig. 5.18                          | Configurations produites après l'exécution d'opérations en parallèle |
|---------------------|---|--|
| 1                   | $\langle Add(C_i), Add(C_{i+1}), \dots, Add(C_{i+k_1-1}) \rangle$             | $\langle Add(C_i) \rangle$   |
| 2                   | $\langle Comb(C_{i+k_1}) \rangle$   | $\langle Comb(C_{i+1}) \rangle$                                      |
| 3                   | $\langle Del(C_{i+k_1+1}), Del(C_{i+k_1+2}), \dots, Del(C_{i+k_2-1}) \rangle$ | $\langle Del(C_{i+2}) \rangle$                                       |
| 4                   | $\langle Comb(C_{i+k_2}) \rangle$   | $\langle Comb(C_{i+3}) \rangle$                                      |
| 5                   | $\langle Del(C_{i+k_2+1}), Del(C_{i+k_2+2}), \dots, Del(C_{i+k_3-1}) \rangle$ | $\langle Del(C_{i+4}) \rangle$                                       |

L'algorithme 5.5 est la fonction de reconfiguration d'une paire de branches en parallèle. L'utilisation de cette fonction permet de réduire le nombre d'étapes de reconfiguration, et donc la durée de reconfiguration.

La figure 5.22 présente les étapes de reconfiguration produites par l'algorithme *BpBAR\_2* pendant la résolution de l'instance de reconfiguration de la figure 5.13.

**Algorithme 5.5 : Fonction de configuration de paire de branches utilisée par BpBAR\_2.****Exécution d'opérations en parallèle**


---

*Reconf\_Paire\_Br\_Modifié* ( $T_i, T_z, d$ )

1. {
2. *Sw\_Node* ( $T_i, T_z, d$ )      /\*détermination de nœud de basculement\*/
3. Etablir la nouvelle branche  $Br(T_z, S, d_j)$  en configurant en parallèle les ports de sortie des nœuds entre la destination  $d_j$  et la source S pour commuter la longueur d'onde (par exemple  $\lambda_1$ ). L'opération d'ajout de commutation est utilisée pour tous les nœuds sauf la destination. L'opération de convergence est utilisée pour la destination.
4. **Si** (le nœud de basculement est différent du nœud source)
5. {
6.     Etablir le segment entre la source S et le nœud de basculement sur l'ancienne branche  $Br(T_i, S, d_j)$  en configurant en parallèle les ports de sortie des nœuds pour commuter la longueur d'onde (par exemple  $\lambda_1$ ).
7. }
8. Configurer le nœud de basculement pour basculer le flux de l'ancienne branche  $Br(T_i, S, d_j)$  vers la nouvelle branche  $Br(T_z, S, d_j)$ .
9. Configurer en parallèle les ports de sortie des nœuds du segment de la branche reconfigurable initiale entre le nœud de destination et le nœud de basculement pour le supprimer. L'opération de suppression de commutation est utilisée pour tous les nœuds sauf la destination. L'opération de non-convergence est utilisée pour la destination.
10. }

---

L'algorithme *BpBAR\_2* présenté ci-dessus configure en parallèle des ports dans la phase de reconfiguration d'une paire de branches. Une seule branche est reconfigurée par une phase avec *BpBAR\_2*. L'algorithme *PRoTB* que nous présentons dans la section suivante permet de reconfigurer plusieurs branches en parallèle dans une phase de reconfiguration.

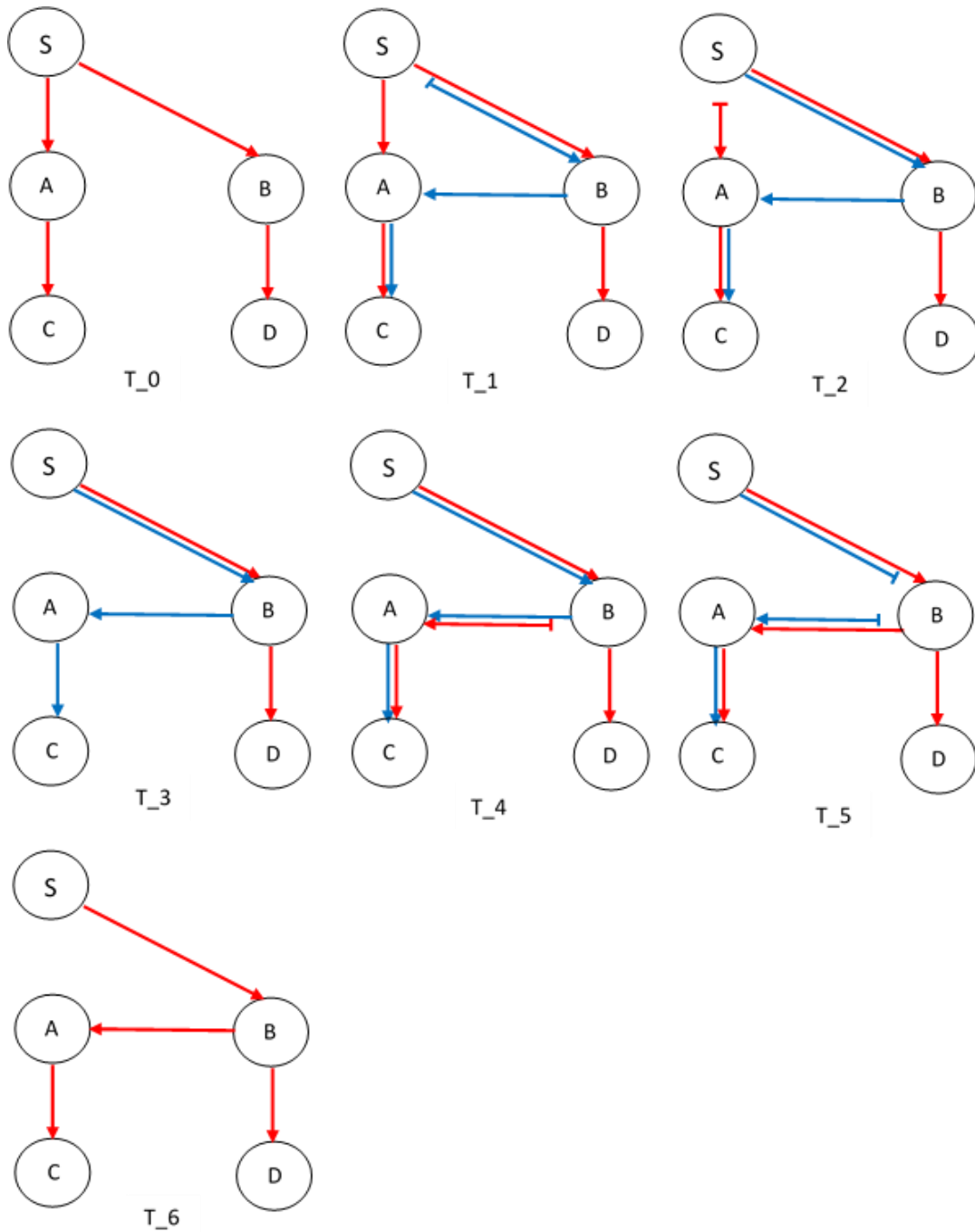


Fig. 5.22 : Séquence de reconfiguration par branche avec *BpBAR\_2*

### 5.6.5.2 Algorithme de reconfiguration *PRoTB*

*PRoTB* est basé sur l'approche de reconfiguration par branche *BpBAR*. Il réduit la durée de reconfiguration en configurant en parallèle plusieurs branches. Une phase de reconfiguration dans cette approche permet de reconfigurer un ensemble de paires de branches. Chaque étape garantit l'existence d'un arbre couvrant la source et toutes les destinations. *PRoTB* est présenté

par l'algorithme 5.6. Il se déroule en deux tours : ALLER et RETOUR. Il utilise la fonction  $Conf\_Arbre\_modif (T_0, T_z)$  pour la reconfiguration dans chaque tour.

**Algorithme 5.7: Algorithme de reconfiguration par branche *PRoTB***

---

```

PRoTB( $T_0, T_z$ )
1. {
2.  $T=T_0$ 
3. Conf_Arbre_modif ( $T, T_z$ )      /*tour ALLER*/
4. Conf_Arbre_modif ( $T, T_z$ )      /*tour RETOUR*/
5. }

```

---

**Algorithme 5.8: Fonction de reconfiguration de branches en parallèle**

---

```

Conf_Arbre_modif ( $T_0, T_z$ )
1. {
2.  $T=T_0$  /*T est l'arbre courant de la connexion*/
3. Tant qu'il existe une destination non reconfigurée
4. {
5.   Reconf_Paral_Paire_Br ( $T, T_z$ )
6. }
7. }

```

---

Nous décrivons ci-dessous la manière dont les branches à reconfigurer en parallèle sont déterminées et décrivons par la suite le processus de reconfiguration de ces branches.

- Détermination des paires de branches à reconfigurer en parallèle

Les branches à reconfigurer en parallèle doivent être choisies de sorte qu'il existe un arbre couvrant la source et les destinations après chaque étape (c.-à-d. qu'on ne provoque pas d'interruption). Cela impose que la reconfiguration de chaque ancienne branche de l'ensemble de branches à reconfigurer dans la phase garantit la continuité du flux. Nous avons vu dans les sections passées que la reconfiguration par branche avec le processus *BpBAR* se fait sans interrompre le flux.

Dans cette approche, nous déterminons dans le réseau des nœuds qui respectent la propriété de nœuds de basculement. C'est-à-dire des nœuds pouvant servir à configurer une paire de branches sans interruption du flux. Chaque nœud de basculement permettant de reconfigurer une seule paire de branches, le nombre de nœuds de basculement choisis sera égal au nombre de paires de branches à reconfigurer en parallèle. Les nœuds de basculement sont déterminés avec l'algorithme qui a déjà été décrit (Cf. algorithme 5.1).

## 5. Reconfiguration par branche du routage multicast

Considérons l'exemple du problème de reconfiguration de la figure 5.23.a. L'arbre initial est en rouge et l'arbre final est en bleu. Le tableau 5.4 représente l'ensemble des nœuds de basculement de l'arbre courant (ici l'arbre initial représenté sur la figure 5.23.b et les branches initiales correspondantes). L'ensemble des nœuds de basculement choisis sur l'arbre est  $\{A; C; E; F\}$ . Donc quatre anciennes branches peuvent être reconfigurées en parallèle à cette étape.

La figure 5.23.c est l'arbre obtenu à la fin du tour ALLER du processus. Le tableau 5.5 représente l'ensemble des nœuds de basculement de l'arbre courant de la figure 5.23.d et les anciennes branches correspondantes. L'ensemble des nœuds de basculement choisis sur l'arbre est  $\{C; E; F\}$ . Donc trois anciennes branches peuvent être reconfigurées en parallèle à cette étape.

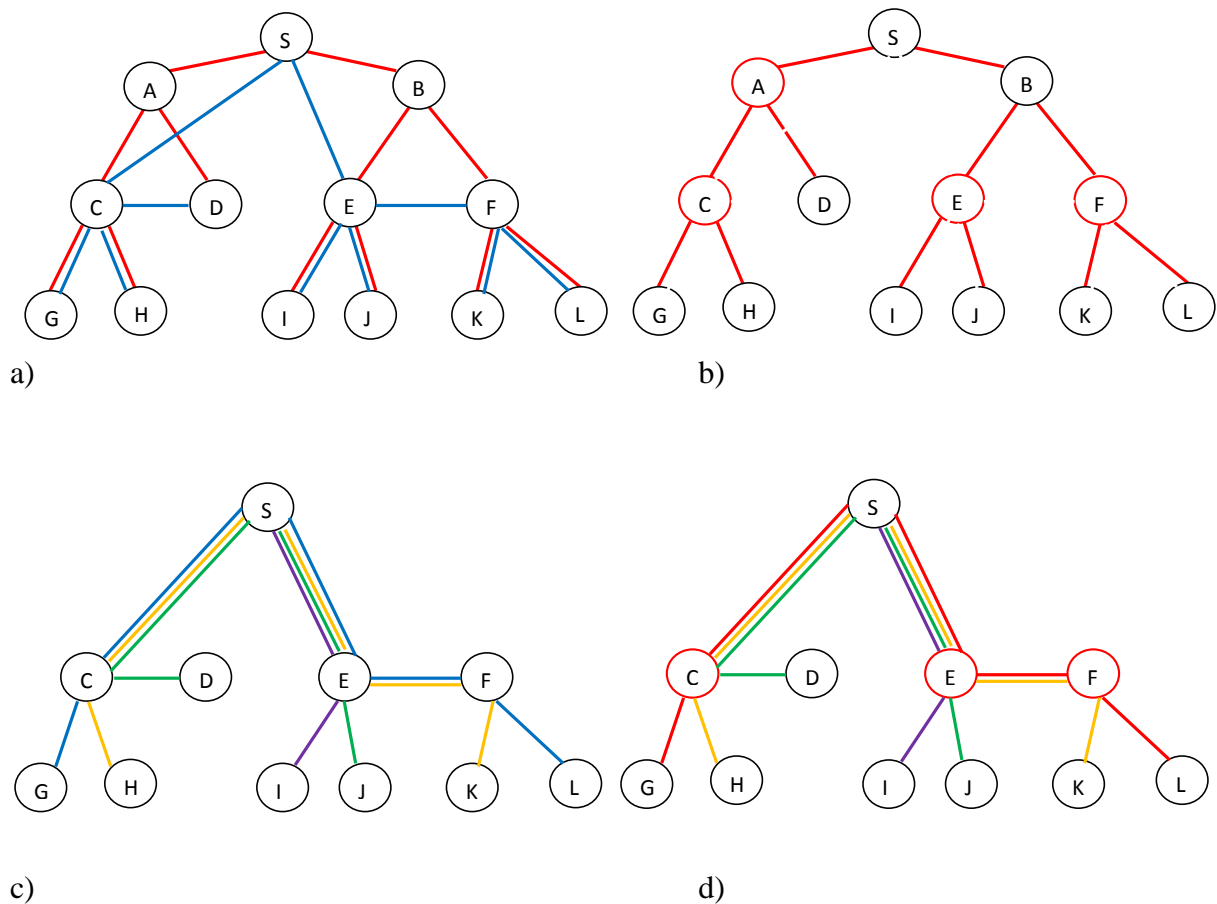


Fig. 5.23 : Exemple de problème de reconfiguration

- a) Arbre initial  $T_0$  (en rouge) et arbre final  $T_z$  (en bleu); b) nœuds de basculement (en rouge) sur l'arbre initial  $T_0$ . c) Arbre obtenu à la fin du tour ALLER; d) Arbre obtenu après la reconfiguration des branches de destination G et L dans la phase RETOUR (Ces deux branches utilisent la même longueur d'onde)

Tableau 5.4 : Correspondance entre les nœuds de basculement et les différentes branches initiales associées dans l'arbre de la figure 5.23.b

| Branche             | $Br(T_i, S, G)$ | $Br(T_i, S, H)$ | $Br(T_i, S, D)$ | $Br(T_i, S, I)$ | $Br(T_i, S, J)$ | $Br(T_i, S, K)$ | $Br(T_i, S, L)$ |
|---------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Nœud de Basculement | C               | C               | A               | E               | E               | F               | F               |

Tableau 5.5 : Correspondance entre les nœuds de basculement et les différentes branches initiales associées dans l'arbre de la figure 5.23.d

| Branche             | $Br(T_i, S, H)$ | $Br(T_i, S, D)$ | $Br(T_i, S, I)$ | $Br(T_i, S, J)$ | $Br(T_i, S, K)$ |
|---------------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Nœud de basculement | C               | C               | E               | E               | F               |

- Reconfiguration des branches en parallèle

Les branches déterminées sont configurées en parallèle avec la fonction  $Reconf\_Paral\_Paire\_Br(T_i, T_z)$  présentée par l'algorithme 5.9.

**Algorithme 5.9 : Fonction de configuration de plusieurs paires de branches en parallèle utilisée par l'algorithme *PRoTB*.**

---

$Reconf\_Paral\_Paire\_Br(T_i, T_z)$

1. {
  2. Déterminer les nœuds de basculement de  $T_i$  /\*détermination des nœuds de basculement\*/
  3. Déterminer l'ensemble des paires de branches correspondant aux nœuds de basculement
  4. Pour toutes les paires de branches, établir en parallèle les nouvelles branches  $Br(T_z, S, d_j)$  en configurant en parallèle les ports de sortie des nœuds entre les destinations  $d_j$  et la source S pour commuter la longueur d'onde (par exemple  $\lambda_1$ ). L'opération d'ajout de commutation est utilisée pour tous les nœuds sauf les destinations. L'opération de convergence est utilisée pour les destinations.
  5. **Si** (les nœuds de basculement sont différents du nœud source)
  6. {
  7. Etablir le segment entre la source S et le nœud de basculement sur l'ancienne branche  $Br(T_i, S, d_j)$  en configurant en parallèle les ports de sortie des nœuds pour commuter la longueur d'onde (par exemple  $\lambda_1$ ).
  8. }
  9. Configurer en parallèle les nœuds de basculement pour basculer le flux des anciennes branches  $Br(T_i, S, d_j)$  vers les nouvelles branches  $Br(T_z, S, d_j)$ .
  10. Configurer en parallèle les ports de sortie des nœuds des segments des anciennes branches entre les nœuds de destination et les nœuds de basculement pour les supprimer. L'opération de suppression de commutation est utilisée pour tous les nœuds sauf la destination. L'opération de non-convergence est utilisée pour la destination.
  11. }
-

Les ports des nœuds des nouvelles branches n'étant pas utilisés par le flux, ils sont configurés en parallèle avec l'opération d'ajout de commutation sur les nœuds pour les branches. Après le pré-établissement des nouvelles branches, on exécute en parallèle l'opération de basculement sur les nœuds de basculement. Ces opérations interrompent simultanément le flux sur les anciennes branches et alimentent les nouvelles branches. Enfin, configurer les ports des nœuds des anciennes branches pour supprimer les segments dont le flux est interrompu.

### 5.6.6 Evaluations

Les algorithmes proposés sont évalués à partir de simulations sur des topologies bien connues. Nous présentons les différents critères d'évaluation, les paramètres de la simulation et les résultats obtenus.

#### 5.6.6.1 Critères d'évaluation

L'objectif principal de nos travaux est de reconfigurer un arbre optique sans interrompre le flux. Mais l'algorithme proposé doit également réduire la durée du processus et le coût des ressources additionnelles utilisées. Nous définissons trois critères d'évaluation. Le premier critère est la durée d'interruption d'une séquence de reconfiguration. Le deuxième critère est la durée d'une séquence de reconfiguration et le troisième critère est le coût des ressources additionnelles utilisées par une séquence de reconfiguration. Une séquence de reconfiguration, commençant à l'arbre initial  $T_0$  et prenant fin à l'arbre final  $T_z$  sera notée  $Seq(T_0, T_z)$ .

- Durée d'interruption du flux

L'absence ou la présence d'interruption est une propriété inhérente au processus de reconfiguration : certains processus de reconfiguration produisent des interruptions. Nous mesurons donc cette durée d'interruption par processus. La durée d'interruption d'une étape  $k$  à une étape  $k + 1$  est supposée égale à la durée unitaire d'une étape de reconfiguration. La durée d'interruption mesurée est la durée totale des étapes de la séquence de reconfiguration ne contenant pas un arbre couvrant la source et tous les nœuds destinataires.

Si *duration* ( $O(T_k)$ ) désigne la durée d'une étape de reconfiguration appliquée à l'arbre  $T_k$ , elle est égale à la durée d'interruption de l'étape  $k$  à une étape  $k + 1$ .

Si *interrupt\_duration* ( $Seq(T_0, T_z)$ ) désigne la durée d'interruption d'une séquence de reconfiguration  $Seq(T_0, T_z)$ , elle sera déterminée par :

## 5. Reconfiguration par branche du routage multicast

$$interrupt\_duration (Seq(T_0, T_z)) = \sum_{k=0}^{|Seq(T_0, T_z)|-1} b_k \times duration (O(T_k))$$

$$\text{avec } b_k = \begin{cases} 1 & \text{s'il n'existe pas d'arbre couvrant la source} \\ & \text{et les destinations à l'étape } k. \\ 0 & \text{s'il existe un arbre couvrant la source} \\ & \text{et les destinations à l'étape } k. \end{cases}$$

Par définition, toutes les opérations d'une étape de reconfiguration sont exécutées en parallèle. Si on suppose que la durée de chaque opération de reconfiguration est connue, on a :

$$Min\_step\_duration (Seq(T_0, T_z)) = \min\{duration (O(T_k)) / T_k \in Seq(T_0, T_z)\}$$

où  $Min\_step\_duration (Seq(T_0, T_z))$  représente la durée minimum d'une étape de reconfiguration.

Si au moins une interruption est produite durant le processus de reconfiguration, la durée d'interruption du processus sera au moins égale à  $Min\_step\_duration (Seq(T_0, T_z))$ .

