HAL
open science

Reconfiguration du Routage Multicast dans les Réseaux Optiques WDM ayant la Propriété de Conversion Partielle de Longueur d'onde

Amanvon Ferdinand Atta

To cite this version:

Amanvon Ferdinand Atta. Reconfiguration du Routage Multicast dans les Réseaux Optiques WDM ayant la Propriété de Conversion Partielle de Longueur d'onde. Informatique [cs]. Institut National Polytechnique Félix Houphouët Boigny de Yamoussoukro (Côte d'Ivoire), 2021. Français. NNT : . tel-03544920

HAL Id: tel-03544920

https://theses.hal.science/tel-03544920v1

Submitted on 27 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.

Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique

République de Côte d'Ivoire
Union - Discipline - Travail

Institut National Polytechnique
Félix HOUPHOUËT-BOIGNY

EDP
École Doctorale Polytechnique

LARIT
LABORATOIRE DE RECHERCHE EN INFORMATIQUE ET TELECOMMUNICATION

N° d'ordre : 100/2021

THÈSE UNIQUE

Pour l'obtention du grade de

Docteur de l'Institut National Polytechnique Félix Houphouët Boigny de Yamoussoukro

Mention : Sciences et Technologies

Spécialité : Informatique

ATTA Amanvon Ferdinand

Reconfiguration du Routage Multicast dans les Réseaux Optiques WDM ayant la Propriété de Conversion Partielle de Longueur d'onde

Soutenue le 09/12/2021 devant le jury :

M. BABRI Michel	Professeur Titulaire	Institut National Polytechnique Félix Houphouët Boigny, Côte d'Ivoire	Président du Jury
M. MOLNAR Miklos	Professeur Titulaire	Université de Montpellier, France	Rapporteur
M. PALANGA Eyouléki Tcheyi Gnadi	Maître de Conférences	Université de Lomé, Togo	Rapporteur
M. DOSSO Mouhamadou	Maître de Conférences	Université Félix Houphouët Boigny, Côte d'Ivoire	Examinateur
M. OUMTANAGA Souleymane	Professeur Titulaire	Institut National Polytechnique Félix Houphouët Boigny, Côte d'Ivoire	Directeur de thèse
M. COUSIN Bernard	Professeur Titulaire	Université de Rennes 1, France	Co-directeur de thèse

” Les joies de la découverte sont certainement les plus vives que l’esprit de l’homme puisse jamais ressentir ”

— Claude Bernard

DÉDICACE

A la mémoire de ma très chère mère N'DA Kouadio B. Antoinette.
A la mémoire de mon très cher père ATTA Kouamé.
A toute ma famille.
A tous ceux que j'aime et qui me sont chers.

REMERCIEMENTS

Ce manuscrit présente le travail de mes années de thèse, qui a rassemblé de nombreux acteurs à qui je souhaite rendre hommage.

Je tiens à témoigner en premier lieu, ma reconnaissance et mes remerciements distingués aux responsables de l'Institut National Polytechnique Félix Houphouët Boigny (INP-HB) de Yamoussoukro avec à sa tête le Directeur général dudit Institut M. KOFFI N'Guessan pour avoir accepté mon inscription dans son établissement. Je ne saurai oublier le Directeur de l'Ecole Doctorale Polytechnique (EDP) de l'INP-HB, Prof. YAO Kouassi Benjamin et le Directeur de l'UMRI 78, Prof. ZOUEU Thouakesseh Jérémie qui contribuent par leurs actions à la promotion de la recherche fondamentale et appliquée en Côte d'Ivoire.

Je tiens à exprimer toute ma reconnaissance à Prof. BABRI Michel, Directeur du Laboratoire de Recherche en Informatique et Télécommunication (LARIT/UMRI 78) de l'Institut National Polytechnique Félix Houphouët Boigny (INP-HB) de Yamoussoukro, pour m'avoir permis de travailler dans d'excellentes conditions dans ce laboratoire auquel j'appartiens.

Je remercie Prof. Miklos MOLNAR et Dr. PALANGA Eyouléki Tcheyi Gnadi pour les remarques et suggestions constructives effectuées en tant que les rapporteurs de cette thèse, ainsi qu'à Dr. DOSSO Mouhamadou d'avoir accepté d'examiner ma thèse et Prof. BABRI Michel d'avoir accepté de présider le Jury de ma thèse. J'adresse mes plus sincères remerciements à Prof. OUMTANAGA Souleymane pour m'avoir offert l'opportunité d'effectuer cette thèse dans de très bonnes conditions. Il a eu les mots qu'il fallait au bon moment pour me permettre de rester concentré sur le travail. Je tiens aussi à témoigner ma gratitude à Prof. Bernard COUSIN, en tant que co-directeur de thèse, qui a su par sa grande expérience, ses conseils et son sens de la motivation me guider dans la réalisation de ce travail. Je tiens à remercier également Dr. ADEPO Joël Christian pour les échanges fructueux et ses relectures pertinentes. Tous mes remerciements également à l'ensemble des membres du LARIT pour leurs critiques et suggestions qu'ils ont bien voulu faire sur mon travail.

Ma reconnaissance va également à Rennes Métropole et le Service de Coopé-

ration et d'Action Culturelle (SCAC) de l'ambassade de France en Côte d'Ivoire pour m'avoir octroyé des bourses qui ont permis de financer mes séjours à l'Institut de Recherche en Informatique et Systèmes Aléatoires (IRISA) de Rennes. Je remercie Prof. Jean Marc JEZEQUEL, Directeur de l'IRISA pour m'avoir permis de séjourner dans son institut. Je remercie également Prof. Xavier LAGRANGE, responsable de l'équipe ADOPNET de l'IRISA ainsi que tous les membres de cette équipe pour m'avoir accueilli dans cette équipe et pour les échanges bénéfiques auxquels j'ai eu droit. Je remercie Gaëlle TWORKOWSKI (Assistance d'équipes à l'IRISA) et N'TAKPE Juliette (chargé de bourse au SCAC) pour leurs assistances dans la prise en charge administrative de tous mes séjours à Rennes.

Je ne saurai oublier de remercier Antoine LEMAITRE, Héloïse MOREZ ainsi que tous les autres membres de l'aumônerie catholique de Beaulieu à Rennes, Olivier PICHON et tous les autres membres de la section Rennaise de l'Ordre de Malte de France, Emilie GESNYS, Luc MANOEUVRIER ainsi que tous les autres membres du Centre de Mobilité Internationale (CMI) de Rennes, Varvara GROSHEVA et tous autres les membres du Club Erasmus de Rennes, M.Yannick OSSUIH, Mme. Grâce OSSUIH et tous les membres de l'AMicale des Ivoiriens de Rennes (AMIR) pour m'avoir aidé à m'intégrer socialement durant mes séjours à Rennes.

Dernier point, mais non le moindre, je suis profondément et à jamais redevable à toute ma famille pour leur amour sans fin, leur soutien constant et leurs encouragements incommensurables tout au long de ma vie. Je suis fier de reconnaître leurs prières tout au long des années d'efforts vers cette thèse. Evidemment, je tiens à remercier ma compagne, AMANI Aya Anne, pour son soutien affectif, son amour et son indulgence durant ces années de thèse. Enfin, je tiens à remercier tous ceux qui ont contribué à la réalisation de cette thèse et je m'excuse si je ne peux pas les mentionner tous.

RÉSUMÉ

Les réseaux optiques munis des technologies de répartition en longueur d'onde (WDM : *Wavelength Division Multiplexing*) et de conversion de longueur d'onde représentent une solution performante pour les réseaux cœurs à très haut débit. Ils permettent de faire face aux exigences très élevées notamment en bande passante des applications telles que la télémédecine, l'apprentissage à distance, les calculs distribués, les jeux en ligne. Cependant, pour réduire les coûts de dimensionnement du réseau, il est assez pratique que seulement certains nœuds du réseau soient dotés de la capacité de conversion de longueur d'onde. Un tel réseau est appelé réseau optique WDM ayant la propriété de conversion partielle de longueur d'onde.

Dans les réseaux WDM, les opérateurs établissent des connexions pour leurs clients qui désirent utiliser des applications unicast ou multicast. Il est courant qu'un client, vu le caractère critique de son activité exige de l'opérateur une faible interruption voire une non-interruption de la livraison du flux optique qu'il génère. Par ailleurs, à la suite de la survenue d'évènement tel qu'une augmentation des demandes de connexions, l'opérateur réseau est amené à effectuer une reconfiguration du routage en vue d'améliorer les performances du réseau. Cependant, l'exigence de continuité de flux n'est pas toujours facile à respecter durant le processus de reconfiguration si cela n'est pas effectué avec soin. La reconfiguration devient encore plus délicate pour le routage multicast dans un réseau optique WDM ayant la propriété de conversion partielle de longueur d'onde. Les travaux existants abordent la reconfiguration du routage de connexions unicasts ou multicasts dans les réseaux ayant la propriété de conversion totale de longueur d'onde.

Dans cette thèse, nous traitons d'une part le problème de reconfiguration du routage d'une seule connexion multicast dans un réseau optique WDM ayant la propriété de conversion partielle de longueur d'onde et d'autre part le problème de reconfiguration du routage d'un ensemble de plusieurs connexions multicasts. Dans le premier cas, on désire migrer le flux d'un routage multicast initial vers un routage final sans interruption de flux. Dans le second cas, on désire migrer le flux optique du routage multicast initial vers le routage multicast final en générant un faible nombre d'interrup-

tions de flux. Notons que le routage multicast initial et le routage multicast final sont supposés être chacun un ensemble d'arbres mono-optiques.

La méthode notée Para_SbSRA a été proposée pour résoudre plus efficacement le problème de reconfiguration du routage relatif à une seule connexion multicast et les méthodes LSRA et LSRA_2 ont été proposées pour résoudre le problème de reconfiguration du routage relatif à un ensemble de connexions multicasts. L'efficacité de ces méthodes a été confirmée à travers des simulations.

Mots-clés : Reconfiguration, interruption de flux, routage multicast, réseau optique WDM, arbre mono-optique, conversion de longueur d'onde

ABSTRACT

Optical networks using Wavelength Division Multiplexing (WDM) and wavelength conversion technologies are an efficient solution for very high-speed core networks. They make it possible to cope with the very high bandwidth and short delay requirements of applications such as telemedicine, distance learning, distributed computing, and online games. However, to reduce the costs of network dimensioning, it is quite practical that only some nodes of the network are equipped with wavelength conversion capability. Such a network is called a sparse wavelength converter network.

In the WDM networks, operators establish optical connections for their customers who need to use unicast or multicast applications. It is common for a customer, given the critical nature of its business, to require from the operator little or no interruption in the delivery of the optical flow it generates. In addition, following the occurrence of events such as an increase in connection requests, the network operator could be required to reconfigure the routing in order to improve network performance. However, the requirement for continuity of flow is not always easy to meet during the reconfiguration process if it is not done carefully. Reconfiguration becomes even more challenging for multicast routing in a WDM optical network with the property of partial wavelength conversion, which is also called a sparse wavelength converter network. Existing works address the reconfiguration of unicast or multicast connection routing in networks having the property of full wavelength conversion.

In this thesis, we deal on the one hand with the problem of reconfiguring the routing of a single multicast connection in a sparse wavelength converter network and on the other hand with the problem of reconfiguring the routing of a set of several multicast connections. In the first case, one aims to migrate the flow from an initial multicast routing to a final multicast routing without flow interruption. In the second case, we aim to migrate the optical flow from the initial multicast routing to the final multicast routing with a low number of flow interruptions. Note that the initial multicast routing and the final multicast routing are each supposed to be a set of light-trees.

The method denoted Para_SbSRA (Parallel Subtree by Subtree Reconfiguration Algorithm) has been proposed to solve the problem of routing reconfiguration relating

to a single multicast connection and the methods LSRA (Light-tree Set Reconfiguration Algorithm) and LSRA_2 have been proposed to solve the problem of routing reconfiguration relating to a set of multicast connections. The effectiveness of these methods has been confirmed through simulations.

Keywords : Reconfiguration, flow interruption, multicast routing, WDM optical network, light-tree, wavelength conversion.

TABLE DES MATIÈRES

Dédicace i

Remerciements ii

Résumé iv

Abstract vi

Liste des figures xiv

Liste des tableaux xv

Liste des algorithmes xvi

Liste des acronymes xvii

1 Introduction générale 1

1.1 Contexte 1

1.2 Problématique générale 3

1.3 Organisation de la thèse 6

1.4 Principales contributions et liste des publications 8

1.4.1 Principales contributions 8

1.4.2 Liste des publications 9

2 Notions de base sur les réseaux optiques WDM 11

2.1 Introduction 12

2.2 Architecture d'un réseau optique WDM 13

2.3 Technologie WDM 16

2.4 Technologie de conversion de longueur d'onde 17

2.5 Typologie des répartiteurs optiques 19

2.5.1 Architecture générale des répartiteurs optiques 20

2.5.2 Répartiteurs de type duplicateur et transmetteur pour unicast et multicast . . . 21
2.5.3 Répartiteurs de type duplicateur et transmetteur pour multicast . 22
2.5.4 Répartiteurs à convertisseurs de longueur d'onde . . . 24
2.6 Routage multicast et assignation de longueurs d'onde dans les réseaux optiques WDM . . . 27
2.6.1 Arbre mono-optique . . . 29
2.6.2 Arbre multi-optique . . . 30
2.6.3 Forêt optique . . . 31
2.6.4 Hiérarchie optique . . . 32
2.6.5 Classification des solutions de routage multicast et d'assignation de longueurs d'onde . . . 33
2.7 Conclusion . . . 33

3 Problèmes de reconfiguration de paires d'arbres mono-optiques 36
3.1 Introduction . . . 37
3.2 Concepts fondamentaux . . . 37
3.2.1 Opérations de configuration d'un nœud optique . . . 37
3.2.2 Définitions . . . 39
3.2.3 Illustration des concepts . . . 40
3.3 Problème de reconfiguration d'une paire d'arbres mono-optiques . . . 42
3.3.1 Description du problème de reconfiguration d'une paire d'arbres mono-optiques . . . 42
3.3.2 Travaux connexes à la reconfiguration d'une paire d'arbres mono-optiques . . . 44
3.3.2.1 Reconfiguration sans utilisation d'une longueur d'onde additionnelle . . . 45
3.3.2.2 Reconfiguration avec utilisation de longueurs d'onde additionnelles . . . 48
3.3.3 Récapitulatif et implications . . . 51
3.4 Problème de reconfiguration d'un ensemble de plusieurs paires d'arbres mono-optiques . . . 52
3.4.1 Description du problème de reconfiguration d'un ensemble de plusieurs paires d'arbres mono-optiques . . . 53

3.4.2 Travaux connexes à la reconfiguration d'un ensemble de plusieurs paires d'arbres mono-optiques 54
3.4.3 Récapitulatif et implications 59
3.5 Conclusion 61

4 Reconfiguration d'une paire d'arbres mono-optiques 63
4.1 Introduction 63
4.2 Spécification du problème 65
4.3 Méthodes de reconfiguration d'une paire d'arbres mono-optiques 67
4.3.1 Caractérisation de paires de sous-arbres 68
4.3.1.1 Sélection d'une paire de sous-arbres à liens disjoints . 69
4.3.1.2 Sélection d'une paire de sous-arbres à liens partagés . 74
4.3.2 Méthode de reconfiguration SbSRA 77
4.3.2.1 Justification de l'ordre de reconfiguration des catégories de paires sous-arbres 79
4.3.2.2 Présentation de la méthode SbSRA 82
4.3.3 Méthode de parallèlisation de la reconfiguration Para_SbSRA . . 93
4.3.3.1 Justification de la parallélisation de la reconfiguration de paires de sous-arbres 93
4.3.3.2 Algorithme de reconfiguration Para_SbSRA 94
4.4 Evaluation de performance 101
4.4.1 Critères d'évaluation 101
4.4.2 Description des simulations 101
4.4.3 Analyse des résultats 104
4.5 Conclusion 107

5 Reconfiguration d'un ensemble de paires d'arbres mono-optiques 110
5.1 Introduction 110
5.2 Spécification du problème 112
5.3 Méthodes de reconfiguration pour un ensemble de plusieurs paires d'arbres mono-optiques 114
5.3.1 Méthode de reconfiguration réduisant le coût d'interruptions de flux 115
5.3.1.1 Graphe orienté pondéré de dépendances 115
5.3.1.2 Algorithme proposé 116
5.3.2 Méthode de reconfiguration réduisant les pénalités financières . 123

5.3.2.1 Motivation 123
5.3.2.2 Présentation de la méthode 125
5.4 Evaluation de performance 126
5.4.1 Critères d'évaluation 127
5.4.1.1 Reconfiguration visant à réduire les interruptions de flux 127
5.4.1.2 Reconfiguration visant à réduire les pénalités financières 127
5.4.2 Description des simulations 127
5.4.2.1 Reconfiguration visant à réduire les interruptions de flux 127
5.4.2.2 Reconfiguration visant à réduire les pénalités financières 130
5.4.3 Analyse des résultats 131
5.4.3.1 Reconfiguration visant à réduire les interruptions de flux 131
5.4.3.2 Reconfiguration visant à réduire les pénalités financières 136
5.5 Conclusion 140

6 Conclusion générale 142
6.1 Resumé des travaux 142
6.2 Perspectives 143

Références 147

TABLE DES FIGURES

1.1 Itération de reconfiguration et quelques questions majeures à résoudre à chaque phase selon Li et Wu [18] 4

2.1 Une architecture typique de réseau optique WDM [52] 14
2.2 Principe de transmission WDM 16
2.3 Conversion de longueur d'onde 17
2.4 Réduction de la probabilité de blocage 18
2.5 Architecture générale d'un OXC [52] 20
2.6 Architecture d'un $N * N$ SaD-OXC [64] 22
2.7 Concept de partage des duplicateurs [67] 23
2.8 Architecture d'un MOSaD-OXC [66] 23
2.9 Architecture dite dédiée d'un $2 * 2$ WC-OXC [68] 25
2.10 Architecture dite de partage par nœud d'un $2 * 2$ WC-OXC [68] 26
2.11 Architecture dite de partage par lien d'un 2*2 WC-OXC [68] 26
2.12 Exemple d'arbre mono-optique pour la connexion $< s, \{d_1, d_2\} >$ 30
2.13 Exemple d'admission de connexion par un arbre multi-optique 31
2.14 Exemple de forêt optique 32
2.15 Structures de routage et optimalité 33
2.16 Classification des problèmes à traiter 35

3.1 Illustration de l'exécution d'opérations de configuration 41
3.2 Exemple d'amélioration de performance par reconfiguration d'une paire d'arbres mono-optiques 45
3.3 Exemple de reconfiguration du routage unicast par MBB 46
3.4 Exemple de séquence de configurations obtenue par MBB_1 48
3.5 Illustration d'une séquence de configurations avec BpBAR 50
3.6 Exemple de problème de reconfiguration d'un ensemble de paires d'arbres mono-optiques 54

3.7 Instance de problème de reconfiguration d'un ensemble de paires de chemins (unicasts) mono-optiques 58
3.8 Graphe orienté non pondéré de dépendances associé au problème de reconfiguration d'un ensemble de paires de chemins mono-optiques obtenu par MFVSA 58
3.9 Séquence de configurations retournée par MFVSA 59
3.10 Graphe orienté non pondéré de dépendances associé à l'instance de problème 60
3.11 Séquence de configurations de MFVSA pour une instance de problème de reconfiguration d'un ensemble de paires d'arbres mono-optiques . . 61

4.1 Exemple de problème contenant une paire de sous-arbres à liens disjoints 73
4.2 Exemple de problème contenant une paire de sous-arbres à liens partagés 77
4.3 Illustration d'une séquence de configurations avec SbSRA 92
4.4 Instance de problème à reconfigurer avec Para_SbSRA 99
4.5 Illustration d'une séquence de configurations avec Para_SbSRA 100
4.6 Topologies réseaux 102
4.7 Comparaison des taux d'interruptions moyen causées par les trois méthodes 105
4.8 Comparaison du coût des longueurs d'onde additionnelles causé par les trois méthodes 106
4.9 Comparaison des durées de reconfiguration causées par les trois méthodes 107

5.1 Graphe orienté de dépendances pondéré par les nœuds, associé à l'instance de problème 120
5.2 Graphe obtenu après suppression de MCFVS 121
5.3 Séquence de configurations générée par LSRA 122
5.4 Instance de problème avec des pénalités différentes 124
5.5 Graphe orienté pondéré de dépendances si les pénalités financières sont différentes 126
5.6 Moyenne du nombre d'interruptions produite par les deux approches lorsque la plage du nombre de nœuds destinations est [2 ;8] 133
5.7 Moyenne du nombre d'interruptions produite par les deux approches lorsque la plage du nombre de nœuds destinations est [9 ;15] 133

5.8 Moyenne du nombre d'interruptions produite par les deux approches lorsque la plage du nombre de nœuds destinations est [16 ;22] 134
5.9 Moyenne du nombre d'étapes produite par les deux approches lorsque la plage du nombre de nœuds destinations est [2 ;8] 135
5.10 Moyenne du nombre d'étapes produite par les deux approches lorsque la plage du nombre de nœuds destinations est [9 ;15] 135
5.11 Moyenne du nombre d'étapes produite par les deux approches lorsque la plage du nombre de nœuds destinations est [16 ;22] 136
5.12 Moyenne des pénalités financières produite par les deux approches dans le cadre d'une distribution à connexions fortement prioritaires majoritaires 138
5.13 Moyenne des pénalités financières produite par les deux approches dans le cadre d'une distribution à connexions fortement prioritaires minoritaires 138
5.14 Moyenne du nombre d'interruptions produite par les deux approches dans le cadre d'une distribution à connexions fortement prioritaires majoritaires 139
5.15 Moyenne du nombre d'interruptions produite par les deux approches dans le cadre d'une distribution à connexions fortement prioritaires majoritaires 140

6.1 Classification des problèmes abordés et les solutions proposées 144

LISTE DES TABLEAUX

2.1 Récapitulatif des résultats des méthodes de routage multicast 34

4.1 Informations sur les topologies 102
4.2 Taux d'interruptions moyen (%) causé par les trois méthodes 104
4.3 Coût des longueurs d'onde additionnelles causé par les trois méthodes 106
4.4 Durée (nombre d'étapes de reconfiguration) causée par les trois méthodes 107

Liste des Algorithmes

4.1 Select_PSLD 71
4.2 Select_PSLP 75
4.3 Canaux_preetablir 83
4.4 Canaux_migrer 84
4.5 Supprimer_Canaux 85
4.6 SbSRA 88
4.7 Para_SbSRA 96

5.1 LSRA 117

LISTE DES ACRONYMES

ATM	Asynchronous Transfer Mode
BFS	Breadth-First Search
DFS	Depth-First Search
FVS	Feedback Vertex Set
IPTV	Internet Protocol TeleVision
MBB	Make Before Break
MC-OXC	Multicast Capable Optical Cross-Connect
MCFVS	Minimum Cost Feedback Vertex Set
MFVS	Minimum Feedback Vertex Set
Mono-RRM	Mono-Reconfiguration du Routage Multicast
MOSaD-OXC	Multicast-Only Splitter-and-Delivery Optical Cross-connect
Multi-RRM	Multi-Reconfiguration du Routage Multicast
OADM	Optical Add/Drop Multiplexeur
OEO	Opto-Électro-Optique
OLT	Optical Line Terminal
OSW	Optical SWitch
OXC	Optical Cross-Connect
PCE	Path Computation Element
PON	Passive Optical Network
Rx	Receiver
RMAL	Routage Multicast et Assignation de Longueur d'onde
ROADM	Reconfigurable Optical Add/Drop Multiplexeur
SaD	Splitter-and-Delivery

SDON	Software Defined Optical Networks
SDS	Space Division Switch
SLA	Service Level Agreement
SSB	Split-Switch Bank
Tx	Transceiver
UM	Unité Monétaire
WCB	Wavelength Converter Bank
WC-OXC	Wavelength Converter Optical Cross Connect
WDM	Wavelength Division Multiplexing

CHAPITRE 1

INTRODUCTION GÉNÉRALE

Sommaire

1.1 Contexte **1**

1.2 Problématique générale **3**

1.3 Organisation de la thèse **6**

1.4 Principales contributions et liste des publications **8**

1.4.1 Principales contributions 8

1.4.2 Liste des publications 9

1.1 Contexte

Les réseaux optiques, réseaux utilisant comme support de transmission la fibre optique, sont des réseaux à haut débit qui jouent un rôle essentiel dans nos vies quotidiennes. Ils ont rendu possible la révolution mondiale des communications qui a rassemblé le monde, et a permis aux pays émergents de rejoindre l'économie numérique mondiale. Ils ont donc contribué à l'amélioration de la qualité de vie des personnes. La popularité grandissante des applications telles qu'Internet Protocol TeleVision (IPTV), la vidéoconférence, le calcul distribué, la télémédecine et la formation à distance obligent les opérateurs réseau à utiliser plusieurs technologies afin de répondre aux besoins de leurs clients. La technologie de multiplexage à répartition en longueur d'onde ou *Wavelength Division Multiplexing* (WDM) [1, 2] crée de nombreux canaux optiques qui permettent de transmettre simultanément plusieurs signaux (ou flux) optiques sur une seule fibre optique. Notons que chaque canal optique est caractérisé par sa longueur d'onde. La conversion de longueur d'onde est une autre technologie utilisée afin de réduire la probabilité de blocage de nouvelles demandes de connexion et d'ajouter plus de flexibilité dans la gestion du réseau [3-5]. Elle consiste à convertir un signal de longueur d'onde λ_1 reçu par un nœud optique en un signal

d'une longueur d'onde λ_2 à la sortie avec λ_1 différent de λ_2 [6]. Pour ce faire, ce nœud optique doit être équipé de convertisseurs de longueur d'onde. Il est important de savoir que malgré leurs avantages, les convertisseurs de longueur d'onde sont coûteux et sont destinés à le rester pendant assez longtemps [7, 8]. Par conséquent, il est plus pratique que seulement certains nœuds du réseau soient dotés de convertisseurs de longueur d'onde [9, 10]. Plusieurs stratégies de placement des convertisseurs de longueur d'onde sur certains nœuds du réseau ont été proposées [11-15]. Un réseau optique WDM dans lequel seulement certains nœuds ont la capacité de conversion de longueur d'onde est appelé réseau optique WDM ayant la propriété de conversion partielle de longueur d'onde [9]. Ce type de réseau permet une meilleure utilisation des ressources tout en induisant un coût acceptable de dimensionnement du réseau [16].

Dans ces réseaux, différents types de connexions sont établies à savoir les connexions unicasts (d'une source vers une destination) et les connexions multicasts (d'une source vers plusieurs destinations). L'établissement de ces connexions nécessite la résolution du problème de routage et d'assignation de longueurs d'onde qui consiste à déterminer une succession de liens (ou fibre) dans le réseau et d'assigner une longueur d'onde à chacun de ces liens. Par souci de concision, dans la suite de ce chapitre, le terme routage (unicast/multicast) désigne la solution du problème de routage et d'assignation de longueurs d'onde.

Par défaut, lorsque des connexions sont ajoutées ou supprimées du réseau, le routage des connexions existantes dans le réseau n'est pas modifié. Par conséquent, après quelques ajouts et retraits de connexions, l'utilisation globale des ressources que sont les longueurs d'onde du réseau n'est plus optimale [17]. Ainsi, une nouvelle demande de connexion peut être bloquée parce qu'un routage ne peut être défini pour celle-ci. C'est dans ce contexte que la reconfiguration du routage a été introduite dans les réseaux optiques WDM. Elle est effectuée en configurant un nouveau routage appelé routage final tout en supprimant ou en modifiant le routage actuel ou routage initial utilisé par le flux de données [18]. Cela est rendu possible en effectuant des opérations de conversion de longueur d'onde par exemple sur certains nœuds du réseau.

La reconfiguration du routage est un mécanisme important pour les opérateurs de réseau parce qu'elle permet d'optimiser la performance du réseau et la transmission de flux optique [19]. Elle est menée en réponse à une augmentation du nombre de demandes de connexion, à des défaillances sur le réseau ou encore à des déploiements de nouvelles ressources réseau telles que de nouvelles fibres optiques, nœuds

[20, 21]. La reconfiguration du routage est un mécanisme très important dans les réseaux WDM et donc doit être méthodiquement effectuée. Dans cette perspective de réalisation méthodique, la reconfiguration du routage se compose de trois phases [22] qui sont le choix de la politique de déclenchement de la reconfiguration (phase 1), la définition d'un nouveau routage (phase 2) et la migration du flux sur le nouveau routage (phase 3). Le choix de la politique de déclenchement concerne le processus de décision du moment où la phase 2 doit être réalisée. L'opérateur a le choix entre une politique de déclenchement proactive qui déclenche la phase 2 de la reconfiguration à un moment prédéfini ou à des intervalles de temps réguliers [23, 24], et une politique de déclenchement réactive qui déclenche la phase 2 après l'atteinte d'un seuil critique de niveau de performance [25, 26]. La phase 2 qui consiste à définir un nouveau routage se réalise lorsque les conditions de la politique de déclenchement sont remplies. Ce nouveau routage doit optimiser un critère prédéfini par l'opérateur de réseau comme minimiser le nombre de longueurs d'onde utilisées, minimiser la probabilité de blocage [25]. Après avoir défini le nouveau routage ou routage final, la dernière phase de la reconfiguration consiste à migrer en douceur le flux optique du routage initial au routage final déterminé à la phase 2 [18]. Mais des questions doivent être traitées à chacune des phases de la reconfiguration afin que cette dernière puisse être réalisée efficacement.

1.2 Problématique générale

La reconfiguration du routage est une tâche délicate qui doit être effectuée avec beaucoup de précautions si l'on veut bénéficier de ses avantages. La Figure 1.1 donne un aperçu des questions les plus usuelles qui doivent être prises en compte à chaque phase de la reconfiguration. Pour les résumer, la phase 1 consiste à déterminer la fréquence de reconfiguration appropriée dans le cas d'une politique proactive ou le seuil critique de déclenchement de la reconfiguration dans le cas d'une politique réactive. La seconde phase consiste à déterminer d'une part la liste des connexions à reconfigurer et d'autre part un nouveau routage en fonction d'un critère d'optimisation défini par l'opérateur. La dernière phase consiste d'une part à identifier la manière dont les nœuds optiques vont participer au processus de reconfiguration et d'autre part à trouver la séquence de configurations de telle sorte à migrer le flux optique du routage initial sur le nouveau routage.

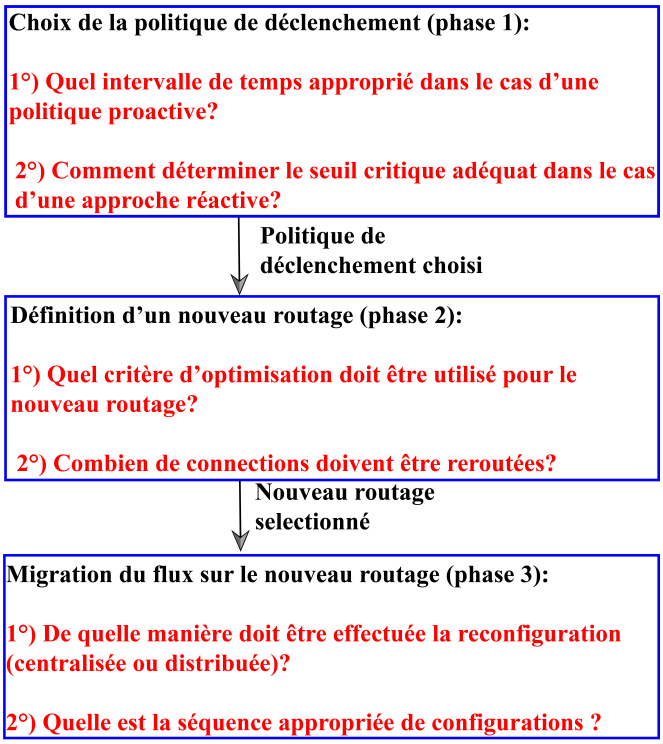


FIGURE 1.1 – Itération de reconfiguration et quelques questions majeures à résoudre à chaque phase selon Li et Wu [18]

Vu que les questions relatives à la phase 1 et la phase 2 de la reconfiguration du routage ont été traitées en profondeur dans [27-30] et [31-34], les questions relatives à la phase 3 retiennent plus l'attention des chercheurs depuis ces deux dernières décennies. Aborder la question 1 de la phase 3 se résume à indiquer le nombre d'entités chargées de la mise en œuvre des différentes phases de la reconfiguration. Ainsi, il y a deux approches de base : l'approche centralisée et l'approche distribuée. Dans l'approche centralisée, une entité centrale (par exemple le *Path Computation Element* (PCE) [35]) est chargée de collecter les informations (demandes de connexion, pannes) sur l'état du réseau en temps réel afin d'implémenter la politique choisie à la phase 1. Puis, elle calcule les nouveaux chemins optiques optimisant le critère de performance prédéfinie. Ensuite, elle détermine la séquence de configurations et enfin envoie les ordres de configurations aux différents nœuds du réseau. Dans l'approche distribuée, les nœuds coopèrent à l'exécution des phases. Cette coopération peut créer des problèmes de convergence qui impliquent une longue durée de reconfiguration en

pratique. Par conséquent, une bonne solution est d'utiliser une approche mixte. Dans l'approche mixte, le réseau est subdivisé en plusieurs domaines de reconfiguration. Puis, l'approche centralisée est appliquée à chaque domaine de reconfiguration et l'approche distribuée pour la gestion de l'ensemble des entités centrales (ou PCEs) [36]. En effet, lorsque la reconfiguration du routage concerne des connexions établies entre des nœuds appartenant à plusieurs domaines, il est inévitable d'avoir une coordination entre les PCEs de ces différents domaines [37]. L'approche mixte a pour avantages la réduction de la charge du réseau, la résilience aux pannes vu le nombre de PCEs supérieur à un, et l'amélioration du temps de convergence grâce à une division du travail de reconfiguration entre les différents domaines de reconfiguration.