### - Durée d'une séquence de reconfiguration

La durée d'une séquence de reconfiguration est la durée totale de toutes les étapes de reconfiguration constituant la séquence. Si on désigne par  $duration (Seq(T_0, T_z))$  la durée d'une séquence de reconfiguration, elle est déterminée par :

$$duration (Seq(T_0, T_z)) = \sum_{k=0}^{|Seq(T_0, T_z)|-1} duration (O(T_k))$$

Si on suppose que la durée maximum de chaque opération de reconfiguration est connue, on a :

$$Maj\_step\_duration = \max\{duration (O(T_k)) / T_k \in Seq(T_0, T_z)\}$$

où  $Maj\_step\_duration$  représente la durée maximum d'une étape de reconfiguration.

La durée maximum d'une séquence de reconfiguration sera alors majorée comme suit :

$$duration (Seq(T_0, T_z)) \leq |Seq(T_0, T_z)| \times Maj\_step\_duration$$



- Coût des ressources additionnelles

Le coût des ressources additionnelles est la somme totale des ressources additionnelles utilisées sur chaque lien du réseau et à chaque étape du processus. Si  $add\_cost(Seq(T_0, T_z))$  désigne le coût des ressources additionnelles de la séquence de reconfiguration  $Seq(T_0, T_z)$  et  $add\_cost(O(T_k))$  désigne le coût des ressources additionnelles d'une étape  $k$  à une étape  $k + 1$ , nous avons :

$$add\_cost(Seq(T_0, T_z)) = \sum_{k=0}^{|Seq(T_0, T_z)|-1} add\_cost(O(T_k))$$

### 5.6.6.2 Description de la simulation

Les topologies de réseau utilisées pour évaluer les algorithmes sont : COST 239 (Fig. 5.9), NSF Network (Fig. 5.10) et US Backbone Network (Fig. 5.11). Ces topologies présentent différentes caractéristiques. COST 239 est formé de 11 nœuds et 26 liens, NSF Network est formé de 14 nœuds et 21 liens et US Backbone Network est formé de 24 nœuds et 43 liens (Tableau 5.1).

Pour une topologie donnée, nous sélectionnons de façon aléatoire deux arbres comme arbre initial et arbre final du problème de reconfiguration. L'arbre des plus courts chemins entre la source et chaque destination (SPT) et l'arbre minimum de Steiner (arbre partiel couvrant la source et les destinations de poids minimum, MST) ont été choisis pour représenter les deux arbres. La source de la connexion est choisie de façon aléatoire (loi uniforme) parmi les nœuds du réseau considéré. Le nombre  $k$  de destinations du groupe multicast est déterminé de façon aléatoire (loi uniforme) parmi les nœuds restants du réseau, avec  $k \in [1; n - 1]$  où  $n$  est le nombre de nœuds du réseau. L'algorithme de Dijkstra [Dij57] est appliqué entre le nœud source et chaque destination pour calculer l'arbre des plus courts chemins. La détermination de l'arbre minimum de Steiner est connue comme un problème difficile à résoudre. Nous utilisons l'algorithme de Prim [Pri57] pour déterminer une approximation de l'arbre minimum de Steiner couvrant la source et les nœuds destinataires.

Dans la première partie des évaluations, l'arbre initial est l'arbre SPT et l'arbre final est l'approximation de l'arbre MST. Dans la seconde partie des simulations, nous échangeons les algorithmes permettant de calculer les arbres. L'arbre initial devient l'approximation de l'arbre MST et l'arbre final devient l'arbre SPT.

Nous supposons que les nœuds ont une capacité de *splitting* (duplication optique) appropriée. Cela signifie que la capacité de *splitting* de chaque nœud est supérieure ou égale au nombre maximum de nœuds voisins à un nœud du réseau moins un et que le nombre de *splitters* dans le nœud est supérieur au nombre de connexions multicast utilisant ce nœud comme nœud de branchement. Cela permet de construire l'arbre initial et l'arbre final de notre problème de reconfiguration sans contrainte de nombre de nœuds voisins à chaque nœud et du nombre de sessions multicast établies dans le réseau.

Le nombre de reconfigurations peut être élevé durant le processus de reconfiguration, même pour les petites topologies. Alors nous utilisons la méthode de Monte-Carlo pour évaluer les performances des différents critères retenus (durée d'interruption, coût de ressources additionnelles et durée d'interruption dans chaque topologie). Dans le paragraphe suivant, nous définissons les différents paramètres utilisés.

Le couple constitué de l'arbre initial et l'arbre final est déterminé avec les algorithmes de Dijkstra et de Prim. Pour chaque paire d'arbres, on vérifie si les deux arbres sont différents ; sinon on détermine une nouvelle paire d'arbres. Pour chaque détermination d'arbres, différentes tailles du groupe multicast sont utilisées. La taille du groupe multicast est déterminée de façon aléatoire. Le nombre de paires d'arbre initial et d'arbre final représentant les problèmes de reconfiguration par simulation est fixé à 5000. Ce nombre a été fixé de façon expérimentale. Nous avons calculé la moyenne pour un nombre varié d'exécutions d'une même simulation. L'erreur relative entre deux moyennes issues d'une même simulation mais calculée sur un nombre d'exécutions différents est inférieure à 3% pour tous les critères lorsque le nombre d'exécutions de base est 5000.

### 5.6.6.3 Résultats et analyses des résultats

Les tableaux 5.6 à 5.10 présentent les résultats des évaluations. Les notations utilisées sont :

- CONF. PROCESS: Algorithme de reconfiguration
- AVG: Moyenne
- SD: Ecart type
- WL: Longueur d'onde
- STEP : étape

Tableau 5.6 : Durée d'interruption

| CONF. PROCESS  | COST 239      |              |                   | NSF NETWORK   |              |                   |
|----------------|---------------|--------------|-------------------|---------------|--------------|-------------------|
|                | AVG<br>(STEP) | SD<br>(STEP) | MIN/MAX<br>(STEP) | AVG<br>(STEP) | SD<br>(STEP) | MIN/MAX<br>(STEP) |
| <i>BpBAR_2</i> | 0             | 0            | 0/0               | 0             | 0            | 0/0               |
| <i>PRoTB</i>   | 0             | 0            | 0/0               | 0             | 0            | 0/0               |
| <i>MBB_1</i>   | 9,63          | 5,60         | 0/18              | 13,18         | 7,16         | 0/24              |

Tableau 5.7 : Durée de reconfiguration

| CONF. PROCESS  | COST 239      |              |                   | NSF NETWORK   |              |                   |
|----------------|---------------|--------------|-------------------|---------------|--------------|-------------------|
|                | AVG<br>(STEP) | SD<br>(STEP) | MIN/MAX<br>(STEP) | AVG<br>(STEP) | SD<br>(STEP) | MIN/MAX<br>(STEP) |
| <i>BpBAR_2</i> | 18,07         | 12,09        | 6/51              | 22,62         | 17,35        | 6/74              |
| <i>PRoTB</i>   | 15,06         | 7,66         | 6/37              | 15,71         | 7,49         | 6/38              |
| <i>MBB_1</i>   | 12,79         | 5,42         | 3/21              | 16,25         | 7,06         | 3/27              |

Tableau 5.8 : Coût des ressources additionnelles

| CONF. PROCESS  | COST 239    |            |                 | NSF NETWORK |            |                 |
|----------------|-------------|------------|-----------------|-------------|------------|-----------------|
|                | AVG<br>(WL) | SD<br>(WL) | MIN/MAX<br>(WL) | AVG<br>(WL) | SD<br>(WL) | MIN/MAX<br>(WL) |
| <i>BpBAR_2</i> | 166,84      | 195,57     | 10/861          | 370,14      | 448,1      | 9/2273          |
| <i>PRoTB</i>   | 164,80      | 180,14     | 10/834          | 312,1       | 329,5      | 9/1431          |
| <i>MBB_1</i>   | 0           | 0          | 0/0             | 0           | 0          | 0/0             |

Le tableau 5.7.a montre que le nombre d'étapes produit par l'algorithme *PRoTB* est supérieur à celui produit par l'algorithme *MBB\_1*. Cela est dû au fait que *PRoTB* s'effectue en deux tours, chaque tour étant composé de plusieurs étapes. Le tableau 5.8.a montre que la reconfiguration avec l'algorithme *PRoTB* utilise plus de ressources additionnelles que l'algorithme *MBB\_1*. En effet, pour établir une nouvelle branche, l'algorithme *PRoTB* utilise de nouvelles longueurs d'onde (différentes de celles déjà utilisées) sur les liens déjà utilisés par l'arbre courant (arbre utilisé pour la transmission du flux). Ces ressources additionnelles sont utilisées durant toute une phase de reconfiguration pour reconfigurer une branche ou un ensemble de branches. Le tableau 5.6 présente la durée d'interruption de chaque algorithme. La durée d'interruption produite par *PRoTB* et *BpBAR\_2* est nulle. Ces deux algorithmes reconfigurent donc l'arbre sans interruption de flux comme le montre la figure 5.24. Cela est dû au fait que les algorithmes *PRoTB* et *BpBAR\_2* sont basés sur la technique de reconfiguration par branche. Cette technique détermine judicieusement des nœuds de basculement permettant de reconfigurer une paire de branches sans interrompre le flux.

## 5. Reconfiguration par branche du routage multicast

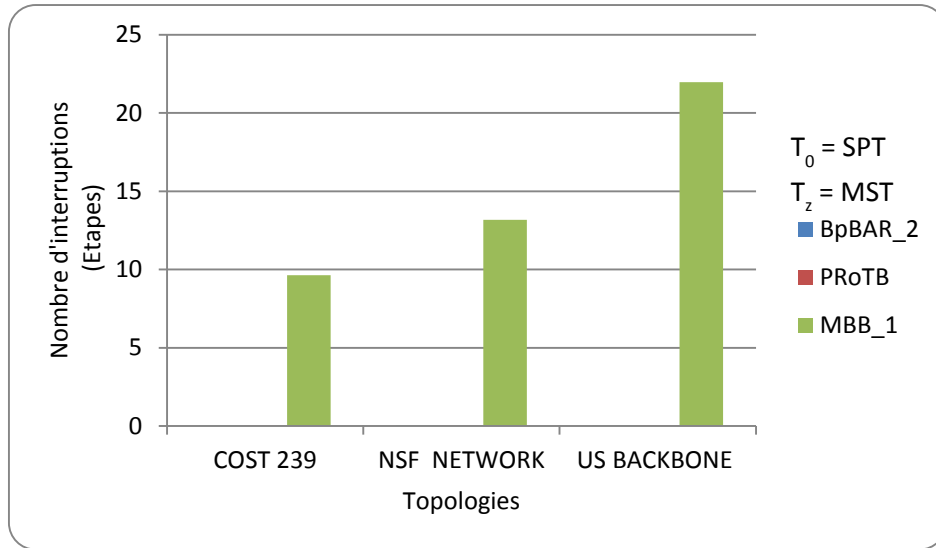


Fig. 5.24 : Durée d'interruption

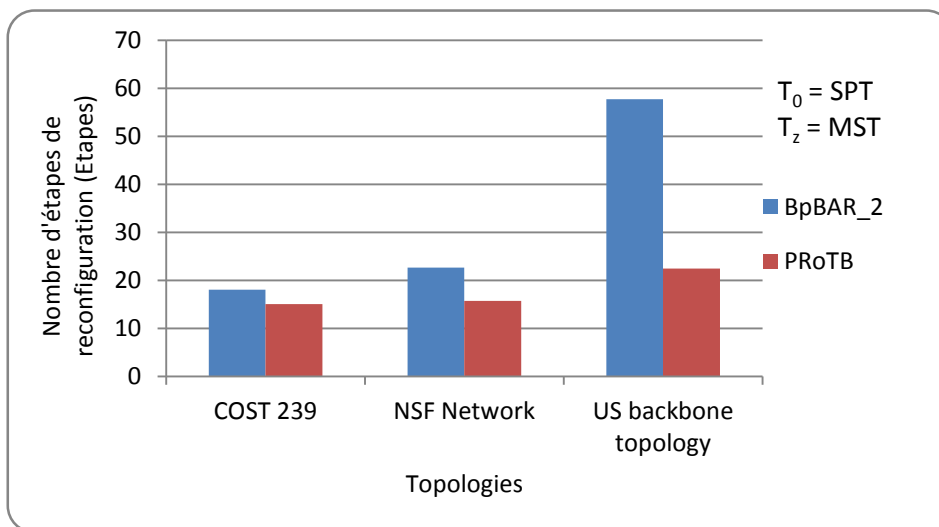


Fig. 5.25 : Durée de reconfiguration

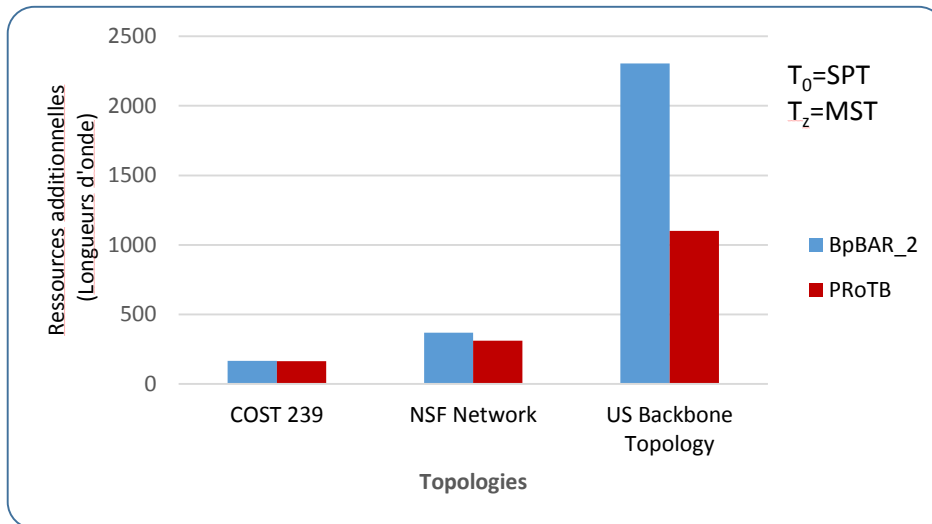


Fig. 5.26 : Coût de ressources additionnelles

L'objectif principal de ces travaux est de reconfigurer l'arbre optique sans interrompre le flux. La durée d'interruption des algorithmes *BpBAR\_2* et *PRoTB* étant nulle, ces deux algorithmes sont plus efficaces que *MBB\_1*. *BpBAR\_2* et *PRoTB* garantissant la continuité du flux vers toutes les destinations (sans interruption de flux), nous allons comparer la durée de reconfiguration et le coût des ressources additionnelles de ces deux algorithmes. La durée de reconfiguration produite par l'algorithme *PRoTB* dans la topologie COST 239 est inférieure à celle produite par *BpBAR\_2* (Fig. 5.25). Le gain est entre 16% et 19,5%. Dans la topologie NSF Network, le gain de durée de reconfiguration est plus important (entre 30 et 33,5%). La réduction de la durée de reconfiguration par *PRoTB* s'explique par la reconfiguration en parallèle de plusieurs branches. Les évaluations montrent que le gain devient important lorsque la taille du réseau augmente. En effet, lorsque la taille du réseau augmente, la taille de l'arbre initial et l'arbre final augmente. Certaines paires d'arbres peuvent avoir plusieurs branches à reconfigurer lorsque le réseau est de grande taille. L'algorithme *PRoTB* reconfigurant plusieurs branches en parallèle, permet de réduire la durée de reconfiguration. Cette assertion est vérifiée par les résultats des simulations effectuées dans le réseau US Backbone (Tableau 5.9). La topologie du réseau US Backbone est de taille plus grande que les deux précédents réseaux. La réduction de la durée de reconfiguration dans ce réseau est de 61%. *PRoTB* réduisant la durée de reconfiguration dans les trois topologies (qui sont de caractéristiques différents), on peut induire que l'algorithme *PRoTB* réduit la durée de reconfiguration de façon probablement générale (quel que soit la topologie des réseaux).

## 5. Reconfiguration par branche du routage multicast

La figure 5.26 montre que l'algorithme *PRoTB* utilise moins de ressources additionnelles que *BpBAR\_2*. Le gain en coût de ressources additionnelles est entre 1% et 49%. Ce gain est plus important dans la topologie US Backbone, soit 49% (Tableau 5.10). *PRoTB* reconfigure plus rapidement l'arbre et utilise moins de ressources additionnelles.

Les résultats obtenus avec l'algorithme de reconfiguration ne dépendent pas de la détermination de l'arbre initial et de l'arbre final. Les figures 5.27 et 5.28 montrent les résultats des évaluations lorsqu'on permute les algorithmes ayant servi à calculer l'arbre initial et l'arbre final dans les simulations ci-dessus.

Tableau 5.9 : Durée de reconfiguration

| CONF. PROCESS  | US BACKBONE TOPOLOGY |               |                    |
|----------------|----------------------|---------------|--------------------|
|                | AVG<br>(STEPS)       | SD<br>(STEPS) | MIN/MAX<br>(STEPS) |
| <i>BpBAR_2</i> | 57,69                | 33,79         | 6/139              |
| <i>PRoTB</i>   | 22,42                | 08,18         | 6/57               |

Tableau 5.10 : Coût de ressources additionnelles

| CONF. PROCESS  | US BACKBONE TOPOLOGY |            |                 |
|----------------|----------------------|------------|-----------------|
|                | AVG<br>(WL)          | SD<br>(WL) | MIN/MAX<br>(WL) |
| <i>BpBAR_2</i> | 2260,5               | 2431       | 10/13047        |
| <i>PRoTB</i>   | 1147,83              | 1109       | 10/8544         |

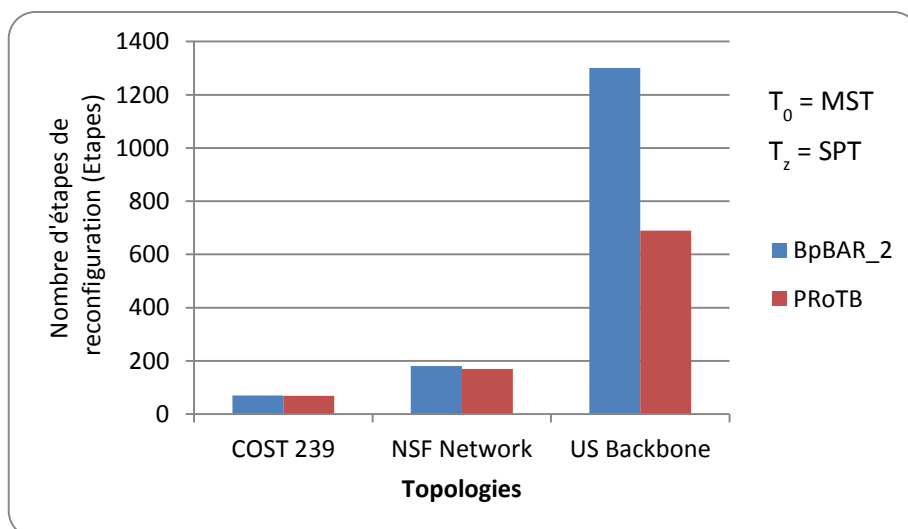


Fig. 5.27 : Durée de reconfiguration

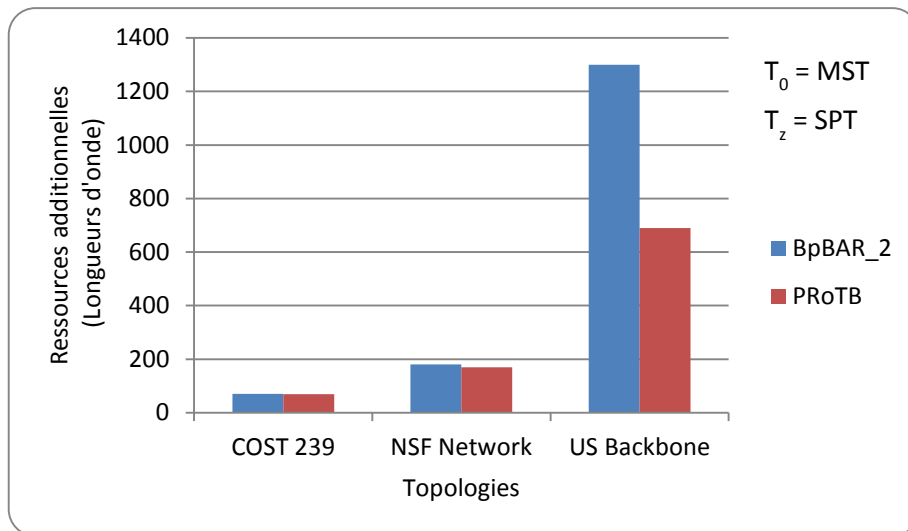


Fig. 5.28 : Coût de ressources additionnelles

Maintenant, l'arbre initial est l'arbre minimum de Steiner et l'arbre final est l'arbre des plus courts chemins entre la source et chaque destination. Les autres paramètres restent inchangés. Les résultats montrent que *PProTB* reconfigure plus rapidement l'arbre et utilise moins de ressources additionnelles.