La recherche de la séquence de configurations (voir question 2 de la phase 3) demeure le défi le plus épineux de la phase 3. Elle consiste à trouver la séquence de configurations qui permet de faire migrer le flux optique du routage initial vers le routage final de chaque connexion sans interrompre (ou très peu) le flux optique vers les nœuds destinataires. Les interruptions de flux sont très critique dans les réseaux optiques en raison de l'énorme capacité des canaux optiques [38]. De plus, elles peuvent entraîner des pénalités financières pour l'opérateur de réseau qui ne respecte pas certains Service-Level Agreements (SLAs) ou contrats fournisseur-client interdisant les interruptions de flux [39, 40]. Par conséquent, les interruptions de flux sont à éviter autant que possible lors de la reconfiguration du routage.

Plusieurs réponses ont été données à cette question dans le contexte de la reconfiguration du routage unicast [41-43]. Cependant, selon la revue de littérature de Wu [36], très peu de réponses ont été apportées dans le cadre de la reconfiguration du routage multicast. Selon le rapport annuel (2018-2023) de Cisco [44], la popularité naissante des applications multicasts (telles que le calcul distribué avec le MapReduce pour le Big Data, la formation en ligne et la télémédecine) risque de s'accroître encore pour plusieurs années. Par ailleurs, les travaux majeurs en rapport avec la reconfiguration du routage multicast ont été effectués dans le contexte des réseaux optiques WDM dont tous les nœuds ont la capacité de conversion de longueur d'onde. Ce type de réseau n'est pas très réaliste en partie à cause de leur coût de dimensionnement élevé. Par conséquent, dans cette thèse nous nous intéressons à la détermination de la séquence de configurations dans le cadre de la reconfiguration du routage multicast dans les réseaux WDM ayant la propriété de conversion partielle de longueur d'onde.

Le problème général de reconfiguration du routage multicast dans les réseaux

WDM ayant la propriété de conversion partielle de longueur d'onde est formulé comme suit :

— **Entrées** : Un ensemble de connexions multicasts M , la configuration initiale C_0 du routage multicast initial des éléments de M et la configuration finale C_f (à établir) du routage multicast final des éléments de M.

— **Sortie** : Une séquence de configurations $C =< C_0, C_1, \ldots, C_{f-1}, C_f >$ qui induit un faible nombre d'interruptions de flux ; avec C_i la configuration nécessaire pour la transmission du flux optique à l'étape i du processus de reconfiguration.

Par convention, nous parlons de problème de Mono-Reconfiguration du Routage Multicast (Mono-RRM) si $|M| = 1$ et de problème de Multi-Reconfiguration du Routage Multicast (Multi-RRM) si $|M| > 1$. La résolution de ces deux types de problèmes est délicate à cause de dépendances qui peuvent exister entre certains canaux optiques établis qui appartiennent à C_0 et certains canaux optiques à établir qui appartiennent à C_f. En clair, dans le problème Mono-RRM, la configuration du routage multicast final de la seule connexion multicast m élément de M peut nécessiter des canaux optiques déjà utilisés par la configuration du routage multicast initial de m. Ce qui rend le problème Mono-RRM délicat à traiter. Dans le cas du problème Multi-RRM, en plus de la difficulté héritée du problème Mono-RRM, il faut tenir compte du fait que la configuration du routage multicast final d'un élément de M peut nécessiter des canaux optiques déjà utilisés par la configuration du routage multicast initial d'un autre élément de M.

1.3 Organisation de la thèse

L'objectif de cette thèse est de proposer des méthodes pour résoudre les problèmes généraux cités à la fin de la section 1.2 à savoir le problème de Mono-RRM et le problème de Multi-RRM. Pour ce faire, cette thèse est organisée comme suit : les chapitres 1 et 2 fournissent les prérequis pour entamer l'analyse en profondeur et la résolution des problèmes traités dans cette thèse. Le chapitre 3 fournit une analyse et une revue de littérature relative d'une part au problème de Mono-RRM et d'autre part au problème de Multi-RRM. Quant aux chapitres 4 et 5, ils présentent nos contributions relatives d'une part au problème de Mono-RRM (voir chapitre 4) et d'autre part au problème de Multi-RRM (voir chapitre 5). Le chapitre 6 livre les conclusions de cette thèse.

Le chapitre 1 est une introduction générale au sujet de la thèse. Il évoque l'atout que représente un réseau WDM ayant la propriété de conversion partielle de longueur d'onde pour un opérateur de réseau. Puis, ce chapitre justifie l'importance du problème général de reconfiguration traité dans cette thèse avant de l'énoncer. Cette justification peut se résumer au fait que si le problème de reconfiguration n'est pas bien traité alors la reconfiguration va entraîner des interruptions de flux qui vont dégrader la qualité du service fourni par un opérateur réseau à certains de ses clients. Cela peut dans certains cas entraîner des pénalités financières pour l'opérateur de réseau.

Le chapitre 2 présente le réseau tout-optique WDM en insistant sur les équipements qui composent chaque nœud du réseau. Ensuite, nous présentons en détail le routage multicast dans le cas où la contrainte de continuité de longueur d'onde [45] serait respectée et enfin dans le cas où cette contrainte serait relâchée. Ces différentes structures de routage, ayant la forme d'un arbre, sont les résultats de la phase 2 du processus de reconfiguration. Leur présentation est importante dans la mesure où ces structures de routage (à forme d'arbre) sont des éléments clés de notre problème.

Le chapitre 3 apporte une description claire à chaque type de problème puis ce chapitre présente les travaux existants en rapport avec chaque type de problème. Il ressort que les travaux en rapport avec le problème Mono-RRM proposent des méthodes qui sont peu adaptées lorsque tous les nœuds n'ont pas la capacité de conversion de longueur d'onde. Quant aux travaux en rapport avec le problème Multi-RRM, ils proposent des méthodes qui sont efficaces dans le cas de connexions unicasts alors que cette étude concerne les connexions multicasts.

Le chapitre 4 présente nos contributions au problème Mono-RRM. En clair, nous proposons deux méthodes de reconfiguration qui sont basées sur une approche par sous-arbres contrairement aux méthodes de la littérature qui se basent sur une approche par branches. Les résultats des simulations ont montré que nos méthodes ne causent pas d'interruption de flux alors que dans certains cas, les méthodes de la littérature interrompent le flux. Les contributions évoquées dans ce chapitre on fait l'objet d'articles scientifiques dans [J1] et [J3].

Le chapitre 5 présente nos contributions au problème Multi-RRM. En clair, nous proposons une méthode de reconfiguration qui supprime temporairement les routes d'un ensemble de connexions multicasts de coût faible alors que les méthodes de la littérature suppriment les routes d'un ensemble de cardinalité minimale. Le coût d'une connexion multicast est le nombre de nœuds destinations de cette connexion. Nous

montrons analytiquement et par simulation que notre méthode interrompt moins de flux que les méthodes existantes. Vu que l'un des objectifs cruciaux pour l'opérateur de réseau est de réduire ses dépenses, nous proposons également une autre méthode dont l'objectif principal est de réduire les pénalités financières pour cause d'interruptions de l'opérateur de réseau. Les contributions de ce chapitre ont fait l'objet d'articles scientifiques dans [J2] et [C1].

Le chapitre 6, dans un premier temps, donne les conclusions de ce travail qui concerne le problème Mono-RRM appliqué à un routage multicast qui respecte la contrainte de continuité de longueur d'onde et le problème Multi-RRM appliqué à un routage multicast qui respecte la contrainte de continuité de longueur d'onde. Dans un second temps, ce chapitre livre des perspectives concernant les problèmes Mono-RRM et Multi-RRM appliquées à un routage multicast qui respectent ou pas la contrainte de continuité de longueur d'onde.

1.4 Principales contributions et liste des publications

1.4.1 Principales contributions

Les contributions proposées dans cette thèse supposent que le routage multicast initial et le routage multicast final de chaque connexion concernée par la reconfiguration ont une structure d'arbre. Nos contributions principales peuvent être résumées comme suit :

— **Pour traiter le problème Mono-RRM, nous avons effectué les contributions suivantes** (voir chapitre 4) :

1. La caractérisation des différentes paires de sous-arbres que l'on peut sélectionner dans la paire d'arbres (arbre initial, arbre final) à reconfigurer : deux catégories de paires de sous-arbres à savoir la catégorie des paires de sous-arbres à liens disjoints et la catégorie des paires de sous-arbres à liens partagés ;
2. Une méthode de reconfiguration d'une paire d'arbres mono-optiques notée SbSRA qui s'appuie sur la précédente caractérisation : elle procède par une reconfiguration séquentielle des paires de sous-arbres à liens disjoints puis une reconfiguration séquentielle des paires de sous-arbres à liens partagés ;

3. Une preuve de la reconfigurabilité en parallèle des paires de sous-arbres d'une même catégorie ;
4. La méthode Para_SbSRA de reconfiguration d'une paire d'arbres qui permet de mieux réduire la longueur du processus de reconfiguration. Cette méthode s'appuie sur la précédente preuve afin de reconfigurer en parallèle les paires de sous-arbres à liens disjoints puis de reconfigurer en parallèle les paires de sous-arbres à liens partagés.

— **Pour traiter le problème Multi-RRM, nous avons effectué les contributions suivantes** (voir chapitre 5) :

1. Un algorithme de reconfiguration noté LSRA induisant un plus faible nombre d'interruptions de flux que celles de la littérature : il s'appuie sur la construction d'un graphe orienté des dépendances (entre les connexions) valué par les nœuds afin de déduire la séquence de configurations recherchée. Le poids associé à un nœud représente le nombre de nœuds destinations de la connexion que représente ce nœud dans le graphe ;
2. Un algorithme de reconfiguration noté LSRA_2 induisant de faibles pénalités financières ; LSRA_2 est une variante de LSRA dans laquelle le poids d'une connexion tient compte de la pénalité financière associée à l'interruption du flux vers les nœuds destinations de cette connexion.

1.4.2 Liste des publications

Cette thèse a abouti à trois (03) articles scientifiques de journal :

[J1] **A. F. Atta**, B. Cousin, J. C. Adépo, S. Oumtanaga, 'Light-tree Reconfiguration in Sparse Wavelength Converter Network', *International Journal of Communication Networks and Distributed Systems*. Statut : *Accepté et sous presse, parution en Janvier 2022*.

[J2] **A. F. Atta**, J. C. Adépo, B. Cousin, S. Oumtanaga, 'Minimize Flow Interruptions during Reconfiguration of a set of Light-trees in All-optical WDM Network', *International Journal of Computer Science and Network Security*, vol. 20, no. 7, pp. 77–85, 2020.

[J3] **A. F. Atta**, J. C. Adépo, B. Cousin, S. Oumtanaga, 'Sub-tree-based Approach for

Reconfiguration of a Light-tree Pair without Flow Interruption in Sparse Wavelength Converter Network', *Information*, vol 12, no. 5, pp. 211-230, 2021.

Par ailleurs, nous avons également participé à une (01) conférence internationale :

[C1] **A. F. Atta**, G.A. Keupondjo-Satchou, J. C. Adépo, S. Oumtanaga, 'Minimize Penalty Fees During Reconfiguration of a Set of Light-tree Pairs in an All-Optical WDM Network', *6th International Conference on Information, Communication & Computing Technologies (ICICCT 2021)*. Proceedings dans *Communications in Computer and Information Science (CCIS)*, Volume 1417, Springer, Octobre 2021.

CHAPITRE 2

NOTIONS DE BASE SUR LES RÉSEAUX OPTIQUES WDM

Sommaire

2.1 Introduction **12**

2.2 Architecture d'un réseau optique WDM **13**

2.3 Technologie WDM **16**

2.4 Technologie de conversion de longueur d'onde **17**

2.5 Typologie des répartiteurs optiques **19**

2.5.1 Architecture générale des répartiteurs optiques 20

2.5.2 Répartiteurs de type duplicateur et transmetteur pour unicast et multicast 21

2.5.3 Répartiteurs de type duplicateur et transmetteur pour multicast 22

2.5.4 Répartiteurs à convertisseurs de longueur d'onde 24

2.6 Routage multicast et assignation de longueurs d'onde dans les réseaux optiques WDM **27**

2.6.1 Arbre mono-optique 29

2.6.2 Arbre multi-optique 30

2.6.3 Forêt optique 31

2.6.4 Hiérarchie optique 32

2.6.5 Classification des solutions de routage multicast et d'assignation de longueurs d'onde 33

2.7 Conclusion **33**

2.1 Introduction

Les réseaux WDM sont couramment classés en fonction de leur architecture en deux groupes : les réseaux WDM à diffusion et sélection ou *broadcast-and-select* et les réseaux WDM à routage en longueur d'onde. Les réseaux WDM à diffusion et sélection [46] sont basés sur des topologies en étoile, en bus ou en anneau. Dans ces réseaux, un émetteur diffuse simplement le signal optique sur différents canaux, chacun utilisant une longueur d'onde dédiée sur une fibre optique et les récepteurs s'accordent pour recevoir le bon signal en utilisant un filtre optique adapté à la longueur d'onde du canal. L'avantage des réseaux WDM à diffusion et sélection réside dans leur simplicité et leur capacité de multicasting naturelle (capacité à transmettre un message vers plusieurs destinations). Toutefois, ils ont des insuffisances. Tout d'abord, ils nécessitent un grand nombre de longueurs d'onde, généralement au moins autant que le nombre de nœuds dans le réseau, car il n'y a pas ou peu de réutilisation possible des longueurs d'onde dans le réseau [47]. Ainsi, les réseaux ne sont pas extensibles au-delà du nombre de longueurs d'onde prises en charge : on dit que ces réseaux ne supportent pas le passage à l'échelle ou *scalability* [7]. Pour terminer, ils ne peuvent pas couvrir de très longues distances, car la puissance émise est répartie entre différents nœuds et chaque nœud ne reçoit qu'une petite fraction de la puissance émise, qui devient plus petite à mesure que le nombre de nœuds augmente. Pour ces raisons, les réseaux à diffusion et sélection conviennent aux réseaux locaux, mais ne sont pas extensibles aux réseaux étendus qui nous concernent dans cette thèse. Ils sont également utilisés dans les réseaux d'accès récents, par exemple les réseaux optiques passifs ou *Passive Optical Network* (PON) [48].

Afin de pallier les insuffisances des réseaux WDM à diffusion et sélection, les réseaux WDM à routage en longueur d'onde ont été introduits. Dans un réseau WDM à routage en longueur d'onde, les nœuds sont interconnectés via des fibres optiques de sorte à former une topologie physique maillée. Chaque nœud est équipé d'un ensemble de transpondeurs qui sont des émetteurs/récepteurs permettant de transmettre ou recevoir des signaux optiques. Contrairement à un réseau WDM à diffusion et sélection, dans un réseau WDM à routage en longueur d'onde, une succession de canaux optiques non partagés est requise par un émetteur pour transmettre des signaux optiques. Ce type de réseau WDM convient aux réseaux étendus et est plus sophistiqué que les réseaux WDM à diffusion et sélection. En effet, il permet de mettre en œuvre

davantage de fonctionnalités de réseau : routage (unicast et multicast), groupage de trafic, reconfiguration, résistance aux pannes, etc. Dans cette thèse qui traite des réseaux WDM à routage en longueur d'onde, nous utilisons le terme de réseau optique WDM pour indiquer un réseau optique WDM à routage en longueur d'onde. Une brève introduction à l'architecture des réseaux optiques WDM ainsi que certaines technologies clés de ces réseaux sont présentées dans les sections 2.2, 2.3 et 2.4. Puis une présentation succinte de l'architecture des nœuds d'un réseau optique WDM est faite dans la section 2.5.

Le chapitre précédent souligne que la reconfiguration du routage est une fonctionnalité importante dans un réseau optique WDM. Aussi, nous avons énoncé le problème général de cette thèse. Ce problème utilise comme entrée, le routage obtenu à la fin de la phase 2 de la reconfiguration. Pour mieux appréhender les entrées de notre problème, la section 2.6 présente les différents types de résultats possibles retournés par la phase 2 de la reconfiguration. Nous terminons ce chapitre par une conclusion qui fait une ouverture sur les prochains chapitres.

2.2 Architecture d'un réseau optique WDM

La Figure 2.1 présente un exemple de réseau optique WDM. En général, un réseau optique WDM est composé de trois éléments fondamentaux que sont les répartiteurs optiques ou *Optical Cross Connects* (OXCs), les fibres optiques et les nœuds d'accès encore appelés nœuds de bordure [49] :

— Les OXCs sont chargés des principales fonctionnalités, notamment le multiplexage, le démultiplexage, la commutation, la conversion de longueur d'onde, etc. Grâce au démultiplexage du signal optique entrant, un OXC peut commuter chacune des longueurs d'onde d'un port d'entrée vers un port de sortie particulier, indépendamment des autres longueurs d'onde. Certains OXCs particuliers peuvent également commuter une longueur d'onde vers plusieurs ports de sortie simultanément en utilisant un duplicateur optique [50] (ou *light splitter*) pour prendre en charge les applications nécessitant le multicasting.

— Les fibres optiques acheminent les signaux optiques via des canaux optiques, fournissant ainsi un moyen de transmission à haut débit. En effet, plusieurs signaux optiques peuvent être transmis simultanément sur la même fibre optique

via des canaux optiques qui sont caractérisés par des longueurs d'onde différentes. Notons qu'un canal optique donné est une «voie» de communication optique caractérisée par une longueur d'onde donnée sur une fibre optique donnée [51].

— Les nœuds d'accès jouent le rôle d'interface entre le réseau optique WDM cœur et les réseaux clients tels que les réseaux IP/MPLS et les réseaux de type *Asynchronous Transfer Mode* (ATM). Un nœud d'accès peut être soit émetteur, soit récepteur dans une certaine communication. Du côté de l'émetteur, il reçoit chaque signal via un port d'entrée. Ensuite, il effectue la conversion Electrique/Optique (E/O) du signal reçu et l'émet enfin dans le cœur du réseau vers un OXC. Du côté du récepteur, il effectue la conversion Optique/Electrique (O/E) du signal destiné aux réseaux d'accès.

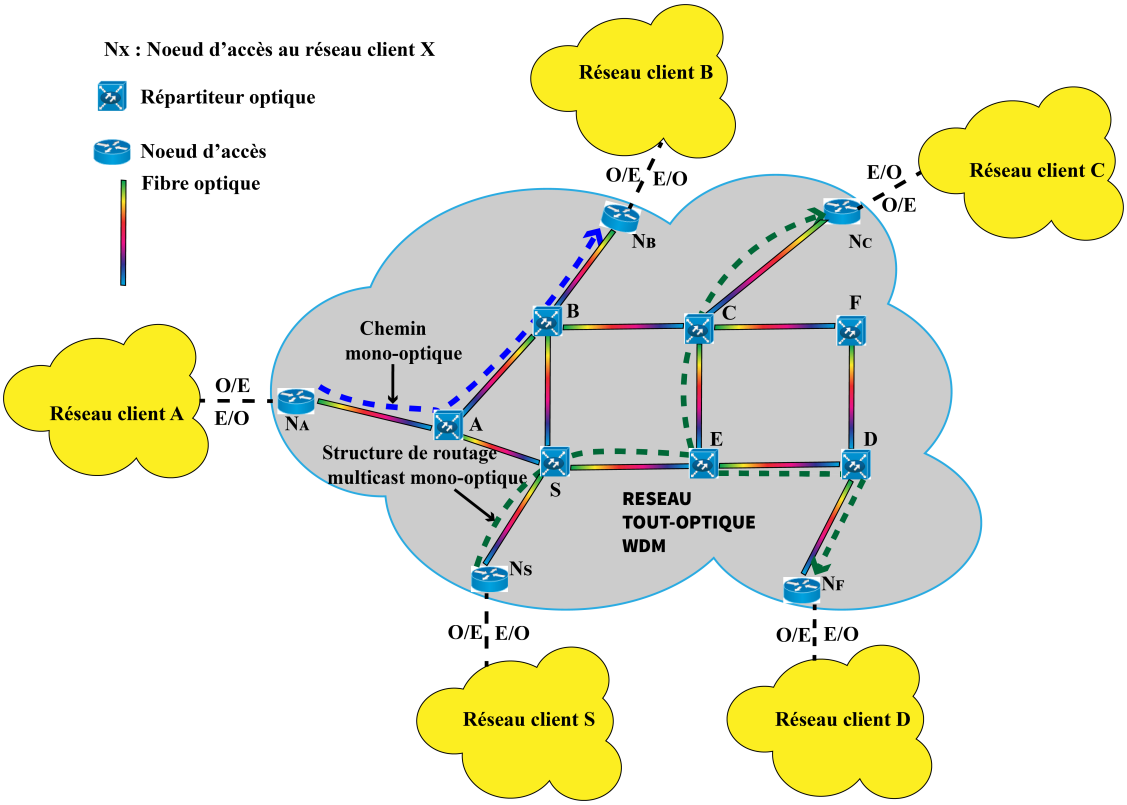


FIGURE 2.1 – Une architecture typique de réseau optique WDM [52]

Pour prendre en charge une communication unicast (respectivement multicast) optique à très haut débit dans le réseau optique WDM, une structure de routage uni-

cast (respectivement multicast) entièrement optique doit être établie entre les nœuds d'accès liés aux réseaux clients dans lesquelles les nœuds désirant communiquer se trouvent. Cela signifie que le signal reste entièrement dans le domaine optique entre les OXCs. Autrement dit, la conversion Optique/Electrique (O/E) et vice-versa n'est possible uniquement qu'au niveau des nœuds d'accès [53] : le réseau est appelé un réseau tout-optique WDM. Par souci de concision, on garde le terme réseau optique WDM pour faire référence au réseau tout-optique WDM. Cependant, les structures de routage optiques sont sujettes aux contraintes suivantes [54] :

— Contrainte d'unicité de longueur d'onde : deux structures de routage ne peuvent pas utiliser la même longueur d'onde dans une fibre optique partagée.

— Contrainte de continuité de la longueur d'onde : en l'absence de toute conversion de longueur d'onde, tous les canaux optiques constituant une structure de routage unicast (respectivement multicast) doivent utiliser la même longueur d'onde : on parle de structure de routage unicast (respectivement multicast) mono-optique. Si cette contrainte est relâchée alors on parle de structure de routage unicast (respectivement multicast) multi-optique. Dans la suite de notre document, le terme chemin fait référence à une structure de routage unicast.

Sur la Figure 2.1, afin de permettre une communication unicast entre un nœud du réseau client A et un nœud du réseau client B, un chemin mono-optique est établi. Ce chemin relie le nœud d'accès au réseau client A noté N_A au nœud d'accès au réseau client B noté N_B. Il est matérialisé par l'ensemble des canaux optiques qui sont représentés par des traits discontinus de couleur bleue. Il passe par les nœuds (répartiteurs optiques) A et B. Notons que ce chemin mono-optique impose que la communication unicast soit unidirectionnelle à savoir depuis un nœud du réseau client A vers un nœud du réseau client B. Par ailleurs, une structure de routage multicast mono-optique est établie dans le but de permettre une communication multicast entre un nœud du réseau client S et des nœuds situés respectivement dans les réseaux clients C et D. Cette structure de routage relie le nœud d'accès au réseau client S noté N_S aux nœuds d'accès des réseaux clients C et D respectivement noté N_C et N_D. Elle est matérialisée par l'ensemble des canaux optiques qui sont représentés par des traits discontinus de couleur verte. Notons que cette structure de routage multicast mono-optique impose que la communication multicast est unidirectionnelle à savoir depuis un nœud du réseau client S vers des nœuds situés respectivement dans le réseau client C et le réseau client D.

2.3 Technologie WDM

Une fibre optique a une énorme bande passante, donc un débit théorique maximal de l'ordre du Térabit par seconde [55]. Cependant, le débit généré ou absorbé par un réseau client contenant quelques postes de travail est souvent limité, à quelques Gigabits par seconde [7]. Par conséquent, il est extrêmement difficile d'exploiter toute la bande passante de communication d'une seule fibre en utilisant un seul canal optique par fibre optique. Le multiplexage par répartition en longueur d'onde ou WDM est une technique qui permet de gérer la disparité entre la bande passante d'une fibre et le débit généré ou absorbé par un réseau client en multiplexant des canaux optiques de longueurs d'onde différentes sur une seule fibre. Le WDM crée de nombreux canaux optiques qui permettent de transmettre simultanément plusieurs signaux (ou flux) optiques sur une même fibre optique. Chaque canal optique permet un débit maximal avoisinant le débit que peut générer (ou absorber) un réseau client. Comme le montre la Figure 2.2, au niveau du nœud émetteur, chaque émetteur (compris dans un transpondeur) EM_i envoie tout signal optique suivant une longueur d'onde distincte λ_i, où $1 \leq i \leq n$. Un multiplexeur collecte tous les signaux optiques et les transmet à une sortie commune. Au niveau du nœud récepteur, un démultiplexeur sépare les signaux optiques et transmet tout signal optique de longueur d'onde λ_i à un récepteur RE_i compris dans un transpondeur.

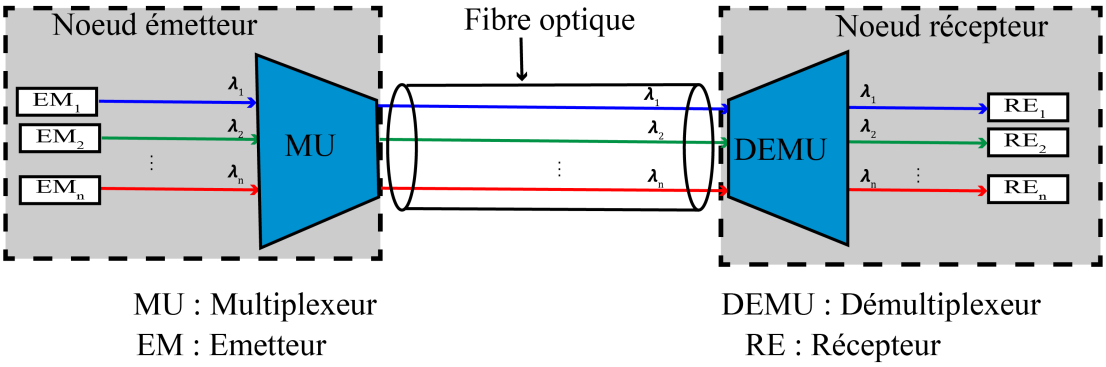


FIGURE 2.2 – Principe de transmission WDM

2.4 Technologie de conversion de longueur d'onde

Dans un réseau optique WDM, par défaut on requiert la continuité des longueurs d'onde le long d'une succession de fibres. Autrement dit, si un signal optique utilise une longueur d'onde donnée dans une fibre à un point du réseau, on continuera à utiliser cette même longueur d'onde le long des autres fibres que doit emprunter le signal optique. Cependant, les équipements des nœuds du réseau peuvent avoir la capacité de convertir une longueur d'onde : cela signifie que si un signal optique emprunte un canal optique qui est caractérisé par la longueur d'onde λ_i dans une fibre connectée à un nœud du réseau alors ce signal optique peut être réémis via un canal optique caractérisé par une longueur d'onde λ_o (avec λ_o différent de λ_i) dans une autre fibre connectée à ce nœud. La conversion est assurée par un équipement appelé convertisseur de longueur d'onde qui est inclus dans un nœud du réseau. La Figure 2.3 présente un nœud optique ou OXC qui grâce à son convertisseur de longueur d'onde convertit la longueur d'onde λ_2 en la longueur d'onde λ_3.

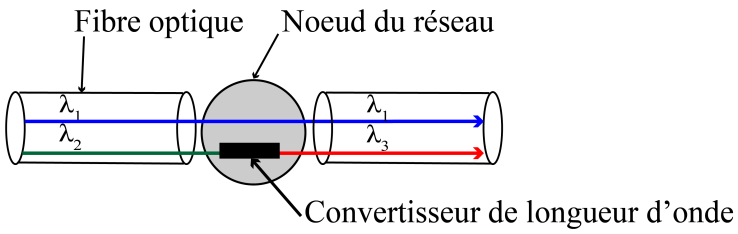


FIGURE 2.3 – Conversion de longueur d'onde

La conversion de longueur d'onde ajoute de la flexibilité dans la gestion du réseau, car elle peut être requise dans la reconfiguration du routage précisément aux phases 2 et 3 afin de réduire la probabilité de blocage dans le réseau [56, 57]. Considérons l'exemple de la Figure 2.4. L'ensemble des liens (c'est-à-dire les traits en noir) ainsi que les nœuds (c'est-à-dire les ronds en noir) forment la topologie physique du réseau optique WDM. Notons que dans la suite de cette thèse, un nœud de toute topologie physique désigne une combinaison d'un répartiteur optique et du nœud d'accès qui lui est associé comme proposé dans [58]. Dans cet exemple, il y a deux connexions en cours (voir Figure 2.4.a). Une connexion du nœud a vers le nœud d dont le chemin mono-optique utilise la longueur d'onde λ_1 et une connexion du nœud e vers le nœud c dont le chemin mono-optique utilise la longueur d'onde λ_2. Nous supposons dans cet

exemple que le nombre maximum de longueurs d'onde sur chacun des liens (ou fibres optiques) est égal à 2. Ces longueurs d'onde sont λ_1 et λ_2. Dans ce cas de figure, une nouvelle connexion du nœud a vers le nœud c est bloquée bien qu'il existe une longueur d'onde disponible sur chacun des liens $a - b$ et $b - c$. Ce blocage est dû à la contrainte de continuité de la longueur d'onde : il n'existe pas une longueur d'onde disponible continue de a vers c. Afin d'éviter ce blocage, on peut envisager d'installer dans le nœud b des convertisseurs de longueur d'onde afin de pouvoir accepter la connexion $< a\{c\} >$. Dans ce cas, le chemin multi-optique de la connexion du nœud a vers le nœud c utilisera sur le lien $a - b$ la longueur d'onde λ_2 et sur le lien $b - c$ la longueur d'onde λ_1 comme on peut le voir sur la Figure 2.4.b.

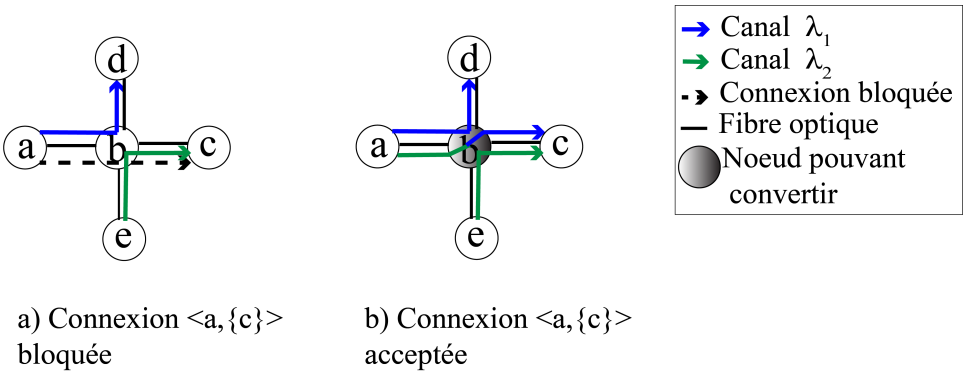


FIGURE 2.4 – Réduction de la probabilité de blocage

Les technologies de conversion de longueur d'onde peuvent être classées en deux grandes catégories : la conversion opto-électronique de longueur d'onde, dans laquelle le signal optique doit d'abord être converti en un signal électronique puis un routeur de paquets intervient avant d'effectuer une conversion inverse du signal électronique en signal optique ; et la conversion de longueur d'onde tout-optique, dans laquelle le signal reste dans le domaine optique [59]. La conversion opto-électronique combinée à la conversion inverse dans le routeur de paquets introduit un délai supplémentaire, elle est plus coûteuse. Dans les réseaux optiques WDM que nous avons supposés tout-optique, la conversion de longueur d'onde tout-optique est employée afin de conserver un haut débit.

2.5 Typologie des répartiteurs optiques

Dans les réseaux optiques WDM, les nœuds optiques pouvant effectuer des commutations sont indispensables pour exploiter le grand potentiel de transmission des fibres optiques. Parallèlement à la révolution des réseaux optiques, on a assisté à un grand développement de divers types de nœuds optiques : depuis les terminaux de ligne optique ou *Optical Line Terminals* (OLTs), en passant par les multiplexeurs optiques d'insertion/extraction ou *Optical Add/Drop Multiplexers* (OADMs) pour arriver aux répartiteurs optiques (OXCs) [60].

Selon l'architecture des réseaux optiques WDM (voir Figure 2.1), les nœuds d'accès servent d'interface entre le réseau optique WDM cœur (constitué par l'ensemble des OXCs ou des OADMs et liens entre ces derniers) et les réseaux clients. Ces nœuds d'accès sont en pratique des OLTs [47]. Un OLT contient des transpondeurs (ou émetteurs/récepteurs), des multiplexeurs et optionnellement des amplificateurs optiques [60].