*PProTB* a de meilleures performances comparé à l'algorithme *MBB\_1* et *BpBAR\_2*. *PProTB* a de meilleures performances par rapport à *MBB\_1* car il reconfigure sans interruption de flux. En effet *MBB\_1* exhibe certaines interruptions durant la reconfiguration. *PProTB* a de meilleures performances comparé à l'algorithme *BpBAR\_2* car il reconfigure plus rapidement et utilise moins de ressources additionnelles. *PProTB* produit de très bonnes performances lorsque l'arbre initial a une structure d'arbre binaire. Lorsque l'arbre courant a une structure d'arbre binaire, les nœuds de branchement qui ont la plus grande hauteur sur chaque branche ont la propriété de basculement. Le nombre de nœuds de basculement augmente dans ce cas en fonction du nombre de destinataires du groupe multicast. Le nombre de branches pouvant être reconfigurées en parallèle étant égal au nombre de nœuds de branchement, plusieurs branches sont configurées en parallèle dans le cas où l'arbre initial a une structure d'arbre binaire.

Dans le pire des cas, *PProTB* a les mêmes performances que *BpBAR\_2* lorsque l'arbre initial a un seul nœud de branchement. Dans ce cas, un seul nœud va avoir la propriété de basculement, donc une seule branche ne peut être reconfigurée en parallèle.

*PProTB* garantit certes la continuité du flux mais présente quelques désavantages par rapport à *MBB\_1*. Le tableau 5.7 montre que le nombre d'étapes produit par l'algorithme *PProTB* est

supérieur à celui produit par l'algorithme *MBB\_1*. Cela est dû au fait que *PRoTB* s'effectue en deux tours, chaque tour étant composé de plusieurs étapes. Le tableau 5.8 montre que la reconfiguration avec l'algorithme *PRoTB* utilise plus de ressources additionnelles que l'algorithme *MBB\_1*. En effet pour établir une nouvelle branche, l'algorithme *PRoTB* utilise de nouvelles longueurs d'onde (différentes de celles déjà utilisées) sur les liens déjà utilisés par l'arbre courant (arbre utilisé pour la transmission du flux). Ces ressources additionnelles sont utilisées durant toute une phase de reconfiguration pour reconfigurer une branche ou un ensemble de branches.

En conclusion, *PRoTB* reconfigure l'arbre sans interruption de flux. De plus, il produit de bonnes performances en termes de durée de reconfiguration et de coût de ressources additionnelles et de continuité du flux quel que soit l'algorithme utilisé pour le calcul des arbres initial et final. Nous induisons que ces performances sont probablement vraies quel que soit l'arbre initial ou final.

### 5.6.7 Conclusion

Cette section s'est intéressée à la reconfiguration sans interruption de flux d'un arbre optique. La technique de reconfiguration proposée pour la reconfiguration de connexion unicast, *MBB*, n'est pas prévue pour la reconfiguration de connexion multicast. Une extension de cette technique pour la reconfiguration multicast appelée *MBB\_1* interrompt le flux dans certaines étapes lors de la reconfiguration. La technique de reconfiguration proposée dans cette section est la reconfiguration par branche appelé *BpBAR*. Cette technique de reconfiguration consiste à déterminer dans l'arbre courant, une paire d'arbres constituée d'une branche de l'arbre courant et d'une branche de l'arbre final dont la reconfiguration se fait sans interruption de flux. Pour chaque paire d'arbres, l'algorithme détermine un nœud respectant la propriété dite de basculement permettant de basculer le flux de l'ancienne branche à la nouvelle branche sans interrompre le flux. Une première approche de reconfiguration par branche appelé *BpBAR\_2* a consisté à reconfigurer une seule paire de branches en parallèle. Dans une seconde approche, plusieurs paires de branches sont déterminées et reconfigurées en parallèle avec l'algorithme *PRoTB*. Cet algorithme permet de reconfigurer sans interruption de flux un arbre et réduit de façon considérable la durée de reconfiguration. Il réduit également le coût de ressources additionnelles. Les gains réalisés avec cette approche varient de 16% à 61% pour la durée de reconfiguration et de 1% à 49% pour le coût de ressources additionnelles.



## 5.7 Reconfiguration dans un réseau avec des ressources limitées

### 5.7.1 Introduction

Nous supposons dans cette section que les liens du réseau ont un nombre limité de ressources disponibles. L'arbre initial de la connexion multicast est  $T_0$  et utilise la longueur d'onde initiale  $\lambda_0$ . L'arbre final sur lequel on veut basculer le flux de la connexion est  $T_z$ . Nous proposons un algorithme de reconfiguration par branche permettant d'établir une connexion multicast dans un environnement de ressources limitées tout en réduisant la durée du processus de reconfiguration et le coût des ressources.

### 5.7.2 Algorithme de reconfiguration *TRwRC*

*TRwRC* est un algorithme par branche qui s'effectue en deux tours : ALLER et RETOUR. *TRwRC* produit une série de forêts durant la reconfiguration. Dans le processus de reconfiguration d'un arbre, l'ensemble des destinations du groupe multicast peut être subdivisé en deux sous-groupes :

- L'ensemble des destinations dont les branches ont été reconfigurées. Cet ensemble est noté *Dest\_Config*.
- L'ensemble des destinations dont les branches n'ont pas été reconfigurées. Cet ensemble est *Dest\_NConfig*.

La nouvelle approche de reconfiguration limite le nombre d'arbres (constituant une forêt) nécessaires pour couvrir la source et toutes les destinations du groupe à une étape de reconfiguration. Cette approche utilise deux arbres au plus pour couvrir la source et les destinations du groupe multicast: un arbre couvrant la source et l'ensemble de destinations *Dest\_Config* et un arbre couvrant la source et l'ensemble de destinations *Dest\_NConfig*. On remarque que l'union des ensembles *Dest\_Config* et *Dest\_NConfig* constitue l'ensemble des nœuds destinataires de la connexion.

L'algorithme 5.10 représente *TRwRC*. *TRwRC* utilise la fonction *ReconfBr* pour reconfigurer une paire de branches (une ancienne branche et une nouvelle branche) dans le premier tour. Dans ce tour, la reconfiguration d'une paire de branches se fait en deux phases. Dans la première phase, la nouvelle branche est établie avec une longueur d'onde additionnelle sur chaque lien (par exemple avec la longueur d'onde  $\lambda_2$ ). Dans la seconde phase, la branche est établie avec une longueur d'onde additionnelle  $\lambda_1$ . La longueur d'onde  $\lambda_2$  est ensuite libérée

sur les liens de la branche. L'arbre obtenu à la fin de la reconfiguration de toutes les branches avec ce procédé est une copie de l'arbre final. Cette copie de l'arbre final appelé  $T'$  utilise la même longueur d'onde sur chaque lien. Dans le second tour,  $TRwRC$  configure simultanément les ports des nœuds de l'arbre  $T'$  obtenu pour convertir les longueurs d'onde utilisées en la longueur d'onde initiale  $\lambda_0$ . La fonction  $ReconfBr$  est présentée par l'algorithme 5.11.

**Algorithme 5.10 : Algorithme de reconfiguration par branche  $TRwRC$**

---

$TRwRC(T_0, T_z)$

1. {
2. Pour chaque destination  $d$
3. {
4.  $ReconfBr(T_0, T_z, d, T)$
5.  $ReconfBr(T, T_z, d, T')$
6. }
7. Configurer en parallèle les ports des nœuds de l'arbre  $T'$  pour convertir les longueurs d'onde en la longueur d'onde initiale
8. }

---

**Algorithme 5.11 : Fonction de configuration de paire de branches utilisée par  $TRwRC$**

---

$ReconfBr(T_0, T_z, d, T')$

1. {
2. **Si** les branches  $Br(T_0, S, d)$  et  $Br(T_z, S, d)$  sont différentes
3. {
4.  $Sw\_Node(T_i, T_z, d)$  /\*détermination de nœud de basculement\*/
5. Préétablir en parallèle la nouvelle branche  $Br(T_z, S, d)$  en configurant en parallèle les nœuds entre la destination  $d$  et la source  $S$  pour commuter la longueur d'onde. L'opération d'ajout de commutation est utilisée pour tous les nœuds sauf la destination. L'opération de convergence est utilisée pour la destination.
6. **Si** (le nœud de basculement est différent du nœud source)
7. {
8. Préétablir le segment entre la source  $S$  et le nœud de basculement sur l'ancienne branche  $Br(T_0, S, d_j)$  en configurant en parallèle les nœuds pour commuter la longueur d'onde.
9. }
10. Configurer le nœud de basculement pour basculer le flux de l'ancienne branche  $Br(T_0, S, d)$  vers la nouvelle branche  $Br(T_z, S, d)$ .
11. Configurer en parallèle les nœuds de l'ancienne branche entre le nœud de destination et le nœud de basculement pour supprimer ce segment dont le flux est interrompu. L'opération de suppression de commutation est utilisée pour tous les nœuds sauf la destination. L'opération de non-convergence est utilisée pour la destination.
12. }
13. }

---

### 5.7.3 Evaluations

Nous présentons les différents critères d'évaluation des algorithmes, les paramètres de la simulation et les résultats obtenus.

#### 5.7.3.1 Critères d'évaluation

En plus de la durée d'interruption, de la durée de reconfiguration et du coût de ressources additionnelles, un quatrième critère d'évaluation est défini. Le quatrième critère est le nombre d'instances non résolues. Une instance non résolue dans la simulation se définit comme étant constituée d'un arbre initial et un arbre final ne pouvant être reconfigurés sans interruption par un algorithme donné. Nous supposons dans la simulation que les liens du réseau ont un nombre de longueurs d'onde fixé. Alors un problème de reconfiguration ne pourra pas être résolu par un algorithme donné si les ressources optiques dont a besoin l'algorithme ne sont pas disponibles sur les liens. Le nombre d'instances non résolues dépend alors du nombre de ressources disponibles sur les liens. Il sera noté  $nb\_unsolv\_pb(w)$  où  $w$  représente le nombre de longueurs d'onde disponibles par lien. Si le nombre de paires d'arbres générés est noté  $nb\_tirage$  :

$$nb\_unsolv\_pb(w) = \sum_{k=1}^{nb\_tirage} \alpha_k^w(T_0, T_z)$$

avec :

$$\alpha_k^w(T_0, T_z) = \begin{cases} 1 & \text{si } w \text{ longueurs d'onde par lien ne permet pas} \\ & \text{de reconfigurer le } k^{i\grave{e}me} \text{ problème sans} \\ & \text{interruption de flux vers toutes les destinations.} \\ 0 & \text{sinon.} \end{cases}$$

Le nombre de paires d'arbres générés dans les simulations est fixé de façon expérimentale à 5000. Les topologies utilisées sont COST 239 et NSF Network.

#### 5.7.3.2 Résultats et analyses

Les résultats des évaluations sont présentés dans les tableaux 5.11 à 5.15. Les tableaux 5.11 à 5.13 présentent les résultats lorsque l'arbre initial est l'arbre des plus courts chemins de la source à chaque destination et l'arbre final est une approximation de l'arbre minimum de Steiner.

## 5. Reconfiguration par branche du routage multicast

Les résultats des simulations montrent que l'algorithme *MBB\_1* est plus rapide et utilise moins de ressources que l'algorithme *TRwRC*. *MBB\_1* reconfigure sans utiliser de longueur d'onde additionnelle tandis que *TRwRC* utilise des ressources additionnelles. Mais cet algorithme produit des interruptions durant la reconfiguration de certains arbres comme le montre le tableau 5.13. *MBB\_1* ne répond donc pas à l'objectif qui est de reconfigurer tout arbre sans interruption de flux.

Nous allons donc comparer *TRwRC* avec les algorithmes *BpBAR\_2* et *PRoTB* qui sont des algorithmes de reconfiguration par branche présentés dans les sections précédentes, qui reconfigurent sans interruption de flux tout arbre. La garantie de la continuité du flux par *TRwRC* s'explique par le fait que c'est un algorithme de reconfiguration par branche qui utilise la propriété de basculement de la famille des algorithmes *BpBAR*.

Tableau 5.11 : Durée de reconfiguration

| CONF. PROCESS | COST 239   |           |                | NSF NETWORK |           |                |
|---------------|------------|-----------|----------------|-------------|-----------|----------------|
|               | AVG (STEP) | SD (STEP) | MIN/MAX (STEP) | AVG (STEP)  | SD (STEP) | MIN/MAX (STEP) |
| <i>BpBAR2</i> | 18,07      | 12,09     | 6/51           | 22,62       | 17,35     | 6/74           |
| <i>TRwRC</i>  | 18,31      | 11,61     | 7/54           | 22,68       | 16,15     | 7/79           |
| <i>MBB1</i>   | 12,79      | 5,42      | 3/21           | 16,25       | 7,06      | 3/27           |
| <i>PRoTB</i>  | 15,06      | 7,66      | 6/37           | 15,71       | 7,49      | 6/38           |

Tableau 5.12 : Coût de ressources additionnelles

| CONF. PROCESS | COST 239 |         |              | NSF NETWORK |         |              |
|---------------|----------|---------|--------------|-------------|---------|--------------|
|               | AVG (WL) | SD (WL) | MIN/MAX (WL) | AVG (WL)    | SD (WL) | MIN/MAX (WL) |
| <i>BpBAR2</i> | 166,84   | 195,57  | 10/861       | 370,1       | 448,1   | 9/2273       |
| <i>TRwRC</i>  | 107,52   | 96,45   | 13/430       | 227,8       | 220,4   | 13/1104      |
| <i>MBB1</i>   | 0        | 0       | 0/0          | 0           | 0       | 0/0          |
| <i>PRoTB</i>  | 164,80   | 180,14  | 10/834       | 312,1       | 329,5   | 9/1431       |

Tableau 5.13 : Durée d'interruption

| CONF. PROCESS | COST 239   |           |                | NSF NETWORK |           |                |
|---------------|------------|-----------|----------------|-------------|-----------|----------------|
|               | AVG (STEP) | SD (STEP) | MIN/MAX (STEP) | AVG (STEP)  | SD (STEP) | MIN/MAX (STEP) |
| <i>BpBAR2</i> | 0          | 0         | 0/0            | 0           | 0         | 0/0            |
| <i>TRwRC</i>  | 0          | 0         | 0/0            | 0           | 0         | 0/0            |
| <i>MBB1</i>   | 9,63       | 5,60      | 0/18           | 13,18       | 7,16      | 0/24           |
| <i>PRoTB</i>  | 0          | 0         | 0/0            | 0           | 0         | 0/0            |

## 5. Reconfiguration par branche du routage multicast

La figure 5.29 montre que *PRoTB* est plus rapide que *TRwRC* et la durée du processus de reconfiguration produite par *TRwRC* est approximativement égale à celle produite par *BpBAR\_2*. Cela s'explique par le fait que *PRoTB* configure plusieurs branches en parallèle alors que *TRwRC* et *BpBAR\_2* configurent une seule branche en parallèle.

Les performances importantes de l'algorithme sont présentées par les figures 5.30 à 5.32. La figure 5.30 exhibe le gain en coût de ressources additionnelles de l'algorithme *TRwRC*. *TRwRC* utilise moins de ressources additionnelles que les algorithmes *BpBAR\_2* et *PRoTB*. Cela est dû au fait que *TRwRC* utilise au plus deux arbres pour couvrir la source et les destinations du groupe multicast durant le processus de reconfiguration, soit au plus deux longueurs d'onde par lien. Or le nombre de longueurs d'onde additionnelles utilisées par *BpBAR\_2* et *TRwRC* sur un lien augmentent avec le nombre de branches partageant ce lien. Les figures 5.31 et 5.32 montrent l'évolution du pourcentage de problèmes non résolus par chaque algorithme dans les topologies COST 239 et NSF Network en fonction du nombre de longueurs d'onde disponibles par lien. Ces résultats révèlent que lorsque le nombre de longueurs d'onde disponibles par lien est inférieur ou égal à deux, la reconfiguration par branche ne peut pas résoudre tout problème sans interruption de flux. Toutefois, le pourcentage de problèmes non résolus par *TRwRC* est inférieur à celui des autres algorithmes. Ces mêmes figures montrent que lorsque le nombre de longueurs d'onde disponibles par lien est 3, *TRwRC* reconfigure tout arbre sans interruption de flux, quel que soit la topologie tandis que *BpBAR\_2* et *PRoTB* requièrent plus de ressources disponibles.

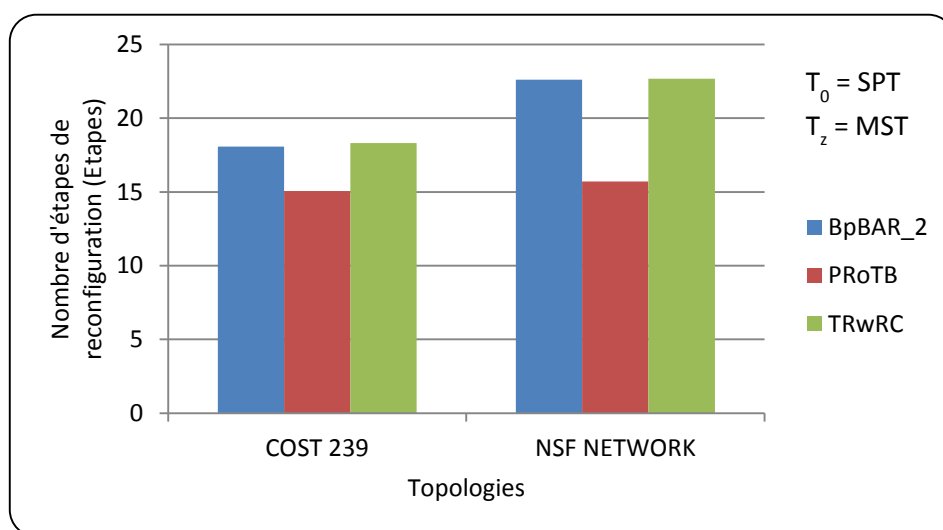


Fig. 5.29 : Durée de reconfiguration.

## 5. Reconfiguration par branche du routage multicast

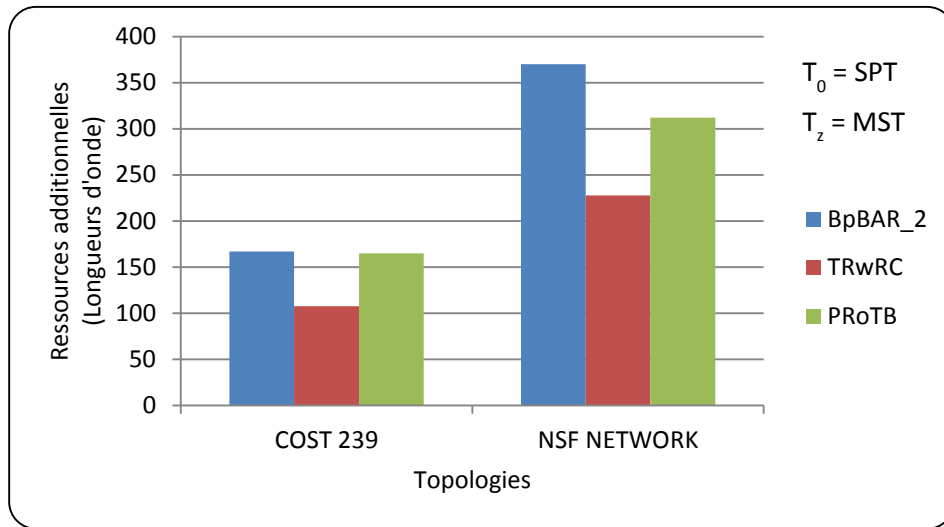


Fig. 5.30 : Coût de ressources additionnelles.