Le cœur du réseau optique peut être constitué par un ensemble de OADMs. Un OADM est un équipement qui permet d'extraire un signal optique caractérisé par la longueur d'onde λ_i (on parle de fonction *Drop*) et de le remplacer par un autre signal optique caractérisé par la même longueur d'onde λ_i (on parle de fonction *Add*) provenant d'un autre équipement [61]. Ainsi, un OADM permet d'effectuer les fonctions de commutation, de routage et de reconfiguration du routage (on parle alors de *Reconfigurable Add /Drop Multiplexers* noté ROADM) dans le réseau optique WDM cœur. Il contient des multiplexeurs, des démultiplexeurs, et des ports d'insertions et d'extractions des signaux optiques [61]. Cependant, il contient un nombre très limité de ports et ne prend pas en charge la conversion de longueur d'onde [60]. Autrement dit, on ne peut pas concevoir un réseau optique WDM ayant la propriété de conversion partielle (qui nous concerne dans cette thèse) en utilisant des OADMs. Par ailleurs, les insuffisances de l'OADM le rend inadapté pour la constitution du réseau optique WDM cœur pouvant supporter plusieurs connexions. Pour pallier cela, les répartiteurs optiques (ou OXCs) ont vu le jour. Les OXCs supportent un grand nombre de ports que les OADMs, ils peuvent prendre en charge la conversion de longueur et sont hautement reconfigurables. Ainsi, dans cette thèse qui traite de la reconfiguration du routage, les nœuds formant le cœur du réseau optique WDM sont supposés être des OXCs. Cette section décrit quelques architectures de répartiteurs optiques sur lesquelles se basent les so-

lutions du routage multicast et de l'assignation de longueur d'onde présentées dans la section 2.6.

2.5.1 Architecture générale des répartiteurs optiques

Un OXC comprend différents composants notamment des démultiplexeurs, des commutateurs optiques ou *Optical Switchs* (OSWs) et des multiplexeurs. Un commutateur optique fonctionne en commutant sélectivement un signal optique délivré par une fibre optique sur une certaine longueur d'onde vers une autre fibre [62]. Ainsi, un répartiteur optique à N ports d'entrée et N ports de sorties noté $N * N$ OXC prend en charge W longueurs d'onde et est mis en œuvre par N démultiplexeurs, W commutateurs optiques (ou OSWs) qui sont des $N * N$ commutateurs optiques (c'est-à-dire à N ports d'entrée et N ports de sortie) et N multiplexeurs comme le montre la Figure 2.5, avec N et W des entiers. Chaque port d'entrée d'un OXC est suivi d'un démultiplexeur et chaque port de sortie est précédé d'un multiplexeur. Les W OSWs sont placés au milieu de l'OXC, chacun étant responsable d'une longueur d'onde (de λ_1 à λ_w). En clair, chaque fibre connectée au port d'entrée d'un démultiplexeur optique

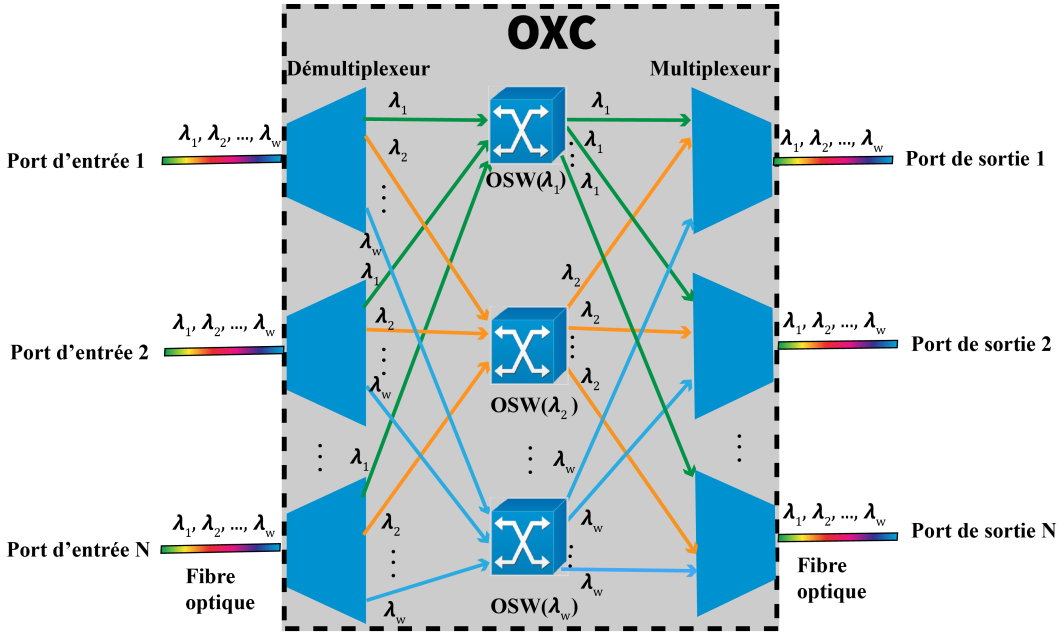


FIGURE 2.5 – Architecture générale d'un OXC [52]

de l'OXC contient W longueurs d'onde différentes. Le démultiplexeur optique sépare les longueurs d'onde entrantes en W directions. Chacune de ces directions est caractérisée par une longueur d'onde et passe par l'OSW associée à sa longueur d'onde avant d'être combinée avec les sorties des autres $W - 1$ OSWs dans un multiplexeur [63]. Ainsi, en configurant les OXCs de manière appropriée le long d'une succession de fibres, une connexion peut être établie entre n'importe quelle paire de nœuds d'accès du réseau optique WDM. Cette architecture pour OXC est adaptée pour prendre en charge les connexions unicasts. En revanche, notre thèse traite une problématique liée aux connexions multicasts. Par conséquent, nous présentons dans ce qui suit quelques OXCs qui sont compatibles également avec des connexions multicasts.

2.5.2 Répartiteurs de type duplicateur et transmetteur pour unicast et multicast

Afin de supporter les connexions multicasts, un dispositif optique appelé duplicateur optique (ou *light splitter*) doit être intégré dans un OXC. Ainsi, un commutateur de type duplicateur et transmetteur (ou *Splitter-and-Delivery*) noté SaD a été proposé dans [64] pour les commutateurs dans l'architecture d'un OXC. On obtient alors un répartiteur de type duplicateur et transmetteur (pour unicast et multicast) noté SaD-OXC. La Figure 2.6.a montre un commutateur de type duplicateur et transmetteur qui est composé de N duplicateurs optiques, de N^2 amplificateurs afin de réduire les pertes de puissance dues à la duplication optique et de N^2 commutateurs à deux ports d'entrées et un port de sortie ou $2*1$ commutateurs. Un tel commutateur de type duplicateur et transmetteur se note $N*N$ SaD. La Figure 2.6.b présente une architecture d'un SaD-OXC. Un signal optique en entrée est initialement dupliqué en N branches. Chaque branche est dirigée vers un port de sortie qui lui est associée grâce à un $2*1$ commutateur. Par conséquent, toute entrée peut être connectée à aucun, à un, à plusieurs ou à tous les ports de sortie : on dit qu'un $N * N$ SaD à la propriété non bloquante [65] et une capacité de multicasting.

Cependant, ces commutateurs SaD non bloquants ne sont ni efficaces en matière de puissance ni rentables en raison du grand nombre de duplicateurs optiques et d'amplificateurs nécessaires, ce qui entraîne une perte de puissance importante, même pour une connexion unicast. Pour pallier cet inconvénient, une autre architecture OXC a été proposée dans [66] afin de prendre en charge à la fois les connexions unicast et

multicast avec moins de pertes de puissance que les SaD-OXC.

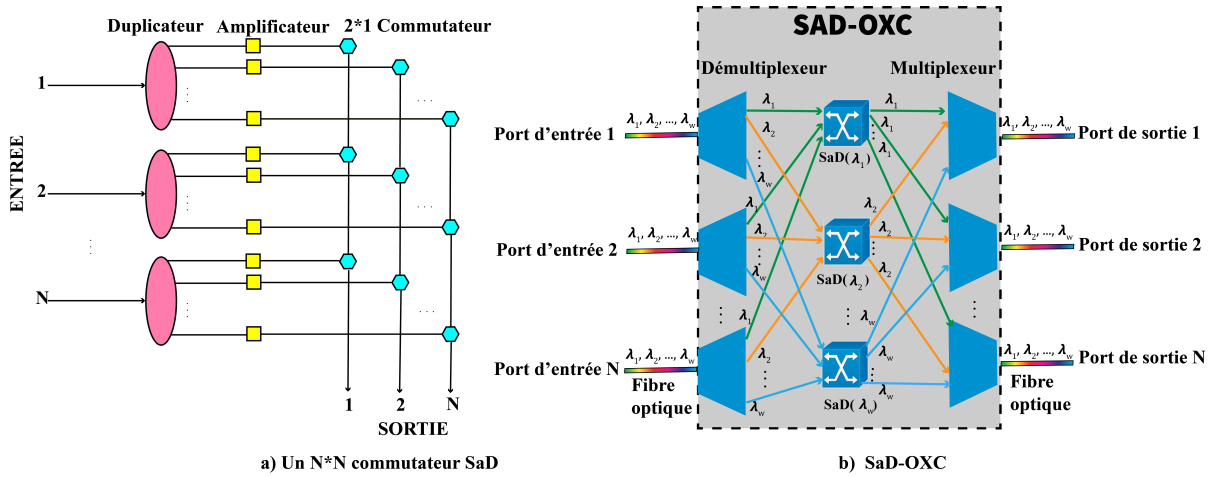


FIGURE 2.6 – Architecture d'un $N * N$ SaD-OXC [64]

2.5.3 Répartiteurs de type duplicateur et transmetteur pour multicast

Un répartiteur de type duplicateur et transmetteur pour multicast (ou *Multicast-Only Splitter-and-Delivery Optical cross-connect*) noté MOSaD-OXC, basé sur le concept de partage des duplicateurs [67] a été proposé afin de réduire la complexité de l'architecture et donc le coût financier d'un SaD-OXC. Comme le montre la Figure 2.7, le signal optique de longueur d'onde λ_a d'une connexion unicast est envoyé directement (sans transiter par un duplicateur) au multiplexeur pour un port de sortie souhaité, tandis que le signal optique de longueur d'onde λ_b d'une connexion multicast est envoyé au groupe de duplicateurs ou *Split-Switch Bank* (SSB). Dans cette architecture, chaque duplicateur d'un SSB est partagé par différentes longueurs d'onde d'entrée.

La Figure 2.8 montre comment l'architecture proposée pour les répartiteurs MOSaD-OXC applique le concept de partage des duplicateurs. Un $N * N$ MOSaD supportant W longueurs d'onde, se compose de N démultiplexeurs pour séparer les signaux optiques en entrée, de W commutateurs ou *Space Division Switch* (SDS) pour effectuer la commutation proprement dite, de N multiplexeurs pour agréger les signaux optiques sur une fibre de sortie, de W SSBs, de $W.N$ commutateurs (de type $2*1$ commutateurs) [66]. En clair, les signaux optiques en entrée sont d'abord démultiplexés et

les signaux optiques utilisant la même longueur d'onde sont dirigés vers le commutateur de type SDS correspondant. Notons qu'un SDS peut également traiter un signal

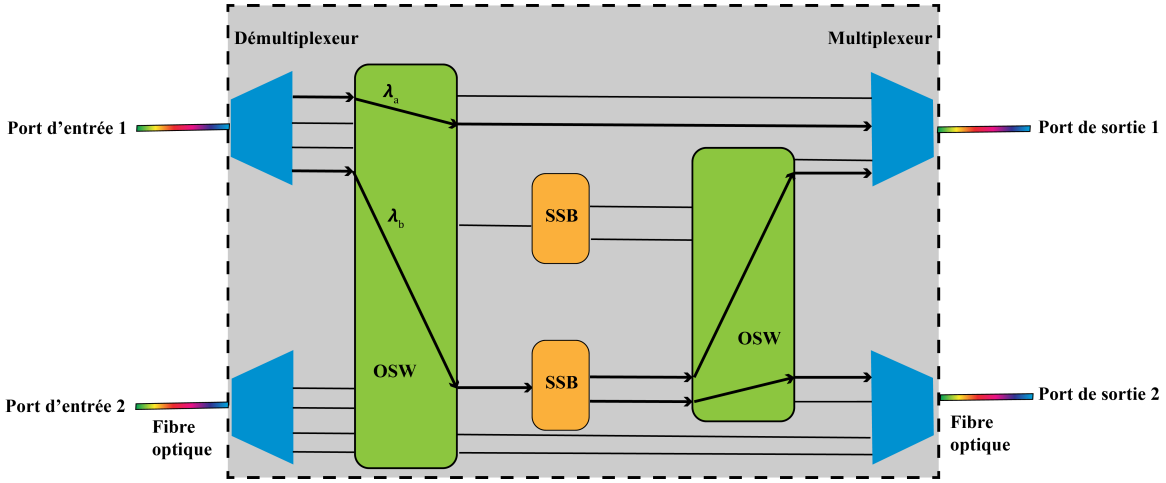


FIGURE 2.7 – Concept de partage des duplicateurs [67]

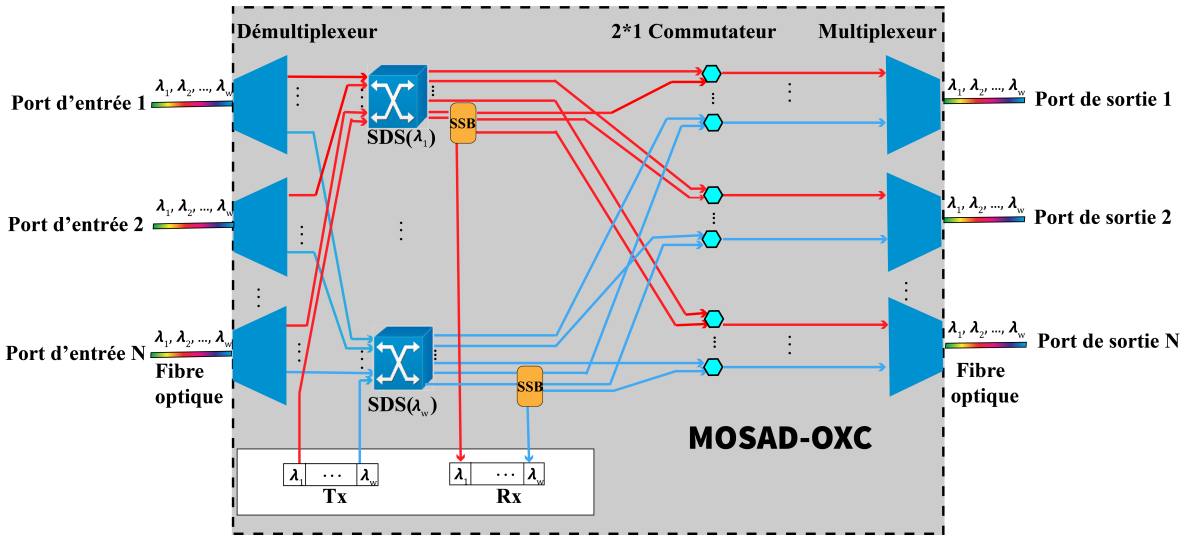


FIGURE 2.8 – Architecture d'un MOSaD-OXC [66]

optique provenant directement d'un transmetteur ou *Transceiver* noté Tx. Chaque SDS commute à son tour un signal optique d'entrée soit vers un lien (ou fibre optique) de sortie correspondant via un $2 * 1$ commutateur si ce signal concerne une connexion unicast, soit vers un SSB si ce signal concerne une connexion multicast. Un seul SSB

est attaché à chaque SDS. Un duplicateur optique appartenant au SSB duplique le signal optique (arrivant au SSB) et commute chacun des signaux optiques résultant de l'opération de duplication soit vers un lien de sortie via un $2 * 1$ commutateur ou vers un receveur (ou *Receiver* noté Rx) local.

Les répartiteurs à base de duplicateurs optiques tels que les SaD-OXCs et les MOSaD-OXCs sont appelés également des nœuds optiques ayant la capacité de multicasting ou Multicast Capable Optical cross connects (MC-OXCs). Dans cette thèse, nous supposons que tous les nœuds d'un réseau optique WDM sont des nœuds optiques ayant la capacité de multicasting.

2.5.4 Répartiteurs à convertisseurs de longueur d'onde

Afin de tirer profit des avantages de l'utilisation d'un convertisseur de longueur d'onde ou *Wavelength Converter* (WC), des architectures de répartiteurs contenant des convertisseurs de longueurs d'onde ou *Wavelength Converter Optical Cross Connect* (WC-OXC) ont vu le jour. Pour voir comment les WC-OXCs peuvent être réalisés, nous décrivons brièvement plusieurs architectures de WC-OXC proposées dans [68].

La Figure 2.9 présente une architecture de WC-OXC dite dédiée où un WC est dédié à chaque longueur d'onde au niveau de chaque port de sortie. Pour un $N * N$ WC-OXC supportant W longueurs d'onde, un total de $N.W$ convertisseurs de longueur d'onde est nécessaire. Les signaux optiques arrivant à chaque entrée sont d'abord démultiplexés en différentes longueurs d'onde. Ensuite, chaque longueur d'onde est dirigée vers le port de sortie souhaité par un commutateur OSW. Un WC dédié à chaque sortie du commutateur convertit ensuite le signal optique, si nécessaire, à la longueur d'onde de sortie souhaitée ; un multiplexeur agrège enfin les W longueurs d'onde résultantes au niveau de la fibre optique de sortie. Il est évident que cette architecture n'est pas économique, car certains signaux ne nécessitent pas forcément de conversion de longueur d'onde. Une méthode efficace consiste à partager les convertisseurs de longueur d'onde en appliquant la technique de partage (voir Figure 2.7), ce qui donne deux architectures de partage des convertisseurs de longueur d'onde : partage par nœud (voir Figure 2.10) et partage par lien (voir Figure 2.11), selon la manière dont les convertisseurs de longueur d'onde sont partagés.

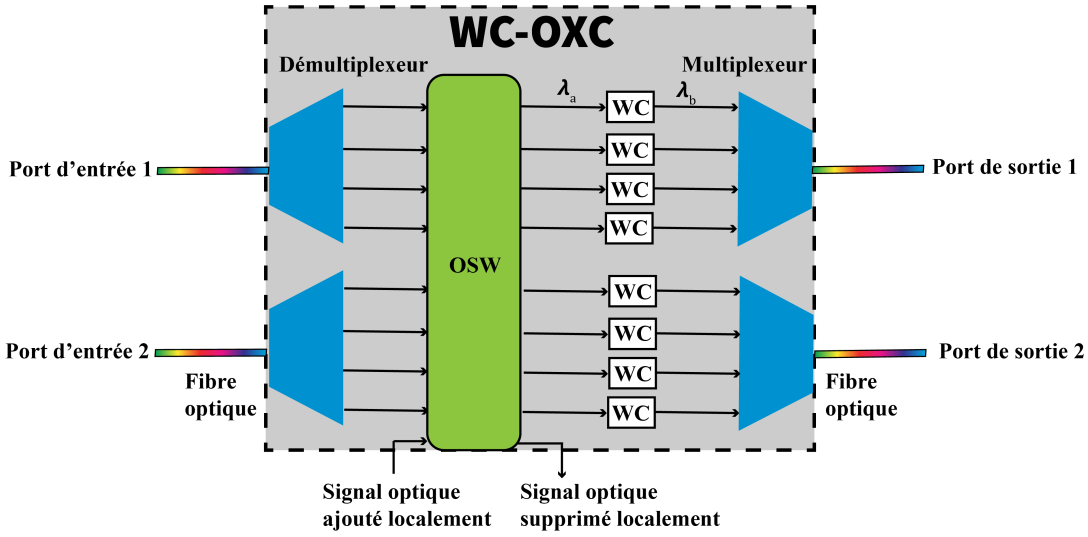


FIGURE 2.9 – Architecture dite dédiée d'un $2 * 2$ WC-OXC [68]

Dans la Figure 2.10, les convertisseurs au niveau de l'OXC sont rassemblés dans un groupe ou *Wavelength Converter Bank* (WCB). Ce groupe est accessible par n'importe quelle longueur d'onde entrante à partir de n'importe quelle entrée en configurant le grand commutateur optique (c'est-à-dire OSW1). Dans cette architecture, seules les longueurs d'onde nécessitant une conversion sont acheminées vers le WCB, ce qui permet d'économiser en matière de nombre de convertisseurs et de délai de conversion (puisque la conversion à la même longueur d'onde (par exemple, λ_a) n'est pas nécessaire). Les longueurs d'onde converties sont ensuite commutées vers les ports de sortie appropriés à l'aide du petit commutateur optique (c'est-dire OSW2). Malgré les coûts supplémentaires liés à l'ajout d'un petit commutateur et d'un plus grand nombre de ports d'entrée au niveau des multiplexeurs, les économies réalisées sur le nombre de convertisseurs dans l'architecture à partage par nœud compensent largement ces coûts supplémentaires.

La Figure 2.11 présente l'architecture d'un WC-OXC dite de partage par lien, qui prévoit un WCB pour les signaux d'un même lien (fibre) sortant. Par rapport à l'architecture à partage par nœud, cette approche à partage par lien nécessite un coût supplémentaire plus élevé, mais permet d'obtenir de meilleures performances. Elle offre un bon compromis entre le coût de fabrication et les performances obtenues avec les deux autres architectures.

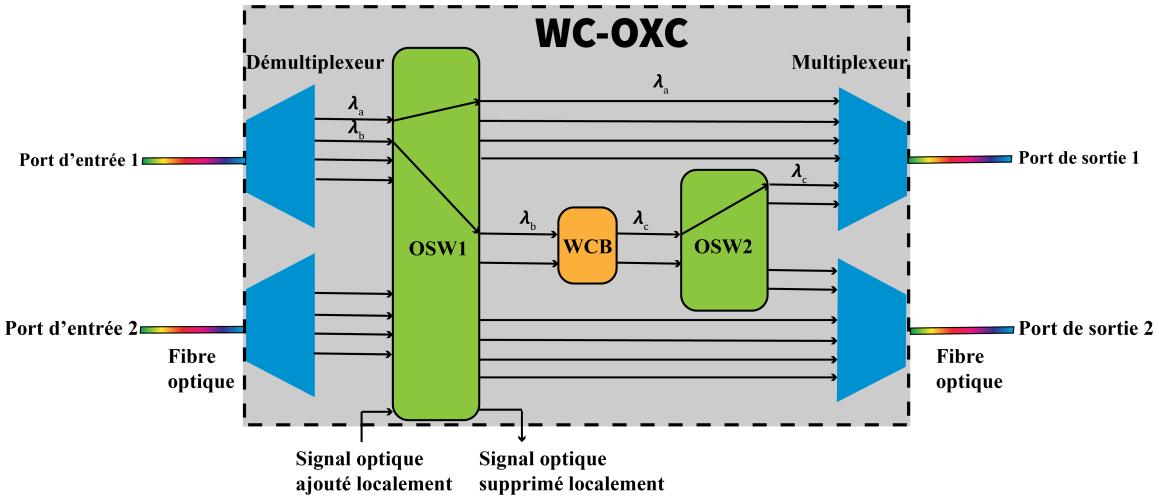


FIGURE 2.10 – Architecture dite de partage par nœud d'un $2 * 2$ WC-OXC [68]

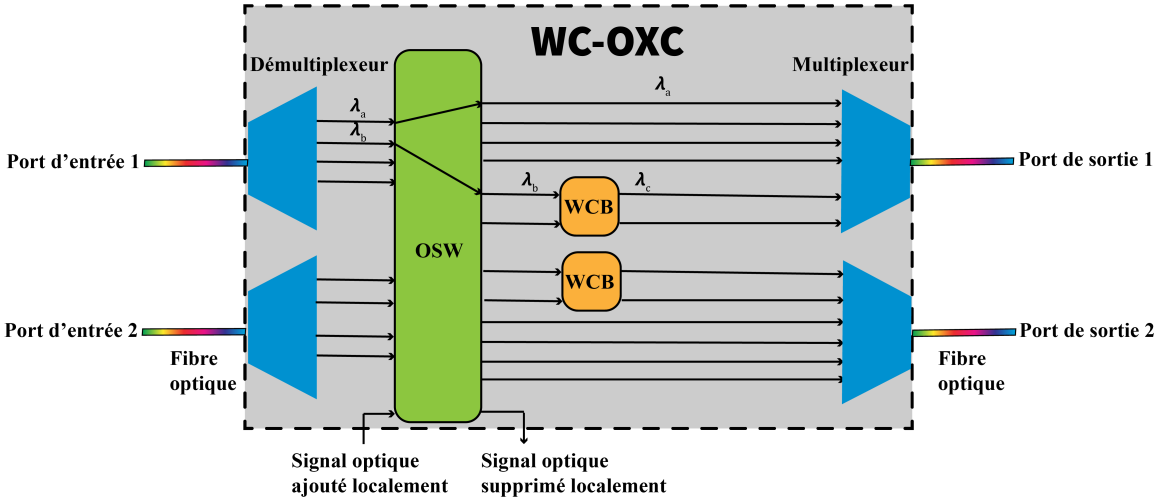


FIGURE 2.11 – Architecture dite de partage par lien d'un 2*2 WC-OXC [68]

2.6 Routage multicast et assignation de longueurs d'onde dans les réseaux optiques WDM

L'objectif du multicast est de fournir des services de communication efficaces pour des applications qui nécessitent la transmission d'un même message [1] d'une source à plusieurs destinations [69]. Dans les réseaux optiques WDM, un ensemble de données constituant un message est acheminé grâce à un flux de signaux optiques propagés dans des fibres optiques via des canaux optiques. Comme aucune conversion O/E/O n'a lieu dans les OXCs (voir section 2.2), la transmission et la duplication des signaux (correspondant aux données) optiques se font toutes dans le domaine optique. Dans ce cas, le multicast dans les réseaux optiques WDM est appelé multicast tout-optique [70].

Le multicast tout-optique présente de nombreux avantages potentiels [71]. Tout d'abord, le multicast tout-optique est efficace en matière de bande passante puisque les chemins entre la source et chaque nœud destination peuvent partager des canaux optiques [72]. Deuxièmement, la réplication des données dans les réseaux optiques WDM est plus efficace que dans les réseaux IP. Dans les réseaux optiques WDM, un MC-OXC duplique les données en utilisant directement des duplicateurs optiques, tandis que, dans les réseaux IP, les routeurs le font en copiant électroniquement les paquets en les dupliquant un par un dans la mémoire du routeur [73]. L'utilisation de duplicateurs optiques élimine également le besoin de mise en mémoire tampon qui est généralement nécessaire pour la duplication des données dans le domaine électronique. Comme il n'y a pas besoin de conversion O/E/O, le multicast tout-optique a une faible latence. Par souci de concision, le terme multicast tout-optique sera remplacé par le terme multicast dans tout le reste du manuscrit.

Au vu des nombreux avantages du multicast, des méthodes de résolution du problème de Routage Multicast et d'Assignation de Longueurs d'onde (RMAL) ont été proposées pour les réseaux optiques WDM. Le problème de RMAL est souvent découplé en deux sous-problèmes distincts : le sous-problème du routage multicast et le sous-problème de l'assignation des longueurs d'onde.

Le sous-problème du routage multicast consiste à trouver une succession de fibres de coût minimal permettant de relier un nœud source et plusieurs nœuds destinations

1. Le message est un flux de données numériques de longueur quelconque, c'est-à-dire des milliers ou des milliards de paquets

dans la topologie physique du réseau optique WDM. Autrement dit, ce sous-problème est équivalent au problème de l'arbre de Steiner [74]. Diverses heuristiques efficaces et bien connues, telles que l'arbre des plus courts chemins et l'arbre couvrant de poids minimum, peuvent être adoptées pour résoudre ce sous-problème. C'est pourquoi nous avons choisi dans cette thèse de ne pas nous focaliser sur ce point.

Après avoir résolu le sous-problème du routage multicast, le sous-problème d'assignation de longueurs d'onde vise à assigner une longueur d'onde disponible pour chaque lien de l'arbre. Cette assignation est indispensable pour établir les canaux optiques nécessaires au transfert du flux optique depuis un nœud source vers les nœuds destinations. Les algorithmes d'assignation de longueurs d'onde se subdivisent en deux groupes : les algorithmes de mono-assignation et les algorithmes de multi-assignation. Dans le premier groupe, une seule longueur d'onde est choisie pour être assignée à tous les liens de l'arbre obtenu comme solution au sous-problème du routage multicast. Dans ce groupe, on retrouve des algorithmes usuels à savoir l'algorithme du premier convenable (ou *first fit*) qui consiste à choisir la longueur d'onde disponible ayant l'indice le plus bas pour réaliser l'assignation et l'algorithme d'assignation aléatoire (ou *random fit*) qui consiste à choisir au hasard une longueur d'onde parmi l'ensemble des longueurs d'onde disponibles [75]. Quant au second groupe, plusieurs longueurs d'onde peuvent être utilisées pour réaliser l'assignation des longueurs d'onde pour un arbre. Les algorithmes de ce groupe ont pour objectif d'utiliser le moins de longueurs d'onde possible. Ces algorithmes peuvent être basés sur la programmation dynamique comme dans [76] ou basés sur l'approche de l'effet minimum en premier (ou *minimum effect first*) pour sélectionner les longueurs d'onde comme dans [77].

Comme mentionné plus haut, généralement le problème de RMAL est découplé en deux sous-problèmes (le sous-problème du routage multicast et le sous-problème de l'assignation des longueurs d'onde) qui sont résolus séparément. Par ailleurs, certaines méthodes (telles que celles dans [78, 79]) intègrent simultanément les deux sous-problèmes. Cependant, ces dernières méthodes sont assez complexes à programmer [80].

Les méthodes de résolution du problème de RMAL visent généralement à minimiser la probabilité de blocage [81]. La probabilité de blocage se définit comme le rapport du nombre de demandes de connexions bloquées sur le nombre total de demandes de connexions [75]. Une demande de connexion est dite bloquée si aucune structure

de routage ne peut être trouvée par la méthode choisie pour la résolution du problème de RMAL.

La solution finale retournée par une méthode de résolution du problème de routage multicast et d'assignation de longueurs d'onde peut prendre quatres formes possibles à savoir l'arbre mono-optique [67], l'arbre multi-optique [82], la forêt optique [83] ou la hierarchie optique [84] lorsque le réseau optique possède des répartiteurs optiques présentés à la section 2.5 qui contiennent des duplicateurs optiques pour réaliser le multicasting. Nous allons décrire dans les sections suivantes chacune de ces quatre formes.

2.6.1 Arbre mono-optique

Un arbre mono-optique (ou *light-tree*) est une structure de routage qui est un arbre et qui permet d'acheminer le flux optique en provenance d'un nœud dit nœud source vers plusieurs autres nœuds appelés nœuds destinations. La racine de cet arbre est le nœud source. Cet arbre est dit mono-optique, car tous les canaux optiques le constituant utilisent la même longueur d'onde [85]. Cela implique que pour trouver un arbre mono-optique, il faut respecter la contrainte de continuité de longueur d'onde. Autrement dit, les méthodes de routage multicast et d'assignation de longueurs d'onde retournant un arbre mono-optique ne sollicitent pas de convertisseurs de longueur d'onde.

Sur la Figure 2.12, l'ensemble des liens en noir ainsi que les nœuds forment la topologie physique du réseau optique WDM. Sur ce réseau, l'on veut établir une connexion multicast de source le nœud s et de destinations les nœuds d_1 et d_2. Supposons qu'il existe au moins une même longueur d'onde qui est disponible sur chacun des liens du réseau. Cette hypothèse élimine effectivement le besoin de conversion de longueur d'onde. Ainsi, un arbre mono-optique peut être déterminé pour transmettre le flux optique depuis le nœud s vers les destinations d_1 et d_2. Sur cette figure, l'ensemble des canaux optiques en bleu forme l'arbre mono-optique. Comme on peut le voir cet arbre est enraciné en s et les canaux optiques formant cet arbre utilisent une même longueur d'onde c'est-à-dire la longueur d'onde λ_1.

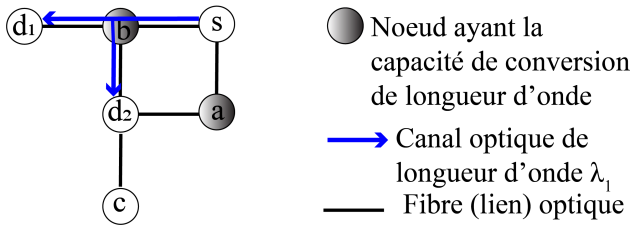


FIGURE 2.12 – Exemple d'arbre mono-optique pour la connexion $< s, \{d_1, d_2\} >$

Un arbre mono-optique est le choix privilégié pour des connexions multicasts vu sa relative simplicité de mise en œuvre. Cependant, il n'est pas toujours évident de pouvoir trouver une même longueur d'onde disponible sur des liens. Dans ce cas, l'on s'autorise à relâcher la contrainte de continuité de longueur d'onde si le réseau contient des nœuds munis de convertisseurs de longueurs d'onde.

2.6.2 Arbre multi-optique

Une structure de routage autorisant un relâchement de la contrainte de continuité de longueur d'onde est l'arbre multi-optique. Un arbre multi-optique (ou *semi-light-tree*) est également une structure de routage qui est un arbre, mais dont les canaux optiques n'utilisent pas tous la même longueur d'onde [86].

Sur la Figure 2.13.a et la Figure 2.13.b, l'ensemble des liens en noir ainsi que les nœuds forment le réseau optique WDM ayant la propriété de conversion partielle de longueur d'onde. Nous supposons que chaque lien à une capacité maximale de deux longueurs d'onde (λ_1 et λ_2). Sur la Figure 2.13.a, nous admettons deux connexions unicasts $c_1 =< d_1, \{b\} >$ et $c_2 =< d_2, \{b\} >$ via les chemins mono-optiques respectivement constitués d'un canal optique utilisant la longueur d'onde λ_1 et d'un canal optique utilisant la longueur d'onde λ_2. Supposons que l'opérateur désire établir une connexion multicast $c_3 =< s, \{d_1, d_2\} >$. On constate sur la Figure 2.13.a qu'il n'existe pas une et même longueur d'onde disponible sur les fibres reliant le nœud s aux nœuds destinations d_1 et d_2. Par conséquent, en s'autorisant à relâcher la contrainte de continuité de longueur d'onde par l'utilisation de la conversion de longueur d'onde au nœud b on obtient un arbre multi-optique. Cet arbre est matérialisé par les traits en pointillés sur la Figure 2.13.b. Comme on peut le voir, la longueur d'onde λ_2 est requise sur le lien $b - d_1$ alors que sur les liens $s - b$ et $b - d_2$ c'est la longueur d'onde λ_1 qui est requise ; cet arbre requiert donc deux longueurs d'onde différentes d'où son caractère

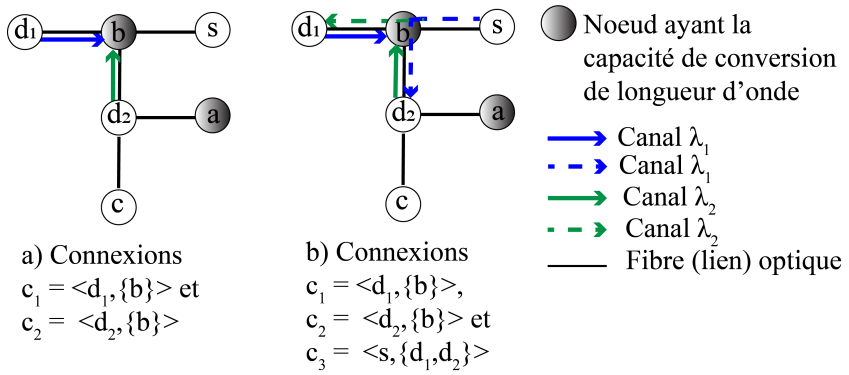


FIGURE 2.13 – Exemple d'admission de connexion par un arbre multi-optique

multi-optique.

2.6.3 Forêt optique

L'arbre mono-optique utilise la capacité de chaque nœud à dupliquer le signal dans le domaine optique. Aussi, il n'y a pas de conversion de longueur d'onde exigée au niveau d'un nœud intermédiaire de l'arbre mono-optique. À chaque nœud de branchement, le signal optique est copié en un certain nombre de signaux optiques. Notons que sur une structure de routage qui est un arbre, un nœud de branchement est un nœud ayant plusieurs liens incidents émanant de ce nœud. Autrement dit, un nœud de branchement d'un arbre est un nœud qui possède plusieurs nœuds enfants sur cet arbre. Si tous les nœuds d'un réseau sont des nœuds ayant la capacité de duplication optique alors un arbre mono-optique (ou multi-optique) est utilisé pour acheminer les données vers toutes les destinations. Dans le cas contraire, un ensemble d'arbres mono-optiques, regroupés sous forme de forêt optique (ou *light-forest*) est utilisé. Pour ce faire, la solution usuelle est d'utiliser plusieurs arbres mono-optiques enracinés au même nœud source [87].

Sur la Figure 2.14, l'ensemble des liens en noir ainsi que les nœuds forment la topologie physique du réseau optique WDM. Sur ce réseau, l'on veut établir une connexion multicast ayant pour source le nœud s et pour destinations les nœuds d_1, d_2 et d_3. Comme on peut le voir sur cette figure, le nœud b n'a pas la capacité de duplication de signal. Par conséquent, on construit un premier arbre mono-optique (voir ensemble des traits de couleur bleue) enraciné au nœud s afin de couvrir les destinations d_1 et

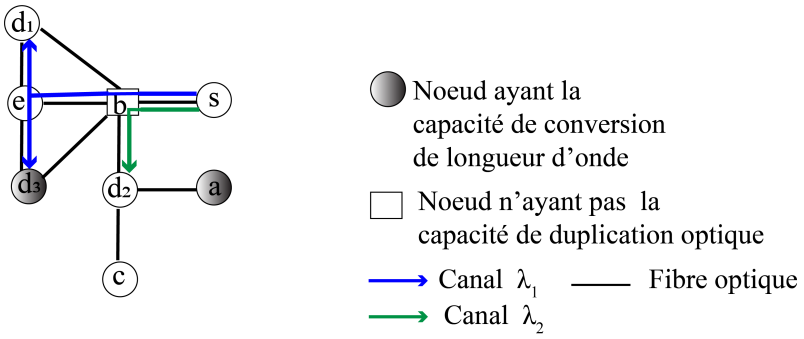


FIGURE 2.14 – Exemple de forêt optique

d_3. Puis, un second arbre mono-optique (voir ensemble des traits de couleur verte) enraciné au nœud s afin de couvrir la destination d_2. Ainsi, pour transmettre le flux optique provenant du nœud s vers les nœuds d_1, d_2 et d_3, un ensemble de deux arbres mono-optiques enracinés au nœud s a été créé : cela s'appelle une forêt optique.

2.6.4 Hiérarchie optique

La forêt optique a été proposée comme structure de routage lorsqu'un nombre réduit de nœuds sont dotés de duplicateurs optiques. Cependant, cette structure de routage n'est pas optimale lorsque certains nœuds n'ayant pas de duplicateurs optiques sont des nœuds de degré supérieur ou égale à 4 dans le réseau [84]. Par conséquent, une autre structure de routage appelé la hiérarchie optique (ou *light-hierarchy*) a été introduite. La hiérarchie optique est une extension de l'arbre mono-optique dans laquelle les cycles sont permis, car le signal optique peut passer plusieurs fois par le même nœud, mais jamais deux fois par le même canal optique [88]. Autrement dit, la hiérarchie optique est une forme d'arbre non-élémentaire [89].

Sur la Figure 2.15, l'ensemble des liens en noir ainsi que les nœuds forment la topologie physique du réseau optique WDM comportant des nœuds sans duplicateur optique. Cette figure montre différentes structures de routage utilisable par la connexion multicast $< s, \{d_1, d_2, d_3\} >$. Ainsi, la Figure 2.15.a présente la hiérarchie optique matérialisée par l'ensemble des traits continus de couleur bleue. Comme on peut le voir, le signal optique passe deux fois par le nœud b sans jamais utiliser deux fois le même canal optique. Quant à la Figure 2.15.b, elle présente la forêt optique matérialisée par l'ensemble des traits continus de couleur bleue et des traits continus de couleur verte.

Il est évident, d'après ces deux sous-figures que la hiérarchie optique est optimale, car elle requiert moins de canaux optiques que la forêt optique.

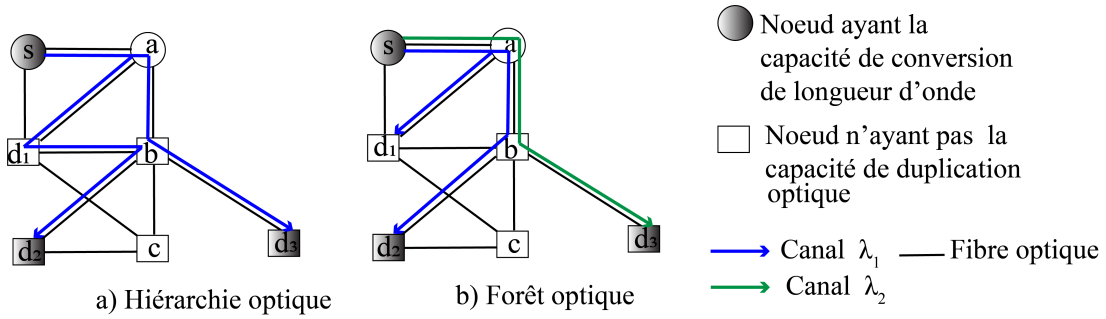


FIGURE 2.15 – Structures de routage et optimalité

2.6.5 Classification des solutions de routage multicast et d'assignation de longueurs d'onde

Les méthodes de routage multicast et d'assignation de longueur d'onde se basent sur la présence totale ou partielle ou l'absence de certains équipements au niveau des nœuds de ce réseau. En effet, certaines méthodes considèrent que des duplicateurs optiques sont présents au niveau de tous les nœuds (c'est-à-dire une présence totale) du réseau optique WDM tandis que d'autres considèrent que des duplicateurs optiques sont présents uniquement sur certains nœuds (c'est-à-dire une présence partielle). De plus, certaines méthodes considèrent la distribution des convertisseurs de longueurs d'onde optiques (présence totale ou partielle) sur l'ensemble des nœuds du réseau optique WDM ou non. Le Tableau 2.1 récapitule les résultats possibles des méthodes de routage multicast et d'assignation de longueurs d'onde dans les réseaux optiques WDM en fonction de la présence (totale ou partielle) des duplicateurs optiques et de la présence (totale ou partielle) ou l'absence des convertisseurs de longueur d'onde.

2.7 Conclusion

Dans ce chapitre, il a été question d'abord de présenter l'architecture générale d'un réseau optique WDM. Puis les technologies de base utilisées par ce type de réseau ont été abordées. Il s'agit du WDM et de la conversion de longueur d'onde.

Tableau 2.1 – Récapitulatif des résultats des méthodes de routage multicast

Convertisseurs de longueur d'onde \ Duplicateurs optiques	Présence totale	Présence partielle
Présence partielle ou totale	Arbre mono-optique [67], Arbre multi-optique [82]	Forêt optique [83], Hiérarchie optique [84]
Absence ou non présence	Arbre mono-optique [67]	Forêt optique [83], Hiérarchie optique [84]

Après cela, une présentation minutieuse des répartiteurs optiques a été faite. Cela nous a permis de comprendre que les répartiteurs optiques sont des éléments très importants dans un réseau optique WDM. En effet, ils participent au transport des flux (de signaux) optiques de connexions (unicast et multicast). Dans le cadre d'une connexion multicast, un type de répartiteurs noté MC-OXC est nécessaire. Un MC-OXC est doté d'un équipement appelé duplicateur optique qui permet d'envoyer un signal reçu en entrée sur plusieurs ports de sortie. Notons qu'un MC-OXC peut contenir des convertisseurs optiques afin de contribuer à la réduction de la probabilité de blocage dans le réseau.

Dans cette thèse, on fait l'hypothèse que tous les nœuds ont la capacité de duplication optique. Autrement dit, tous les nœuds sont des MC-OXCs. Par ailleurs, notre thèse porte sur les réseaux optiques WDM ayant la propriété de conversion partielle de longueur d'onde. Autrement dit, c'est seulement un nombre restreint de nœuds qui ont la capacité de conversion de longueur d'onde. Par conséquent, les deux solutions possibles au problème du routage multicast et d'assignation de longueur d'onde dans un réseau optique WDM ayant la propriété de conversion partielle de longueur d'onde sont l'arbre mono-optique et l'arbre multi-optique. La transmission du flux optique via un arbre multi-optique peut entraîner des temps de transmission longs. En effet, les MC-OXCs moins coûteux utilisent une structure de partage de convertisseurs de longueurs d'onde (voir Figure 2.10 et Figure 2.11) qui peuvent entraîner des retards de transmissions de flux optique si aucun convertisseur de longueur d'onde n'est disponible à un moment donné. L'arbre mono-optique à cause de sa simplicité entraîne des temps de transmission moins longs que les arbres multi-optiques. Il en résulte que l'arbre mono-optique est très avantageux pour les opérateurs réseau. Par conséquent,

dans cette thèse, nous supposons uniquement que la solution de routage multicast et d'assignation de longueur d'onde est l'arbre mono-optique.

Le chapitre 1 a mis en lumière le fait que le problème général étudié dans cette thèse est double : le problème général de mono-reconfiguration du routage multicast et le problème général de multi-reconfiguration du routage multicast. Dans les deux cas, la solution au problème de routage multicast et d'assignation de longueurs d'onde doit être prise en compte. Par souci de concision, la solution est désignée par le terme routage multicast. Ainsi, dans le cas de la mono-reconfiguration du routage multicast, nous considérons que les routages multicasts initial et final sont des arbres mono-optiques. Autrement dit, la paire de routages (routage initial, routage final) est une paire d'arbres mono-optiques : on parle alors de reconfiguration d'une paire d'arbres mono-optiques. De même, dans le cas de la multi-reconfiguration du routage multicast, nous considérons que les routages multicasts initial et final de chaque connexion sont des arbres mono-optiques : on parle alors de reconfiguration d'un ensemble de plusieurs paires d'arbres mono-optiques. La Figure 2.16 présente un résumé des caractéristiques des deux problèmes spécifiques qui sont traités dans les chapitres suivants.

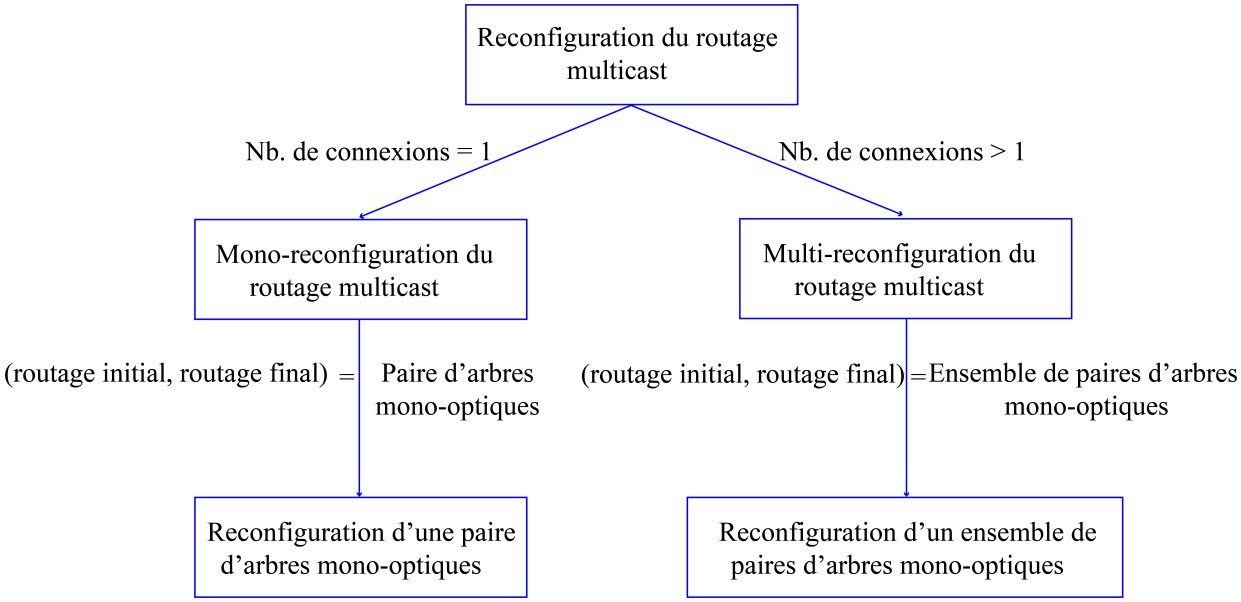


FIGURE 2.16 – Classification des problèmes à traiter

CHAPITRE 3

PROBLÈMES DE RECONFIGURATION DE PAIRES D'ARBRES MONO-OPTIQUES

Sommaire

3.1 Introduction **37**

3.2 Concepts fondamentaux **37**

3.2.1 Opérations de configuration d'un nœud optique 37

3.2.2 Définitions 39

3.2.3 Illustration des concepts 40

3.3 Problème de reconfiguration d'une paire d'arbres mono-optiques **42**

3.3.1 Description du problème de reconfiguration d'une paire d'arbres mono-optiques 42

3.3.2 Travaux connexes à la reconfiguration d'une paire d'arbres mono-optiques 44

3.3.3 Récapitulatif et implications 51

3.4 Problème de reconfiguration d'un ensemble de plusieurs paires d'arbres mono-optiques **52**

3.4.1 Description du problème de reconfiguration d'un ensemble de plusieurs paires d'arbres mono-optiques 53

3.4.2 Travaux connexes à la reconfiguration d'un ensemble de plusieurs paires d'arbres mono-optiques 54

3.4.3 Récapitulatif et implications 59

3.5 Conclusion **61**

3.1 Introduction

La reconfiguration du routage multicast est un mécanisme important qu'un opérateur réseau utilise afin d'optimiser les performances de son réseau comme indiqué au chapitre 1 de cette thèse. Le chapitre 1 a également mis en lumière le fait que plusieurs problèmes doivent être traités à chaque phase du processus de reconfiguration afin que ce mécanisme soit profitable à l'opérateur réseau. Comme mentionné à la fin du chapitre précédent, nous traitons dans cette thèse le problème lié à la reconfiguration d'une paire d'arbres mono-optiques puis celui lié à la reconfiguration d'un ensemble de plusieurs paires d'arbres mono-optiques. Le premier problème implique que l'opérateur décide de reconfigurer une seule connexion multicast tandis que le second problème implique que l'opérateur décide de reconfigurer au moins deux connexions multicasts. En pratique, la reconfiguration passe par l'exécution d'opérations dites de configuration permettant l'établissement ou la suppression de canaux optiques. Par conséquent, nous présentons d'abord les concepts liés aux différentes opérations de configurations (voir section 3.2). Puis nous présentons dans ce chapitre (voir section 3.3) le problème de reconfiguration d'une seule paire d'arbres mono-optiques et les travaux afférant à ce problème. Enfin, nous présentons le problème de reconfiguration d'un ensemble de plusieurs paires d'arbres mono-optiques et les travaux relatifs à ce problème dans la section 3.4 avant de conclure ce chapitre dans la section 3.5.

3.2 Concepts fondamentaux

Dans cette section, nous présentons l'ensemble des opérations de configuration de nœuds [90] et quelques définitions qui sont utiles pour comprendre le reste de ce chapitre.

3.2.1 Opérations de configuration d'un nœud optique

Six opérations de configuration sont applicables à un nœud optique x. Ce sont :

— L'ajout de commutation de longueur d'onde noté ADD. On désigne par $ADD\left(x_{\lambda}^{pe,\{ps_1,\ldots,ps_k\}}\right)$, l'opération d'ajout de commutation de longueur d'onde appliquée au nœud optique x. Cette opération appliquée au nœud optique x,

confère à ce nœud optique la capacité de commuter un flux optique de longueur d'onde λ reçu par son port d'entrée pe vers un ou plusieurs ports de sortie $(ps_1, \ldots, ps_k)$ avec la même longueur d'onde λ. En d'autres termes, cette opération notifie au nœud x de commuter un flux optique donné utilisant une longueur d'onde donnée d'un port d'entrée donné vers un ou plusieurs ports de sortie donnés sans changer la longueur d'onde.

— La suppression de commutation de longueur d'onde noté DEL. On désigne par $DEL\left(x_{\lambda}^{pe,\{ps_1,\ldots,ps_k\}}\right)$, l'opération de suppression de commutation de longueur d'onde appliquée au nœud optique x. Cette opération appliquée au nœud optique x, ôte à ce nœud optique la capacité de commuter un flux optique de longueur d'onde λ reçu par son port d'entrée pe vers un ou plusieurs ports de sortie $(ps_1, \ldots, ps_k)$. En d'autres termes, cette opération notifie au nœud x de ne plus commuter un flux optique donné utilisant une longueur d'onde donnée d'un port d'entrée donné vers un ou plusieurs ports de sortie donnés.

— La conversion de longueur d'onde notée $CONV$. On désigne par $CONV\left(x_{\lambda_i,\lambda_o}^{pe,\{ps_1,\ldots,ps_k\}}\right)$, l'opération de conversion de longueur d'onde appliquée au nœud optique x. Cette opération appliquée au nœud optique x, confère à ce nœud optique la capacité de commuter un flux optique de longueur d'onde λ_i reçu via son port d'entrée pe vers un ou plusieurs ports de sortie $(ps_1, \ldots, ps_k)$ après avoir converti la longueur d'onde λ_i en une autre longueur d'onde λ_o.

— Le basculement noté $BASC$. On désigne par $BASC\left(x_{\lambda_i,\lambda_o}^{pe,\{ps_1,\ldots,ps_k\},\{ps'_1,\ldots,ps'_k\}}\right)$, l'opération de basculement appliquée au nœud optique x. Ainsi, si un flux optique de longueur d'onde λ_i arrivant au port d'entrée pe du nœud optique x est initialement commuté vers un ou plusieurs ports de sortie $(\{ps_1, \ldots, ps_k\})$, alors cette opération va permettre la commutation du flux optique non plus désormais vers ces précédents ports, mais plutôt vers plusieurs autres ports de sortie $(\{ps'_1, \ldots, ps'_k\})$ sur la longueur d'onde λ_0. La longueur d'onde λ_i peut être différente de la longueur d'onde λ_o. Il est important de noter que, $BASC\left(x_{\lambda_i,\lambda_o}^{pe,\{ps_1,\ldots,ps_k\},\{ps'_1,\ldots,ps'_k\}}\right)$ est composée d'une suppression de commutation de longueur d'onde λ_i au nœud optique x et soit de l'ajout de commutation de longueur d'onde λ_o (dans le cas où $\lambda_o = \lambda_i$) soit de la conversion de la longueur d'onde λ_i en la longueur d'onde λ_o (dans le cas où $\lambda_o \neq \lambda_i$).

— La convergence notée $CONVG$. On désigne par $CONVG\left(x_{\lambda}^{\{pe_1,pe_2\},\{ps_1,\ldots,ps_k\}}\right)$, l'opération de convergence appliquée au nœud optique x. Ainsi, le flux optique

utilisant la longueur d'onde λ peut converger désormais au nœud optique x par deux de ses ports d'entrées (pe_1 et pe_2) sans changer les ports de sortie ($\{ps_2, \ldots, ps_k\}$) vers lesquels le flux est commuté.

— La non-convergence notée $NCONVG$. On désigne par $NCONVG\left(x_\lambda^{\{pe_1,pe_2\},\{ps_1,\ldots,ps_k\}}\right)$, l'opération de non-convergence appliquée au nœud optique x. Cette opération est appliquée à la suite de l'opération $CONVG\left(x_\lambda^{\{pe_1,pe_2\},\{ps_1,\ldots,ps_k\}}\right)$. En clair, la non-convergence a pour effet d'annuler l'effet de l'opération de convergence. Autrement dit, l'opération de non-convergence sur le nœud optique x assure que le flux optique utilisant la longueur d'onde λ arrive effectivement via un seul port d'entrée c'est-à-dire le port d'entrée pe_2 (en interdisant une arrivée du flux optique via le port d'entrée pe_1) sans changer les ports de sortie ($\{ps_2, \ldots, ps_k\}$) vers lesquels le flux est commuté.

Avant de décrire le problème de la reconfiguration, donnons quelques définitions utiles pour la suite du manuscrit.

3.2.2 Définitions

Définition 3.1 Soit $T = (V, E)$ un arbre enraciné au nœud r, où E est l'ensemble des liens, V est l'ensemble des nœuds de T et $r \in V$. $ST = (V', E')$ est un sous-arbre de l'arbre T si ST est un arbre et que $V' \subseteq V, E' \subseteq E$. Cela implique que ST est enraciné en un nœud (pas forcément la racine r) de T et les nœuds feuilles (qui sont des nœuds de T) de ST ne sont pas forcément identiques aux nœuds feuilles de T. En clair, un sous-arbre ST de l'arbre T est un arbre inclus dans T. Une branche B de l'arbre T enracinée au nœud r est un chemin sur T reliant le nœud r à un nœud feuille (ou nœud sans successeur) de T. Par conséquent, une branche est un type particulier de sous-arbre.

Définition 3.2 Un nœud de basculement est un nœud configuré avec l'opération de basculement.

Définition 3.3 Un canal optique noté $f_{x,y}(\lambda)$ est une voie de transmission caractérisée par la longueur d'onde λ.

Ce canal est établi sur la fibre (ou lien) optique f afin d'acheminer un flux de signaux optiques depuis le nœud x vers le nœud y. Pour établir ce canal, il faut d'une part que

le nœud x soit configuré pour acheminer le flux optique vers le port dit port de sortie de x auquel est connecté la fibre f. D'autre part, il faut que le nœud y soit configuré pour recevoir le flux optique sur le port dit port d'entrée auquel est connecté la fibre f.

Remarque 3.1 Si l'un de ces nœuds n'est pas configuré alors on parle de semi-canal optique. Notons qu'un semi-canal ne permet pas une transmission de flux entre les deux nœuds extrémités de ce canal vu qu'un de ces deux nœuds n'est pas encore configuré.

Remarque 3.2 Le pré-établissement d'un sous-arbre (et donc aussi d'une branche) consiste à le préparer à transporter un flux optique en configurant les nœuds de ce sous-arbre hormis son nœud racine. Ainsi, un sous-arbre préétabli contient au moins un semi-canal. En d'autres termes, juste après avoir préétabli un sous-arbre, aucun flux optique ne peut le traverser.

3.2.3 Illustration des concepts

La Figure 3.1 présente l'effet de l'exécution de certaines opérations de configuration sur le réseau. Sur chacune des sous-figures, l'ensemble des liens et des nœuds en noir forme la topologie physique du réseau. Sur la Figure 3.1.a, il est impossible au flux optique provenant du nœud s d'être commuté du nœud a vers le nœud b. Ainsi, la configuration du nœud a via l'opération $ADD(a_{\lambda_1}^{s,\{b\}})$ permet d'établir un canal optique caractérisé par la longueur d'onde λ_1 entre le nœud a et le nœud b tel que présenté par la Figure 3.1.b. Ainsi, le flux optique provenant du nœud s sera commuté du nœud a vers le nœud b.

Puisqu'une seule fibre relie toute paire de nœuds alors pour chaque opération illustrée sur la Figure 3.1, sans perte de généralité, les ports de sortie (respectivement d'entrée) ont été remplacés par les nœuds fils (respectivement parents) du nœud auquel est appliquée ladite opération. C'est possible sans ambiguïté, car dans notre réseau il y a au plus un seul lien entre toutes paires de nœuds. Ainsi, pour $ADD(a_{\lambda_1}^{s,\{b\}})$, le port d'entrée est remplacé par le nœud s et le port de sortie est remplacé par le nœud b. On obtient une illustration de l'opération de suppression de commutation de longueur d'onde en considérant la Figure 3.1.b comme étant la situation avant l'exécution de $DEL(a_{\lambda_1}^{s,\{b\}})$ et la Figure 3.1.a comme étant la situation après l'exécution de $DEL(a_{\lambda_1}^{s,\{b\}})$.

Sur la Figure 3.1.c, au nœud c, le flux de signaux optiques arrivant par le port d'entrée reliant le nœud c au nœud b sort par le port de sortie reliant le nœud c au nœud d. L'exécution de $CONVG\left(c_{\lambda_1}^{\{b,a\},\{d\}}\right)$ accorde la capacité au nœud c de recevoir le flux optique via le nœud a comme on peut le voir sur la Figure 3.1.d. Cependant, le nœud a n'est pas encore configuré pour commuter le flux vers le nœud c. Cela se matérialise par un semi-canal optique reliant le nœud a au nœud c sur la Figure 3.1.d. On obtient une illustration de l'opération de non-convergence en considérant la Figure 3.1.d comme étant la situation avant l'exécution de $NCONVG\left(c_{\lambda_1}^{\{b,a\},\{d\}}\right)$ et la Figure 3.1.c comme étant la situation après l'exécution de $NCONVG\left(c_{\lambda_1}^{\{b,a\},\{d\}}\right)$.

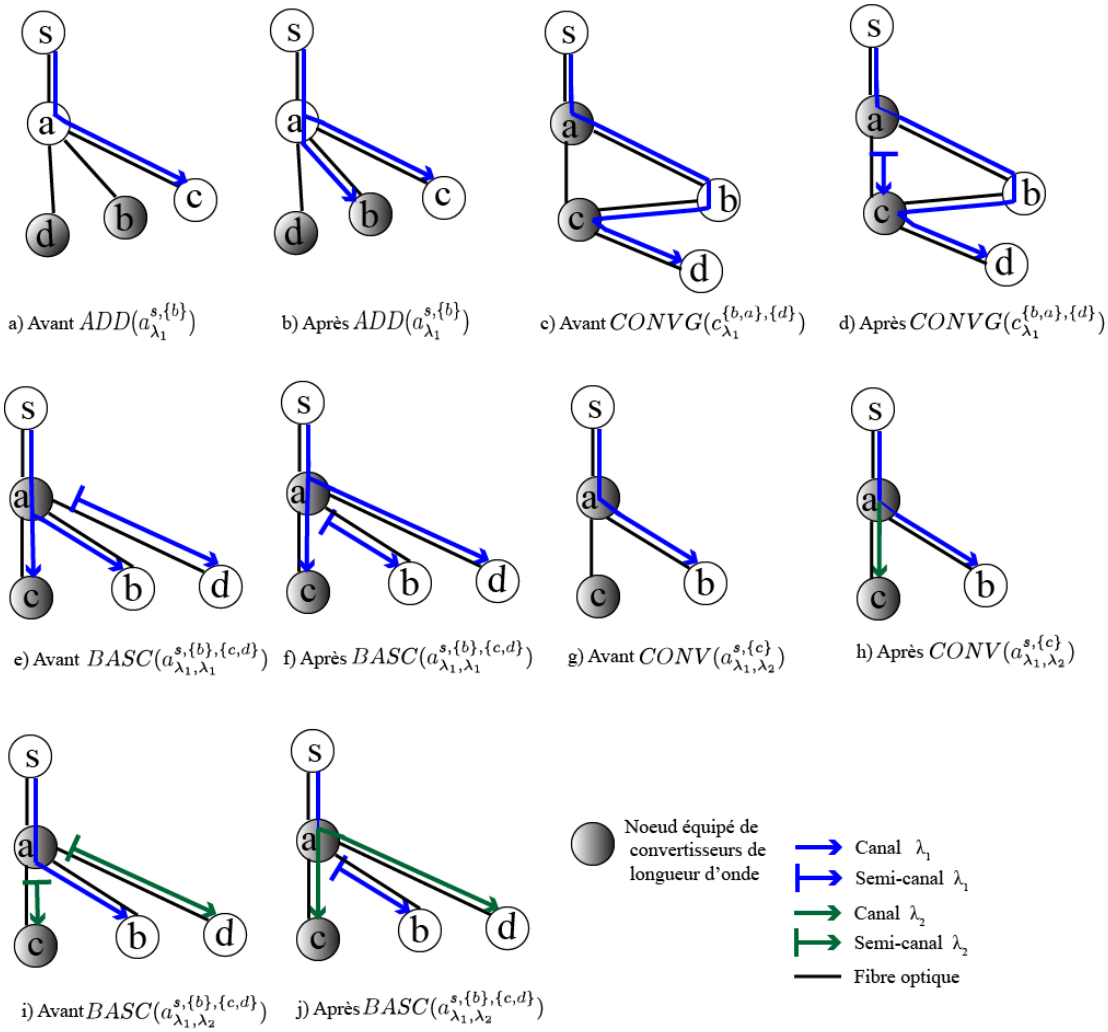


FIGURE 3.1 – Illustration de l'exécution d'opérations de configuration

Sur la Figure 3.1.g, il est impossible au flux optique provenant du nœud s d'être commuté (au nœud a) vers le nœud c. En effet, aucun canal optique n'est établi entre le nœud a et le nœud c. Ainsi, la configuration du nœud a via l'opération $CONV(a^{s,\{c\}}_{\lambda_1,\lambda_2})$ permet de créer un canal optique caractérisé par la longueur d'onde λ_2 entre le nœud a et le nœud c. En clair, le flux optique provenant du nœud s est commuté (au nœud a) vers le nœud c après conversion de la longueur d'onde utilisée λ_1 en la longueur d'onde λ_2 comme on peut le voir sur la Figure 3.1.h.

Sur la Figure 3.1.e, au nœud a, le flux optique (utilisant la longueur d'onde λ_1) arrive par le port d'entrée reliant le nœud s au nœud a et est transféré avec la même longueur d'onde λ_1 au port de sortie reliant le nœud a au nœud b. Pour que le flux optique bascule du port de sortie reliant le nœud a au nœud b vers le port de sortie reliant le nœud a au nœud c et le port de sortie reliant le nœud a au nœud d, on configure le nœud a avec l'opération $BASC\left(a^{s,\{b\},\{c,d\}}_{\lambda_1,\lambda_1}\right)$. La Figure 3.1.f montre le résultat de cette configuration. Si l'on veut que le flux optique parvienne aux nœuds c et d via la longueur d'onde λ_2 (voir Figure 3.1.j) lors du basculement alors c'est l'opération $BASC\left(a^{s,\{b\},\{c,d\}}_{\lambda_1,\lambda_2}\right)$ qui est utilisée pour configurer le nœud a.

3.3 Problème de reconfiguration d'une paire d'arbres mono-optiques

Le premier problème de reconfiguration du routage multicast étudié dans cette thèse est la reconfiguration d'une seule paire d'arbres mono-optiques. Ainsi, chacune des sous-sections suivantes est consacrée respectivement à la description du problème et à la présentation des travaux qui proposent des méthodes pouvant servir de solution à ce problème.