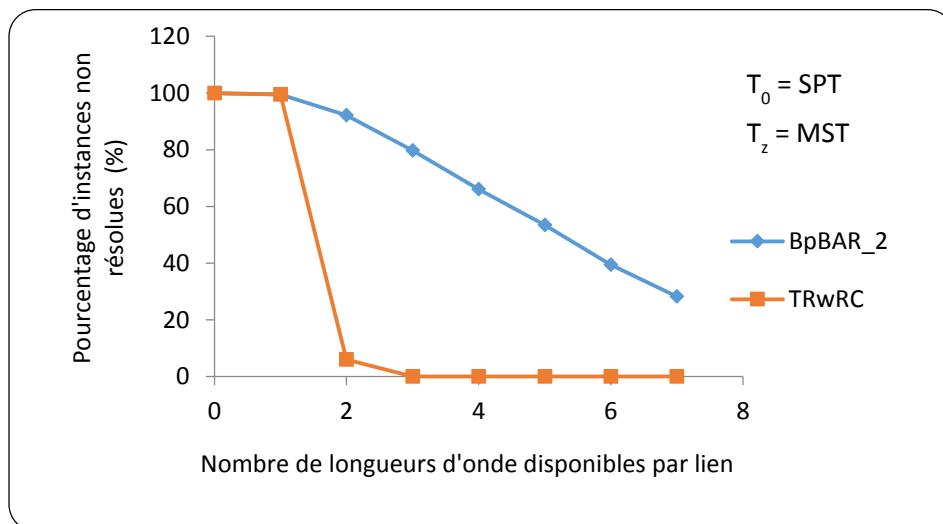


Fig. 5.31 : Pourcentage d'instances de reconfiguration non résolues dans la topologie COST

## 5. Reconfiguration par branche du routage multicast

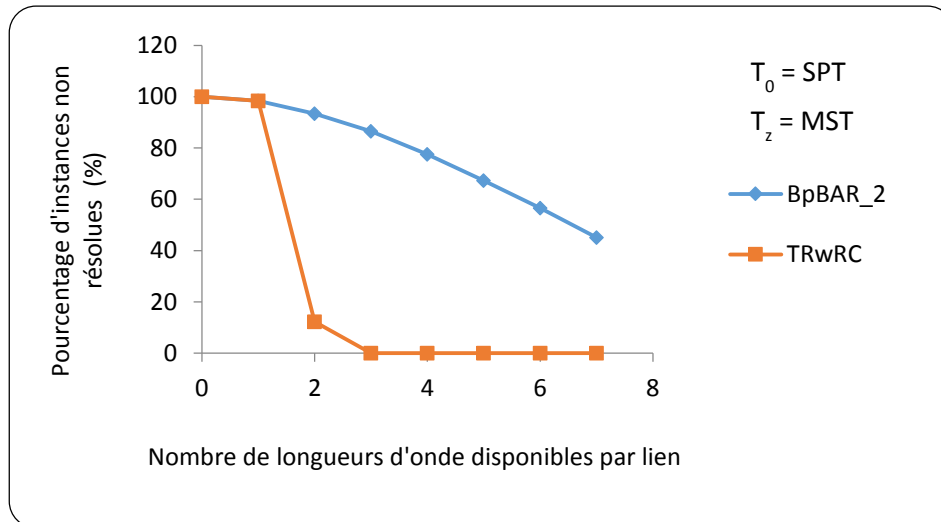


Fig. 5.32 : Pourcentage d'instances de reconfiguration non résolues dans la topologie NSF Network

Tableau 5.14 : Durée de reconfiguration

| CONF. PROCESS | COST 239      |              |                   | NSF NETWORK   |              |                   |
|---------------|---------------|--------------|-------------------|---------------|--------------|-------------------|
|               | AVG<br>(STEP) | SD<br>(STEP) | MIN/MAX<br>(STEP) | AVG<br>(STEP) | SD<br>(STEP) | MIN/MAX<br>(STEP) |
| BpBAR2        | 19,02         | 12,81        | 6/61              | 22,59         | 16,43        | 6/80              |
| TRwRC         | 20,33         | 12,80        | 7/62              | 22,21         | 12,92        | 7/62              |
| PRoTB         | 15,06         | 7,66         | 6/37              | 15,71         | 7,49         | 6/38              |

Tableau 5.15 : Coût de ressources additionnelles

| CONF. PROCESS | COST 239    |            |                 | NSF NETWORK |            |                 |
|---------------|-------------|------------|-----------------|-------------|------------|-----------------|
|               | AVG<br>(WL) | SD<br>(WL) | MIN/MAX<br>(WL) | AVG<br>(WL) | SD<br>(WL) | MIN/MAX<br>(WL) |
| BpBAR2        | 92,75       | 100,65     | 5/542           | 183,8       | 2,6        | 5/1088          |
| TRwRC         | 93,15       | 89,67      | 8/427           | 116,4       | 127,7      | 9/701           |
| PRoTB         | 164,8       | 180,14     | 10/834          | 312,17      | 329,5      | 9/1431          |

Lorsque nous échangeons les algorithmes permettant de calculer l'arbre initial et l'arbre final des problèmes, les figures 5.33 à 5.35 montrent que les mêmes performances sont maintenues pour l'algorithme *TRwRC*. *TRwRC* reconfigure sans interruption de flux tout problème lorsque le nombre de longueurs d'onde par lien est supérieur ou égal à trois tout en utilisant moins de ressources additionnelles.

## 5. Reconfiguration par branche du routage multicast

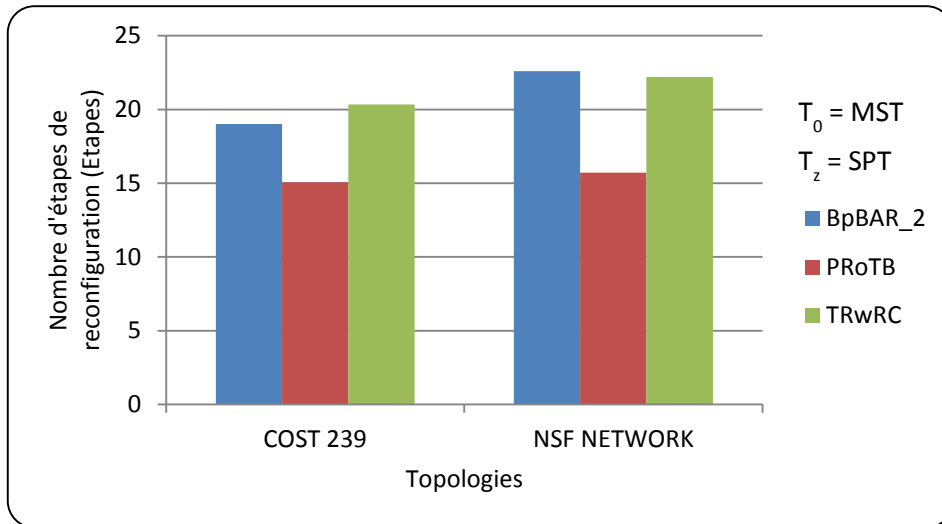


Fig. 5.33 : Durée de reconfiguration.

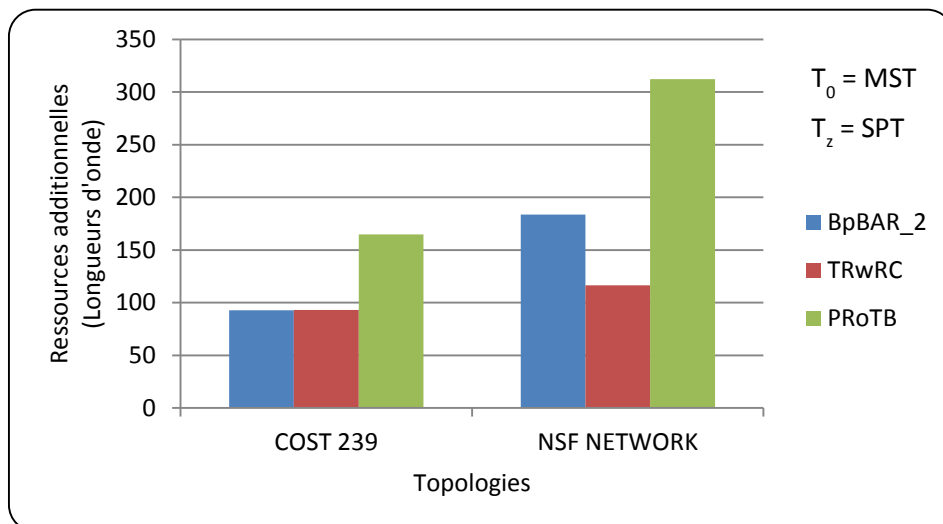


Fig. 5.34 : Coût de ressources additionnelles.

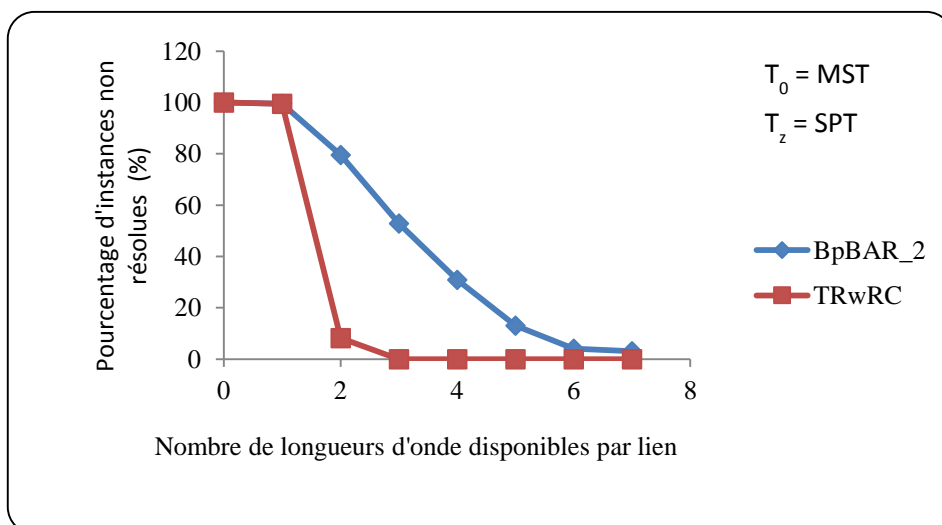


Fig. 5.35 : Pourcentage d'instances de reconfiguration non résolues dans le réseau COST 239.



### 5.7.4 Conclusion

Cette section s'est intéressée à la reconfiguration sans interruption de flux dans un réseau ayant des ressources limitées. L'algorithme *TRwRC* présenté est basé sur l'approche de reconfiguration par branche. *TRwRC* produit une durée de reconfiguration approximativement égale à *BpBAR\_2* et est moins rapide que *ProTB*. Mais ces algorithmes utilisent plus de ressources. *TRwRC* utilise mieux les ressources disponibles. Cet algorithme permet de reconfigurer sans interruption de flux tout problème de reconfiguration lorsqu'il existe trois longueurs d'onde disponibles par lien.

## 5.8 Conclusion

Ce chapitre a présenté une approche de reconfiguration sans interruption de flux basée sur la reconfiguration branche. Cette approche de reconfiguration est appelée *BpBAR*. Les premières sections de ce chapitre ont étudié la reconfiguration rapide sans interruption de flux. Différents algorithmes réduisant la durée de reconfiguration avec l'exécution de plusieurs opérations en parallèle ont été présentés. Les algorithmes *BpBAR\_2* et *ProTB* sont tous basés sur l'approche de reconfiguration *BpBAR* afin de garantir la continuité du flux. Mais cette approche de reconfiguration reste limitée lorsque le réseau présente des contraintes de ressources disponibles sur les liens. La dernière section s'est intéressée à la reconfiguration sans interruption de flux dans un réseau ayant des ressources limitées. L'algorithme *TRwRC* présenté produit de bonnes performances lorsque les ressources par lien sont limitées. C'est un algorithme de reconfiguration par branche qui maintient la continuité du flux en utilisant la propriété de basculement durant le processus de reconfiguration d'une connexion multicast. Il reconfigure sans interruption de flux tout problème de reconfiguration lorsque le nombre de longueurs d'onde disponibles par lien est supérieur ou égale à trois. De plus, il utilise moins de ressources additionnelles que les algorithmes présentés dans la section précédente

# 6. Reconfiguration par sous-arbre d'une connexion multicast

---

## Sommaire

---

|   |     |
|---|-----|
| 6.1 Introduction.....   | 114 |
| 6.2 Contexte et objectifs.....  | 115 |
| 6.3 Approche de reconfiguration par sous-arbre.....   | 116 |
| 6.3.1 Définition.....   | 116 |
| 6.3.2 Challenges de reconfiguration par sous-arbre.....                                     | 116 |
| 6.4 Garantie de la continuité du flux vers toutes les destinations du groupe multicast..... | 119 |
| 6.5 Garantie de la reconfiguration sans boucle.....   | 123 |
| 6.6 Réduction des ressources additionnelles.....  | 124 |
| 6.7 Réduction de la durée de reconfiguration.....   | 125 |
| 6.8 Algorithme de reconfiguration par sous-arbre $STBReconf(P\_of\_T, n, L)$ .....          | 126 |
| 6.8.1 Fonction de construction d'une paire de sous-arbres.....                              | 128 |
| 6.8.2 Description de la fonction de reconfiguration de paire de sous-arbres... ..           | 131 |
| 6.9 Evaluations.....  | 132 |
| 6.9.1 Critères d'évaluation.....  | 132 |
| 6.9.2 Description de la simulation.....   | 132 |
| 6.9.3 Résultats et analyses des résultats.....  | 133 |
| 6.10 Conclusion.....  | 139 |

---

## 6.1 Introduction

La reconfiguration sans interruption de flux a fait l'objet des chapitres antérieurs. Nous y avons montré que la reconfiguration d'arbre sans interruption et sans longueur d'onde additionnelle n'est pas toujours possible. Toutefois, la reconfiguration peut se faire sans interruption de flux en réduisant les ressources utilisées. Ce chapitre présente une nouvelle approche de reconfiguration pour réduire les ressources optiques utilisées. La technique proposée est de reconfigurer l'arbre par sous-arbre. L'idée est de déterminer dans l'arbre initial et l'arbre final à reconfigurer, des paires de sous-arbres constituées d'anciens sous-arbres et de nouveaux sous-arbres pouvant être reconfigurés sans longueur d'onde additionnelle. Ce chapitre étudie l'approche de reconfiguration par sous-arbre et présente un algorithme de reconfiguration par sous-arbre. La section 6.2 présente le contexte et les objectifs de l'approche de reconfiguration

par sous-arbre. La section 6.3 présente les différents challenges de la reconfiguration par sous-arbre. Les chapitres 6.4 à 6.7 présentent les différentes méthodes permettant d'atteindre les différents objectifs de la reconfiguration par sous-arbre. La section 6.8 présente l'algorithme de reconfiguration par sous-arbre proposé et les performances de cet algorithme sont évaluées dans la section 6.9.

### 6.2 Contexte et objectifs

La reconfiguration par sous-arbre est une approche de reconfiguration très importante. En effet, le calcul de l'arbre final dépend en grande partie de l'arbre initial. Ainsi, certains algorithmes de calcul peuvent privilégier la ressemblance entre les deux arbres. En d'autres termes, l'arbre initial et l'arbre final peuvent avoir de nombreux liens identiques ; cela permet de réduire le nombre de nœuds à reconfigurer avec les techniques de reconfiguration existantes dans la littérature (MBB ou BBM). Cela est possible en ne configurant pas les liens identiques.

Dans les chapitres précédents, nous avons étudié la reconfiguration sans interruption de flux. Nous avons proposé des approches de reconfiguration par branche. La reconfiguration par branche d'une instance où l'arbre initial et l'arbre final ont de nombreux segments en commun utilisera beaucoup de ressources additionnelles à cause des segments identiques sur les deux arbres. Cela a pour conséquence directe l'augmentation du taux de blocage durant le processus de reconfiguration de la connexion multicast et donc la réduction de la performance du réseau. Une approche de reconfiguration par sous-arbre permettra de reconfigurer l'arbre avec une bonne performance durant le processus de reconfiguration.

Avec l'approche de reconfiguration par sous-arbre, seulement les segments de l'arbre initial et l'arbre final nécessitant une reconfiguration seront reconfigurés. Cette approche consiste à déterminer une paire de sous-arbres constituée d'un sous-arbre de l'arbre courant (arbre utilisé pour la transmission du flux optique) et d'un sous-arbre de l'arbre final. Les paires de sous-arbres doivent être déterminées de sorte que leur reconfiguration réponde non seulement à la contrainte de reconfiguration sans interruption de flux mais aussi réduise les ressources utilisées. Le problème consistera à déterminer les sous-arbres à configurer sans interruption de flux tout en réduisant le coût de ressources additionnelles utilisées.

La reconfiguration par sous-arbre est un challenge car elle présente plusieurs problèmes, notamment :

- Quelles paires de sous-arbres peuvent être reconfigurées sans interruption de flux et sans boucle dans le réseau tout en réduisant le coût des ressources utilisées?
- Quelle paire de sous-arbres reconfigurer avant quelle autre ?

### 6.3 Approche de reconfiguration par sous-arbre

#### 6.3.1 Définition

- Paire de sous-arbres

Dans ce chapitre, une paire de sous-arbres est constituée d'un sous-arbre de l'arbre courant et d'un sous-arbre de l'arbre final.

Dans la suite, le sous-arbre de l'arbre courant sera appelé ancien sous-arbre et le sous-arbre de l'arbre final sera appelé nouveau sous-arbre.

- Phase de reconfiguration d'une paire de sous-arbres

Une phase de reconfiguration est l'ensemble des étapes permettant de reconfigurer une paire de sous-arbres. En d'autres termes, c'est l'ensemble des étapes qui permettent de passer de l'ancien sous-arbre au nouveau sous-arbre.

#### 6.3.2 Challenges de reconfiguration par sous-arbre

La reconfiguration par sous-arbre consiste à déterminer dans l'arbre initial et l'arbre final du problème un ancien sous-arbre et un nouveau sous-arbre tels que la reconfiguration de cette paire :

- Garantisse la continuité du flux vers toutes les destinations ;
- Garantisse l'unicité du canal optique amont de tout canal (à l'exception des canaux optiques de la source de la connexion multicast) ;
- Ne crée pas de boucle dans le réseau ;
- Minimise le coût des ressources utilisées ;
- Minimise la durée du processus de reconfiguration.

La détermination et la configuration d'une paire de sous-arbres respectant les contraintes citées plus haut nécessitent la résolution de sous-problèmes. Les sous-arbres à reconfigurer dans une phase partagent des ressources optiques comme des nœuds ou des liens avec le complémentaire

de cette paire de sous-arbres dans le réseau qui ne doit pas être configuré dans cette phase de reconfiguration. La reconfiguration de la paire de sous-arbres, lorsqu'elle est mal choisie ou mal menée, peut interrompre le flux vers certaines destinations partageant certaines de ses ressources. De plus, lorsque la paire de sous-arbres n'est pas bien calculée ou lorsque la reconfiguration n'est pas bien menée, cela peut entraîner des boucles entre les nœuds du réseau.

Supposons que l'ancien arbre est noté  $Old\_T$  et que le nouvel arbre est noté  $New\_T$ . La reconfiguration peut créer une boucle dans le réseau lorsqu'il existe deux nœuds (par exemple  $a$  et  $b$ ) tel que l'ordre des nœuds sur l'ancien arbre est différent sur le nouvel arbre. En d'autres termes, une boucle ne peut exister durant le processus de reconfiguration que si :

Les nœuds  $a$  et  $b$  appartiennent à l'arbre initial et l'arbre final et vérifie

$$hauteur(Old\_T, b) < hauteur(Old\_T, a) \text{ et } hauteur(New\_T, a) < hauteur(New\_T, b).$$

En effet, supposons que  $R$  est le nœud de basculement (la racine de la paire de sous-arbres). Après le pré-établissement du nouveau sous-arbre et l'exécution de l'opération de basculement sur le nœud de basculement  $R$ , nous aurons :

$$hauteur(New\_T, a) < hauteur(New\_T, b) < hauteur(New\_T, a).$$

Considérons le problème de reconfiguration de la figure 6.1, dans lequel l'arbre initial est en rouge et l'arbre final en bleu. Supposons qu'on veuille résoudre ce problème par sous-arbre. Les paires de sous-arbres entre les nœuds  $a$  et  $D_1$  et les nœuds  $b$  et  $D_2$  sont identiques donc ne nécessitent pas de reconfiguration. Notons que  $hauteur(New\_T, a) = 1$  ;  $hauteur(New\_T, b) = 2$  ;  $hauteur(Old\_T, a) = 3$  ;  $hauteur(Old\_T, b) = 1$ . Les nœuds qui nécessitent d'être reconfigurés sont  $S$ ,  $a$ ,  $b$  et  $c$ . La paire de sous-arbres restante à configurer est («  $S$ - $b$ - $c$ - $a$  » ; «  $S$ - $a$ - $b$  ») et sa racine  $S$  est le nœud de basculement de la paire. Supposons que dans la première étape, on préétablit le nouveau sous-arbre «  $S$ - $a$ - $b$  ». Dans la seconde étape, on bascule le flux de l'ancien sous-arbre au nouveau sous-arbre à partir du nœud de basculement  $S$ . L'arbre obtenu est représenté par la figure 6.2.b. On constate dans l'arbre obtenu à la figure 6.2.b que  $hauteur(New\_T, a) < hauteur(New\_T, b) < hauteur(New\_T, a)$ . Donc il existe une boucle entre les nœuds  $a$  et  $b$ .

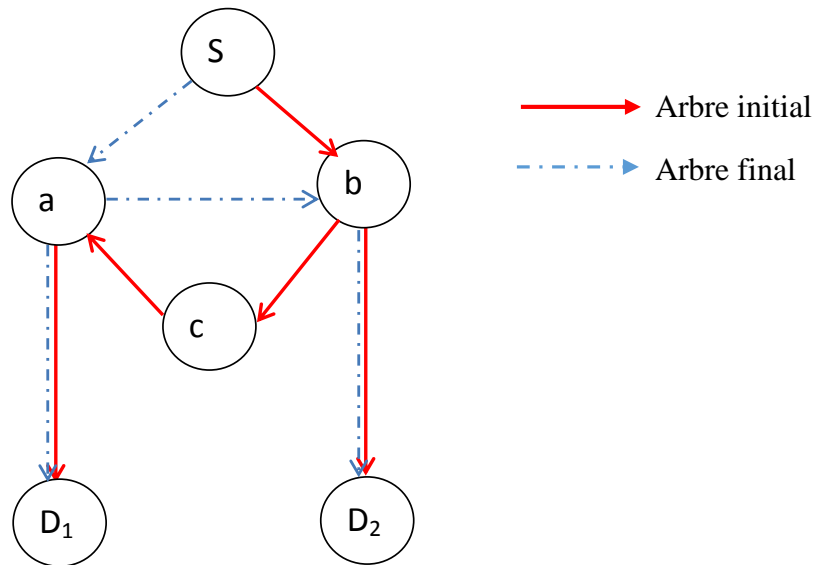


Fig. 6.1: Instance d'un problème de reconfiguration

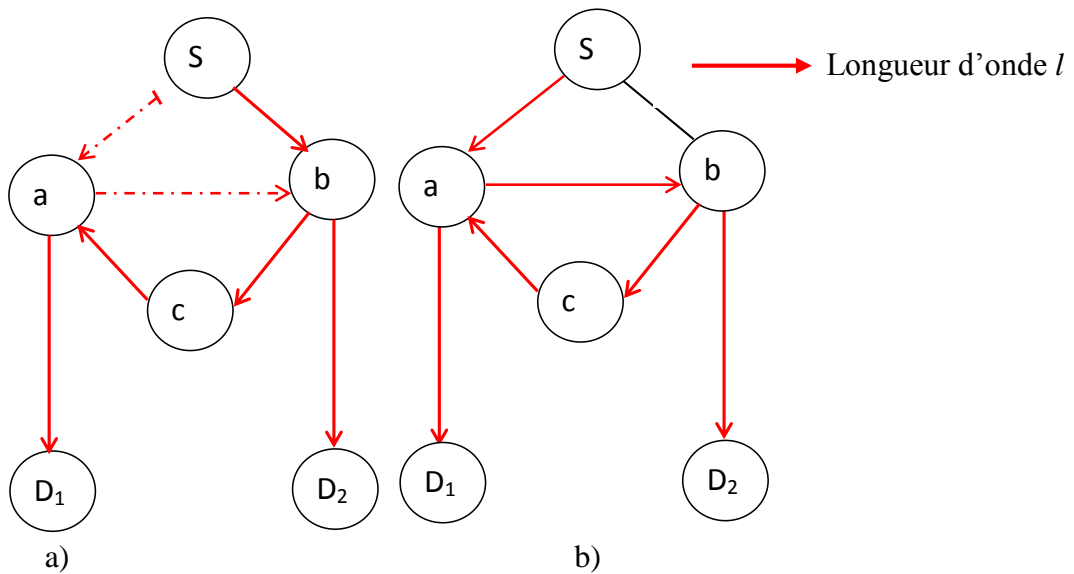


Fig. 6.2 : Exemple de reconfiguration par sous-arbre avec boucle

Un autre problème qui doit être résolu se présente lorsque la reconfiguration est mal conduite. Certains canaux optiques dans ce cas peuvent être alimentés par plus d'un canal optique amont. Cela ne respecte pas les contraintes de reconfiguration. Sur la figure 6.2.b, le canal optique entre les nœuds  $a$  et  $D_1$  est alimenté à la fois par les canaux optiques entre  $S$  et  $a$ , et  $c$  et  $a$ .

Les sections suivantes étudient les différents sous-problèmes de la reconfiguration par sous-arbre.

## 6.4 Garantie de la continuité du flux vers toutes les destinations du groupe multicast

La continuité du flux vers toutes les destinations du groupe multicast est garantie s'il existe un arbre couvrant la source de la connexion et les destinations à chaque étape du processus de reconfiguration. Cela implique qu'il doit exister une branche de l'arbre courant entre la source de la connexion et chaque destination à chaque étape.

Dans le graphe du problème de reconfiguration, on peut distinguer deux types de paires de sous-arbres :

- les paires de sous-arbres identiques,
- les paires de sous-arbres distincts.

Une paire de sous-arbres identiques est constituée d'un ancien sous-arbre et d'un nouveau sous-arbre ayant les mêmes nœuds et les mêmes liens (fibres).

Une paire de sous-arbres distincts est constituée d'un ancien sous-arbre et d'un nouveau sous-arbre ayant au moins un lien différent.

### i) Paires de sous-arbres identiques

Les paires de sous-arbres identiques constituent le premier type de paires de sous-arbres reconfigurables sans interruption de flux. En effet, lorsque l'ancien sous-arbre et le nouveau sous-arbre sont identiques, aucune opération de reconfiguration n'est nécessaire.

### ii) Paires de sous-arbres distincts

Lorsque l'ancien sous-arbre et le nouveau sous-arbre sont distincts, la paire de sous-arbres est reconfigurable sans interruption de flux si:

- les deux sous-arbres ont la même racine,
- l'ensemble des destinations du groupe multicast atteint par les canaux optiques du nouveau sous-arbre est le même que celui atteint par l'ancien sous-arbre.

### Preuve

- Unicité de la racine des sous-arbres

Soit  $ST_1$  et  $ST_2$  deux sous-arbres respectivement ancien sous-arbre et nouveau sous-arbre constituant une paire de sous-arbres. La racine du sous-arbre  $ST_1$  est  $r_1$  et la racine du sous-arbre  $ST_2$  est  $r_2$ .

Par définition, l'exécution de l'opération de basculement sur le nœud racine d'une paire de sous-arbres simultanément interrompt le flux sur l'ancien sous-arbre et alimente le nouveau. Supposons que les nœuds  $r_1$  et  $r_2$  sont différents et montrons par contradiction que les deux sous-arbres ont la même racine.

Quatre différents cas de figures peuvent exister :

- $r_1 \notin ST_2$  et  $r_2 \in ST_1$
- $r_1 \notin ST_2$  et  $r_2 \notin ST_1$
- $r_1 \in ST_2$  et  $r_2 \in ST_1$
- $r_1 \in ST_2$  et  $r_2 \notin ST_1$

Nous analyserons les 4 différents cas ci-dessus. Pour chaque cas, il existe deux sous-cas.  $r_1$  et  $r_2$  étant différents, soit on exécute l'opération de basculement sur  $r_1$  avant  $r_2$ , soit on l'exécute sur  $r_2$  avant  $r_1$ .

**Cas 1 :**  $r_1 \notin ST_2$  et  $r_2 \in ST_1$

Si nous configurons en premier le nœud  $r_1$  pour alimenter le nouveau sous-arbre, alors il existe un nœud  $x \in ST_2$  tel que  $hauteur(New\_T, r_1) < hauteur(New\_T, x) < hauteur(New\_T, r_2)$  (car  $r_1 \notin ST_2$ ).  $r_2$  est la racine du nouveau sous-arbre  $ST_2$  donc  $\forall y \in ST_2$ ,  $hauteur(New\_T, r_2) < hauteur(New\_T, y)$ . Alors  $hauteur(New\_T, r_2) < hauteur(New\_T, x)$  (car  $x \in ST_2$ ). On obtient  $hauteur(New\_T, x) < hauteur(New\_T, r_2)$  et  $hauteur(New\_T, r_2) < hauteur(New\_T, x)$ . Cela est une contradiction.

Si nous configurons en premier le nœud  $r_2$  pour interrompre le flux sur l'ancien sous-arbre  $ST_1$ , alors  $hauteur(Old\_T, r_2) < hauteur(Old\_T, r_1)$ .  $r_1$  est la racine de  $ST_1$ . Nous avons supposé que  $r_2 \in ST_1$  donc  $hauteur(Old\_T, r_1) < hauteur(Old\_T, r_2)$ . On obtient  $hauteur(Old\_T, r_2) < hauteur(Old\_T, r_1)$  et  $hauteur(Old\_T, r_1) < hauteur(Old\_T, r_2)$ . Cela est une contradiction.

**Cas 2 :**  $r_1 \notin ST_2$  et  $r_2 \notin ST_1$

Si nous configurons en premier le nœud  $r_1$  pour alimenter le nouveau sous-arbre, alors il existe un nœud  $x \in ST_2$  tel que  $hauteur(New\_T, r_1) < hauteur(New\_T, x) < hauteur(New\_T, r_2)$  (car  $r_1 \notin ST_2$ ).  $r_2$  est la racine de  $ST_2$  donc  $\forall y \in ST_2$ ,  $hauteur(New\_T, r_2) < hauteur(New\_T, y)$ .



## 6. Reconfiguration par sous-arbre du routage multicast

Alors  $hauteur(New\_T, r_2) < hauteur(New\_T, x)$  (car  $x \in ST_2$ ). On obtient  $hauteur(New\_T, x) < hauteur(New\_T, r_2)$  et  $hauteur(New\_T, r_2) < hauteur(New\_T, x)$ . Cela est une contradiction.

Si nous configurons en premier le nœud  $r_2$  pour interrompre le flux sur l'ancien sous-arbre  $ST_1$ , alors  $hauteur(Old\_T, r_2) \leq hauteur(Old\_T, r_1)$  (car  $r_2 \in ST_1$ ).  $r_1$  est la racine de l'ancien sous-arbre  $ST_1$  alors  $hauteur(Old\_T, r_1) \leq hauteur(Old\_T, r_2)$ . On obtient  $hauteur(Old\_T, r_2) \leq hauteur(Old\_T, r_1)$  et  $hauteur(Old\_T, r_1) \leq hauteur(Old\_T, r_2)$ . Alors  $r_1 = r_2$  car un sous-arbre a une racine unique.  $r_1 \neq r_2$  et  $r_1 = r_2$ , ce qui est une contradiction.

**Cas 3 :**  $r_1 \in ST_2$  et  $r_2 \in ST_1$

Si nous configurons en premier le nœud  $r_1$  pour alimenter le nouveau sous-arbre, alors  $hauteur(New\_T, r_1) \leq hauteur(New\_T, r_2)$ .  $r_2$  est la racine du nouveau sous-arbre  $ST_2$  alors  $hauteur(New\_T, r_2) \leq hauteur(New\_T, r_1)$ . On a  $hauteur(New\_T, r_1) \leq hauteur(New\_T, r_2)$  et  $hauteur(New\_T, r_2) \leq hauteur(New\_T, r_1)$ . Donc  $r_1 = r_2$  car un sous-arbre a une racine unique.  $r_1 \neq r_2$  et  $r_1 = r_2$ , ce qui est une contradiction.

Si nous configurons en premier le nœud  $r_2$  pour interrompre le flux sur l'ancien sous-arbre  $ST_1$ , alors  $hauteur(Old\_T, r_2) \leq hauteur(Old\_T, r_1)$  (car  $r_2 \in ST_1$ ).  $r_1$  est la racine de l'ancien sous-arbre  $ST_1$  alors  $hauteur(Old\_T, r_1) \leq hauteur(Old\_T, r_2)$ . On a  $hauteur(Old\_T, r_2) \leq hauteur(Old\_T, r_1)$  et  $hauteur(Old\_T, r_1) \leq hauteur(Old\_T, r_2)$ . Donc  $r_1 = r_2$  car un sous-arbre a une racine unique.  $r_1 \neq r_2$  et  $r_1 = r_2$ , ce qui est une contradiction.

**Cas 4 :**  $r_1 \in ST_2$  et  $r_2 \notin ST_1$

Si nous configurons en premier le nœud  $r_1$  pour alimenter le nouveau sous-arbre, alors  $hauteur(New\_T, r_1) \leq hauteur(New\_T, r_2)$ .  $r_2$  est la racine du nouveau sous-arbre  $ST_2$  alors  $hauteur(New\_T, r_2) \leq hauteur(New\_T, r_1)$ . On a  $hauteur(New\_T, r_1) \leq hauteur(New\_T, r_2)$  et  $hauteur(New\_T, r_2) \leq hauteur(New\_T, r_1)$ . Donc  $r_1 = r_2$  car un sous-arbre a une racine unique.  $r_1 \neq r_2$  et  $r_1 = r_2$ , ce qui est une contradiction.

Si nous configurons en premier le nœud  $r_2$  pour interrompre le flux sur l'ancien sous-arbre  $ST_1$ , alors il existe un nœud  $x \in ST_1$  tel que  $hauteur(Old\_T, r_2) < hauteur(Old\_T, x) < hauteur(Old\_T, r_1)$  (car  $r_2 \notin ST_1$ ).  $r_1$  est la racine de l'ancien sous-arbre  $ST_1$  donc  $\forall y \in ST_1$ ,  $hauteur(Old\_T, r_1) \leq hauteur(Old\_T, y)$ .

## 6. Reconfiguration par sous-arbre du routage multicast

Alors  $hauteur(Old\_T, r_1) \leq hauteur(Old\_T, x)$  (car  $x \in ST_1$ ). On a  $hauteur(Old\_T, x) < hauteur(Old\_T, r_1)$  et  $(Old\_T, r_1) \leq hauteur(Old\_T, x)$ . Ce qui est une contradiction.

En conclusion, pour garantir la reconfiguration sans interruption de flux, les sous-arbres de la paire qu'on doit reconfigurer doivent avoir la même racine.

- Identité de l'ensemble de destinations des sous-arbres

Soit  $Old\_D$  et  $New\_D$  respectivement l'ensemble de destinations couvertes par le nouveau sous-arbre et l'ancien sous-arbre d'une paire de sous-arbres à reconfigurer. Soit  $D$  l'ensemble de destinations de la connexion multicast. Nous montrerons par contradiction que  $Old\_D = New\_D$ .

Supposons que la reconfiguration d'une paire de sous-arbres garantisse la continuité du flux vers toutes les destinations de la connexion multicast. Supposons qu'il existe un seul arbre couvrant toutes les destinations de la connexion multicast durant le processus de reconfiguration d'une paire de sous-arbres. En d'autres termes, chaque nœud (hormis la source) a un unique nœud amont avant et après la reconfiguration.

Supposons que  $Old\_D \neq New\_D$ . Par définition,  $Old\_D \subset D$  et  $New\_D \subset D$ ; les branches des sous-arbres sont à liens disjoints. Alors, soit il existe un nœud  $d \in D$  tel que  $d \in Old\_D$  et  $d \notin New\_D$ ; soit il existe un nœud  $d' \in D$  tel que  $d' \notin Old\_D$  et  $d' \in New\_D$ .

Durant la reconfiguration d'une paire de sous-arbres, l'exécution de l'opération de basculement à la racine de la paire simultanément interrompt le flux optique des anciennes branches de destinations  $Old\_D$  et alimente les nouvelles branches de destinations  $New\_D$ .

Si  $d \in Old\_D$  et  $d \notin New\_D$ : la continuité du flux optique étant maintenue vers toutes les destinations de la connexion multicast  $D$ , en particulier pour le nœud  $d$ , permet de dire que  $d \in New\_D$ . On a  $d \notin New\_D$  et  $d \in New\_D$ : cela est une contradiction.

Si  $d' \notin Old\_D$  et  $d' \in New\_D$ : avant la reconfiguration, nous savons d'après les hypothèses qu'un nœud (différent de la source) possède un unique nœud amont. Notons  $x$  le nœud amont d'un nœud destinataire  $d'$  avant la reconfiguration.

Après la reconfiguration, sachant que  $d' \in New\_D$ ,  $d'$  a un nœud amont (par exemple  $y$ ). Nous savons par hypothèses que les branches des sous-arbres sont à liens disjoints. Alors  $x \neq y$ .

Par conséquent, le nœud destinataire  $d'$  a deux nœuds amont après la reconfiguration. Cela est une contradiction.

Finalement,  $Old\_D = New\_D$ . Les deux sous-arbres constituant une paire de sous-arbres à reconfigurer sans interruption de flux ont le même (sous-)ensemble de destinations.

## 6.5 Garantie de la reconfiguration sans boucle

Certains nœuds du réseau peuvent appartenir à la fois à l'arbre initial et l'arbre final d'un problème de reconfiguration. Un mauvais ordonnancement de la reconfiguration de ces nœuds ou une mauvaise configuration de ces nœuds peut créer une boucle dans le réseau. L'ordre défini par la hauteur des nœuds sur les différents arbres est un paramètre très important dans la création de boucle lors de la reconfiguration. En effet, une boucle ne peut être créée dans le réseau que s'il existe des nœuds par exemple  $a$  et  $b$  appartenant à la fois à l'arbre initial ( $Old\_T$ ) et l'arbre final ( $New\_T$ ) tel que  $hauteur(Old\_T, b) < hauteur(Old\_T, a)$  et  $hauteur(New\_T, a) < hauteur(New\_T, b)$  tel que montré en 6.3.2.

Nous proposons deux approches de résolution du problème de reconfiguration sans boucle.

- La première approche consiste à déterminer la paire de sous-arbres à reconfigurer de telle sorte que s'il existe deux nœuds de la paire appartenant à la fois à l'arbre courant et l'arbre final, l'ordre de leurs hauteurs respectives sur l'arbre courant soit le même sur l'arbre final.
- La seconde solution intervient au niveau du processus de configuration lorsque la paire de sous-arbres contient des nœuds appartenant à la fois à l'arbre initial et l'arbre final. Ces nœuds doivent être configurés de telle sorte qu'ils commutent le flux reçu du port d'entrée vers leurs ports de sortie sauf vers les ports de sortie appartenant exclusivement à l'ancien sous-arbre. Cette procédure de configuration des nœuds garantit que s'il existe des nœuds par exemple  $a$  et  $b$  appartenant à la fois à l'arbre initial et l'arbre final tel que  $hauteur(Old\_T, b) < hauteur(Old\_T, a)$  et  $hauteur(New\_T, a) < hauteur(New\_T, b)$ , la configuration de la paire ne crée pas de boucle. Nous prouvons ci-dessous cette assertion.

En effet, le chemin formé par les chemins entre les nœuds  $b$  et  $a$  sur l'ancien sous-arbre et celui entre les nœuds  $a$  et  $b$  sur le nouveau sous-arbre forme un circuit. Cette approche consiste à configurer les nœuds de telle sorte que tous les canaux optiques du circuit ne transmettent pas le flux au même instant.

Supposons que chaque nœud de la paire est configuré pour commuter le flux sur ces ports de sortie sauf sur ceux appartenant exclusivement au sous-arbre initial. Lorsque le chemin entre les nœuds  $a$  et  $b$  sur le nouveau sous-arbre est alimenté, simultanément le flux sur le chemin entre les nœuds  $b$  et  $a$  sur l'ancien chemin est interrompu. Le circuit entre les nœuds  $a$  et  $b$  n'est pas alimenté au même moment ; il n'y a donc pas de boucle dans le réseau après la configuration de cette paire de sous-arbres.

### 6.6 Réduction des ressources additionnelles

Une ressource additionnelle sur un lien (fibre) est une longueur d'onde de ce lien, autre que celle utilisée par l'arbre courant, servant à l'établissement du nouveau sous-arbre sur ce lien lorsque celui-ci est déjà utilisé par l'arbre courant.

La procédure de reconfiguration d'une paire de sous-arbres comprend trois étapes :

- Etablissement du nouveau sous-arbre,
- Basculement du flux sur le nouveau sous-arbre et interruption sur l'ancien sous-arbre,
- Suppression de l'ancien sous-arbre.

Pour réduire les ressources additionnelles utilisées, il faut déterminer et reconfigurer les paires de sous-arbres dont la reconfiguration ne nécessite pas de ressources additionnelles. Les paires de sous-arbres permettant d'atteindre cet objectif de reconfiguration sans ressource additionnelle sont les paires de sous-arbres identiques et les paires de sous-arbres à liens disjoints.

Les paires de sous-arbres identiques ne requièrent aucune opération de reconfiguration ; elles n'utilisent donc aucune ressource additionnelle.

Lorsque les sous-arbres sont distincts, les paires de sous-arbres dont la reconfiguration se fait sans ressource additionnelle sont telles que le nouveau sous-arbre et l'arbre courant ont des liens tous disjoints. En effet, le nouveau sous-arbre ne peut être établi entièrement sans longueur d'onde additionnelle que si les liens qu'il doit utiliser sont libres. En d'autres termes, le nouveau sous-arbre et l'arbre courant utilisé pour la transmission du flux sont à liens disjoints.