3.3.1 Description du problème de reconfiguration d'une paire d'arbres mono-optiques

L'amélioration de la performance du réseau par la réduction de la probabilité de blocage est une des motivations pour laquelle l'opérateur réseau a recours à la reconfiguration du routage en général et en particulier à la reconfiguration d'une paire d'arbres mono-optiques. Sur les Figures 3.2.a à 3.2.c, l'ensemble des liens et des

nœuds en noir forme la topologie physique du réseau. Ainsi, on suppose ici, qu'une seule fibre relie toute paire de nœuds. La Figure 3.2.a montre trois connexions établies. On a la connexion multicast $< s, \{d_1, d_2\} >$ dont la structure de routage établi est un arbre mono-optique dit arbre mono-optique initial, c'est-à-dire l'arbre $s \rightarrow a \rightarrow \{d_1, d_2\}$. Cette configuration d'arbre est formée par l'ensemble des canaux optiques établis (utilisant la longueur d'onde λ_1) qui correspond à l'ensemble des traits en bleu continu. Par ailleurs, il y a la connexion unicast $< a, \{c\} >$ dont le chemin établi, est formé par un ensemble de canaux optiques utilisant la longueur d'onde λ_2. Cet ensemble de canaux est matérialisé par l'ensemble des traits continus en vert successivement sur la fibre optique reliant le nœud a au nœud d_2 et sur celle reliant le nœud d_2 au nœud c. Puis, il y a la connexion unicast $< b, \{d_2\} >$ dont le chemin établi est formé par un canal optique utilisant la longueur λ_2. Il est représenté par le trait continu en vert sur la fibre optique reliant le nœud b au nœud d_2. En admettant que chaque fibre optique a une capacité de deux longueurs d'onde disponibles pour l'opération de routage et d'assignation de longueur d'onde, une nouvelle connexion unicast $< a, \{d_2\} >$ ayant pour source le nœud a et pour destination le nœud d_2 est une connexion bloquée. En effet, aucun chemin (mono ou multi-) optique ne peut être obtenu, car aucune succession de canaux optiques d'une même longueur d'onde (c'est-à-dire en respectant la contrainte de continuité) ou non (c'est-à-dire par relâchement de la contrainte de la continuité de longueur d'onde) n'est possible pour relier le nœud a au nœud d_2. L'opérateur réseau décide donc de recalculer un autre routage pour la connexion multicast $< s, \{d_1, d_2\} >$. Ce nouveau routage est un arbre mono-optique dit arbre mono-optique final. Cette configuration d'arbre est formée par l'ensemble des canaux optiques à établir en bleu discontinu sur la Figure 3.2.b. Ainsi, après avoir configuré cet arbre mono-optique final pour établir ses canaux optiques et configurer l'arbre mono-optique initial pour supprimer ses canaux optiques, la demande de connexion unicast $< a, \{d_2\} >$ est alors acceptée car il y a un chemin mono-optique représenté par le canal optique (en bleu continu) sur la fibre optique entre le nœud a et le nœud d_2 comme illustré par la Figure 3.2.c. Le remplacement de la configuration de l'arbre initial (ou configuration initiale) par la configuration de l'arbre final (ou configuration finale) est appelé reconfiguration d'une paire d'arbres mono-optiques, car certains canaux optiques peuvent appartenir aux deux arbres. Ce partage de canaux optiques oblige à passer par des configurations intermédiaires tout en évitant des interruptions de flux optiques vers les nœuds destinations de la connexion multicast. Il est important de souligner qu'une configura-

tion est définie comme un ensemble composé de canaux optiques et éventuellement de semi-canaux optiques.

Le problème de reconfiguration d'une paire d'arbres mono-optiques est énoncé comme suit : étant donné un arbre mono-optique initial (ou configuration initiale) utilisé pour transmettre le flux d'une connexion multicast et l'arbre mono-optique final (ou configuration finale) à établir pour transmettre le flux optique de la même connexion multicast, quelle est la séquence de configurations permettant de migrer de l'arbre mono-optique initial à l'arbre mono-optique final sans interrompre la transmission de flux en cours vers les nœuds destinations de la connexion multicast ? Comme mentionné au chapitre 1, les interruptions de flux sont à éviter. En effet, elles peuvent entraîner une grande gêne aux clients et des pénalités financières que l'opérateur réseau doit verser aux clients sinistrés. Aussi, dans une certaine mesure, l'opérateur peut perdre des clients pour cause de mauvaise qualité réseau.

3.3.2 Travaux connexes à la reconfiguration d'une paire d'arbres mono-optiques

Les premiers travaux (dans [91-94]) portant sur la reconfiguration d'une paire d'arbres mono-optiques se sont focalisés sur comment calculer l'arbre mono-optique final de façon optimale sans insister sur la manière de migrer le flux optique de la configuration initiale (ou arbre mono-optique initial) vers la configuration finale (ou arbre mono-optique final). Cela a motivé une série de travaux fortement liés au problème de reconfiguration (d'une paire d'arbres mono-optiques) énoncé à la section 3.3.1.

Les travaux fortement en lien avec le problème de reconfiguration d'une paire d'arbres mono-optiques peuvent être classés en deux groupes selon que ces travaux requièrent ou non des longueurs d'onde additionnelles ou *spare wavelengths*. Une longueur d'onde additionnelle est une longueur d'onde non requise par l'ensemble des structures de routage établies contenant dans notre cas l'arbre mono-optique initial et l'ensemble des structures de routage (à établir) contenant dans notre cas l'arbre mono-optique final [95]. Notons que les longueurs d'onde additionnelles correspondent à des ressources critiques donc utilisables en dernier recours et de façon temporaire pour améliorer les performances du réseau. A part la reconfiguration, elles sont utilisables pour des opérations de résilience du réseau telles que la protection de connexions [96].

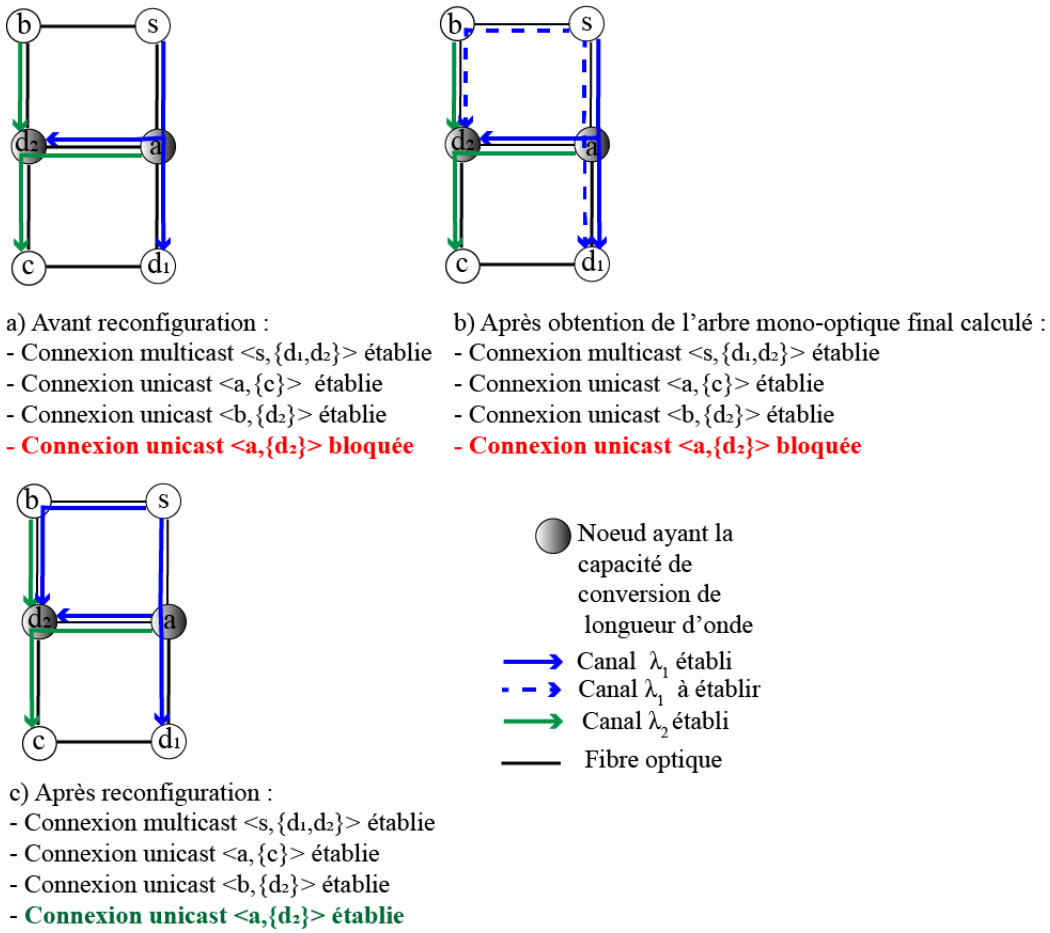


FIGURE 3.2 – Exemple d'amélioration de performance par reconfiguration d'une paire d'arbres mono-optiques

3.3.2.1 Reconfiguration sans utilisation d'une longueur d'onde additionnelle

Pour reconfigurer le routage unicast dans les réseaux optiques WDM sans interruption de flux, *Make-Before-Break* (MBB) [19, 97] est l'une des politiques les plus utilisées. MBB consiste à préétablir d'abord le nouveau chemin, puis à migrer le flux optique vers le nouveau chemin avant de libérer les ressources (c'est-à-dire les canaux optiques) utilisées par l'ancien chemin [98]. Sur Les Figures 3.3.a à 3.3.d, l'ensemble des liens et des nœuds en noir forme la topologie physique du réseau. Soit la connexion unicast $< a, \{c\} >$. La Figure 3.3.a montre l'instance de problème de reconfiguration du routage unicast relatif à la connexion $< a, \{c\} >$, où l'ensemble des traits continus de couleur bleue forme l'ancien chemin $a \rightarrow b \rightarrow c$ et l'ensemble des

traits discontinus de couleur bleue forme le nouveau chemin $a \rightarrow c$. La séquence de configurations retournée par MBB est illustrée par les Figures 3.3.b, à 3.3.d. La méthode MBB est efficace pour reconfigurer le routage unicast. Cependant, si la paire de chemins (ancien chemin, nouveau chemin) partage des canaux optiques alors la méthode MBB dans sa forme de base ne peut être utilisée pour la reconfiguration, car le préétablissement devient impossible. Cela oblige à supprimer l'ancien chemin avant de créer le nouveau chemin. Ce qui va créer des interruptions de flux [99, 100].

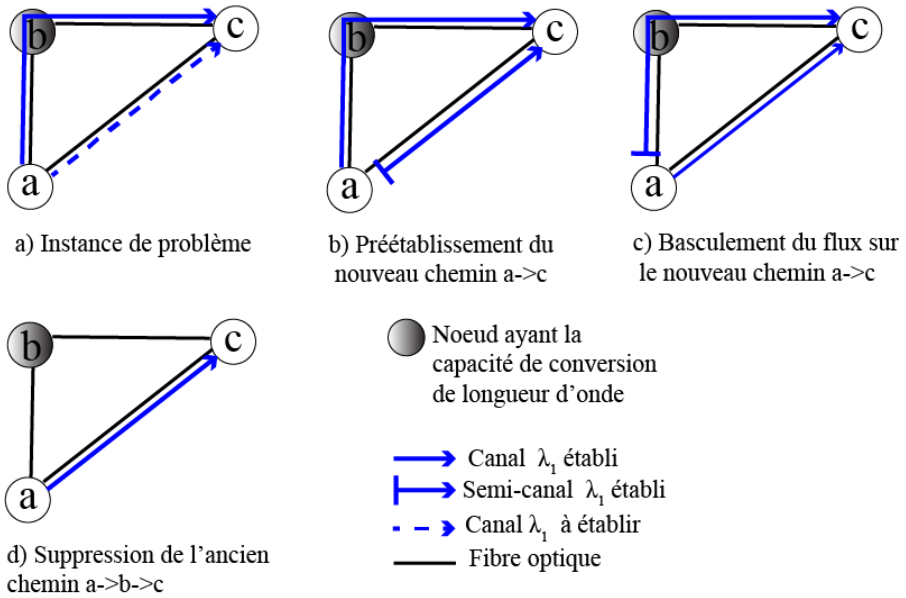


FIGURE 3.3 – Exemple de reconfiguration du routage unicast par MBB

Dans le contexte de la reconfiguration du routage multicast, une adaptation du MBB notée MBB_1 [101] peut être appliquée en considérant chaque branche d'un arbre mono-optique comme un chemin. Ainsi, MBB_1 reconfigure une paire d'arbres mono-optiques (un arbre mono-optique initial, un arbre mono-optique final) en appliquant MBB pour chaque paire de branches (ancienne branche, nouvelle branche). Notons que l'ancienne branche d'une paire de branches appartient à l'arbre mono-optique actuel et la nouvelle branche d'une paire de branches appartient à l'arbre mono-optique final. L'arbre mono-optique actuel est celui utilisé par le flux à une étape donnée du processus de reconfiguration. Au début du processus de reconfiguration, l'arbre mono-optique actuel est équivalent à l'arbre mono-optique initial. Le préétablissement de chaque nouvelle branche s'obtient à l'aide d'une succession d'opérations d'ajout de

commutation de longueur d'onde sans utilisation d'une longueur d'onde additionnelle sur les nœuds de cette branche. La migration du flux sur une nouvelle branche se fait à l'aide de l'opération de basculement sans utilisation d'une longueur d'onde additionnelle, appliquée à la racine de la paire d'arbres mono-optiques. Pour terminer, la suppression de l'ancienne branche se fait à l'aide d'une succession d'opérations de suppression de commutation de longueur d'onde (sans utilisation d'une longueur d'onde additionnelle) appliquée aux nœuds de l'ancienne branche qui ne sont pas membres de la nouvelle branche. MBB_1 ne requiert pas de longueur d'onde d'additionnelle et utilise la racine de la paire d'arbres mono-optiques comme nœud de basculement pour chaque paire de branches. Par conséquent, si la nouvelle branche d'une paire de branches partage des canaux optiques utilisés par l'ancienne branche de ladite paire de branches ou d'une autre paire de branches alors MBB_1 peut provoquer des interruptions du flux.

Soit l'instance de problème présentée à la section 3.3.1 qui concerne la connexion multicast $< s, \{d_1, d_2\} >$. La Figure 3.4.a montre l'arbre mono-optique initial et l'arbre mono-optique final de la connexion $< s, \{d_1, d_2\} >$. L'arbre mono-optique initial est représenté par l'ensemble de traits en bleu continu et l'arbre mono-optique final (à établir) est représenté par l'ensemble des traits en bleu discontinu. Cette paire d'arbres est formé par deux paires de branches à savoir la paire de branches $PB_1 = (s \rightarrow a \rightarrow d_1, s \rightarrow a \rightarrow d_1)$ et la paire de branches $PB_2 = (s \rightarrow a \rightarrow d_2, s \rightarrow b \rightarrow d_2)$. Donc le canal optique $s \rightarrow a$ est partagé par l'ancienne branche de PB_2 et la nouvelle branche de PB_1. MBB_1 produit la séquence de configurations illustrée par les Figures 3.4.b, à 3.4.d. En effet, seule la paire de branches $PB_2 = (s \rightarrow a \rightarrow d_2, s \rightarrow b \rightarrow d_2)$ menant au nœud d_2 requiert d'être reconfigurée. Comme on peut le voir sur la Figure 3.4.c, la configuration requise pour le basculement du flux optique sur la nouvelle branche $s \rightarrow b \rightarrow d_2$ va interrompre le flux vers le nœud d_1, car le semi-canal établit entre le nœud s et le nœud a ne permet pas au nœud a de recevoir le flux optique. Aussi, la configuration requise pour la suppression de l'ancienne branche $s \rightarrow a \rightarrow d_2$, va interrompre le flux vers le nœud d_1 malgré le fait qu'il y a un canal optique établi entre le nœud a et le nœud d_1, car aucun canal optique n'est établi entre le nœud s et le nœud a et donc aucun chemin configuré n'est présent entre la source s et le nœud d_1.

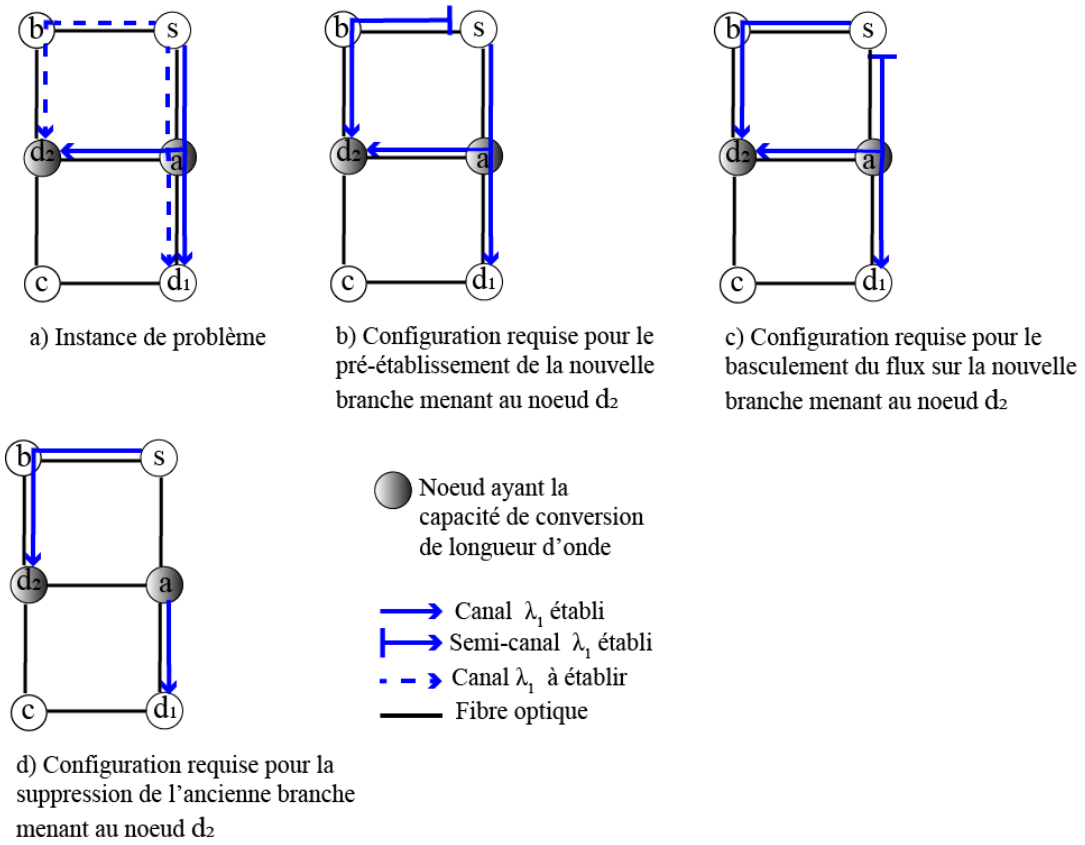


FIGURE 3.4 – Exemple de séquence de configurations obtenue par MBB_1

3.3.2.2 Reconfiguration avec utilisation de longueurs d'onde additionnelles

Afin d'éviter les interruptions de flux lors de la reconfiguration, une approche par branches notée BpBAR a été proposée. Contrairement à MBB_1, cette approche propose une méthode de détermination d'un nœud de basculement. Autrement dit, le nœud de basculement pour une paire de branches n'est pas systématiquement la racine de la paire d'arbres mono-optiques. Aussi, cette approche utilise un processus en deux phases pour la reconfiguration d'une paire de branches. En clair, BpBAR consiste à reconfigurer la paire d'arbres mono-optiques en la reconfigurant par paire de branches. Chaque paire de branches (ancienne branche, nouvelle branche) est constituée d'une branche (c'est-à-dire l'ancienne branche) issue de l'arbre mono-optique actuel et de son correspondant (c'est-à-dire la nouvelle branche) sur l'arbre mono-optique final. Reconfigurer chaque paire de branches consiste à :

1. Choisir un nœud de basculement en suivant un algorithme dans [90] : Ce nœud

doit être choisi de sorte à garantir la continuité du flux vers les autres destinations après l'opération de basculement ;

2. Préétablir la nouvelle branche à partir de ce nœud de basculement avec une longueur d'onde additionnelle : L'ajout de commutation de longueur d'onde est utilisé successivement sur tous les nœuds situés entre le nœud de basculement et le nœud destination de la paire de branche. La convergence est appliquée au nœud destination ;
3. Basculer au niveau du nœud de basculement le flux de l'ancienne branche sur la nouvelle branche grâce à l'opération de basculement en utilisant une longueur d'onde additionnelle ;
4. Supprimer l'ancienne branche : La suppression de commutation de longueur d'onde est utilisée sur tous les nœuds situés entre le nœud de basculement et le nœud destination de la paire de branches. La non-convergence est appliquée au nœud destination ;
5. Les points 1 à 4 sont repris afin d'avoir une nouvelle branche utilisant la même longueur d'onde que l'ancienne branche.

Notons que pour une paire de branches, l'opération de basculement doit être effectuée sur un seul et unique nœud afin de garantir l'absence d'interruption de flux. En effet, plusieurs opérations de basculement (presque) simultanées sur des nœuds différents pourrait engendrer des cycles et donc une interruption de flux. Il est donc primordial de choisir un seul nœud de basculement (pour une paire de branches) conformément à l'algorithme de détermination du nœud de basculement qui a été proposé dans [90].

La Figure 3.5 décrit une séquence de configurations obtenue par BpBAR pour l'instance de problème (de reconfiguration) présentée à la section 3.3.1 qui concerne la connexion multicast $< s, \{d_1, d_2\} >$. Rappelons que sur la Figure 3.5.a, l'arbre mono-optique initial est représenté par l'ensemble des traits continus de couleur bleue et l'arbre mono-optique final est représenté par l'ensemble des traits discontinus de couleur bleue. Pour cette instance, seule la paire de branches $(s \rightarrow a \rightarrow d_2, s \rightarrow b \rightarrow d_2)$ menant au nœud d_2 a besoin d'être reconfiguré. Selon l'algorithme de détermination du nœud de basculement dans [90], le nœud de basculement de cette paire de branches est le nœud a. Les Figures 3.5.b à 3.5.d mettent en évidence les différentes configurations obtenues après la dernière instruction effectuée de chacune des étapes pré-

sentées par les points 2 à 4 de l'approche BpBAR (décrit précédemment). Quant à la Figure 3.5.e, elle illustre la configuration obtenue après l'étape présentée par le point 5 de l'approche BpBAR décrit précédemment. Comme on le voit sur la Figure 3.5.e, les canaux optiques établis constituant l'arbre mono-optique actuel sont identiques aux canaux optiques de l'arbre mono-optique final présenté sur la Figure 3.5.a (voir traits discontinus en blue) : BpBAR a effectivement migré le flux de l'arbre mono-optique initial vers l'arbre mono-optique final. Cette migration s'est effectuée sans interruption de flux, car aucune configuration de la séquence générée ne prive un nœud destination du flux. Autrement dit, il existe un arbre établi couvrant la source s et chacune des destinations (d_1 et d_2) à chaque instant du processus de reconfiguration.

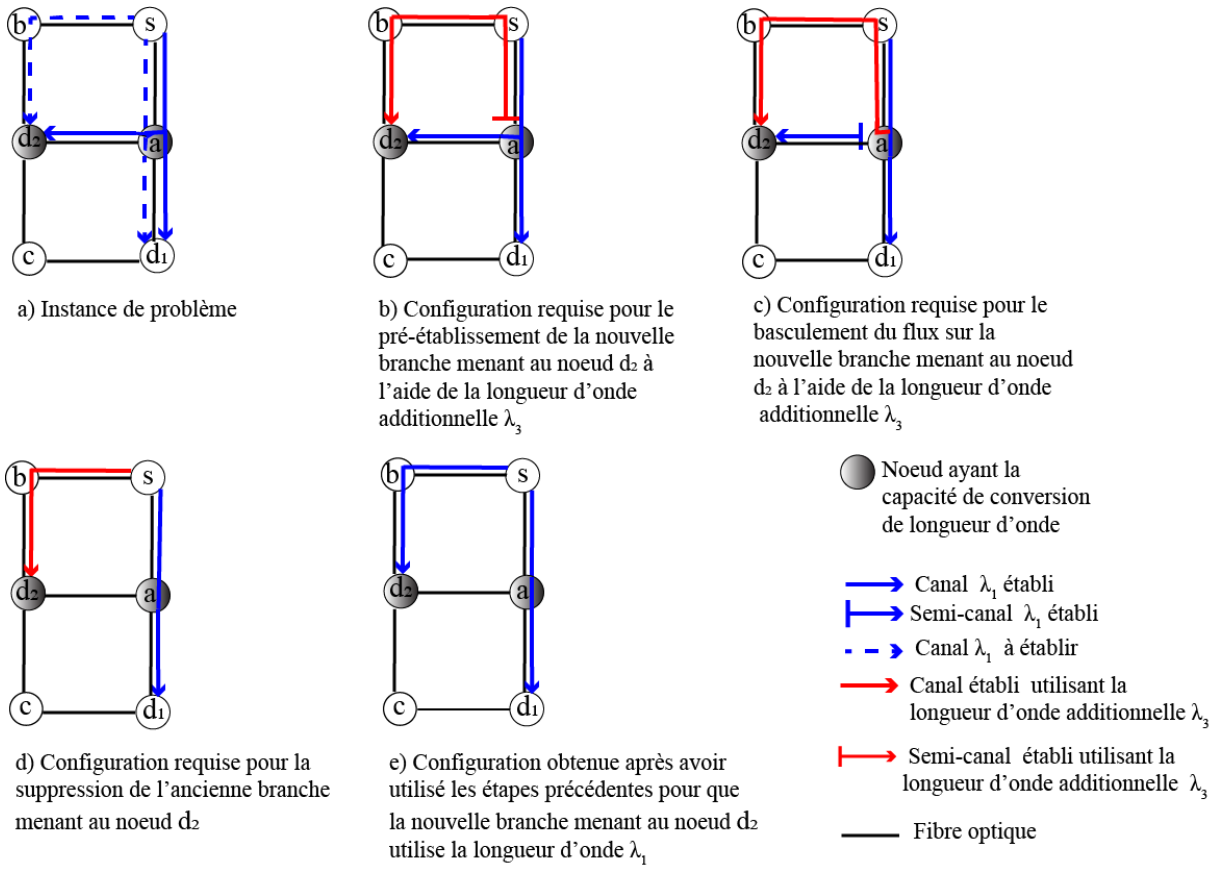


FIGURE 3.5 – Illustration d'une séquence de configurations avec BpBAR

Néanmoins, l'approche de base BpBAR peut induire une longue séquence de configurations avec une forte utilisation d'une longueur d'onde additionnelle si la paire d'arbres mono-optiques à reconfigurer contient plusieurs paires de branches à recon-

figurer. Afin d'éviter d'avoir une séquence de configurations trop longue, plusieurs méthodes basées sur cette approche ont été proposées dans [90, 102, 103]. Dans [90], la méthode BpBAR_2 réalise les opérations de configuration concernées par l'étape 2 en parallèle. BpBAR_2 procède de même pour l'étape 4 de l'approche BpBAR. Cela se justifie par le fait que les ports requis pour chaque opération à l'étape 2 et à l'étape 4 sont des ports isolés. Autrement dit, ces ports ne sont pas utilisés par l'arbre mono-optique actuel. Ainsi, BpBAR_2 réduit mieux la longueur de la séquence de configurations plutôt que BpBAR. Afin de réduire le coût d'utilisation de ressources additionnelles, la méthode TRwRC [102] a été proposée. Pour ce faire, l'ensemble des destinations de la connexion multicast est subdivisé en deux groupes : l'ensemble des destinations dont les branches ont été reconfigurées et l'ensemble des destinations dont les branches n'ont pas été reconfigurées. Cependant, elle peut engendrer une durée de reconfiguration assez longue du fait qu'elle génère une série de forêts optiques à reconfigurer. Afin d'améliorer TRwRC en réduisant sa durée de reconfiguration, la méthode RCBRwPR [103] a été proposée. Elle reconfigure en parallèle les paires de branches ayant des nœuds de basculement différents.

3.3.3 Récapitulatif et implications

En résumé, MBB_1 peut causer des interruptions de flux. Cela est le cas lorsque pour une paire de branches (une ancienne branche, une nouvelle branche), la nouvelle branche de cette paire partage au moins un lien avec l'ancienne branche de ladite paire de branches ou d'une autre paire de branches. Concernant les méthodes basées sur l'approche BpBAR telle que RCBRwPR, elles utilisent systématiquement une longueur d'onde additionnelle à plusieurs reprises pour la reconfiguration d'une paire de branches même si cette paire de branches est à liens disjoints. De plus, avec les méthodes basées sur l'approche BpBAR, si le nœud de basculement d'une paire de branches est différent de la racine de la paire d'arbres mono-optiques alors ce nœud doit forcément avoir la capacité de conversion de longueur d'onde afin de permettre d'effectuer une opération de basculement nécessitant une conversion de longueur d'onde. Par conséquent, ces méthodes sont efficaces pour la reconfiguration d'une paire d'arbres mono-optiques dans un réseau optique WDM où tous les nœuds ont la capacité de conversion de longueur d'onde aussi appelé réseau optique WDM ayant la propriété de conversion totale de longueur d'onde. Cependant, dans un réseau dans

lequel seulement un nombre réduit de nœuds a la capacité de conversion de longueur d'onde (aussi appelé réseau optique WDM ayant la propriété de conversion partielle de longueur d'onde), il est évident que plusieurs nœuds n'ont pas la capacité de conversion de longueur d'onde. Sur l'instance de problème de la Figure 3.5.a, le seul nœud de basculement (pour la paire de branches menant au nœud destination d_2) garantissant la continuité du flux est le nœud a. Comme on peut le voir sur la Figure 3.5.c, l'opération de basculement au nœud a implique une conversion de longueur d'onde au niveau du nœud a. Si le nœud a n'avait pas la capacité de conversion de longueur d'onde alors il aurait fallu supprimer le canal optique (utilisant la longueur d'onde λ_1) établi entre le nœud s et le nœud a pour que l'opération de basculement puisse se faire sans impliquer de conversion de longueur d'onde. Par ailleurs, cette suppression du canal optique établi entre le nœud s et le nœud a via l'opération de suppression de commutation de longueur d'onde aurait pour conséquence qu'il n'y aurait pas une structure de routage configuré entre la source s et les différents nœuds destinations d_1 et d_2. Autrement dit, la suppression de ce canal optique aurait entraîné des interruptions de flux. Par conséquent, l'utilisation d'une méthode basée sur l'approche BpBAR pour la reconfiguration d'une paire d'arbres mono-optiques dans un réseau ayant la propriété de conversion partielle de longueur d'onde peut générer des interruptions de flux.

3.4 Problème de reconfiguration d'un ensemble de plusieurs paires d'arbres mono-optiques

Comme mentionné au chapitre 1, l'opérateur réseau peut utiliser la reconfiguration afin d'améliorer les performances de son réseau. Dans la section 3.3, nous supposons qu'une seule connexion multicast est concernée par la reconfiguration. Ainsi le problème de reconfiguration d'une paire d'arbres mono-optiques et les méthodes existantes utilisées pour le résoudre ont été présentés. Cependant, dans la pratique, plusieurs connexions multicasts peuvent être concernées par la reconfiguration afin de réduire la probabilité de blocage. Ainsi, dans cette section, nous présentons le problème de reconfiguration d'un ensemble de plusieurs paires d'arbres mono-optiques dans un premier temps puis les méthodes de résolution existantes.

3.4.1 Description du problème de reconfiguration d'un ensemble de plusieurs paires d'arbres mono-optiques

Au lieu de traiter une seule connexion multicast, le problème de reconfiguration d'un ensemble de plusieurs paires d'arbres mono-optiques concerne plusieurs connexions multicasts. En effet, chaque paire d'arbres mono-optiques à reconfigurer concerne une connexion multicast donnée. Soit M, un ensemble de connexions multicasts de sorte que chaque connexion multicast est mise en œuvre par un arbre mono-optique initial qui transporte un flux optique. L'ensemble de ces arbres mono-optiques forme l'ensemble de configurations initiales ou plus simplement la configuration initiale. Supposons que les arbres mono-optiques finaux (devant remplacer les arbres mono-optiques initiaux) sont pré-calculés. L'ensemble de ces arbres mono-optiques forme l'ensemble de configurations finales ou plus simplement la configuration finale. Le problème de multi-reconfiguration étudié ici consiste à trouver la séquence de configurations permettant de faire migrer le flux optique de la configuration initiale à la configuration finale (à mettre en œuvre) tout en réduisant les interruptions de flux.

Ce problème est délicat à résoudre, car généralement un arbre mono-optique final nécessite un canal optique qui est déjà utilisé par un arbre mono-optique initial. Cette situation peut créer un état de blocage dans le processus de reconfiguration. Nous illustrons dans ce qui suit une situation de blocage. Soit m_1, m_2 et m_3, trois connexions multicasts établies sur un réseau optique WDM ayant la propriété de conversion partielle de longueur d'onde. Nous supposons qu'un seul canal optique (par exemple celui utilisant la longueur d'onde λ_1) peut être établi sur chaque fibre optique. La Figure 3.6.a contient l'arbre mono-optique initial de m_1 (dont l'ensemble des canaux optiques établis est représenté par l'ensemble des traits continus et de couleur rose), l'arbre mono-optique initial de m_2 (dont l'ensemble des canaux optiques établis est représenté par l'ensemble des traits continus et de couleur bleue) et l'arbre mono-optique initial de m_3 (dont l'ensemble des canaux optiques établis est représenté par l'ensemble des traits continus et de couleur verte). En outre, la Figure 3.6.b contient l'arbre mono-optique final (dont l'ensemble des canaux optiques à établir, est représenté par l'ensemble des traits continus et de couleur rose) de m_1, l'arbre mono-optique final (dont l'ensemble des canaux optiques à établir, est représenté par l'ensemble des traits continus et de couleur bleue) de m_2 et l'arbre mono-optique final (dont l'ensemble des canaux optiques à établir, est représenté par l'ensemble des traits continus et de couleur verte)

de m_3. Nous supposons qu'aucune fibre optique non requise par les paires d'arbres ne peut être utilisée pour établir un chemin temporaire. Ainsi, l'arbre mono-optique final de la connexion m_2 nécessite le canal optique $s_1 \rightarrow a$ qui est déjà utilisé par l'arbre mono-optique initial de la connexion m_1. En outre, l'arbre mono-optique final de m_1 nécessite les canaux optiques $h \rightarrow b$ et $b \rightarrow c$ qui sont déjà utilisés par l'arbre mono-optique initial de m_2. Cette situation d'interdépendance entre les deux connexions crée un état de blocage. Cette situation de blocage ne peut être surmontée sans interruption du flux vers au moins un nœud destination des connexions multicasts. Il faut donc faire au mieux en réduisant le nombre d'interruptions de flux lors du processus de reconfiguration.

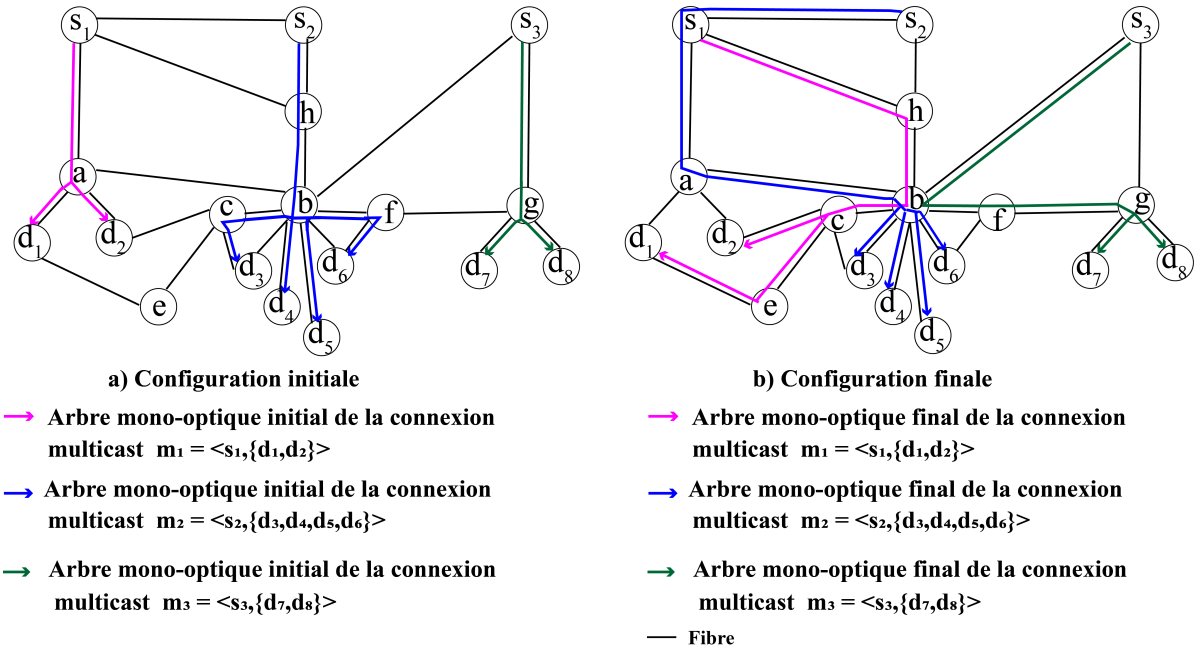


FIGURE 3.6 – Exemple de problème de reconfiguration d'un ensemble de paires d'arbres mono-optiques

3.4.2 Travaux connexes à la reconfiguration d'un ensemble de plusieurs paires d'arbres mono-optiques

Le problème de reconfiguration d'un ensemble de paires d'arbres mono-optiques étudié dans ce chapitre est une extension du problème de reconfiguration d'un ensemble de paires de chemins mono-optiques (unicasts). Par ailleurs, le problème de

reconfiguration d'un ensemble de plusieurs paires d'arbres mono-optiques n'a jamais encore été abordé à notre connaissance (il le sera au chapitre 5 de ce manuscrit). Par conséquent, nous présentons dans ce sous-chapitre, des travaux existants proposant uniquement les solutions au problème de reconfiguration d'un ensemble de paires de chemins mono-optiques.

Un chemin mono-optique [104] (ou *lightpath*) est un cas particulier d'un arbre mono-optique. En effet, un chemin mono-optique peut être vu comme un arbre mono-optique comprenant une seule branche. Ainsi, le problème de la reconfiguration d'un ensemble de paires de chemins mono-optiques est un cas particulier du problème de reconfiguration d'un ensemble de paires d'arbres mono-optiques étudié.

Certains travaux de reconfiguration d'un ensemble de paires de chemins mono-optiques (c'est-à-dire [97, 98, 105-107]) ont proposé des méthodes intégrant une formulation sous forme de programme linéaire et une politique *Make Before Break* pour rechercher une configuration finale permettant de résoudre le problème en évitant des interruptions de flux. Des méthodes exactes ont été proposées dans [98, 105, 106]. Vu que ces méthodes sont peu adaptées pour des instances de problème de grande taille, des heuristiques ont été proposées dans [97, 107]. Les travaux énoncés dans ce paragraphe peuvent conduire à une probabilité de blocage élevé vu que certaines demandes de connexions sont rejetées si elles sont susceptibles de provoquer un grand nombre d'interruptions [108]. Par ailleurs, elles ne supposent pas que la configuration finale est connue et donc ne cherchent pas à surmonter réellement les problèmes de dépendances pouvant exister entre des connexions.

Le problème de la reconfiguration d'un ensemble de paires de chemins mono-optiques est également délicat à résoudre, car certains chemins mono-optiques finaux nécessitent des canaux optiques qui sont déjà utilisés par certains chemins mono-optiques initiaux. Les dépendances sont alors modélisées par un graphe orienté non pondéré de dépendances [109].

Le graphe orienté non pondéré de dépendances est un graphe orienté qui est construit comme suit [110] : a) chaque connexion est représenté par un nœud du graphe ; b) si le chemin mono-optique final de la connexion x requiert un ou plusieurs canaux optiques déjà utilisés par le chemin mono-optique de la connexion y, alors il faut créer un arc orienté du nœud x au nœud y. Notons qu'une instance du problème de reconfiguration d'un ensemble de paires de chemins mono-optiques peut être résolue sans interruption du flux si le graphe orienté non pondéré de dépendances de

cette instance est acyclique [111]. Notez également que chaque cycle montre un état de blocage dans l'instance du problème. Un blocage provoque inévitablement des interruptions de flux.

Les travaux dans [110-113] abordent le problème de la reconfiguration d'un ensemble de paires de chemins mono-optiques. Chaque algorithme proposé dans ces travaux est basé sur l'approche MFVSA ou Minimum Feedback Vertex Set Approach. Cette approche est centrée sur l'utilisation d'un ensemble minimum de nœuds de rétroaction (ou *Minimum Feedback Vertex Set* [114] noté MFVS) pour surmonter les états de blocage dans le problème de reconfiguration d'un ensemble de paires de chemins mono-optiques. Étant donné un graphe orienté non pondéré de dépendances $G_{\bar{p}}\left(V_{G_{\bar{p}}}, E_{G_{\bar{p}}}\right)$, un ensemble de nœuds de rétroaction ou *Feedback Vertex Set* noté FVS est un ensemble de nœuds V' (avec $V' \subseteq V_{G_{\bar{p}}}$) tel que le graphe induit par l'ensemble de nœuds $V_{G_{\bar{p}}} \setminus V'$ est acyclique. En d'autres termes, la suppression de tous les éléments de V' détruit tous les cycles du graphe orienté non pondéré de dépendances $G_{\bar{p}}$. MFVSA peut être résumée par les étapes suivantes [112] :

— **Étape 1** : Construire le graphe orienté non pondéré des dépendances $G_{\bar{p}}$.

— **Étape 2** : Calculer un ensemble minimum de nœuds de rétroaction MFVS. Un ensemble minimum de nœuds de rétroaction est un ensemble de nœuds de rétroaction qui a une cardinalité minimum [115]. Les auteurs dans [112] ont proposé une solution heuristique pour calculer MFVS, alors que dans [110, 111, 113], les auteurs utilisent un algorithme exact.

— **Étape 3** : Supprimer le chemin initial de chaque connexion appartenant à MFVS.

— **Étape 4** : Reconfigurer chaque connexion qui n'appartient pas à MFVS. Autrement dit, il faut établir le chemin final puis supprimer le chemin initial de chaque connexion qui n'appartient pas à MFVS.

— **Étape 5** : Etablir le chemin final de chacune des connexions appartenant à MFVS.

Notons que l'étape 2 ne garantit pas qu'à la fin de l'approche MFVSA, la solution obtenue est la solution optimale. En effet, les interruptions de flux causées aux éléments du MFVS, à l'étape 3, peuvent causer un nombre total d'interruptions de flux très élevé à la fin de la reconfiguration si l'étape 4 concerne un grand nombre de configurations contenues dans la séquence de configurations. Autrement dit, la solution optimale serait garantie en procédant par une méthode exhaustive. En clair, il faut tester tous les ensembles de nœuds de rétroaction ou *Feedback Vertex Sets* (FVSs)

possibles sachant que la taille de tout FVS est d'au moins 1 et d'au plus c, où c est le nombre de cycles dans le graphe de dépendances. Cette recherche serait donc très coûteuse en temps et donc garderait le réseau dans un état sous-optimal pendant un temps non acceptable. C'est pourquoi, cette recherche exhaustive n'est pas évoquée par les méthodes basées sur l'approche MFVSA qui fournissent plutôt une bonne solution (en faisant le choix d'un FVS de cardinalité minimum) au problème de la reconfiguration d'un ensemble de paires de chemins mono-optiques, mais pas toujours la solution optimale. Néanmoins la solution proposée dans [110] donne de meilleures performances que les autres solutions car elle fait l'hypothèse qu'il peut exister des ressources (c'est-à-dire des chemins de secours ou des longueurs d'onde additionnelles) disponibles pour réduire davantage le nombre d'interruptions à l'étape 3. Cependant, cette hypothèse n'est pas valable dans notre cas d'étude.

Nous illustrons MFVSA à travers l'instance de problème de la Figure 3.7. Le problème de reconfiguration d'un ensemble de paires de chemins mono-optiques de cette figure concerne trois connexions u_1, u_2 et u_3. On déduit de la Figure 3.7.a et de la Figure 3.7.b ce qui suit : Le chemin mono-optique final de la connexion u_2 requiert le canal optique $a \rightarrow c$ qui est déjà utilisé par le chemin mono-optique initial de la connexion u_1. En outre, le chemin mono-optique final de la connexion u_1 requiert le canal optique $a \rightarrow d$ qui est déjà utilisé par le chemin mono-optique initial de la connexion u_2. Il y a donc un état de blocage dû à ces deux relations de dépendance. Le chemin mono-optique final de la connexion u_3 requiert le canal optique $e \rightarrow f$ qui est déjà utilisé par le chemin mono-optique initial de la connexion u_1. En outre, le chemin mono-optique final de la connexion u_1 requiert le canal optique $d \rightarrow f$ qui est déjà utilisé par le chemin mono-optique initial de la connexion u_3. Il y a donc un état de blocage dû à ces deux relations de dépendance.

Toutes les dépendances énoncées précédemment sont modélisées au travers du graphe orienté non pondéré de dépendances illustré par la Figure 3.8. Le seul ensemble de nœuds de rétroaction FVS est $\{u_1\}$. Par conséquent, le calcul de l'ensemble minimum de nœuds de rétroaction MFVS retourne toujours $\{u_1\}$. La Figure 3.9 illustre la séquence (de configurations) obtenue avec MFVSA pour l'instance de problème de la Figure 3.7. La Figure 3.9.a, illustre la configuration au début du processus de reconfiguration. L'étape 3 de MFVSA va supprimer le chemin de la connexion u_1. Comme, on peut le voir sur la Figure 3.9.b, la configuration obtenue à l'issue de l'étape 3 ne contient que le chemin de la connexion u_2 et celui de la connexion u_3. Aucun chemin n'a été

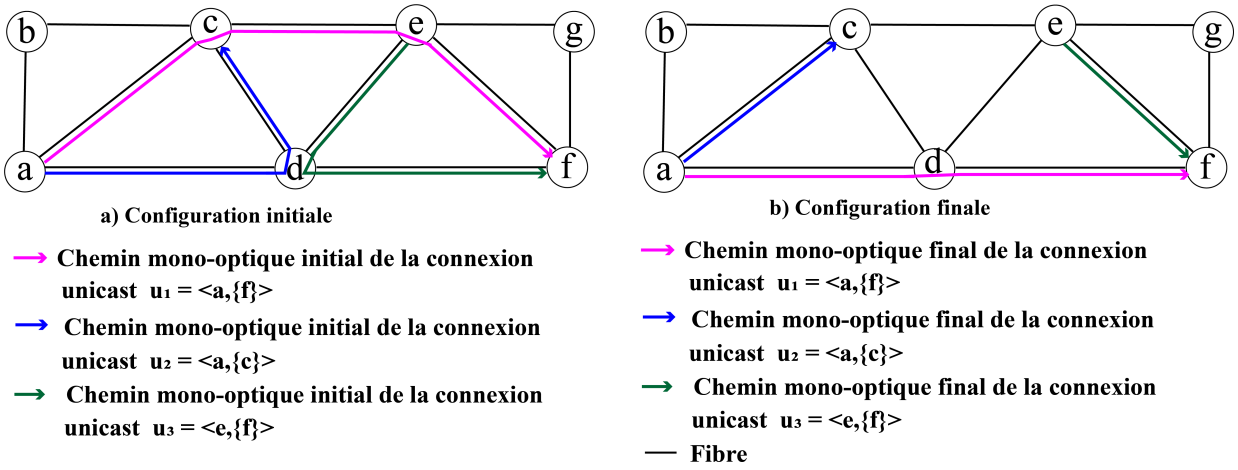


FIGURE 3.7 – Instance de problème de reconfiguration d'un ensemble de paires de chemins (unicasts) mono-optiques

établi pour la connexion u_1. Autrement dit, aucun chemin ne permet de transmettre un flux optique du nœud a vers le nœud f de la connexion u_1. Ainsi, cette configuration cause une interruption de flux vers un nœud destination à savoir le nœud f dans le cadre de la connexion u_1. L'étape 4, n'établit pas un chemin pour la connexion u_1. Ainsi, la configuration obtenue à cette étape (voir Figure 3.9.c) cause la même interruption de flux que celle de l'étape 3. Puis l'étape 5 établit un chemin pour la connexion u_1. Par conséquent, la configuration obtenue à cette étape ne cause pas d'interruption de flux. Comme on peut le voir, cette configuration est identique à la configuration finale de l'instance de problème de la Figure 3.7. Au total le coût d'interruptions (ou le nombre total d'interruptions de flux) de la séquence retournée par l'approche MFVSA est deux (c'est-à-dire une interruption à l'étape 3 et une interruption à l'étape 4). Notons que le coût d'interruptions causé par MFVSA est optimal pour cette instance de problème vu que l'étape 2 retourne un MFVS qui est un singleton et que les étapes 3 et 4 ne concernent chacune qu'une seule configuration (voir Figures 3.9.b et 3.9.c) .

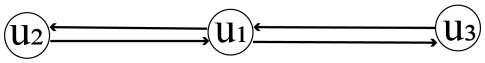


FIGURE 3.8 – Graphe orienté non pondéré de dépendances associé au problème de reconfiguration d'un ensemble de paires de chemins mono-optiques obtenu par MFVSA

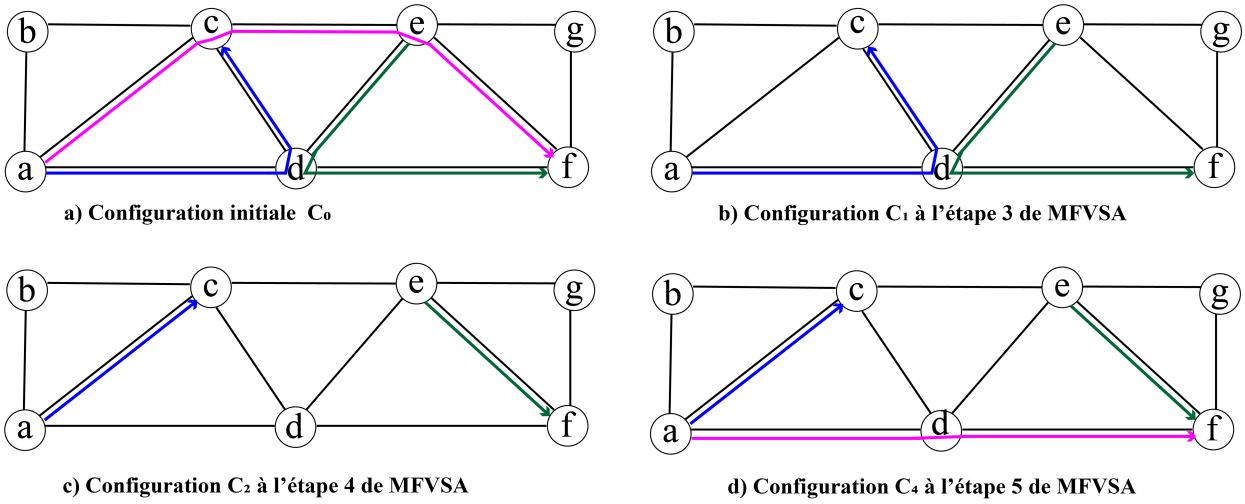


FIGURE 3.9 – Séquence de configurations retournée par MFVSA

3.4.3 Récapitulatif et implications

Le problème de reconfiguration d'un ensemble de paires de chemins mono-optiques (unicasts) qui considère comme objectif la réduction du nombre total d'interruptions de flux a été traité dans la littérature. Pour ce faire, l'approche MFVSA a été proposée. Elle permet de trouver la séquence de configurations qui cause un faible nombre (pas toujours optimal) d'interruptions de flux. Ainsi, MFVSA est une bonne solution pour la résolution du problème de reconfiguration d'un ensemble de paires de chemins mono-optiques. Cependant, MFVSA ne constitue pas généralement une bonne solution pour résoudre le problème de la reconfiguration d'un ensemble de paires d'arbres mono-optiques. En effet, MFVSA s'appuie sur un ensemble minimum de nœuds de rétroaction. Cet ensemble de nœuds est un ensemble de cardinalité minimum qui correspond au nombre minimal de connexions unicasts dont les chemins doivent être temporairement supprimés pour résoudre les situations de blocage dans le processus de reconfiguration. Par ailleurs, supprimer un chemin mono-optique équivaut à interrompre le flux vers un unique nœud destination. Cependant, par définition une connexion multicast (non réduite) implique au moins deux nœuds destinations. De plus, toutes les connexions multicasts n'ont pas forcément le même nombre de nœuds destinations. Ainsi, supprimer temporairement les arbres d'un nombre minimal de connexions multicasts n'implique pas qu'un faible nombre d'interruptions de flux (vers des nœuds destinations) soit obtenu à la fin du processus de reconfiguration.

Autrement dit, un algorithme qui s'appuie sur un ensemble de cardinalité minimum ne permet pas de forcément résoudre efficacement le problème de reconfiguration d'un ensemble de paires d'arbres mono-optiques.

La Figure 3.10 illustre le graphe orienté non pondéré de dépendances pour l'instance de problème (voir Figure 3.6) de reconfiguration d'un ensemble de paires d'arbres mono-optiques vu à la section 3.4.1.

FIGURE 3.10 – Graphe orienté non pondéré de dépendances associé à l'instance de problème

Cette instance de problème concerne trois connexions multicasts $m_1 =< s_1, \{d_1, d_2\} >, m_2 =< s_2, \{d_3, d_4, d_5, d_6\} >$ et $m_3 =< s_3, \{d_7, d_8\} >$. Selon la Figure 3.10, il y a deux ensembles de nœuds de rétroaction de cardinalité minimum à savoir $MFVS_1 = \{m_1\}$ et $MFVS_2 = \{m_2\}$. Ces deux ensembles sont des singletons. Cependant, si nous supposons que l'étape 2 de MFVSA retourne $MFVS_2 = \{m_2\}$ comme ensemble minimum de nœuds de rétroaction MFVS, alors MFVSA cause quatre interruptions de flux (c'est-à-dire une interruption de flux respectivement vers d_3, d_4, d_5 et d_6) à l'étape 3 et aussi à l'étape 4, soit un total de huit interruptions de flux comme illustré par la séquence de configurations (voir Figure 3.11) retournée par MFVSA sur l'instance de problème de reconfiguration d'un ensemble de paires d'arbres mono-optiques de la Figure 3.6. Cependant, un total de six interruptions de flux aurait pu être causé par l'utilisation de $FVS_1 = \{m_1\}$ comme MFVS à l'étape 2. Donc, si MFVSA choisit $FVS_2 = \{m_2\}$ à l'étape 2, alors MFVSA ne renvoie pas une bonne solution pour l'instance du problème de reconfiguration d'un ensemble de paires d'arbres mono-optiques représentée par la Figure 3.6. Ce qui précède confirme qu'un algorithme qui s'appuie sur la recherche d'un ensemble minimum de nœuds de rétroaction n'est pas un algorithme qui retourne toujours une bonne séquence de configurations dans le cas du problème de reconfiguration d'un ensemble de paires d'arbres mono-optiques.

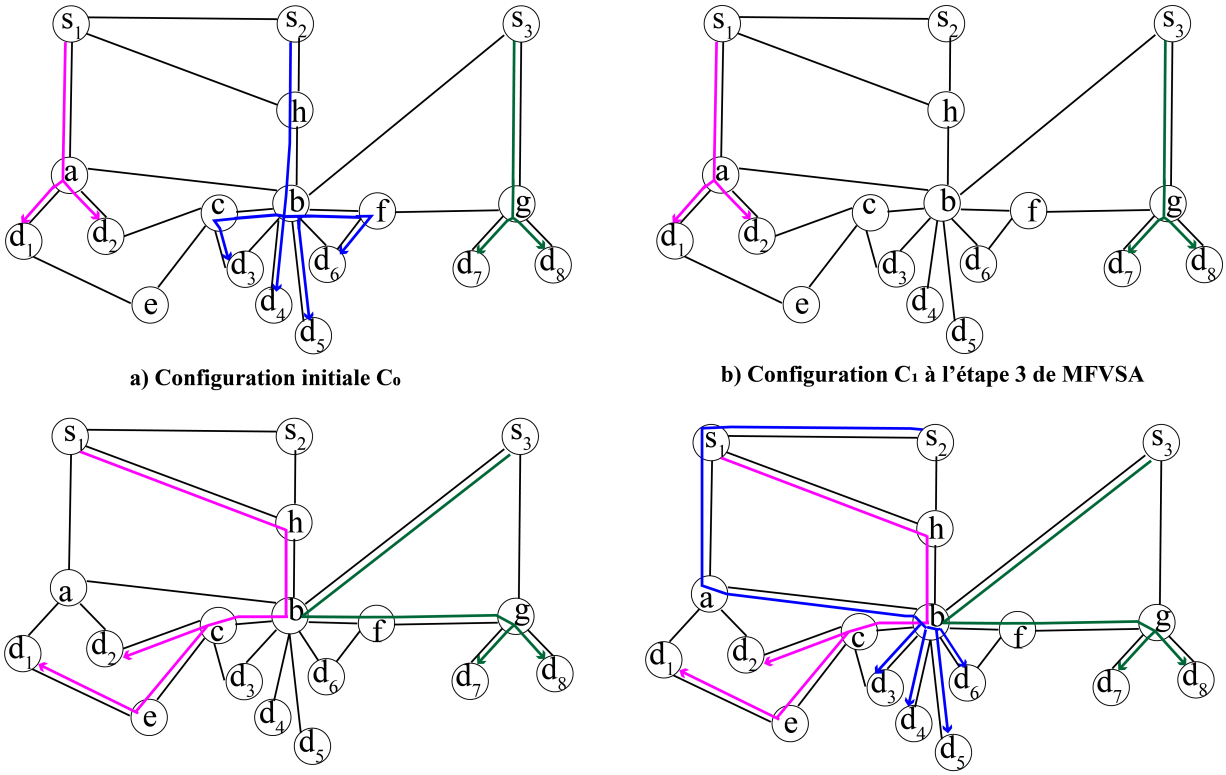


FIGURE 3.11 – Séquence de configurations de MFVSA pour une instance de problème de reconfiguration d'un ensemble de paires d'arbres mono-optiques

3.5 Conclusion

Nous avons présenté dans ce chapitre des notions fondamentales à une meilleure compréhension du processus de reconfiguration puis les problèmes de reconfiguration étudiés dans cette thèse.

Le premier problème évoqué est celui de la reconfiguration d'une paire d'arbres mono-optiques. L'objectif est de trouver une séquence de configurations qui permet de migrer le flux optique d'un arbre mono-optique initial (ou configuration initiale) vers un arbre mono-optique final (ou configuration finale) sans interrompre le flux optique. Il est important de rappeler qu'une configuration est définie comme un ensemble composé de canaux optiques et éventuellement de semi-canaux optiques. Les méthodes existantes garantissent toujours une continuité du flux à condition que tous les nœuds du réseau aient la capacité de conversion de longueur d'onde. Par conséquent, elles ne

sont pas très adaptées à la reconfiguration d'une paire d'arbres mono-optiques dans un réseau optique WDM ayant la propriété de conversion partielle de longueur d'onde. Le chapitre 4 va donc s'atteler à présenter nos contributions afin d'effectuer la reconfiguration d'une paire mono-optiques sans interruption de flux dans un réseau optique WDM ayant la propriété de conversion partielle de longueur d'onde.

Le second problème est celui de la reconfiguration d'un ensemble de plusieurs paires d'arbres mono-optiques. L'objectif est de trouver une séquence de configurations qui permet de migrer le flux optique d'un ensemble d'arbres mono-optiques initiaux (ou configuration initiale) vers la configuration finale tout en réduisant les interruptions de flux. Les méthodes existantes sont des méthodes conçues pour la reconfiguration d'un ensemble de plusieurs paires de chemins mono-optiques. Autrement dit, elles ont été conçues lorsque les connexions à reconfigurer sont des connexions unicasts. Ces méthodes existantes procèdent à la suppression du plus petit nombre de chemins lors de la reconfiguration. Cette suppression provoque un nombre d'interruptions de flux vers des nœuds destinations équivalant au nombre de chemins concernés par cette suppression. Cependant, la suppression du plus petit nombre d'arbres mono-optiques lors de la reconfiguration d'un ensemble de paires d'arbres mono-optiques peut provoquer un grand nombre d'interruptions de flux vers les nœuds destinations. En effet, un arbre mono-optique contient plusieurs nœuds destinations contrairement au chemin qui contient par définition un seul nœud destination. De plus, les arbres mono-optiques à supprimer ne contiennent pas forcément le même nombre de nœuds destinations. Par conséquent, les méthodes existantes ne permettent pas de reconfigurer efficacement un ensemble de plusieurs paires d'arbres mono-optiques. Le chapitre 5 va donc s'atteler à présenter nos contributions afin d'effectuer la reconfiguration d'un ensemble de plusieurs paires mono-optiques dans un réseau optique WDM ayant la propriété de conversion partielle de longueur d'onde.

CHAPITRE 4

RECONFIGURATION D'UNE PAIRE D'ARBRES MONO-OPTIQUES

Sommaire

4.1 Introduction **63**

4.2 Spécification du problème **65**

4.3 Méthodes de reconfiguration d'une paire d'arbres mono-optiques **67**

4.3.1 Caractérisation de paires de sous-arbres 68

4.3.2 Méthode de reconfiguration SbSRA 77

4.3.3 Méthode de parallèlisation de la reconfiguration Para_SbSRA . 93

4.4 Evaluation de performance **101**

4.4.1 Critères d'évaluation 101

4.4.2 Description des simulations 101

4.4.3 Analyse des résultats 104

4.5 Conclusion **107**

4.1 Introduction

Dans le chapitre 1, nous avons présenté la mono-Reconfiguration du routage multicast comme une opération au cours de laquelle un opérateur réseau applique le mécanisme de reconfiguration du routage à une seule connexion multicast. Quant au chapitre 2, il révèle que l'arbre mono-optique et l'arbre multi-optique sont des résultats possibles de l'opération de routage multicast dans les réseaux optiques WDM ayant la propriété de conversion partielle de longueur d'onde. Par ailleurs à la fin du chapitre 2, nous avons décidé après justification de supposer dans notre travail que seul l'arbre mono-optique est le résultat du routage multicast effectué sur le réseau.

Soit une connexion multicast établie par un arbre mono-optique appelé arbre mono-optique initial qui est utilisé pour transporter un flux optique de cette connexion dans un réseau optique WDM ayant la propriété de conversion partielle de longueur d'onde. Le problème de mono-reconfiguration du routage multicast étudié ici consiste à faire migrer le flux optique de l'arbre mono-optique initial vers un nouvel arbre mono-optique pré-calculé ou arbre mono-optique final sans interruption de flux vers les nœuds destinations de la connexion multicast. L'arbre mono-optique initial et l'arbre mono-optique final étant des données du problème, on parle alors de reconfiguration d'une paire d'arbres mono-optiques.

Les méthodes de reconfiguration d'une paire d'arbres mono-optiques les plus efficaces qui sont basées sur l'approche par branches, ont été présentées au chapitre précédent. Cependant, l'approche par branches n'est pas très adaptée au contexte des réseaux optiques WDM ayant la propriété de conversion partielle de longueur d'onde. En effet, le choix d'un nœud de basculement ayant la capacité de conversion de longueur d'onde n'est pas toujours possible vu que seulement certains nœuds de notre réseau ont la capacité de conversion de longueur d'onde. Par conséquent, nous proposons une approche par sous-arbres. Les principales contributions évoquées dans ce chapitre sont :

— La caractérisation des différentes paires de sous-arbres que l'on peut sélectionner dans la paire d'arbres mono-optiques à reconfigurer ;

— Une méthode de reconfiguration d'une paire d'arbres mono-optiques notée SbSRA qui s'appuie sur la précédente caractérisation ;

— Une preuve de la reconfigurabilité en parallèle des paires de sous-arbres ;

— La méthode Para_SbSRA de reconfiguration d'une paire d'arbres mono-optiques qui permet de mieux réduire la longueur du processus de reconfiguration.

Dans ce qui suit, il est question de présenter une formulation du problème étudié dans ce chapitre (voir section 4.2). Après cela, nous exposons nos méthodes pour résoudre le problème (voir section 4.3). L'efficacité de ces méthodes est mise en avant dans la section 5.4. Ce chapitre se termine par une conclusion (voir section 4.5) qui représente l'aide-mémoire de ce chapitre.

4.2 Spécification du problème

Soit $G(V, E)$ un graphe non orienté représentant le réseau optique WDM ayant la propriété de conversion partielle de longueur d'onde, avec V l'ensemble des nœuds et E l'ensemble des liens du réseau. Notons qu'on suppose que chaque lien est constitué d'une seule fibre. Chaque lien $e \in E$ a une capacité de $|\Omega|$ canaux optiques distincts pouvant être établis, où Ω est l'ensemble des longueurs d'onde utilisables sur e. En pratique, Ω est l'union de deux sous-ensembles, à savoir Ω_d et Ω_a. Ω_d désigne le sous-ensemble qui contient toutes les longueurs d'onde disponibles pour calculer une structure de routage optique afin d'établir une connexion. Ω_a désigne le sous-ensemble qui contient les longueurs d'onde additionnelles (ou *spare wavelengths*) qui peuvent être utilisées temporairement dans le processus d'établissement d'une connexion ou dans une tâche de résilience réseau telle que la protection de connexions [110].

Pour une connexion multicast $CM(r, D)$, l'arbre mono-optique initial (établi) désigné par $T_0 = (V_0, E_0(\lambda_0))$ est utilisé pour transférer le flux optique depuis le nœud source r vers les nœuds destinations appartenant à D. Notons que $V_0 \subseteq V$, $E_0 \subseteq E$, $r \in V_0$ et $D \subseteq V_0 \setminus \{r\}$. $E_0(\lambda_0) = \bigcup_{i=1}^{|E_0|} e_i(\lambda_0)$ désigne l'ensemble des canaux optiques établis qui forment T_0, avec $\lambda_0 \in \Omega_d$. L'arbre mono-optique initial établi est aussi appelé la configuration initiale notée C_0.