Nous avons la propriété suivante :

### Propriété

Si deux sous-arbres distincts sont reconfigurables sans longueur d'onde additionnelle alors le nouveau sous-arbre est à liens disjoints de l'arbre courant.

### Preuve

Nous utilisons la preuve par l'absurde. Supposons qu'il existe une paire de sous-arbres distincts reconfigurable sans longueur d'onde additionnelle. Les deux sous-arbres étant distincts, il existe au moins un lien dont les nœuds doivent être configurés. Supposons qu'il existe au moins une fibre optique en commun entre le nouveau sous-arbre et l'arbre courant. Cette fibre est alors utilisée pour la transmission du flux avec la longueur d'onde utilisée par l'arbre initial  $\lambda_0$ .

Toute reconfiguration du nouveau sous-arbre nécessite l'établissement de la connexion multicast sur ce sous-arbre (car le routage est basé sur la longueur d'onde) avec la longueur d'onde  $\lambda_0$ . La longueur d'onde  $\lambda_0$  étant occupée par l'arbre courant de transmission du flux sur la fibre partagée, elle ne peut plus être utilisée par le nouveau sous-arbre (propriété d'unicité de longueur d'onde sur un lien optique). Le sous-arbre ne peut être établi. Il n'est donc pas reconfigurable sans longueur d'onde additionnelle. Cela contredit l'hypothèse.

## 6.7 Réduction de la durée de reconfiguration

La réduction de la durée de reconfiguration peut se faire en exécutant en parallèle toutes les opérations de configuration de nœuds appartenant à une même étape dans le processus de reconfiguration d'une paire de sous-arbres. Le premier gain peut se faire en exécutant en parallèle les opérations de configuration permettant d'établir le nouveau sous-arbre. Cela est possible car les ports que doivent utiliser le nouveau sous-arbre ne sont pas utilisés par l'arbre courant (car le nouveau sous-arbre et l'ancien sous-arbre sont à liens disjoints). Le second gain peut se faire en exécutant en parallèle les opérations de configuration permettant de supprimer l'ancien sous-arbre car il n'est plus utilisé pour la transmission de flux après la reconfiguration de la paire.

Nous présentons ci-dessous l'algorithme de reconfiguration par sous-arbre proposé.

## 6.8 Algorithme de reconfiguration par sous-arbre

### *STBReconf(P\_of\_T, n, L)*

Nous proposons l'algorithme de reconfiguration par sous-arbre *STBReconf(P\_of\_T, n, L)*. C'est un algorithme récursif (algorithme 6.1) qui utilise trois fonctions principales:

- Une fonction permettant de construire une paire de sous-arbres à reconfigurer sans longueur d'onde additionnelle appelée *Select\_Sub\_Tree\_Pair(n)*,
- Une fonction permettant de construire une paire de sous-arbres ayant pour destination un nœud convergent. Cette fonction est appelée *New\_Select\_Sub\_Tree\_Pair(n)* et reconfigure la paire de sous-arbres sans interruption de flux avec éventuellement des ressources additionnelles,
- Une fonction permettant de reconfigurer une paire de sous-arbres appelée *Configure(P\_of\_Sub\_T, Conf\_order)*.

La fonction de reconfiguration par sous-arbre *STBReconf(P\_of\_T, n, L)* prend en entrée une paire constituée d'arbre initial et arbre final, et un nœud  $n$  servant de point de départ de construction de la paire de sous-arbres. Ce nœud représentera la racine de la paire de sous-arbres qui sera construite. Les lignes 1 à 18 permettent de déterminer une paire de sous-arbres pouvant être reconfigurée sans interruption de flux et sans longueur d'onde additionnelle. Si une paire de sous-arbres est retournée par la fonction de calcul de paires de sous-arbres *Select\_Sub\_Tree\_Pair(n)*, elle est reconfigurée avec la fonction *Configure(P\_of\_Sub\_T, Conf\_Order)*. Ensuite on appelle la même fonction de reconfiguration *STBReconf(P\_of\_T, n, L)* avec de nouvelles valeurs en entrée qui sont la paire d'arbres courant et arbre final non configuré et un nœud  $n$  qui est soit la tête de la file de nœuds divergents visités mais non configurés  $L$ , soit la racine de la paire de sous-arbres précédemment configurée, soit encore le nœud successeur de la racine de la paire de sous-arbres précédemment configurée.

Les nœuds divergents visités sont enregistrés dans une file  $L$ . Cette file est mise à jour après chaque reconfiguration d'une paire de sous-arbres. La mise à jour consiste à extraire de la file, les nœuds qui ne sont plus des nœuds divergents. Le nœud qui se trouve à la tête de cette file est le point de départ de la construction de la paire. Ce nœud est ramené par la fonction *pop[L]* (algorithme 6.2).

**Algorithme 6.1: Algorithme récursif de reconfiguration par sous-arbre**


---

```

STBReconf(P_of_T, n, L) {          /* pair of trees P_of_T and n is a node of the initial pair of trees*/
1. If (n is a latch){                /*L is the file of visited latches but not configured*/
2.     If ((P_of_Sub_T = Select_Sub_Tree_Pair(n) && (P_of_Sub_T ≠ null)) {
3.         Configure(P_of_Sub_T, Conf_Order)          /*Conf_Order contains the nodes */
4.         If (L ≠ ∅) {                    /*to configure and their configuration description*/
5.             STBReconf(P_of_T – P_of_Sub_T, pop(L), L)
6.         }
7.         Else {
8.             STBReconf(P_of_T – P_of_Sub_T, next(n), L)
9.         }
10.    }
11.    Else {
12.        push(L, n)
13.        STBReconf(P_of_T, next(n), L)
14.    }
15. }
16. Else{
17.    STBReconf(P_of_T, next(n), L)
18. }
19. If (next(n) = ∅ and P_of_T ≠ null){
20.    n = root of the initial pair of trees
21.    Do {
22.        P_of_Sub_T = NewSelect_Sub_Tree_Pair(n) /*determination of sub-tree pair*/
23.        If ((P_of_Sub_T = null)) { /*such that n is a convergent node and is the leaf*/
24.            n = next(n)
25.        }
26.    } While (P_of_Sub_T = null)
27.    Configure(P_of_Sub_T, Conf_Order)
28.    n = root of the initial pair of trees
29.    STBReconf(P_of_T – P_of_Sub_T, n, L)
30. }
31. If (P_of_T = null){
32.    Return 0;
33. }
34. }

```

---

**Algorithme 6.2: Fonction qui retourne l'élément en tête d'une file de nœud.**


---

```

pop(L) {
1. Retourner le premier élément de la file
2. Supprimer la valeur retournée de la file
}

```

---

Si le nœud  $n$  est un nœud divergent et qu'aucune paire de sous-arbres n'a été construite à partir de celui-ci, alors on applique l'algorithme sur un autre nœud  $next(n)$ , successeur du nœud  $n$  sur l'arbre final et on met le nœud  $n$  à la fin de la file des nœuds divergents visités. Si le nœud  $n$  n'est pas un nœud divergent, alors on applique l'algorithme sur un autre nœud  $next(n)$ ,

successeur du nœud  $n$  dans un parcours en profondeur sur l'arbre final (ligne 16 à 18). Si la condition 19 est vraie, alors aucun nœud divergent ne permet de construire une paire de sous-arbres or les nœuds ne sont pas entièrement configurés. Les lignes 20 à 26 déterminent une paire de sous-arbres ayant pour destination un nœud convergent. La ligne 27 configure cette paire de sous-arbres en utilisant, lorsque c'est impossible de faire autrement, des longueurs d'onde additionnelles. Ensuite la fonction de reconfiguration par sous-arbre  $STBReconf(P\_of\_T, n, L)$  est appelée à la ligne 29 avec l'argument  $n$  égal à la source de la paire d'arbres initiale. La reconfiguration s'achève si l'arbre initial et l'arbre final sont entièrement configurés.

### 6.8.1 Fonction de construction d'une paire de sous-arbres reconfigurables sans longueur d'onde additionnelle

La fonction  $Select\_Sub\_Tree\_Pair(n)$  détermine une paire d'ancien et de nouveau sous-arbres à reconfigurer. La paire de sous-arbres retournée par cette fonction est telle que sa reconfiguration se fait sans interruption de flux et sans longueur d'onde additionnelle. Les paires de sous-arbres identiques ne sont pas retournées par la fonction car elles ne nécessitent pas de configuration.

Cette fonction prend en entrée un nœud qui sera le nœud racine de la paire déterminée. Le nœud racine constitue le point de départ de construction de la paire de sous-arbres. La description de la fonction est la suivante :

- Détermination du nouveau sous-arbre à liens disjoints de l'arbre courant

Cette étape s'exécute si le nœud  $n$  en entrée de la fonction est un nœud divergent. Dans cette étape, nous déterminons à partir du nœud  $n$  suivant les branches de l'arbre final, les premiers nœuds convergents de chaque segment d'extrémité initiale  $n$  qui ne sont pas des nœuds divergents. Cet ensemble de nœuds convergents sera appelé ensemble de nœuds convergents candidats déterminés à partir du nœud divergent  $n$ . Par construction, le nouveau sous-arbre couvrant le nœud racine  $n$  et l'ensemble de nœuds convergents candidats déterminés à partir de  $n$  est à liens disjoints de l'arbre courant.



## 6. Reconfiguration par sous-arbre du routage multicast

- Détermination de l'ancien sous-arbre et du nouveau sous-arbre

Cette étape consiste à déterminer un sous-ensemble de l'ensemble des nœuds convergents candidats tel que le nouveau sous-arbre et l'ancien sous-arbre couvrant le nœud  $n$  et ses nœuds convergents soient reconfigurables sans interruption de flux. Supposons que l'ensemble des nœuds convergents candidats déterminés à partir du nœud divergent  $n$  est  $K(n)$ . Un sous-ensemble de  $K(n)$  contient  $k$  nœuds convergents candidats, avec  $k \in [1; |K(n)|]$ . Le processus de détermination se décrit comme suit :

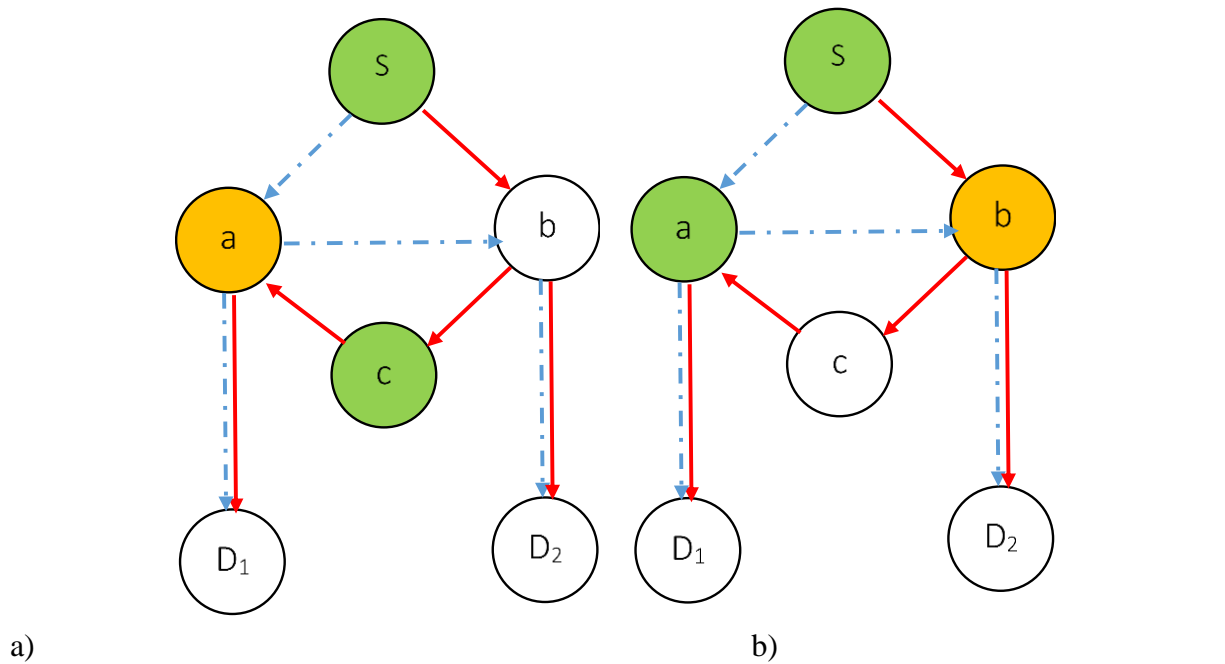
- (i) Pour chaque sous-ensemble de l'ensemble  $K(n)$ , déterminer l'ancien sous-arbre couvrant le nœud  $n$  et ce sous-ensemble de nœuds convergents.
- (ii) Si la paire de sous-arbres déterminée interrompt le flux, alors aller en (i) jusqu'à l'obtention d'une paire de sous-arbres qui n'interrompt pas le flux ou jusqu'à ce que tous les sous-ensembles de  $K(n)$  soit explorés.

La reconfiguration d'une paire de sous-arbres interrompt le flux si l'une des conditions ci-dessous est vérifiée :

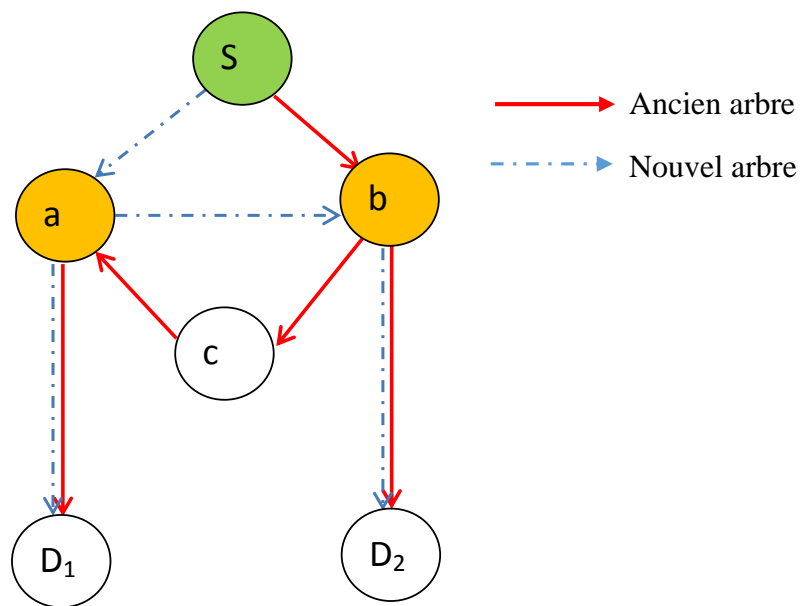
- (a) Si un nœud de destination appartient à l'ancien sous-arbre et s'il n'est pas nœud convergent alors la reconfiguration de cet ancien sous-arbre provoquera l'interruption du flux vers ce nœud destinataire. Dans ce cas, aller à (i).
- (b) Si cet ancien sous-arbre a un nœud de branchement et s'il existe un nœud aval de ce nœud de branchement tel qu'il n'existe pas de chemin sur l'arbre courant entre ce nœud aval et un nœud convergent candidat, alors la reconfiguration de cet ancien sous-arbre provoquera l'interruption du flux vers un nœud destinataire. En effet, l'ensemble de nœuds destinataires de cet ancien sous-arbre et du nouveau sous-arbre couvrant la même racine et le même ensemble de nœuds convergents seront différents. Dans ce cas, aller à (i).

Afin de réduire la durée du processus de reconfiguration, le nombre de nœuds convergents par paire de sous-arbres doit être le plus grand possible. Cela permet d'accroître le nombre de nœuds à configurer en parallèle.

## 6. Reconfiguration par sous-arbre du routage multicast



a) Le nœud  $a$  est un nœud convergent ; les nœuds  $S$  et  $c$  sont respectivement les nœuds amonts sur l'arbre initial et sur l'arbre final. b) Le nœud  $b$  est un nœud convergent ; les nœuds  $S$  et  $a$  sont respectivement les nœuds amonts sur l'arbre initial et sur l'arbre final.



c) Paire de sous-arbres couvrant la racine  $S$  et les nœuds convergents candidats  $a$  et  $b$

Fig. 6.3: Problème de reconfiguration.

Dans la figure 6.3, L'arbre initial est en rouge et l'arbre final est en bleu. les nœuds convergents candidats déterminés à partir du nœud divergent  $S$  sont  $\{a ; b\}$  comme le montre les figures 6.3.a et 6.3.b. le nœud  $c$  n'est pas un nœud divergent car il n'a pas de nœud amont sur l'arbre

final. Les nœuds  $D_1$  et  $D_2$  ne se pas des nœuds convergents car ils ont le même nœud amont sur l'arbre initial et l'arbre final.

L'ancien sous-arbre et le nouveau sous-arbre couvrant la racine  $S$  et les nœuds convergents candidats  $a$  et  $b$  sont respectivement  $Old\_ST = (S \{b \{c \{a\}\})$  et  $New\_ST = (S \{a \{b\})$  comme le montre la figure 6.3.c.

La paire de sous-arbres constituée de  $Old\_ST$  et  $New\_ST$ , enracinée au nœud  $S$  et couvrant les nœuds convergents  $a$  et  $b$  est telle que :

- Aucun nœud destinataire n'appartient à l'ancien sous-arbre :  $D_1 \notin New\_ST$ ;  $D_2 \notin New\_ST$ ;  $D_1 \notin Old\_ST$ ;  $D_2 \notin Old\_ST$ ;
- Le nœud  $b$  est un nœud de branchement appartenant à la paire de sous-arbres mais il est un nœud convergent.

En conclusion, la paire de sous-arbres enracinée en  $S$  et couvrant les nœuds convergents  $\{a ; b\}$  est reconfigurable sans interruption de flux et sans ressource additionnelle.

### 6.8.2 Description de la fonction de reconfiguration d'une paire de sous-arbres

L'algorithme 6.3 présente la fonction  $Configure(P\_of\_Sub\_T, Conf\_Order)$  permettant de reconfigurer une paire de sous-arbres. Cette fonction prend en entrée la paire de sous-arbres à reconfigurer  $P\_of\_Sub\_T$ .  $Conf\_Order$  est un paramètre de sortie qui donne l'ordre de configuration des nœuds et les différentes opérations à exécuter sur chacun des nœuds.

La fonction se décrit comme suit :

#### **Etape 1** : Etablissement du nouveau sous-arbre

Configurer en parallèle les ports des nœuds situés sur le nouveau sous-arbre, entre la racine et les nœuds feuilles. L'opération d'ajout de commutation de longueur d'onde est utilisée sur chaque nœud (sauf les nœuds convergents de la paire et la racine). L'opération de convergence est utilisée sur les nœuds feuilles. On configure avec l'opération de convergence les nœuds convergents appartenant aux sous-arbres pour commuter le flux vers tous leurs canaux optiques avals sauf sur ceux appartenant exclusivement à l'ancien sous-arbre. Cette étape permet d'établir le nouveau sous-arbre.

#### **Etape 2** : Basculement du flux de l'ancien sous-arbre au nouveau sous-arbre

Configurer avec l'opération de basculement la racine de la paire de sous-arbres pour basculer le flux sur le nouveau sous-arbre et l'interrompre sur l'ancien.

### Etape 3 : Suppression de l'ancien sous-arbre

Configurer en parallèle avec l'opération de suppression de commutation les nœuds situés entre la racine et les nœuds feuilles sur l'ancien sous-arbre pour le supprimer. L'opération de suppression de commutation est utilisée sur chaque nœud (sauf les nœuds feuilles). L'opération de non-convergence est utilisée sur les nœuds feuilles.

#### Algorithme 6.3: Algorithme de reconfiguration d'une paire de sous-arbre

---

*Configure(P\_of\_Sub\_T, Conf\_Order)*

1. {
  2. Configurer en parallèle les nœuds de la paire du nouveau sous-arbre de *P\_of\_Sub\_T*.  
L'opération d'ajout de commutation est utilisée (sauf pour les nœuds feuilles et la racine).  
L'opération de convergence est utilisée sur les nœuds feuilles.
  3. Exécuter l'opération de basculement sur le nœud racine
  4. Configurer en parallèle avec l'opération de suppression les ports des nœuds entre la racine et chaque nœud feuille de l'ancien sous-arbre pour supprimer l'ancien sous-arbre de *P\_of\_Sub\_T*.
  5. }
- 

## 6.9 Evaluations

Nous présentons les différents critères d'évaluation des algorithmes, les paramètres de la simulation et les résultats obtenus.

### 6.9.1 Critères d'évaluation

Trois critères d'évaluation sont définis pour évaluer l'algorithme. Le premier est la durée d'interruption dans une séquence de reconfiguration. La durée d'interruption est proportionnelle au nombre d'étapes ne contenant pas d'arbre couvrant la source et tous les nœuds destinataires. Le second est la durée d'une séquence de reconfiguration. La durée d'une séquence de reconfiguration est proportionnelle au nombre d'étapes d'une séquence de reconfiguration. Le troisième est le coût des ressources additionnelles. C'est la somme des ressources additionnelles de chaque étape de la séquence de reconfiguration.

### 6.9.2 Description de la simulation

Les topologies COST 239, NSF Network et US Backbone Network sont utilisées pour les évaluations de l'algorithme.

## 6. Reconfiguration par sous-arbre du routage multicast

Nous sélectionnons de façon aléatoire deux arbres comme arbre initial et arbre final du problème de reconfiguration. L'arbre des plus courts chemins (SPT) et l'arbre minimum de Steiner (MST) représentent les deux arbres.

La source de la connexion multicast est choisie de façon aléatoire (loi de distribution uniforme) parmi les nœuds du réseau. Le nombre de destinations du groupe multicast est déterminé de façon aléatoire ((loi de distribution uniforme) parmi les nœuds restants du réseau. L'algorithme de Dijkstra est appliqué entre le nœud source et chaque destination pour calculer l'arbre des plus courts chemins (SPT). L'algorithme de Prim est utilisé pour calculer une approximation de l'arbre minimum de Steiner (MST).

Dans la première partie des évaluations, l'arbre initial est l'arbre des plus courts chemins (SPT), et l'arbre final est l'approximation de l'arbre de Steiner (MST). Dans la seconde partie des simulations, nous échangeons les algorithmes permettant de calculer les arbres. L'arbre initial devient l'approximation de l'arbre de Steiner et l'arbre final devient l'arbre des plus courts chemins. Nous faisons les évaluations sur plusieurs itérations de problèmes. Ensuite, nous déterminons les moyennes de chaque métrique d'évaluation définie. Le nombre d'itérations est fixé à 5000.

### 6.9.3 Résultats et analyses

Les tableaux 6.1 à 6.5 présentent les résultats des évaluations. Les tableaux 6.1 à 6.3 présentent les résultats obtenus lorsque l'arbre initial est l'arbre des plus courts chemins entre la source et chaque destination du groupe multicast et l'arbre final est l'approximation de l'arbre minimum de Steiner.

Tableau 6.1: Durée d'interruption du processus de reconfiguration

| CONF. PROCESS    | COST 239   |           |                | NSF NETWORK |           |                | US BACKBONE |           |                |
|------------------|------------|-----------|----------------|-------------|-----------|----------------|-------------|-----------|----------------|
|                  | AVG (STEP) | SD (STEP) | MIN/MAX (STEP) | AVG (STEP)  | SD (STEP) | MIN/MAX (STEP) | AVG (STEP)  | SD (STEP) | MIN/MAX (STEP) |
| <i>BpBAR_2</i>   | 0          | 0         | 0/0            | 0           | 0         | 0/0            | 0           | 0         | 0/0            |
| <i>STBReconf</i> | 0          | 0         | 0/0            | 0           | 0         | 0/0            | 0           | 0         | 0/0            |
| <i>MBB_1</i>     | 9,63       | 5,60      | 0/18           | 13,18       | 7,16      | 0/24           | 21,97       | 12,20     | 0/46           |

Tableau 6.2: Durée du processus de reconfiguration

| CONF. PROCESS    | COST 239   |           |                | NSF NETWORK |           |                | US BACKBONE |           |                |
|------------------|------------|-----------|----------------|-------------|-----------|----------------|-------------|-----------|----------------|
|                  | AVG (STEP) | SD (STEP) | MIN/MAX (STEP) | AVG (STEP)  | SD (STEP) | MIN/MAX (STEP) | AVG (STEP)  | SD (STEP) | MIN/MAX (STEP) |
| <i>BpBAR_2</i>   | 20,75      | 10,37     | 6/43           | 22,15       | 13,29     | 6/68           | 57,24       | 34,85     | 6/130          |
| <i>STBReconf</i> | 16,68      | 10,21     | 3/42           | 16,56       | 10,79     | 3/54           | 41,98       | 22,77     | 3/102          |
| <i>MBB_1</i>     | 12,79      | 5,42      | 3/21           | 16,25       | 7,06      | 3/27           | 25,07       | 12,11     | 3/50           |

Tableau 6.3: Coût additionnel du processus de reconfiguration

| CONF. PROCESS    | COST 239    |            |                 | NSF NETWORK |            |                 | US BACKBONE |            |                 |
|------------------|-------------|------------|-----------------|-------------|------------|-----------------|-------------|------------|-----------------|
|                  | AVG<br>(WL) | SD<br>(WL) | MIN/MAX<br>(WL) | AVG<br>(WL) | SD<br>(WL) | MIN/MAX<br>(WL) | AVG<br>(WL) | SD<br>(WL) | MIN/MAX<br>(WL) |
| <i>BpBAR_2</i>   | 94,04       | 126,9      | 4/640           | 203,5       | 302,48     | 4/1712          | 1911,71     | 2063,97    | 6/10793         |
| <i>STBReconf</i> | 33,65       | 28,69      | 0/100           | 56,48       | 45,40      | 0/215           | 117,72      | 80,48      | 0/315           |
| <i>MBB_1</i>     | 0           | 0          | 0/0             | 0           | 0          | 0/0             | 0           | 0          | 0/0             |

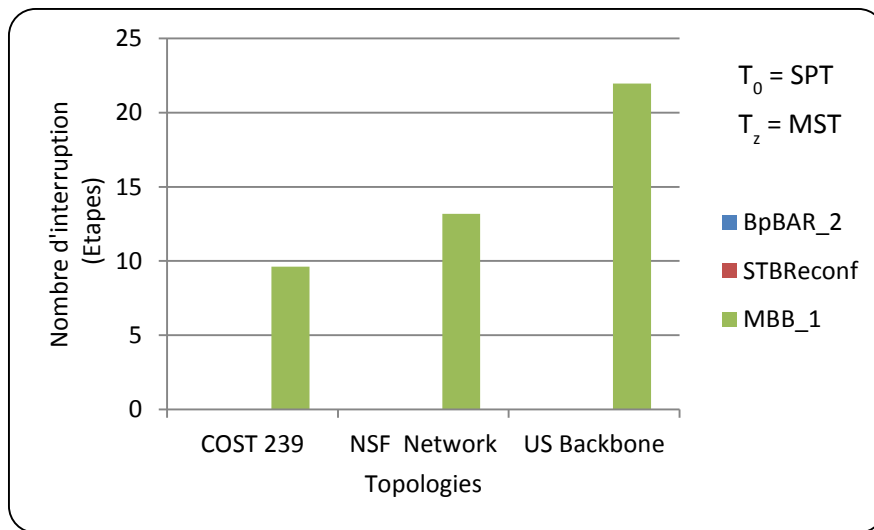


Fig. 6.4 : Durée d'interruption de chaque algorithme

L'algorithme proposé *STBReconf* est comparé aux algorithmes *BpBAR\_2* et *MBB\_1*. Le premier critère de comparaison est la durée d'interruption durant la reconfiguration. La contrainte principale que doit respecter l'algorithme proposé est la continuité du flux vers toutes les destinations durant la reconfiguration. La figure 6.4 présente la durée d'interruption de chaque algorithme de reconfiguration. *MBB\_1* produit des interruptions dans certaines étapes du processus tandis que *STBReconf* et *BpBAR\_2* ne produisent pas d'interruption durant tout le processus (Tableau 6.1). *STBReconf* et *BpBAR\_2* reconfigurant les arbres optiques sans interruption de flux, nous comparons d'une part la durée de reconfiguration produite par chacun d'eux et d'autre part le coût des ressources additionnelles utilisées.

Les tableaux 6.4 et 6.5 présentent les résultats obtenus lorsque l'arbre initial est l'approximation de l'arbre minimum de Steiner et l'arbre final est l'arbre des plus courts chemins entre la source et chaque destination du groupe multicast. Les figures 6.5 et 6.6 présentent la durée de reconfiguration produite par les algorithmes *BpBAR\_2* et *STBReconf*.

## 6. Reconfiguration par sous-arbre du routage multicast

Tableau 6.4: Durée du processus de reconfiguration

| CONF. PROCESS    | COST 239   |           |                | NSF NETWORK |           |                | US BACKBONE |           |                |
|------------------|------------|-----------|----------------|-------------|-----------|----------------|-------------|-----------|----------------|
|                  | AVG (STEP) | SD (STEP) | MIN/MAX (STEP) | AVG (STEP)  | SD (STEP) | MIN/MAX (STEP) | AVG (STEP)  | SD (STEP) | MIN/MAX (STEP) |
| <i>BpBAR_2</i>   | 18,21      | 12,40     | 6/61           | 19,24       | 14,77     | 6/76           | 58,48       | 34,10     | 6/153          |
| <i>STBReconf</i> | 16,61      | 10,13     | 3/42           | 15,86       | 09,90     | 3/48           | 43,00       | 22,32     | 3/102          |

Tableau 6.5: Coût additionnel du processus de reconfiguration

| CONF. PROCESS    | COST 239 |         |              | NSF NETWORK |         |              | US BACKBONE |         |              |
|------------------|----------|---------|--------------|-------------|---------|--------------|-------------|---------|--------------|
|                  | AVG (WL) | SD (WL) | MIN/MAX (WL) | AVG (WL)    | SD (WL) | MIN/MAX (WL) | AVG (WL)    | SD (WL) | MIN/MAX (WL) |
| <i>BpBAR_2</i>   | 47,96    | 55,15   | 2/319        | 90,55       | 119,35  | 2/609        | 924,54      | 930,94  | 2/4358       |
| <i>STBReconf</i> | 29,94    | 26,36   | 0/100        | 54,13       | 43,00   | 0/195        | 120,64      | 85,17   | 0/495        |

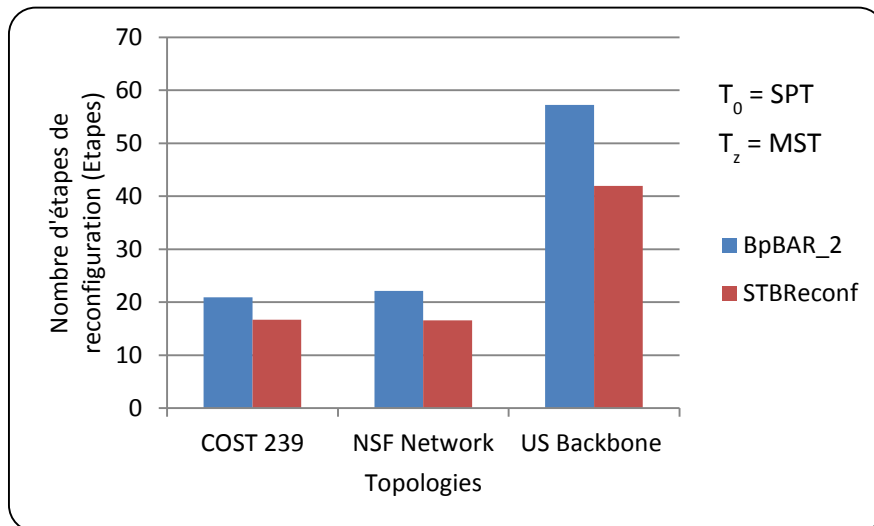


Fig. 6.5 : Durée du processus de reconfiguration de chaque algorithme.

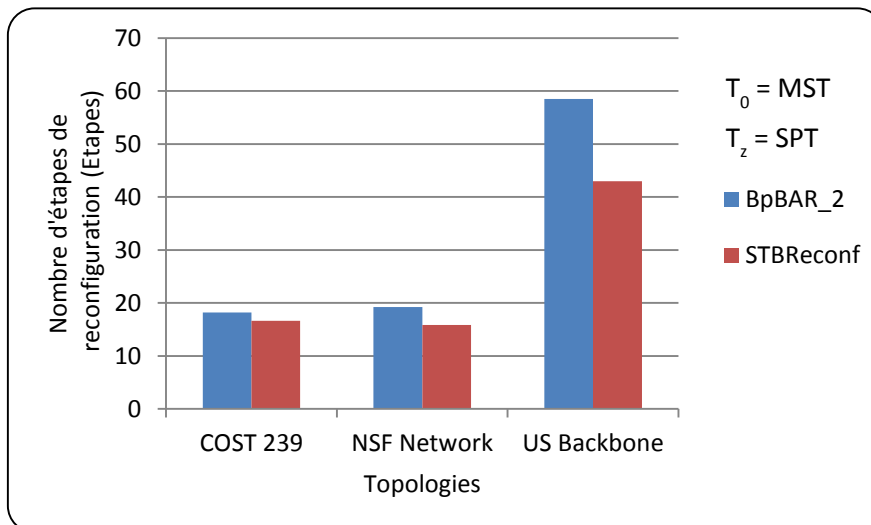


Fig. 6.6 : Durée du processus de reconfiguration avec chaque algorithme.

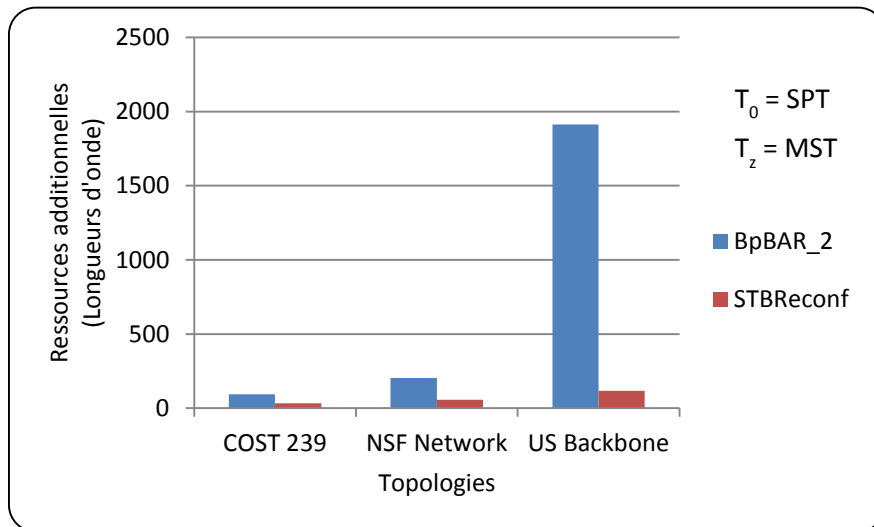


Fig. 6.7 : Coût additionnel de chaque algorithme.

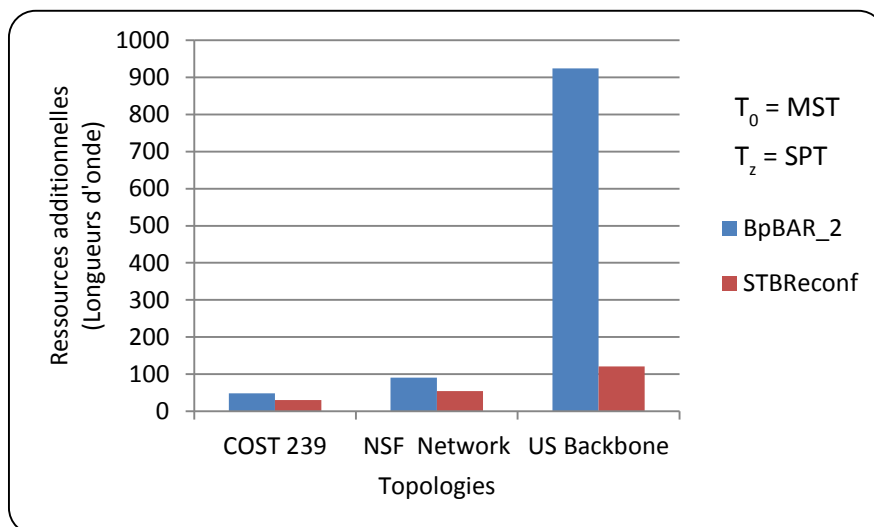


Fig. 6.8 : Coût additionnel de chaque algorithme.

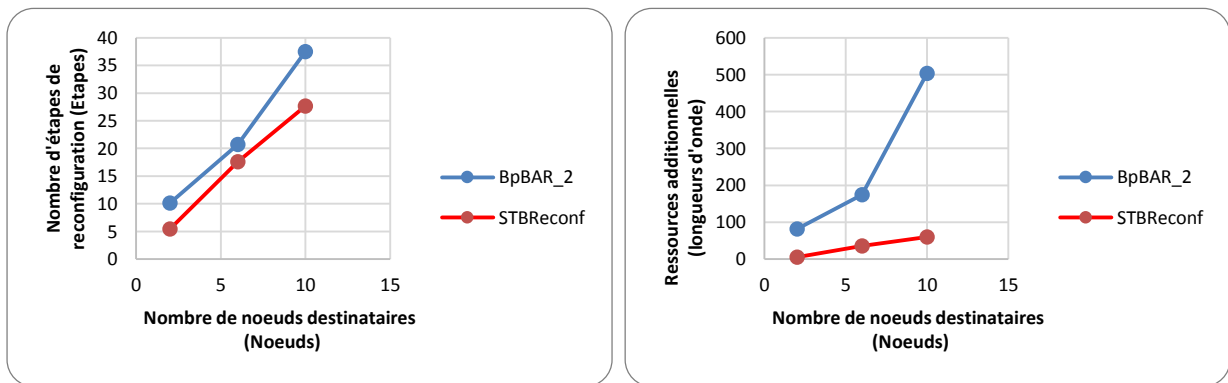
Dans la Fig. 6.5, l'arbre initial est l'arbre des plus courts chemins entre la source et chaque destination et l'arbre final est l'approximation de l'arbre minimum de Steiner. Dans la Fig. 6.6, l'arbre initial est l'approximation de l'arbre minimum de Steiner et l'arbre final est l'arbre des plus courts chemins entre la source et chaque destination. Dans les topologies utilisées, de caractéristiques différentes, la reconfiguration avec *STBReconf* est plus rapide que *BpBAR\_2*. Cette amélioration de la durée de reconfiguration est due au grand nombre de nœuds configurés en parallèle. Lorsque la paire de sous-arbres à reconfigurer contient plusieurs branches, cela augmente le nombre de nœuds à configurer en parallèle. Afin de sélectionner une paire de sous-arbres couvrant plusieurs nœuds à configurer, la fonction de calcul de paires de sous-arbres détermine en premier les sous-ensembles de nœuds convergents candidats ayant le plus grand cardinal. Lorsque nous échangeons les algorithmes permettant le calcul des arbres initial et



## 6. Reconfiguration par sous-arbre du routage multicast

final du problème, la reconfiguration avec *STBReconf* est toujours plus rapide que *BpBAR\_2* dans les topologies de l'évaluation (Fig. 6.6).