À un moment donné, l'opérateur de réseau décide de procéder à une reconfiguration pour optimiser les performances de son réseau (par exemple, prendre en charge un nombre croissant de connexions). On admet que cette reconfiguration concerne uniquement la connexion multicast $CM(r, D)$. Pour effectuer cette reconfiguration, l'opérateur de réseau calcule dans un premier temps un nouvel arbre mono-optique ou arbre mono-optique final. Cet arbre mono-optique final est noté $T_f = \left(V_f, E_f(\lambda_0)\right)$. T_f est enraciné au nœud source r et couvre tous les éléments de D, avec $V_f \subseteq V$, $E_f \subseteq E$, $r \in V_f$ et $D \subseteq V_f \setminus \{r\}$. $E_f(\lambda_0) = \bigcup_{i=1}^{|E_f|} e_i(\lambda_0)$ est l'ensemble des canaux optiques à établir. L'arbre mono-optique final est aussi appelé la configuration finale notée C_f. Ensuite, les opérations de configuration (présentées au chapitre précédent) doivent être exécutées sur les nœuds appartenant à V_f afin d'établir l'arbre mono-optique final. Cependant, l'arbre mono-optique final T_f peut partager quelques canaux optiques avec l'arbre mono-optique initial T_0. Par conséquent, certaines configurations intermédiaires doivent être trouvées avant d'obtenir la configuration finale C_f. En d'autres termes, une séquence de configurations $SC =< C_0, C_1, \ldots, C_{f-1}, C_f >$ doit être construite. Une sé-

quence de configurations utile ne peut pas contenir une configuration intermédiaire C_k qui serait obtenue en appliquant une opération de configuration sur un des nœuds qui n'appartiendrait pas à l'ensemble $V_0 \cup V_f$. Il en est ainsi car on suppose implicitement que tous les liens sont totalement exploités par des connexions sur le réseau. Il est important de rappeler qu'une configuration est définie comme un ensemble composé de canaux optiques et éventuellement de semi-canaux optiques.

Une configuration intermédiaire C_k peut entraîner une interruption de flux vers au moins un des nœuds destinations appartenant à D si certains nœuds n'ont pas été correctement configurés. Dans ce cas, $\exists\ d \in D$ tel qu'il n'y a pas de chemin optique entièrement configuré entre le nœud source r de $CM\,(r, D)$ et le nœud destination d. Notez qu'un chemin optique n'est pas entièrement configuré s'il contient au moins un canal optique qui n'est pas établi ou si ce chemin optique contient au moins un semi-canal optique. Soit $interrupt_rate\,(C_k)$, le taux d'interruptions de flux causé par une configuration intermédiaire C_k. Ce taux est défini par l'Equation (4.1) ci-dessous :

$$interrupt_rate\,(C_k) = \frac{\sum_{d \in D} C_k\,(r, d)}{|D|} \tag{4.1}$$

où $C_k(r, d) = \begin{cases} 1 & \text{si le chemin entre } r \text{ et } d \text{ dans } C_k \text{ n'est pas entièrement configuré} \\ 0 & \text{Sinon.} \end{cases}$

Une configuration intermédiaire C_k est par définition différente de C_0 et C_f, avec par convention $f = |SC| - 1$. Ainsi, on a $1 \leq k \leq |SC| - 2$. Par conséquent, le taux moyen d'interruptions de flux causé par la séquence de configurations $SC =< C_0, C_1, \ldots, C_{f-1}, C_f >$ est défini par l'Equation (4.2) ci-dessous :

$$interrupt_rate\,(SC) = \sum_{k=1}^{|SC|-2} \frac{interrupt_rate\,(C_k)}{|SC| - 2} \tag{4.2}$$

Une configuration intermédiaire C_k peut utiliser une longueur d'onde additionnelle. En d'autres termes, les canaux optiques appartenant à C_k peuvent utiliser une longueur d'onde additionnelle. Cependant, les longueurs d'onde additionnelles doivent être utilisées avec parcimonie. En effet, l'opérateur de réseau pourrait avoir besoin des longueurs d'onde additionnelles pour effectuer une autre tâche pendant que la tâche de reconfiguration a lieu [110]. Soit $add_cost\,(SC)$ le coût des longueurs d'onde additionnelles qui équivaut au nombre total de fois où des canaux optiques utilisent une longueur d'onde additionnelle dans la séquence de configurations SC. Soit $add_cost\,(C_k)$

le nombre de canaux optiques appartenant à la configuration C_k qui utilisent une longueur d'onde additionnelle. $add_cost\,(C_k)$ est défini par l'Equation (4.3) :

$$add_cost\,(C_k) = \sum_{e_i(\lambda_i)\in C_k} S\,(e_i\,(\lambda_i)\,, C_k) \tag{4.3}$$

où $S\,(e_i\,(\lambda_i)\,, C_k) = \begin{cases} 1 & \text{si } \lambda_i \text{ utilisée par le canal } e_i(\lambda_i) \text{ dans } C_k \text{ est tel que } \lambda_i \in \Omega_a \\ 0 & \text{Sinon.} \end{cases}$

Par conséquent, $add_cost\,(SC)$ est défini par l'Equation (4.4) :

$$add_cost\,(SC) = \sum_{k=1}^{|SC|-2} add_cost\,(C_k) \tag{4.4}$$

Notons que chaque configuration C_k dans l'équation (4.4) est une configuration intermédiaire, car la configuration initiale C_0 et la configuration finale C_f par définition ne nécessitent pas de longueur d'onde additionnelle.

Avec toutes les notations précédentes, le problème de la reconfiguration d'une paire d'arbres mono-optiques est formulé comme suit :

— **Données** : L'arbre mono-optique initial (établi) T_0 , l'arbre mono-optique final (à établir) T_f.

— **Solution :** La séquence de configurations $SC =< C_0, C_1, \ldots, C_{f-1}, C_f >$ qui permet de migrer le flux optique de T_0 vers T_f.

— **Objectif** : Produire la séquence de configurations en évitant des interruptions de flux (ce qui correspond à un taux moyen d'interruption de flux égal à zéro) tout en réduisant le coût des longueurs d'onde additionnelles.

4.3 Méthodes de reconfiguration d'une paire d'arbres mono-optiques

La recherche d'une solution optimale au problème formulé précédemment s'avère délicat en pratique. En effet, cela implique de définir d'une part, toutes les paires de sous-structures qu'on peut extraire d'une paire d'arbres mono-optiques. D'autre part, pour chacune de ces sous-structures, il faut tester toutes les séquences possibles d'opérations de configurations sur les nœuds de ces sous-structures. Face à cette

difficulté d'obtention d'une solution optimale, les méthodes existantes ont fait le choix d'une approche basée sur la sous-structure appelée branche qui garantit de bonnes performances dans un réseau où tous les nœuds ont la capacité de conversion de longueur d'onde.

Cependant, nous avons montré dans le chapitre précédent que les méthodes existantes ne permettent pas dans certains cas de reconfigurer une paire d'arbres mono-optiques sans interruption de flux dans un réseau optique WDM ayant la propriété de conversion partielle de longueur d'onde. Par conséquent, nous proposons des méthodes procédant à la reconfiguration d'une paire d'arbres mono-optiques en se basant sur la définition d'une autre sous-structure appelée sous-arbre.

Dans la section 4.3.1, nous identifions les différents types de paires de sous-arbres que l'on peut sélectionner dans une paire d'arbres mono-optiques de sorte que leurs reconfigurations ne causent pas des interruptions de flux tout en utilisant un faible nombre de fois des longueurs d'onde additionnelles. Dans la section 4.3.2, nous proposons une première méthode de reconfiguration se basant sur les résultats de la section 4.3.1. Puis nous terminons par la proposition d'une deuxième méthode (voir section 4.3.3) qui permet de fournir une plus courte séquence de configurations que la première méthode lors de la reconfiguration d'une paire d'arbres mono-optiques.

4.3.1 Caractérisation de paires de sous-arbres

Une méthode de reconfiguration d'une paire d'arbres mono-optiques (arbre mono-optique initial, arbre mono-optique final) par paire de sous-arbres consiste à sélectionner des paires de sous-arbres dans la paire d'arbres mono-optiques (arbre mono-optique actuel, arbre mono-optique final) et reconfigurer ces paires de sous-arbres. Notons qu'au début de tout processus de reconfiguration, l'arbre mono-optique actuel est égal à l'arbre mono-optique initial. Chaque paire de sous-arbres (sous-arbre actuel, nouveau sous-arbre) est telle que le sous-arbre actuel est un sous-arbre de l'arbre mono-optique actuel et le nouveau sous-arbre est un sous-arbre de l'arbre mono-optique final à établir.

Dans la suite de ce chapitre, (ST_c, ST_f) désigne une paire de sous-arbres, où ST_c est un sous-arbre de l'arbre mono-optique actuel T_c appelé sous-arbre actuel et ST_f est un sous-arbre de l'arbre mono-optique final T_f appelé nouveau sous-arbre. Nous admettons que (ST_c, ST_f) est une paire de sous-arbres qui ne nécessite pas d'être

reconfigurée si ST_f et ST_c partagent exactement les mêmes liens (c'est-à-dire ST_f et ST_c sont identiques). Ainsi, les deux seules catégories de paires de sous-arbres à prendre en compte dans la reconfiguration d'une paire d'arbres mono-optiques sont les suivantes :

1. **La catégorie des paires de sous-arbres à liens disjoints** : la paire de sous-arbres (ST_c, ST_f) est une paire de sous-arbres à liens disjoints si le nouveau sous-arbre ST_f n'a aucun lien en commun avec l'arbre mono-optique actuel T_c et donc ST_f ne partage aucun lien avec le sous-arbre actuel ST_c. Par conséquent, la reconfiguration d'une telle paire ne nécessite aucune longueur d'onde additionnelle. Autrement dit, pour une paire de sous-arbres à liens disjoints, une méthode de reconfiguration par paire de sous-arbres n'emploie à aucun instant de longueur d'onde additionnelle.
2. **La catégorie des paires de sous-arbres à liens partagés** : La paire de sous-arbres (ST_c, ST_f) est une paire de sous-arbres à liens partagés si le nouveau sous-arbre ST_f partage des liens (mais pas tous ses liens) avec le sous-arbre actuel ST_c. Par conséquent, la reconfiguration d'une telle paire nécessite l'utilisation de longueurs d'onde additionnelles. En effet, le préétablissement du nouveau sous-arbre nécessite l'utilisation d'au moins une longueur d'onde additionnelle.

4.3.1.1 Sélection d'une paire de sous-arbres à liens disjoints

Une Paire de Sous-arbres (un sous-arbre actuel, un nouveau sous-arbre) à Liens Disjoints (PSLD) à sélectionner doit être reconfigurable sans interruption de flux. Notez que pour une telle paire de sous-arbres, le nouveau sous-arbre ne partage pas de liens avec le sous-arbre actuel. Cela implique que le nœud racine de cette paire doit être un nœud divergent. De même, tous les nœuds feuilles de chaque sous-arbre constituant la paire à sélectionner doivent être des nœuds convergents. Un nœud divergent appartient à l'arbre mono-optique actuel T_c et à l'arbre mono-optique final T_f. Aussi, un nœud divergent possède au moins un nœud enfant sur T_c qui est différent de ses nœuds enfants sur T_f. Un nœud convergent appartient à l'arbre mono-optique actuel T_c et à l'arbre mono-optique final T_f. Aussi, un nœud convergent a son nœud parent sur l'arbre actuel T_c qui est différent de son nœud parent sur l'arbre mono-optique final T_f.

❖ Présentation de l'algorithme

Étant donné une paire d'arbres (T_c, T_f) et un nœud divergent n, l'algorithme $Select_PSLD$ (voir Algorithme 4.1) renvoie une paire de sous-arbres à liens disjoints qui peut être reconfigurée sans interruption du flux. Le processus de sélection d'une telle paire de sous-arbres (ST_c, ST_f) commence par la recherche de l'ensemble des nœuds convergents CG_{ST_c,ST_f} de la paire (T_c, T_f) qui doivent appartenir à la paire de sous-arbres (voir de la ligne 5 à la ligne 13). Si CG_{ST_c,ST_f} est vide, alors l'Algorithme 4.1 renvoie la paire de sous-arbres $null$. Sinon, le sous-arbre actuel ST_c est le sous-arbre de l'arbre actuel T_c qui est enraciné au nœud n et qui couvre tous les éléments de CG_{ST_c,ST_f} (voir de la ligne 15 à la ligne 16). De même, le nouveau sous-arbre ST_f est le sous-arbre de l'arbre mono-optique final T_f qui est enraciné au nœud n et qui couvre tous les éléments de CG_{ST_c,ST_f} (voir de la ligne 17 à la ligne 18). Lors de la reconfiguration de la paire de sous-arbres (ST_c, ST_f), les canaux du nouveau sous-arbre ST_f seront établis afin d'être alimentés par le flux optique et le sous-arbre actuel ST_c sera supprimé. Pour sélectionner correctement l'ensemble des nœuds convergents CG_{ST_c,ST_f} de la paire de sous-arbres (ST_c, ST_f), on désigne par $Est_interrompu\,(T_c, T_f, n, x)$ (voir ligne 9) toute fonction renvoyant VRAI si au moins une des conditions d'interruption de flux suivantes est remplie et FAUX sinon :

— *Condition 1 :* Le nœud convergent x a au moins un ancêtre y sur T_c qui n'appartient pas à T_f et le nœud y est un descendant du nœud divergent n sur T_c. De plus, le nœud y a un descendant n_y qui n'est pas un ancêtre du nœud convergent x et qui n'est pas un descendant de n sur T_f.

— *Condition 2 :* Le nœud convergent x a au moins un ancêtre y sur T_c qui appartient également à T_f et le nœud y est un descendant du nœud divergent n sur T_c mais il n'est pas un descendant de n sur T_f.

— *Condition 3 :* il existe un ancêtre y du nœud convergent x sur le segment de T_f qui relie le nœud divergent n au nœud convergent x de telle sorte que y n'est pas un ancêtre de x sur le segment de T_c qui relie n à x.

➢ Analyse de la correction de l'algorithme

Nous rappelons que Algorithme 4.1 prend en entrées une paire d'arbres mono-optiques (T_c, T_f) et un nœud divergent n puis retourne soit une paire de sous-arbres à liens disjoints qui n'interrompt pas le flux soit la paire de sous-arbres $null$. Nous

Algorithme 4.1 : Select_PSLD

Entrées : T_c, T_f, n// T_c : `Arbre actuel,` T_f : `Arbre final,` n : `nœud divergent`
Sorties : ST_c, ST_f // ST_c : `Sous-arbre actuel de racine` n, ST_f : `Nouveau sous-arbre de racine` n

1 $CG_n \leftarrow$ Ensemble de nœuds convergents qui sont des nœuds descendants de n sur T_f et T_c
2 $Dis_CG_n \leftarrow \{x \setminus x \in CG_n$ et le segment reliant n à x sur T_c est à liens disjoints du segment reliant n à x sur $T_f\}$
3 $ST_c \leftarrow null$; $ST_f \leftarrow null$
4 $CG_{ST_c,ST_f} \leftarrow null$ // `L'ensemble des nœuds convergents de la paire à sélectionner`
5 $OUT_LIENS_{T_c,n} \leftarrow$ Ensemble des liens de l'arbre T_c émanant du nœud n
6 **Tant que** $CG_{ST_c,ST_f} = null$ ***Et*** $OUT_LIENS_{T_c,n}$ *est non vide* **faire**
7 | $out_lien \leftarrow$ Premier élément de $OUT_LIENS_{T_c,n}$
8 | $CG_{ST_c,ST_f} \leftarrow$ Sous-ensemble de Dis_CG_n tel que tous les chemins reliant n aux éléments de CG_{ST_c,ST_f} sur l'arbre T_c partagent le lien out_lien
9 | **Si** $\exists x \in CG_{ST_c,ST_f}$ ***Et*** $Est_interrompu\,(T_c, T_f, n, x)$ **alors**
10 | | $CG_{ST_c,ST_f} \leftarrow null$
11 | **Fin Si**
12 | $OUT_LIENS_{T_c,n} \leftarrow OUT_LIENS_{T_c,n} \setminus \{out_lien\}$
13 **Fin Tant que**
14 **Si** $CG_{ST_c,ST_f} \neq null$ **alors**
15 | $D_c \leftarrow$ Plus grand sous-ensemble de CG_{ST_c,ST_f} tel que chaque élément de ce sous-ensemble ne soit l'ancêtre sur l'arbre T_c d'aucun élément de CG_{ST_c,ST_f}
16 | $ST_c \leftarrow$ Sous-arbre de T_c enraciné au nœud n et ayant D_c comme ensemble de nœuds feuilles
17 | $D_f \leftarrow$ Plus grand sous-ensemble de CG_{ST_c,ST_f} tel que chaque élément de ce sous-ensemble ne soit l'ancêtre sur l'arbre T_f d'aucun élément de CG_{ST_c,ST_f}
18 | $ST_f \leftarrow$ Sous-arbre de T_f enraciné au nœud n et ayant D_f comme ensemble de nœuds feuilles
19 **Fin Si**
20 **Retourner** (ST_c, ST_f)

montrons la correction totale de cet algorithme dans la suite de ce paragraphe. Si la condition de la ligne 14 de l'Algorithme 4.1 est vérifiée alors l'Algorithme 4.1 retourne une paire de sous-arbres à liens disjoints qui n'interrompt pas le flux vers les nœuds destinations de la paire d'arbres (T_c, T_f) (: résultat i). En effet, la fonction $Est_interrompu\,(T_c, T_f, n, x)$ est utilisée dans le choix des nœuds convergents devant permettre de sélectionner une paire de sous-arbres à liens disjoints garantissant la continuité du flux. Si la condition de la ligne 14 de l'Algorithme 4.1 n'est pas vérifiée alors l'Algorithme 4.1 retourne une paire $null$ (: résultat ii). Des résultats i et ii, on déduit la correction partielle de l'Algorithme 4.1 (: résultat iii). L'ensemble des liens émanant du nœud divergent n noté $OUT_LIENS_{T_c,n}$ diminue à chaque itération de la boucle «tant que» (voir ligne 12 de l'Algorithme 4.1). Par conséquent, l'Algorithme 4.1 se termine (: résultat iv). Des résultats iii et iv, on déduit la correction totale de l'Algorithme 4.1.

➢ Analyse de la complexité de l'algorithme

L'analyse de la complexité temporelle dépend des éléments suivants :

— Le cardinal de l'ensemble des nœuds de l'arbre courant $T_c = (V_c, E_c)$ noté $|V_c|$.

— Le cardinal de l'ensemble des nœuds de l'arbre courant $T_f = (V_f, E_f)$ noté $|V_f|$.

Notons que ces notations sont valables dans la suite de ce chapitre.

Lemme 4.1 *Le processus (voir Algorithme 4.1) de sélection d'une paire d'arbres à liens disjoints induit une complexité temporelle (dans le pire des cas) en* $O\left(p.|V_c|^3 + p.|V_f|.|V_c|\right)$, *où* $p = max\left\{|V_c|\,;|V_f|\right\}$.

Preuve On suppose que toute recherche de chemin entre deux nœuds se fait via l'algorithme de recherche en largeur d'abord ou *Breadth-First Search* (BFS). Rappelons que la complexité temporelle (dans le pire des cas) de BFS sur un graphe $G = (V, E)$ quelconque est en $O(|V| + |E|)$. Cependant, si G est un arbre alors cette complexité est de $O(|V|)$. Aussi, un parcours en profondeur d'abord ou *Depth-First Search* (DFS) est requis pour rechercher les ascendants ou les descendants d'un nœud u sur un arbre T avec une complexité également en $O(|V|)$. On convient que $Comp\,(ligne\ x)$ désigne la complexité temporelle de l'instruction indiquée à la ligne x et $Comp\,(ligne\ x - y)$ désigne la complexité temporelle du bloc d'instructions commençant à la ligne x et se terminant à la ligne y. On convient par souci de concision de ne pas mentionner tout $Comp\,(ligne\ x)$ en $O(1)$. $Comp\,(ligne\ 1)$ est en $O(p.|V_c|)$, où $p = max\left\{|V_c|\,;|V_f|\right\}$. Par définition, $|CG_n| \leq q$, où $q = |V_c \cap V_f|$. Ainsi, $Comp\,(ligne\ 2)$ est en $O\left(q.|V_c|.|V_f|\right)$. De plus, $Comp\,(ligne\ 5)$ est en $O(|V_c|)$. Il s'ensuit que $Comp\,(ligne\ 1-5)$ est en $O(l)\ (i)$, où $l = max\{q.|V_c|.|V_f|\,;p.|V_c|\}$. Par ailleurs, $Comp\,(ligne\ 6-13)$ est en $O\left(p.|V_c|^3 + p.|V_f|.|V_c|\right)\ (ii)$ car $Comp\,(ligne\ 8)$ est en $O(q.|V_c|)$, et $Comp\,(ligne\ 9)$ (qui est déterminé par les complexités des trois conditions d'interruptions) est en $O\left(p.|V_c|^2 + p.|V_f|\right)$. De plus, $Comp\,(ligne\ 14-19)$ est en $O\left(q.|V_c|^2 + q.|V_f|^2\right)\ (iii)$ car $Comp\,(ligne\ 15)$ est en $O\left(q^2.|V_c|\right)$, $Comp\,(ligne\ 16)$ est en $O\left(q\,|V_c|^2\right)$, $Comp\,(ligne\ 17)$ est en $O\left(q^2.|V_f|\right)$ et $Comp\,(ligne\ 18)$ est en $O\left(q\,|V_f|^2\right)$. De (i), (ii) et (iii), on conclut que le processus de sélection décrit par l'Algorithme 4.1 est effectivement en $O\left(p.|V_c|^3 + p.|V_f|.|V_c|\right)$.

➢ Illustration de l'algorithme

Nous illustrons maintenant, la sélection d'une paire de sous-arbres à liens disjoints dans une paire d'arbres mono-optiques à reconfigurer. La Figure 4.1 montre une instance de problème d'une paire d'arbres mono-optiques à reconfigurer qui concerne la connexion multicast $< s, \{g, h\} >$. L'ensemble des traits continus de couleur bleue forme l'ensemble des liens de l'arbre mono-optique actuel T_c qui est enraciné au nœud s. L'ensemble des traits discontinus de couleur bleue forme l'ensemble des liens de l'arbre mono-optique final T_f. L'arbre mono-optique T_f est aussi enraciné au nœud s. Dans cette instance, nous n'avons qu'un seul nœud divergent qui est le nœud a. En effet, le nœud a appartient à l'arbre mono-optique actuel et à l'arbre mono-optique final. De plus, b est un nœud enfant du nœud a sur l'arbre mono-optique actuel mais b n'est pas un nœud enfant du nœud a sur l'arbre mono-optique final.

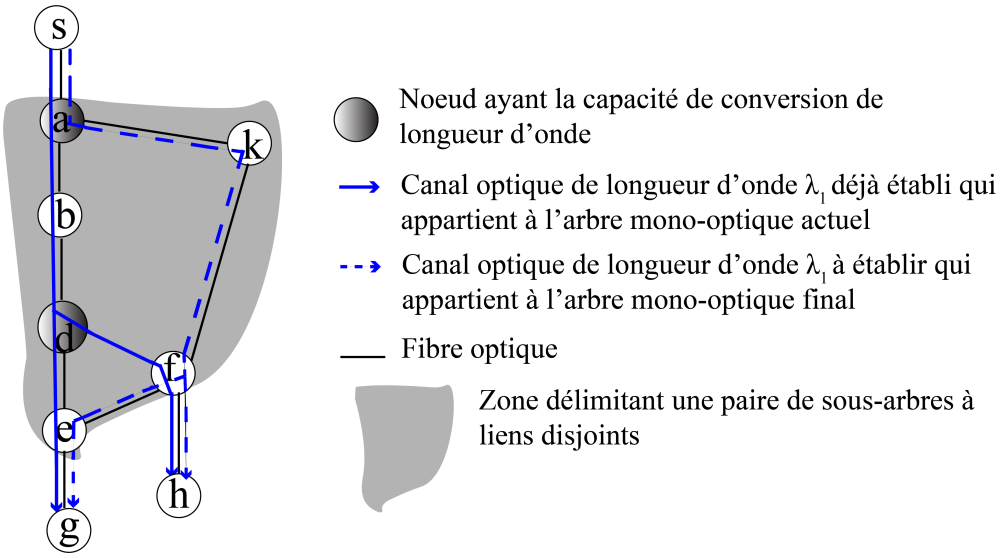


FIGURE 4.1 – Exemple de problème contenant une paire de sous-arbres à liens disjoints

Appliquons l'Algorithme 4.1 avec cette paire d'arbres et le nœud divergent a comme entrées. e et f sont des nœuds convergents descendant du nœud divergent a sur les deux arbres, c'est-à-dire $CG_a = \{e, f\}$. Le chemin reliant le nœud a au nœud e sur l'arbre actuel est à liens disjoints du chemin qui relie le nœud a au nœud e sur l'arbre mono-optique final. Il en est de même pour le nœud f. Par conséquent, $Dis_CG_a = CG_a$. Le lien $a \rightarrow b$ est l'unique lien émanant du nœud a sur T_c. Le chemin

reliant le nœud a au nœud e sur l'arbre mono-optique T_c et le chemin reliant le nœud a au nœud f sur l'arbre mono-optique T_c partagent le lien $a \rightarrow b$. En outre, les nœuds e et f ne remplissent aucune des *conditions 1, 2 et 3*. En effet, l'ensemble des ancêtres du nœud e sur l'arbre mono-optique actuel qui sont des nœuds descendants du nœud a est $\{b, d\}$. Par ailleurs, les nœuds b et d n'appartiennent pas à l'arbre mono-optique final. Mais chacun de ces deux nœuds n'a pas de nœud descendant tel que ce nœud descendant ne serait pas l'ancêtre du nœud e et ne serait pas un nœud descendant du nœud a sur l'arbre mono-optique final. Par conséquent, le nœud e ne remplit pas la *condition 1*. En outre, les nœuds b et d ne se trouvent que sur l'arbre mono-optique actuel. Par conséquent, le nœud e ne remplit pas la *condition 2*. De même, le nœud f ne remplit pas la *condition 1* et la *condition 2*. Le nœud e n'a pas d'ancêtre (appartenant aux deux arbres mono-optiques) sur le chemin dans T_f reliant le nœud a au nœud e qui ne soit pas son ancêtre sur le chemin dans T_c reliant le nœud a au nœud e : le nœud e ne remplit pas la *condition 3*. De même, le nœud f ne remplit pas la *condition 3*. Ainsi, à la fin de l'étape de recherche de l'ensemble des nœuds convergents (voir ligne 13), $CG_{ST_c,ST_f} = CG_a$. Par conséquent, selon la ligne 15 et la ligne 16, le sous-arbre actuel ST_c est égal à $a \rightarrow b \rightarrow d \rightarrow \{e, f\}$. En outre, selon la ligne 17 et la ligne 18, le nouveau sous-arbre ST_f est égal à $a \rightarrow k \rightarrow f \rightarrow e$. L'Algorithme 4.1 se termine et retourne la paire de sous-arbres $(ST_c, ST_f) = (a \rightarrow b \rightarrow d \rightarrow \{e, f\}, a \rightarrow k \rightarrow f \rightarrow e)$.

4.3.1.2 Sélection d'une paire de sous-arbres à liens partagés

❖ **Présentation de l'algorithme**

Afin de sélectionner une Paire de Sous-arbres à Liens Partagés (PSLP), nous proposons la fonction appelée $Select_PSLP$ (voir Algorithme 4.2). Cette fonction prend en entrées la paire d'arbres mono-optiques (T_c, T_f) et un nœud convergent m. En fait, l'existence d'une paire de sous-arbres à liens partagés dans une paire d'arbres mono-optiques implique la présence d'au moins un nœud convergent sur la paire d'arbres mono-optiques. Cela signifie également que T_c et T_f ne sont pas identiques. $Select_PSLP$ retourne une paire de sous-arbres à liens partagés (ST_c, ST_f). Le sous-ensemble des nœuds destinations couvert par ST_c doit être identique au sous-ensemble des nœuds destinations couvert par ST_f afin de ne pas provoquer d'interruptions de flux lors de la reconfiguration de cette paire de sous-arbres à liens partagés.

La sélection d'une paire de sous-arbres à liens partagés (ST_c, ST_f) commence par

la détermination du nœud racine r de la paire de sous-arbres à sélectionner. Par construction, le nœud r doit être un ancêtre du nœud convergent m sur T_c et T_f. En outre, (ST_c, ST_f) est une paire de sous-arbres avec des liens partagés, de sorte que l'utilisation d'une longueur d'onde additionnelle est requise pour sa reconfiguration. Par conséquent, si le nœud r est différent du nœud racine de la paire d'arbres mono-optiques (T_c, T_f) alors le nœud r appartient à l'ensemble des nœuds ayant la capacité de conversion de longueur d'onde noté V_{cl} (c'est-à-dire $r \in V_{cl}$). Soit $D(a, b, T)$, le plus grand sous-ensemble de l'ensemble des nœuds destinations de l'arbre T de sorte que chaque élément de ce sous-ensemble a un ancêtre sur le chemin dans T reliant le nœud a au nœud b. En bref, le nœud racine r d'une paire de sous-arbres à liens partagés (ST_c, ST_f) est égal au nœud n (avec $n \in V_{cl}$) qui est le plus jeune ancêtre commun du nœud convergent m sur T_c et T_f, de sorte que $D(n, m, T_c) = D(n, m, T_f)$ (voir la ligne 2 de l'Algorithme 4.2). Notons que si un tel nœud n n'existe pas, alors le nœud racine de la paire d'arbres mono-optiques (T_c, T_f) est pris comme nœud racine de la paire de sous-arbres (ST_c, ST_f) à sélectionner. Après avoir déterminé le nœud racine et les nœuds destinations d'une paire de sous-arbres à liens partagés, les sous-arbres peuvent être sélectionnés. Le sous-arbre ST_c est le sous-arbre de T_c enraciné au nœud r et ayant les éléments de $D(r, m, T_c)$ comme nœuds feuilles (voir ligne 6). De même, le sous-arbre ST_f est le sous-arbre de T_f enraciné au nœud r et ayant comme nœuds feuilles les éléments de $D(r, m, T_f)$ (voir ligne 7).

Algorithme 4.2 : Select_PSLP

Entrées : T_c, T_f, m,// T_c : `Arbre actuel`, T_f : `Arbre final`, m : `nœud convergent de` (T_c, T_f)
Sorties : ST_c, ST_f. // ST_c : `Sous-arbre actuel`, ST_f : `Nouveau sous-arbre`
1 $V_{cl} \leftarrow$ Ensemble de nœuds ayant la capacité de conversion de longueur d'onde de la paire (T_c, T_f)
2 $r \leftarrow$ Nœud n de V_{cl} qui est le plus jeune ancêtre à la fois sur T_c et T_f de m tel que $D(n, m, T_c) = D(n, m, T_f)$
3 **Si** $r = null$ **alors**
4 | $r \leftarrow Racine(T_c, T_f)$ // $Racine(T_c, T_f)$ `désigne la racine de la paire d'arbres` (T_c, T_f)
5 **Fin Si**
6 $ST_c \leftarrow$ Sous-arbre de T_c enraciné au nœud n et ayant $D(r, m, T_c)$ comme ensemble de nœuds feuilles
7 $ST_f \leftarrow$ Sous-arbre de T_f enraciné au nœud n et ayant $D(r, m, T_f)$ comme ensemble de nœuds feuilles
8 **Retourner** (ST_c, ST_f)

➢ Analyse de la correction de l'algorithme

Nous rappelons que l'Algorithme 4.2 prend en entrées une paire d'arbres (T_c, T_f) et un nœud convergent m puis retourne une paire de sous-arbres à liens partagés qui n'interrompt pas le flux. Nous montrons la correction totale de cet algorithme dans la

suite de ce paragraphe. Les lignes 2 à 5 permettent de trouver dans tous les cas un nœud non nul qui est la racine de la paire à sélectionner (: résultat i). Les lignes 2, 6 et 7 indiquent que les sous-arbres de la paire sélectionnée atteignent le même ensemble de nœuds destinations (:résultat ii). Autrement dit, le basculement sur le nouveau sous-arbre ne risque pas de priver les nœuds destinations du flux. Les résultats i et ii prouvent la correction partielle de l'Algorithme 4.2 (: résultat iii). Il est trivial de dire que l'Algorithme 4.2 se termine (: résultat iv). Les résultats iii et iv prouvent la correction totale de l'Algorithme 4.2.

➢ Analyse de la complexité de l'algorithme

Lemme 4.2 *Le processus (voir Algorithme 4.2) de sélection d'une paire d'arbres à liens partagés induit une complexité temporelle (dans le pire des cas) en* $O\left(|V_c|^4 + |V_f|^3 . |V_c|\right)$.

Preuve Les hypothèses et conventions mentionnées dans la preuve du lemme 4.1 sont également valables ici. Ainsi, $Comp\ (ligne\ 1)$ est en $O\left(|V_c| . |V_f|\right)$, $Comp\ (ligne\ 6)$ est en $O\left(|V_c|^3\right)$ et $Comp\ (ligne\ 7)$ est en $O\left(|V_f|^3\right)$. Par ailleurs, $Comp\ (ligne\ 2)$ est en $O\left(|V_c|^4 + |V_f|^3 . |V_c|\right)$. On conclut que le processus de sélection décrit par l'Algorithme 4.2 est effectivement en $O\left(|V_c|^4 + |V_f|^3 . |V_c|\right)$.