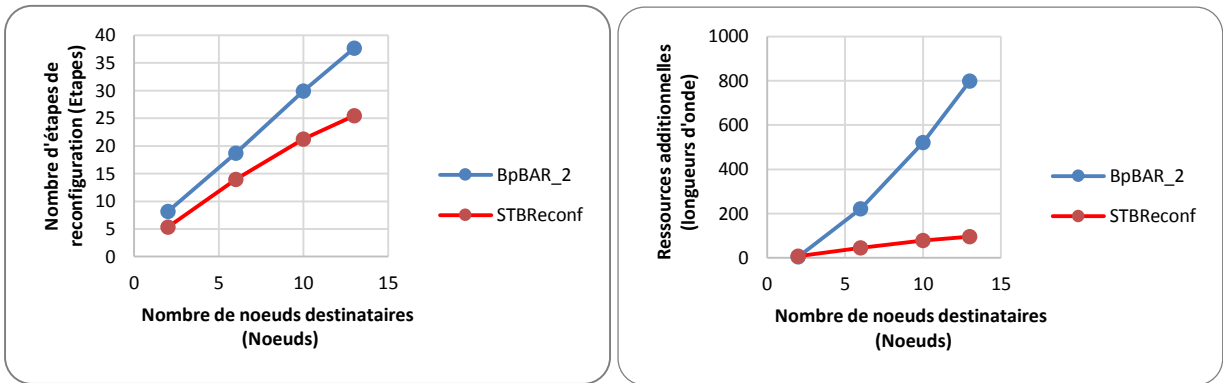
Les figures 6.7 et 6.8 montrent le coût des ressources additionnelles utilisées par les algorithmes *BpBAR\_2* et *STBReconf*. Dans la Fig. 6.7, l'arbre initial est l'arbre des plus courts chemins entre la source et chaque destination et l'arbre final est l'approximation de l'arbre minimum de Steiner. Dans la Fig. 6.8, l'arbre initial est l'approximation de l'arbre minimum de Steiner et l'arbre final est l'arbre de plus court chemin entre la source et chaque destination. Le gain obtenu avec l'algorithme *STBReconf* est important quelle que soit la topologie utilisée. La reconfiguration par sous-arbre utilise alors mieux les ressources disponibles durant le processus de reconfiguration que la reconfiguration par branche. Cette assertion est vérifiée lorsque nous échangeons les algorithmes permettant de calculer l'arbre initial et l'arbre final (Fig. 6.8). Cette réduction du coût des ressources additionnelles est due au fait que la reconfiguration par sous-arbre détermine et reconfigure en priorité les paires de sous-arbres pouvant être reconfigurées sans ressource additionnelle. Alors si plusieurs branches des arbres ont été reconfigurées sans longueur d'onde additionnelle, cela réduit le coût des ressources additionnelles utilisées dans l'ensemble du processus de reconfiguration.



a) Evolution de la durée de reconfiguration    b) Evolution du coût de ressources additionnelles

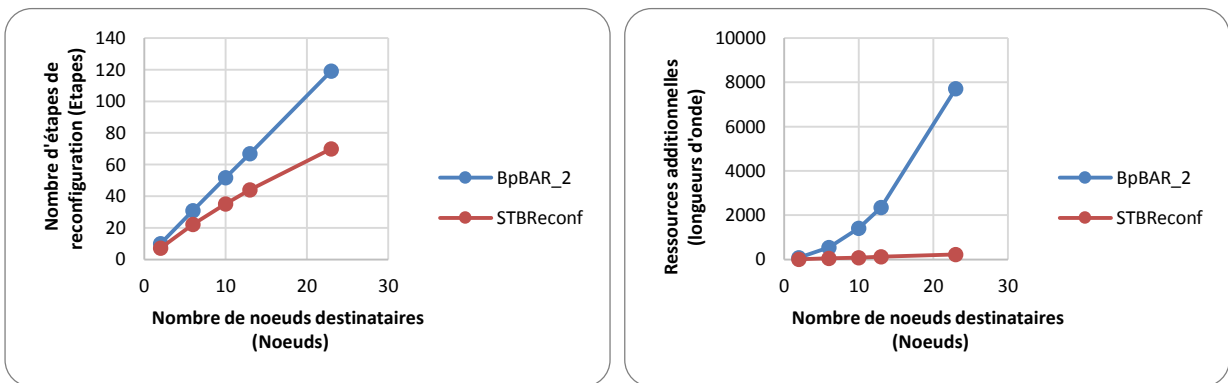
Fig. 6.9 : Evolution de la durée de reconfiguration et du coût de ressources additionnelles en fonction du nombre de nœuds destinataires dans la topologie COST 239

## 6. Reconfiguration par sous-arbre du routage multicast



a) Evolution de la durée de reconfiguration    b) Evolution du coût de ressources additionnelles

Fig. 6.10 : Evolution de la durée de reconfiguration et du coût de ressources additionnelles en fonction du nombre de nœuds destinataires dans la topologie NSF Network



a) Evolution de la durée de reconfiguration    b) Evolution du coût de ressources additionnelles

Fig. 6.11 : Evolution de la durée de reconfiguration et du coût de ressources additionnelles en fonction du nombre de nœuds destinataires dans la topologie US Backbone

Les figures 6.9 à 6.11 présentent l'évolution de la durée de reconfiguration et du coût additionnel de ressources en fonction du nombre de destinataires du groupe multicast. Les figures 6.9, 6.10 et 6.11 présentent les résultats respectivement dans la topologie COST 239, dans la topologie NSF Network et dans la topologie US Backbone. Les figures 6.9.a, 6.10.a et 6.11.a montrent que quel que soit le nombre de destinataires de la connexion multicast, l'algorithme proposé est plus rapide que l'algorithme de reconfiguration par branche. Nous constatons également que lorsque le nombre de destinataires augmente, le gain en durée de reconfiguration également grandit. Les figures 6.9.b, 6.10.b et 6.11.b montrent que *STBReconf* utilise moins de ressources additionnelles que l'algorithme *BpBAR\_2*. Cela est vrai quel que soit le nombre de nœuds destinataires de la connexion et la topologie considérée. Ces résultats montrent que le gain en coût additionnel de ressource est très bon lorsque le nombre de destinataires est grand.

Finalement la reconfiguration par sous-arbre a de bonnes performances : elle maintient la continuité du flux durant la reconfiguration et utilise mieux les ressources du réseau. Elle améliore également la durée de la reconfiguration.

### 6.10 Conclusion

Ce chapitre présente une nouvelle approche de reconfiguration d'arbre optique sans interruption de flux vers les destinations du groupe multicast. L'approche proposée est la reconfiguration par sous-arbre. Elle consiste à déterminer et reconfigurer une paire de sous-arbres constituée d'un sous-arbre de l'arbre courant et d'un sous-arbre de l'arbre final. La paire de sous-arbres est déterminée de telle sorte que sa reconfiguration ne nécessite pas de ressource additionnelle. De plus, les paires sont déterminées de telle sorte qu'elles contiennent plusieurs nœuds à reconfigurer en parallèle afin de réduire la durée du processus. Les résultats des simulations montrent que la reconfiguration par sous-arbre a de bonne performance. Elle réduit de façon considérable le coût des ressources additionnelles utilisées durant les processus. La durée du processus est également très bonne.

## 7. Conclusion et perspectives

---

### Sommaire

---

|                       |     |
|-----------------------|-----|
| 7.1 Conclusion.....   | 140 |
| 7.2 Perspectives..... | 141 |

---

### 7.1 Conclusion

Cette thèse s'est intéressée à la reconfiguration du routage multicast dans les réseaux optiques à multiplexage en longueur d'onde. Le routage multicast permet la communication d'une source vers plusieurs destinataires et est très économe en termes de bande passante utilisée. Le travail que nous avons effectué a pour objectif de proposer des algorithmes de reconfiguration qui permettent de basculer le flux d'un routage initial sur un routage final pré-calculé. La reconfiguration du routage telle qu'étudiée consiste à déterminer un ordre de configuration des nœuds du réseau qui garantit la continuité du flux vers tous les destinataires de la connexion. La reconfiguration étant déclenchée pour faire face à des phénomènes perturbateurs du réseau, doit s'effectuer de façon rapide afin de répondre efficacement à l'événement déclencheur. Dans cette thèse, nous avons donc étudié la reconfiguration rapide sans interruption de flux du routage multicast dans les réseaux optiques WDM.

Les techniques de reconfiguration existantes dans la littérature sont *MTV*, *MBB* et *BBM*. Ces techniques sont à l'origine proposées pour la reconfiguration de connexion unicast. Dans cette thèse, nous avons montré que l'extension des techniques de reconfiguration existantes pour la reconfiguration du routage multicast sans interruption de flux s'avère inefficace. En effet, elles produisent des interruptions de flux pour certaines instances du problème de reconfiguration.

Nous avons prouvé, dans une première partie, qu'il est impossible de reconfigurer n'importe quel arbre optique initial choisi en n'importe quel autre arbre optique final choisi en utilisant une seule longueur d'onde et sans interruption de flux. Nous avons proposé dans la suite des algorithmes qui utilisent des longueurs d'onde additionnelles. Deux approches de

reconfiguration sans interruption de flux ont été proposées: une approche de reconfiguration par branche et une approche de reconfiguration par sous-arbre. Les performances de ces approches ont été évaluées dans différentes topologies communément utilisées dans ce domaine. Ces algorithmes permettent de faire la reconfiguration sans interrompre le flux de la connexion multicast tout en réduisant la durée du processus et les ressources optiques additionnelles utilisées. L'approche de reconfiguration par sous-arbre est plus adaptée aux problèmes de reconfiguration du routage multicast dont l'arbre optique initial et l'arbre optique final ont plusieurs liens en commun.

### 7.2 Perspectives

Le problème traité dans cette thèse est constitué d'un arbre initial et d'un arbre final sur lequel on veut basculer le flux. Dans les réseaux, plusieurs connexions peuvent partager un même lien physique. Une perspective de ce travail est de considérer le problème constitué d'un ensemble de chemins initiaux et un ensemble de chemins finaux. Le problème serait alors de déterminer un ordre de configuration des différents ports des nœuds de ces deux ensembles afin de basculer le flux des chemins initiaux vers les chemins finaux sans l'interrompre tout réduisant la durée du processus et les ressources additionnelles utilisées.

Nous avons proposé des algorithmes qui reconfigurent un arbre initial en un arbre final sans interrompre le flux optique. Mais ces algorithmes ne donnent pas la durée optimale de reconfiguration ou le coût optimal des ressources additionnelles. On pourrait s'intéresser à la détermination de la durée optimale et du coût optimal de reconfiguration à partir des algorithmes de reconfiguration proposés.

Une autre perspective est d'étudier l'architecture des commutateurs pouvant supporter les opérations de convergence et de non-convergence.

## Bibliographie

- [AO12] J. Adépo, S. Oumtanaga, "Reconfiguration du routage multicast," *Colloque Africain sur la recherche en Informatique et Mathématiques appliquées (CARI)*, 2012.
- [Bal97] A. Ballardie, Core Based Trees (CBT version2) Multicast Routing Protocol specification, *RFC 2189*, 1997.
- [CABO14] 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.
- [CAOB12] 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, pp. 85–95, 2012.
- [Bea00] B. Beauquier, *Communications dans les réseaux optiques par multiplexage en longueur d'onde*, Thèse de doctorat, 2000.
- [Blo13] Sofiene BLOUZA, *Etude des potentialités offertes par les technologies de transmission optiques flexibles pour les réseaux métro/cœur*, Thèse de doctorat, 2013.
- [BM00] D. Banerjee and B. Mukherjee, "Wavelength-routed optical networks: Linear formulation, resource budgeting tradeoffs, and a reconfiguration study," *IEEE/ACM Transactions on Networking*, 8(5), pp. 598–607, 2000.
- [Bor97] M. S. Borella et al., "Optical Components for WDM Lightwave Networks," *Proc. IEEE*, vol. 85, no. 8, pp. 1274–1307, 1997.
- [BR01] 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.
- [BR99] I. Baldine and G. N. Rouskas, "Dynamic reconfiguration policies for WDM networks," in: *18<sup>th</sup> Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 1, pp. 313–320, 1999.

- [CBL07] X. Chu, T. Bu, and X.-Y. Li, “A study of lightpath rerouting schemes in wavelength-routed wdm networks,” in: *Proceedings of IEEE International Conference on Communications (ICC)*, pp. 24002–405, 2007.
- [CCMNN10] N. Cohen, D. Coudert, D. Mazauric, N. Nepomuceno, and N. Nisse, “Tradeoffs in routing reconfiguration problems,” in: *12<sup>th</sup> International Symposium on Algorithmic Aspects of Telecommunications (AlgoTel)*, 2010.
- [CHS09] D. Coudert, F. Hue, and J-S. Sereni, “Reconfiguration of the routing in WDM networks with two classes of services,” in: *13<sup>th</sup> International Conference on Optical Network Design and Modeling (ONDM)*, pp. 1–6, 2009.
- [CLT08] M.-T. Chen, B.M.T. Lin, S.-S. Tseng, “Multicast routing and wavelength assignment with delay constraints in WDM networks with heterogeneous capabilities,” *Journal of Network and Computer Applications*, Vol. 31, pp. 47–65, 2008.
- [CM07] B. Cousin, M. Molnar, “Fast reconfiguration of dynamic networks,” *International Workshop on Dynamic Networks*, 2007.
- [Cou01] D. Coudert. *Algorithmique et optimisation de réseaux de communications optiques*. Thèse de Doctorat, Université de Nice Sophia-Antipolis (UNSA), 2001.
- [Dij57] E. W. Dijkstra, “A note on two problems in connection with graphs,” *NumerischeMathematik*, vol. 1, no. 1, pp. 269–271, 1957.
- [Dor02] A. Doria et al. “General switch management protocol (GSMP) V3,” *Internet RFC 3292*, 2002.
- [DP03] A. Ding, G-S Poo, “A survey of optical multicast over WDM networks,” *Computer Communications*, Vol. 26, no. 2, pp. 193–200, 2003.
- [DY12] R.D. Doverspike, J. Yates, “Optical network management and control,” *Proc IEEE*, 100(5), pp. 1092–104, 2012.
- [Fen97] W. Fenner, “Internet Group Management Protocol, Version 2,” *RFC 2236*, 1997.
- [FHH06] B. Fenner, M. Handley, H. Holbrook, I. Kouvelas, “Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)”, *RFC 4601*, 2006.

- [FVA06] A. Farrel, J-P. Vasseur, and J. Ash, “A path computation element (PCE)-based architecture,” *Request for Comments 4655*, 2006.
- [GB04] W. Golab, R. Boothbay, “WDM optical network reconfiguration using automated regression-based parameter value selection,” in: *3rd International Conference on Networking*, 2004.
- [Gea02] N. Geary et al., “The benefits of reconfiguration in optical networks,” in: *10th International Telecommunication Network Strategy and Planning Symposium*, 2002.
- [GM03] A. Gencata and B. Mukherjee, “Virtual-topology adaptation for WDM mesh networks under dynamic traffic,” *IEEE/ACM Transactions on Networking*, 11(2) pp. 236–247, 2003.
- [He14] J. He et al., “A survey on recent advances in optical communications, Computers and Electrical Engineering,” Vol. 40, pp. 216–240, 2014.
- [HCT02] J. He, S.H.G. Chan, and D.H.K. Tsang, “Multicasting in WDM networks,” *IEEE Communications Surveys and Tutorials*, 2002.
- [HD06] R. Hinden, S. Deering, “IP version 6 Addressing Architecture,” *RFC 4291*, 2006.
- [HFZJ04] X. Huang, F. Farahmand, T. Zhang, and J. P. Jue, “Dynamic multicast traffic grooming in WDM networks with reconfigurable light-trees,” in: *Optical Fiber Communication Conference*, 2004.
- [HRW92] F.K. Hwang, D.S. Richards, P. Winter, *The Steiner Tree Problem*, Elsevier Science Publishers, Amsterdam, 1992.
- [HSZ04] X.-D. Hu, T.-P. Shuai, X. Jia, M.-H. Zhang, “Multicast Routing and Wavelength Assignment in WDM Networks with Limited Drop-offs,” *In: proceedings of IEEE INFOCOM*, 2004.
- [IM03] M. Ilyas, H. T. Mouftah, *handbook of optical communication networks*, CRC Press LLC, 2003.
- [JDHLG01] X.-H. Jia, D.-Z. Du, X.-D. Hu, M.-K. Lee, J. Gu, Optimization of wavelength assignment for QoS multicast in WDM networks, *IEEE Transactions on Communications* 49 (2) (2001) 341–350.



- [Kar06] T. Kárász, “Consolidation strategies of provisioning oriented optical networks,” *Journal of Optical Networking*, vol. 5, no. 6, pp. 445–462, 2006.
- [Kar72] R. Karp, “Reducibility among combinatorial problems,” in: *Complexity of Computer Computations*, 1972.
- [KP05] T. Kárász and Z. Pándi, “Optimal reconfiguration of provisioning oriented optical networks,” in: *3<sup>rd</sup> International Working Conference on Performance Modeling and Evaluation of Heterogeneous Networks (HET-NETs)*, 2005.
- [LL96] K.-C. Lee and V.O.K. Li, “A wavelength rerouting algorithm in wide-area all optical networks,” *IEEE/OSA Journal of Lightwave Technology*, 14(6), pp. 1218–1229, 1996.
- [LM04] Y. Lee, B. Mukherjee, “Traffic engineering in next-generation optical networks,” *IEEE Communications Surveys and Tutorials*, 6 (3), pp. 16–33, 2004.
- [LTMY13] L. Liu, T. Tsuritani, I. Morita, S.J.B. Yoo, “Optical network control and management technology using OpenFlow,” in: *OptoElectronics and Communications Conference*, 2013.
- [LWK12] S. Lee, T. Wong, and H. S. Kim, “Improving manageability through reorganization of routing-policy configurations,” *Computer Networks*, Vol. 56, no. 14, pp. 3192–3205, 2012.
- [Maz11] D. Mazauric, *Optimisation discrète dans les réseaux de télécommunication : reconfiguration du routage, routage efficace en énergie, ordonnancement de liens et placement de données*, thèse de Doctorat, 2011.
- [Muk06] B. Mukherjee, *Optical WDM Networks*, Springer, 2006.
- [MM99] G. Mohan and C.S.R. Murthy, “A time optimal wavelength rerouting algorithm for dynamic traffic in WDM networks,” *IEEE/OSA Journal of Lightwave Technology*, 17(3), 406–417, 1999.
- [Muk00] B. Mukherjee, “WDM Optical Communication Network: Progress and Challenges”, *IEEE Journal on Selected Areas in Communication*, Vol. 18 No. 10, pp. 1810–1824, 2000.

- [MZQ98] R. Malli, X. Zhang, C. Qiao, “Benefit of multicasting in all-optical networks,” *SPIE All Optical Networking*, Vol. 2531, pp. 209–220, 1998.
- [PPJ11] Subharthi Paul, Jianli Pan, Raj Jain, “Architectures for the future networks and the next generation Internet: A survey,” *Computer Communications*, Vol. 34, pp. 2–42, 2011.
- [Per05] H-G. Perros. “Connection-oriented networks”, 2005.
- [Pie15] Pierre Lecoy, “Communications sur fibre optique”, Lavoisier hermes, 2015.
- [PCL08] M. Perényi, P. Soproni, T. Cinkler, and D. Larrabeiti, “Regular reconfiguration of light-trees in multilayer optical networks,” in: *12<sup>th</sup> International Conference on Optical Networking Design and Modeling (ONDM)*, pp. 1–6, 2008.
- [Pri57] R. C. Prim, “Shortest connection networks and some generalizations,” *Bell System Technical Journal*, vol. 36, pp. 1389–1401, 1957.
- [Puj14] G. Pujolle; *Les Réseaux*; Eyrolles, 2014.
- [Ram06] R. Ramaswami, “Optical networking technologies: what worked and what didn’t,” *IEEE Communications Magazine*, Vol. 44, No, 9, pp.132–139, 2006.
- [RSS10] R. Ramaswami and K.N. Sivarajan, G.H Sasaki, *Optical Networks: A Practical Perspective, Third Edition*, Morgan Kaufmann, 2010.
- [RSVA10] G. Ramesh, S. Sundara Vadivelu, Jose Anand, “A Survey on Wavelength Division Multiplexing (WDM) Networks,” *ICTACT Journal Of Communication Technology*, Issue 01, 2010.
- [Rou03] G.N. Rouskas, “Optical layer multicast: rationale, building blocks, and challenges,” *Network, IEEE*, 17(1) pp. 60–65, 2003.
- [SA98] G. Sahin, M. Azizoglu, “Multicast routing and wavelength assignment in wide area networks, SPIE Conference on All-Optical Networking: Architecture, Control, and Management,” *SPIE*, 3531 (1998), pp. 196–208.
- [SM99] L.H. Sahasrabudhe, B. Mukherjee, “Light trees: optical multicasting for improved performance in wavelength routed networks,” *Communications Magazine, IEEE*, 37(2), pp. 67–73, 1999.

- [SZ05] M. Saad and Zhi-Quan Luo, "Reconfiguration with no service disruption in multifiber WDM networks," *IEEE/OSA Journal of Lightwave Technology*, 23(10), pp. 3092–3104, 2005.
- [Vac07] J-R Vacca, *Optical Networking Best Practices Handbook*, Wiley, 2007.
- [Web1] Alcatel: Optical Networks Tutorial: Index, <http://www.webproforum.com/alcatel/full.html>.
- [Web2] Web Proforurm tutorial, <http://www.iec.org>.
- [Wei02] J.Y Wei, "Advances in the management and control of optical internet," *IEEE Journal on Selected Areas in Communications*, Vol. 20, No. 4, pp.768–785, 2002.
- [Wu11] J. Wu, "A survey of WDM network reconfiguration: strategies and triggering methods," *Computer Networks*, vol. 55, no. 11, pp. 2622–2645, 2011.
- [YT02] M. Yoshida et H. Tanaka, "The characteristics of WDM systems with hybrid AGC EDFA," in: *the photonics network, chez OFC, Itami, Japan*, 2002.
- [Zha10] Y. Zhang et al., "Energy Efficiency in Telecom Optical Networks," *IEEE communications surveys & tutorials*, vol. 12, no. 4, 2010.
- [Zho10] F. Zhou. *Routage multicast tout optique dans les réseaux WDM*, Thèse de Doctorat, 2010.
- [ZJM00] H. Zang, J. P. Jue, and B. Mukherjee, "A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks," *Optical Network Magazine*, Vol. 1, No. 1, pp. 47–60, 2000.
- [ZWQ00] X. Zhang, J. Wei, C. Qiao, "Constrained multicast routing in WDM networks with sparse light splitting," *IEEE/OSA Journal of Lightwave Technology*, Vol. 18, pp. 1917–1927, 2000.
- [ZYWS07] J. Y. Zhang, O. W. W. Yang, J. Wu, M. Savoie, "Optimization of semidynamic lightpath rearrangements in a WDM network," *IEEE Journal on Selected Areas in Communications*, 25(9), pp. 31–7, 2007.