➢ Illustration de l'algorithme

Dans la Figure 4.2, l'ensemble des traits continus et de couleur bleue forme l'ensemble des liens de l'arbre mono-optique actuel T_c enraciné au nœud s. L'ensemble des traits qui sont discontinus et de couleur bleue forme l'ensemble des liens de l'arbre mono-optique final T_f enraciné au nœud s. Le nœud a est un nœud qui a la capacité de conversion de longueur d'onde et qui est un ancêtre du nœud convergent g à la fois sur T_c et sur T_f. Cependant, l'ensemble des nœuds destinations sur l'arbre mono-optique actuel T_c qui ont un ancêtre sur le chemin reliant le nœud a au nœud g est $D\ (a, g, T_c) = \{f, h, l\}$. L'ensemble des nœuds destinations sur l'arbre mono-optique final T_f qui ont un ancêtre sur le chemin reliant le nœud a au nœud g est $D\left(a, g, T_f\right) = \{f, h\}$. Comme $D\ (a, g, T_c) \neq D\left(a, g, T_f\right)$, nous recherchons un autre ancêtre du nœud g. Le nœud k est également un nœud ayant la capacité de conversion de longueur d'onde qui est un ancêtre du nœud convergent g à la fois sur T_c et sur T_f. En outre, $D\ (k, g, T_c) = D\left(k, g, T_f\right) = \{f, h, l\}$. Par conséquent, la racine de

la paire de sous-arbres à sélectionner est le nœud k. Le sous-arbre actuel ST_c est le sous-arbre de T_c, enraciné au nœud k et couvrant les nœuds destinations l, f et h. Le nouveau sous-arbre ST_f est le sous-arbre de T_f, enraciné au nœud k et couvrant les nœuds destinations l, f et h. La paire de sous-arbres à liens partagés $(ST_c, ST_f) = (k \rightarrow a \rightarrow b \rightarrow \{l, c \rightarrow d \rightarrow \{h, e \rightarrow g \rightarrow f\}\}, k \rightarrow \{l, a \rightarrow b \rightarrow \{g \rightarrow f, c \rightarrow d \rightarrow h\}\})$ est alors obtenue, où $k \rightarrow a \rightarrow b \rightarrow \{l, c \rightarrow d \rightarrow \{h, e \rightarrow g \rightarrow f\}\}$ désigne le sous-arbre actuel ST_c et $k \rightarrow \{l, a \rightarrow b \rightarrow \{g \rightarrow f, c \rightarrow d \rightarrow h\}\}$ désigne le nouveau sous-arbre ST_f.

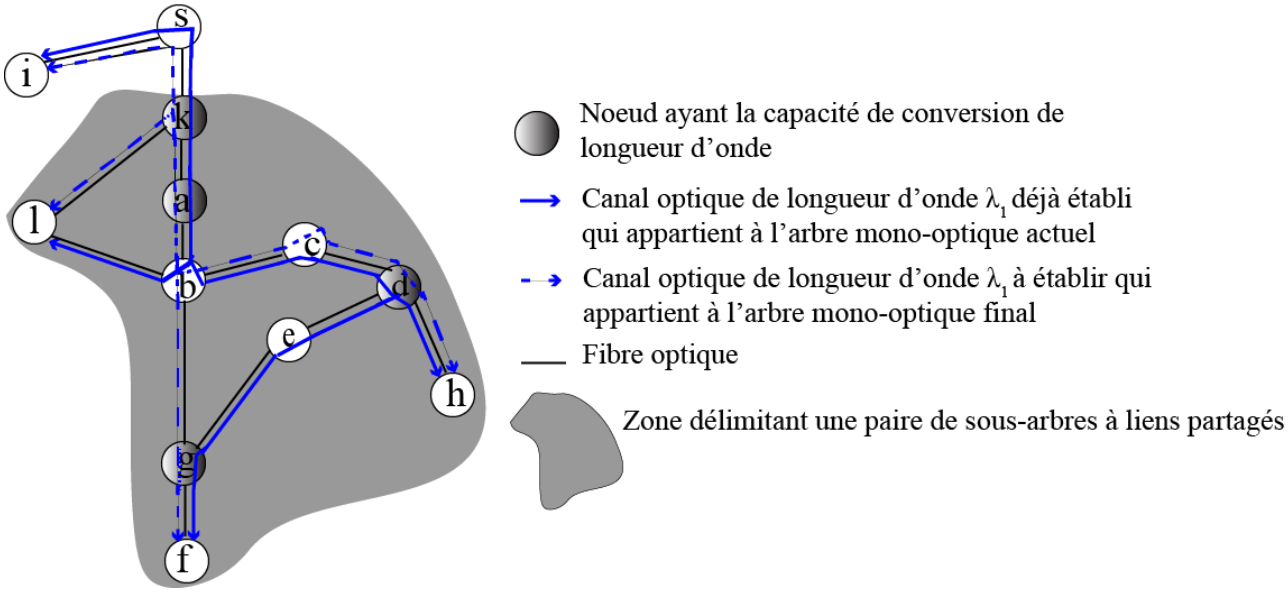


FIGURE 4.2 – Exemple de problème contenant une paire de sous-arbres à liens partagés

4.3.2 Méthode de reconfiguration SbSRA

SbSRA est une méthode de reconfiguration d'une paire d'arbres mono-optiques par paire de sous-arbres. Par ailleurs, la section 4.3.1 nous apprend que les paires de sous-arbres qu'on peut sélectionner dans la paire d'arbres mono-optiques peuvent appartenir à deux catégories à savoir la catégorie des paires de sous-arbres à liens disjoints et celle des paires de sous-arbres à liens partagés. Par conséquent, il est judicieux de s'interroger sur l'ordre dans lequel doivent être reconfigurés les paires

de sous-arbres d'une paire d'arbres mono-optiques à reconfigurer. Il est clair qu'il faut choisir entre les trois ordres de reconfiguration suivants :

— *Cas 1 :* Reconfigurer en plusieurs phases les paires de sous-arbres. A chaque phase, on reconfigure dans un premier temps un certain nombre de paires de sous-arbres à liens partagés (respectivement un certain nombre de paires de sous-arbres à liens disjoints) puis dans un second temps un certain nombre de paires de sous-arbres à liens disjoints (respectivement un certain nombre de paires de sous-arbres à liens partagés) ;

— *Cas 2 :* Reconfigurer dans un premier temps toutes les paires de sous-arbres à liens partagés puis dans un second temps toutes les paires de sous-arbres à liens disjoints ;

— *Cas 3 :* Reconfigurer dans un premier temps toutes les paires de sous-arbres à liens disjoints puis dans un second temps toutes les paires de sous-arbres à liens partagés ;

Le *cas 1* implique qu'il faut connaître le nombre de paires de sous-arbres à liens disjoints et aussi le nombre de paires de sous-arbres à liens partagés à utiliser à chaque phase. Il faudrait déterminer donc ces deux nombres sur la base de critères difficilement objectifs. Par ailleurs, la spécification du problème (voir section 4.2) a mis en lumière le fait que la reconfiguration d'une paire d'arbres mono-optiques doit utiliser avec parcimonie les longueurs d'onde additionnelles. Par conséquent, il est plus efficace de choisir le *cas 3* plutôt que le *cas 2*. Cependant, il faut justifier qu'aucune paire de sous-arbres à liens disjoints qui est non reconfigurable sans interruption de flux à une étape du processus de reconfiguration puisque respectant au moins une des trois conditions d'interruption de flux (voir la section 4.3.1.1) ne devient reconfigurable sans interruption de flux après la reconfiguration d'une paire de sous-arbres à liens partagés. En effet, si cela est possible alors notre méthode devra après la reconfiguration des paires à liens partagés reconfigurer les paires de sous-arbres à liens disjoints devenues éligibles à la reconfiguration sans interruption de flux. Ainsi, dans la section 4.3.2.1, nous justifions avec une preuve par l'absurde qu'aucune paire de sous-arbres à liens disjoints non reconfigurable sans interruption de flux à une étape du processus de reconfiguration puisque respectant au moins une des trois conditions d'interruption de flux (voir la section 4.3.1.1), ne devient reconfigurable sans interruption de flux après la reconfiguration d'une paire de sous-arbres à liens partagés.

4.3.2.1 Justification de l'ordre de reconfiguration des catégories de paires sous-arbres

Théorème 4.1: Théorème d'immuabilité

Aucune paire de sous-arbres à liens disjoints non reconfigurable sans interruption de flux ne devient reconfigurable sans interruption de flux après la reconfiguration d'une paire de sous-arbres à liens partagés.

Preuve (Par l'absurde) Soit une paire de sous-arbres à liens disjoints que nous notons (ST_c^1, ST_f^1) non reconfigurable sans interruption de flux à une étape du processus de reconfiguration. Cela implique que cette paire respecte au moins une des trois conditions (*condition 1, condition 2, condition 3*) d'interruptions de flux.

— **Cas 1 : La *condition 1* est respectée** :
Cela signifie qu'un nœud convergent x appartenant à la paire de sous-arbres à liens disjoints $\left(ST_c^1, ST_f^1\right)$ a au moins un ancêtre y sur l'arbre mono-optique actuel T_c qui n'appartient pas à l'arbre mono-optique final T_f et y est un descendant du nœud divergent n sur T_c. De plus, y a un descendant n_y sur T_c qui n'est pas un ancêtre du nœud convergent x et qui n'est pas un descendant de n sur T_f. Le respect de la *condition 1* implique que le nœud y est un nœud de branchement (c'est-à-dire un nœud dont émane plusieurs liens) appartenant à ST_c^1 et ce nœud est différent de la racine et des nœuds feuilles de $\left(ST_c^1, ST_f^1\right)$. De plus, cela implique que le nœud n_y n'appartient pas à ST_c^1. Soit une paire de sous-arbres à liens partagés que nous notons $\left(ST_c^2, ST_f^2\right)$. Soient r le nœud racine de $\left(ST_c^2, ST_f^2\right)$ et m le nœud convergent à l'origine de la sélection de $\left(ST_c^2, ST_f^2\right)$ comme mentionné par l'Algorithme 4.2. Soit la paire de chemins (Seg_c, Seg_f), avec Seg_c le chemin reliant le nœud r au nœud m sur ST_c^2 et Seg_f le chemin reliant le nœud r au nœud m sur ST_f^2. Par construction, Seg_c appartient exclusivement à l'arbre mono-optique actuel T_c et Seg_f appartient exclusivement à l'arbre mono-optique final T_f.

Supposons que la reconfiguration de la paire de sous-arbres à liens partagés $\left(ST_c^2, ST_f^2\right)$ à une étape du processus de reconfiguration de la paire d'arbres (T_c, T_f) permet de rendre la paire de sous-arbres à liens disjoints $\left(ST_c^1, ST_f^1\right)$ éligible à une reconfiguration sans interruption de flux. Cela implique que la

condition 1 n'est plus respectée pour la paire de sous-arbres à liens disjoints $\left(ST_c^1, ST_f^1\right)$. Autrement dit, après la reconfiguration de $\left(ST_c^2, ST_f^2\right)$, le nœud y n'est plus un nœud de branchement. En clair, le lien (ou canal) reliant le nœud y au nœud n_y sur l'arbre mono-optique actuel T_c a été supprimé. Pour que le nœud y ne soit plus un nœud de branchement sur l'arbre mono-optique actuel T_c il faut alors que le canal (ou lien) reliant le nœud y au nœud n_y soit un canal appartenant à Seg_c. Il s'ensuit que le nœud y est soit :

(i) Un nœud intermédiaire de Seg_c :
Dans ce cas, le canal amont du nœud y sur Seg_c est partagé avec ST_c^1. Cependant, la reconfiguration de $\left(ST_c^2, ST_f^2\right)$ va supprimer ce canal donc il ne peut plus appartenir à ST_c^1. Il s'ensuit que la paire $\left(ST_c^1, ST_f^1\right)$ n'existe plus. Ce qui est contradictoire, car nous supposons que la reconfiguration de $\left(ST_c^2, ST_f^2\right)$ à une étape du processus de reconfiguration de la paire d'arbres (T_c, T_f) permet de rendre la paire de sous-arbres à liens disjoints $\left(ST_c^1, ST_f^1\right)$ éligible à une reconfiguration sans interruption de flux.

(ii) L'extrémité initiale de Seg_c :
Sur un arbre, chaque nœud n'a qu'un seul canal amont. Donc le canal amont du nœud y sur ST_c^2 est identique au canal amont du nœud y sur ST_c^1 car ST_c^2 et ST_c^1 appartiennent à l'arbre mono-optique actuel T_c. Or Seg_c appartient exclusivement à l'arbre mono-optique actuel T_c. Il s'ensuit que le canal amont du nœud y (c'est-à-dire l'extrémité initiale de Seg_c) sur ST_c^2 est partagé avec l'arbre mono-optique final T_f. Ce qui signifie que le canal amont du nœud y sur ST_c^1 est aussi partagé avec l'arbre mono-optique final T_f. Ce qui est une contradiction, car par définition ST_c^1 est le sous-arbre actuel d'une paire de sous-arbres à liens disjoints, donc il ne partage aucun lien avec l'arbre mono-optique final T_f.

Dans (i) et (ii) on arrive à une contradiction.

— **Cas 2 : La *condition 2* est respectée** :
Cela signifie qu'un nœud convergent x appartenant à la paire de sous-arbres à liens disjoints $\left(ST_c^1, ST_f^1\right)$ a au moins un ancêtre y sur l'arbre mono-optique actuel T_c qui est un descendant du nœud divergent n sur T_c et le nœud y appartient à l'arbre mono-optique final T_f mais le nœud y n'est pas un descendant de n sur T_f. Cela équivaut à dire que $y \in ST_c^1$ et $y \notin ST_f^1$. Rappelons que n est la racine de

la paire de sous-arbres à liens disjoints $\left(ST_c^1, ST_f^1\right)$.

Supposons que la reconfiguration de la paire de sous-arbres à liens partagés $\left(ST_c^2, ST_f^2\right)$ à une étape du processus de reconfiguration de la paire d'arbres (T_c, T_f) permet de rendre la paire de sous-arbres à liens disjoints $\left(ST_c^1, ST_f^1\right)$ éligible à une reconfiguration sans interruption de flux. On a donc maintenant $y \in ST_c^1$ et $y \in ST_f^1$. Ce qui implique que le nœud y devient un descendant de n sur T_f. Ce qui est une contradiction, car la reconfiguration d'une paire de sous-arbres en l'occurrence la paire de sous-arbres à liens partagés $\left(ST_c^2, ST_f^2\right)$ ne modifie jamais l'arbre final T_f.

— **Cas 3 : La *condition 3* est respectée** :
Cela signifie qu'un nœud convergent x appartenant à la paire de sous-arbres à liens disjoints $\left(ST_c^1, ST_f^1\right)$ a un nœud ancêtre y sur T_f tel que le nœud y se situe sur le chemin reliant le nœud divergent n au nœud x et le nœud y n'est pas un ancêtre du nœud x se situant sur le chemin qui relie le nœud n au nœud x sur T_c. Cela équivaut à dire que $y \notin ST_c^1$ et $y \in ST_f^1$. Soit Seg_f la partie de ST_f^2 ($ST_f^2 \subseteq T_f$) appartenant exclusivement à l'arbre mono-optique final T_f et Seg_c la partie de ST_c^2 ($ST_c^2 \subseteq T_c$) appartenant exclusivement à l'arbre mono-optique actuel T_c.

Supposons que la reconfiguration de la paire de sous-arbres à liens partagés $\left(ST_c^2, ST_f^2\right)$ à une étape du processus de reconfiguration de la paire d'arbres (T_c, T_f) permet de rendre la paire de sous-arbres à liens disjoints $\left(ST_c^1, ST_f^1\right)$ éligible à une reconfiguration sans interruption de flux. Ainsi, on a maintenant $y \in ST_c^1$ et $y \in ST_f^1$. Donc, le nœud y devient un descendant de n sur T_c. Cela n'est possible que si le chemin reliant n à y sur T_f est inclut dans Seg_f. En effet, la reconfiguration de la paire de sous-arbres à liens partagés $\left(ST_c^2, ST_f^2\right)$ à une étape du processus de reconfiguration de la paire d'arbres (T_c, T_f) remplace Seg_c par Seg_f dans T_c. Dans ce cas, le chemin reliant n à y sur T_f est identique au chemin reliant n à y sur T_c. Autrement dit, le chemin reliant n à y sur ST_f^1 ($ST_f^1 \subseteq T_f$) et le chemin reliant n à y sur ST_c^1 ($ST_c^1 \subseteq T_c$) sont identiques. Ce qui est contradictoire, car $\left(ST_c^1, ST_f^1\right)$ est une paire de sous-arbres à liens disjoints.

Conclusion : Dans les trois cas de figures précédemment traités, on arrive à une contradiction. Par conséquent, on conclut qu'aucune paire de sous-arbres à liens disjoints non reconfigurable sans interruption de flux à une étape du processus ne peut devenir reconfigurable sans interruption de flux après la reconfiguration d'une paire de

sous-arbres à liens partagés. Autrement dit, le théorème d'immuabilité (voir Théorème 4.3.2.1) est prouvé.

4.3.2.2 Présentation de la méthode SbSRA

Nous rappelons que dans ce chapitre, il est question de trouver la séquence de configurations qui permet de migrer le flux optique d'un arbre mono-optique initial vers un arbre mono-optique final sans interruption de flux tout en utilisant le plus petit nombre de fois des longueurs d'onde additionnelles. Notons qu'à une étape donnée du processus de reconfiguration, une nouvelle configuration est obtenue par l'un des moyens suivants :

(i) En ajoutant à la configuration actuelle un ensemble de canaux optiques (y compris des semi-canaux optiques) qui appartiennent exclusivement à l'arbre final afin de préétablir un nouveau sous-arbre.

(ii) En ajoutant à la configuration actuelle un ensemble de canaux optiques afin de basculer (ou migrer) le flux optique d'un sous-arbre actuel à un nouveau sous-arbre.

(iii) En supprimant un ensemble de canaux optiques (qui appartiennent exclusivement à un sous-arbre actuel) de la configuration actuelle.

Ainsi, les différents ensembles de canaux optiques (c'est-à-dire (i), (ii) et (iii)) sont nécessaires pour la constitution de la séquence de configurations. Par ailleurs, la méthode SbSRA est une méthode qui doit procéder à la reconfiguration d'une paire d'arbres mono-optiques par paire de sous-arbres (sous-arbre actuel, nouveau sous-arbre). Par conséquent, nous présentons les trois fonctions permettant de retourner l'ensemble des canaux optiques nécessaire pour trouver les configurations requises à la reconfiguration d'une paire de sous-arbres (ST_c, ST_f). Notons que ces fonctions utilisent les opérations de configurations spécifiées au chapitre précédent. Puis nous présentons l'algorithme SbSRA utilisant ces différentes fonctions pour retourner la séquence de configurations requise pour la reconfiguration de la paire d'arbres mono-optiques (T_c, T_f).

❖ **Ensemble de canaux pour préétablir**

La fonction Canaux_preetablir (voir Algorithme 4.3) prend en entrées la paire de sous-arbres et la longueur d'onde requise pour préétablir le nouveau sous-arbre ST_f.

Puis elle retourne l'ensemble des canaux nécessaire pour préétablir le nouveau sous-arbre ST_f. Cet ensemble de canaux est formé de l'ensemble de canaux devant être

Algorithme 4.3 : Canaux_preetablir

Entrées : ST_c, ST_f, λ // ST_c : Sous-arbre actuel ; ST_f :nouveau sous-arbre ; λ : longueur d'onde
Sorties : Ens_Canaux // Ensemble des canaux à créer pour pré-établir ST_f
1 $Int_noeuds \leftarrow$ Ensemble des nœuds intermédiaires de ST_f
2 $Canaux_ADD \leftarrow$ Ensemble des canaux à obtenir en cas d'application de l'opération d'ajout de commutation de longueur d'onde ADD sur les éléments de Int_noeuds à l'aide de λ
3 $Convg_noeuds \leftarrow$ Ensemble des nœuds convergents de la paire (ST_c, ST_f)
4 $Canaux_CONVG \leftarrow$ Ensemble des canaux à obtenir en cas d'application de l'opération de convergence CONVG sur les éléments de $Convg_noeuds$ à l'aide de λ
5 $Ens_Canaux \leftarrow Canaux_ADD \cup Canaux_CONVG$
6 **Retourner** Ens_Canaux

obtenus par application de l'opération d'ajout de commutation sur les nœuds intermédiaires (c'est-à-dire différent du nœud racine et des nœuds feuilles) du nouveau sous-arbre et de l'ensemble des canaux devant être obtenus par application de l'opération de convergence sur les nœuds convergents de la paire de sous-arbres (voir ligne 5).

➢ Analyse de la correction de l'algorithme

Le nouveau sous-arbre ST_f est composé d'un ensemble de nœuds intermédiaires, de nœuds terminaux (ou nœuds feuilles) qui sont des nœuds convergents et de sa racine. Cependant, la racine est un nœud de basculement donc la racine est concernée uniquement par l'opération de basculement. Ainsi, les lignes 2 et 4, permettent de constituer à la ligne 5 l'ensemble des canaux requis pour préétablir ST_f. Ce qui prouve la correction partielle de l'Algorithme 4.3 (: résultat i). Par ailleurs, pour un nouveau sous-arbre ST_f, l'ensemble de nœuds intermédiaires et l'ensemble de nœuds convergents étant finis, il est donc évident que l'Algorithme 4.3 se termine (: résultat ii). Des résultats i et ii, on déduit la correction totale de l'Algorithme 4.3.

➢ Analyse de la complexité de l'Algorithme

Lemme 4.3 *Le processus (voir Algorithme 4.3) de définition des canaux à préétablir a une complexité temporelle (dans le pire des cas) en* $O\left(|V_f|^2 + p.|V_c|\right)$*, où* $p = max\left\{|V_c|\,;|V_f|\right\}$.

Preuve Les hypothèses et conventions mentionnées dans la preuve du lemme 4.1 sont également valables ici. Ainsi, $Comp\,(ligne\ 1)$ est en $O\left(|V_f|\right)$. La ligne 2 (respectivement la ligne 4) implique de rechercher l'ensemble des successeurs et le

nœud parent sur $ST_f \subseteq T_f$ de chaque nœud appartenant à Int_noeuds (respectivement à $Convg_noeuds \subseteq V_c \bigcap V_f$). Donc, on a $Comp\,(ligne\ 2)$ est en $O\left(|V_f|^2\right)$ et $Comp\,(ligne\ 4)$ est en $O\left(q.|V_f|\right)$, où $q = |V_c \bigcap V_f|$. Aussi, $Comp\,(ligne\ 5)$ est en $O\left(q.|V_f|\right)$. Par ailleurs, $Comp\,(ligne\ 3)$ est en $O\left(p.|V_c|\right)$, où $p = max\left\{|V_c|\,;|V_f|\right\}$. On conclut que l'Algorithme 4.3 a une complexité temporelle qui est en $O\left(|V_f|^2 + p.|V_c|\right)$.

❖ Ensemble de canaux pour migrer le flux de ST_c à ST_f

La fonction Canaux_migrer (voir Algorithme 4.4) prend en entrées la paire de sous-arbres et les longueurs d'onde requise pour migrer le flux optique du sous-arbre actuel ST_c vers le nouveau sous-arbre ST_f . Puis elle retourne l'ensemble des canaux requis pour le basculement du flux (voir ligne 2). Autrement dit, la racine d'une paire de sous-arbres constitue le nœud de basculement utilisé pour la reconfiguration de cette paire de sous-arbres.

Algorithme 4.4 : Canaux_migrer

Entrées : $ST_c, ST_f, \lambda_0, \lambda_1$ `//` ST_c `: Sous-arbre actuel ;` ST_f `:Nouveau sous-arbre ;` λ_0, λ_1 `: Longueurs d'onde`
Sorties : $Canaux_BASC$ `// Ensemble des canaux à créer pour migrer le flux de` ST_c `vers` ST_f

1 $r \leftarrow$ Racine de la paire de sous-arbres (ST_c, ST_f)
2 $Canaux_BASC \leftarrow$ Ensemble des canaux à obtenir en cas d'application de l'opération de basculement BASC sur le nœud r à l'aide de λ_0,λ_1(λ_1 est identique à λ_0 s'il s'agit d'une paire à liens disjoints).
3 **Retourner** $Canaux_BASC$

➢ Analyse de la correction de l'algorithme

La racine commune r de la paire de sous-arbres (ST_c, ST_f) est par construction unique et non nulle. Ainsi, le nœud r a au moins un nœud enfant sur chacun de ses deux sous-arbres. Donc, la ligne 2 contient les canaux nécessaires pour le basculement du flux. D'où la correction partielle de l'Algorithme 4.4 (: résultat i). En outre, la racine r possède un nombre limité de nœuds enfants sur chacun des deux sous-arbres, il est donc évident que l'Algorithme 4.4 se termine (: résultat ii). Des résultats i et ii, on déduit la correction totale de l'Algorithme 4.4.

➢ Analyse de la complexité de l'algorithme

Lemme 4.4 *Le processus (voir Algorithme 4.4) de définition des canaux sur lesquels le flux doit être migré, a une complexité temporelle (dans le pire des cas) en* $O\left(|V_f|\right)$.

Preuve Les hypothèses et conventions mentionnées dans la preuve du lemme 4.1 sont également valable ici. La ligne 2 implique la recherche des successeurs du nœud r sur ST_f. Ainsi, $Comp\ (ligne\ 2)$ est en $O\left(\left|V_f\right|\right)$. On conclut que l'Algorithme 4.4 a une complexité temporelle qui est en $O\left(\left|V_f\right|\right)$.

❖ Ensemble de canaux pour supprimer le flux de ST_c

La fonction Canaux_supprimer (voir Algorithme 4.5) prend en entrées la paire de sous-arbres et la longueur d'onde requise pour supprimer le sous-arbre actuel ST_c. Puis elle retourne l'ensemble des canaux nécessaire pour supprimer le sous-arbre actuel.

Algorithme 4.5 : Supprimer_Canaux

Entrées : ST_c, ST_f, λ // ST_c : Sous-arbre actuel ; ST_f :nouveau sous-arbre ; λ_0 : longueur d'onde
Sorties : Ens_Canaux // Ensemble de canaux à créer pour supprimer ST_c

1 $Int_noeuds \leftarrow$ Ensemble des nœuds intermédiaires de ST_c
2 $Canaux_DEL \leftarrow$ Ensemble des canaux supprimés en cas d'application de l'opération de suppression de commutation de longueur d'onde DEL sur les éléments de Int_noeuds à l'aide de λ
3 $Convg_noeuds \leftarrow$ Ensemble des nœuds convergents de la paire (ST_c, ST_f)
4 $Canaux_NCONVG \leftarrow$ Ensemble des canaux supprimés en cas d'application de l'opération de non-convergence NCONVG sur les éléments de $Convg_noeuds$ à l'aide de λ
5 $Ens_Canaux \leftarrow Canaux_DEL \cup Canaux_NCONVG$
6 **Retourner** Ens_Canaux

➢ Analyse de la correction de l'algorithme

Le sous-arbre actuel ST_c est composé d'un ensemble de nœuds intermédiaires, de nœuds terminaux (ou nœuds feuilles) qui sont des nœuds convergents et de sa racine. Cependant, la racine est un nœud de basculement donc la racine est concernée uniquement par l'opération de basculement. Ainsi, les lignes 2 et 4 permettent de constituer à la ligne 5 l'ensemble des canaux requis pour supprimer ST_c. Ce qui prouve la correction partielle de l'Algorithme 4.5 (: résultat i). Par ailleurs, pour un sous-arbre actuel ST_c, l'ensemble de nœuds intermédiaires et l'ensemble de nœuds convergents étant finis, il est évident que l'Algorithme 4.5 se termine (: résultat ii). Des résultats i et ii, on déduit la correction totale de l'Algorithme 4.5.

➢ Analyse de la complexité de l'algorithme

Lemme 4.5 *Le processus (voir Algorithme 4.5) de définition des canaux à supprimer, a une complexité temporelle (dans le pire des cas) en* $O\left(|V_c|^2 + p.|V_c|\right)$ *où* $p = max\{|V_c|\ ;\left|V_f\right|\}$.

Preuve Les hypothèses et conventions mentionnées dans la preuve du lemme 4.1 sont également valables ici. Ainsi, $Comp\,(ligne\ 1)$ est en $O\,(|V_c|)$. La ligne 2 (respectivement la ligne 4) implique de rechercher l'ensemble des successeurs et le nœud parent sur $ST_c \subseteq T_c$ de chaque nœud appartenant à Int_noeuds (respectivement à $Convg_noeuds \subseteq V_c \bigcap V_f$). Donc, on a $Comp\,(ligne\ 2)$ est en $O\left(|V_c|^2\right)$ et $Comp\,(ligne\ 4)$ est en $O\,(q.\,|V_c|)$, où $q = \left|V_c \bigcap V_f\right|$. Aussi, $Comp\,(ligne\ 5)$ est en $O\,(q.\,|V_c|)$. Par ailleurs, $Comp\,(ligne\ 3)$ est en $O\,(p.\,|V_c|)$, où $p = max\left\{|V_c|\,;\left|V_f\right|\right\}$. On conclut que l'Algorithme 4.5 a une complexité temporelle qui est en $O\left(|V_c|^2 + p.\,|V_c|\right)$.

❖ Algorithme de reconfiguration SbSRA

Nous rappelons que le problème étudié dans ce chapitre consiste à trouver la séquence de configurations qui permet de faire migrer le flux optique de l'arbre mono-optique initial T_0 à l'arbre mono-optique final T_f sans interruption du flux tout en utilisant un faible nombre de fois des longueurs d'onde d'additionnelles. Pour résoudre ce problème, nous proposons l'algorithme de reconfiguration appélé SbSRA (ou Sub-tree by Sub-tree Reconfiguration Algorithm) qui procède par reconfiguration des paires de sous-arbres. Nous avons montré précédemment que l'ordre de reconfiguration approprié des paires de sous-arbres présentes dans une paire d'arbres mono-optiques est le suivant : reconfigurer les paires de sous-arbres à liens disjoints puis après reconfigurer les paires de sous-arbres à liens partagés. Ainsi, le principe de l'algorithme SbSRA est de sélectionner et reconfigurer successivement toutes les paires de sous-arbres à liens disjoints puis de sélectionner et reconfigurer successivement toutes les paires de sous-arbres à liens partagés. Notons que reconfigurer une paire de sous-arbres consiste à préétablir le nouveau sous-arbre, migrer le flux du sous-arbre actuel vers le nouveau sous-arbre et supprimer le sous-arbre actuel.

L'algorithme SbSRA (voir Algorithme 4.6) prend en entrées une paire d'arbres mono-optiques (T_0, T_f) et retourne la séquence de configurations SC résolvant le problème étudié. L'algorithme SbSRA utilise trois phases afin de trouver la séquence de configurations permettant de résoudre le problème étudié : une phase d'initialisation, une phase de recherche des configurations permettant de reconfigurer les paires de sous-arbres à liens disjoints et une phase de recherche des configurations permettant de reconfigurer les paires de sous-arbres à liens partagés. Il est important de rappeler qu'une configuration est définie comme un ensemble composé de canaux optiques et éventuellement de semi-canaux optiques. Par souci de concision, nous in-

troduisons des notations dans l'algorithme SbSRA. Soient T_1, et T_2 deux arbres mono-optiques. On désigne par $Long\,(T_1, T_2)$ une primitive qui donne la longueur d'onde requise par la paire d'arbres mono-optiques (T_1, T_2) et $Long_add\,(T_1, T_2)$ désigne une primitive qui donne une longueur d'onde additionnelle disponible pour la paire d'arbres mono-optiques (T_1, T_2).

La phase d'initialisation de SbSRA (voir lignes 1 et 2 de l'Algorithme 4.6) consiste à définir l'arbre mono-optique initial T_0 comme premier élément de la séquence de configurations SC. L'arbre mono-optique actuel est aussi initialisé avec l'arbre mono-optique initial T_0.

Après la phase d'initialisation, SbSRA exécute la phase de recherche des configurations qui concernent les paires de sous-arbres à liens disjoints (voir lignes 3 à 23 de l'Algorithme 4.6). Cette phase consiste à :

(i) Sélectionner une paire de sous-arbres (sous-arbre actuel, nouveau sous-arbre) à liens disjoints dans la paire d'arbres mono-optiques.

(ii) Rechercher la configuration qui concerne le préétablissement du nouveau sous-arbre de cette paire de sous-arbres. Cette configuration est obtenue en ajoutant l'ensemble des canaux retourné par la fonction Canaux_preetablir à la configuration actuelle. La nouvelle configuration ainsi obtenue est ajoutée à la séquence de configurations.

(iii) Rechercher la configuration qui concerne la migration du flux sur le nouveau sous-arbre de cette paire de sous-arbres. Cette configuration est obtenue en ajoutant l'ensemble des canaux retourné par la fonction Canaux_migrer à la configuration actuelle. La nouvelle configuration ainsi obtenue est ajoutée à la séquence de configurations.

(iv) Rechercher la configuration qui concerne la suppression du sous-arbre actuel de cette paire de sous-arbres. Cette configuration est obtenue en supprimant l'ensemble des canaux retourné par la fonction Canaux_supprimer de la configuration actuelle. La nouvelle configuration ainsi obtenue est ajoutée à la séquence de configurations. Les étapes (i) à (iv) sont ainsi répétées jusqu'à ce qu'il n'y ait plus de paires de sous-arbres à liens disjoints. On note que les étapes (i) à (iv) n'utilisent pas de longueur d'onde additionnelle.

La dernière phase exécutée par SbSRA est la phase de recherche des configurations qui concernent les paires de sous-arbres à liens partagés (voir lignes 21 à 43 de

Algorithme 4.6 : SbSRA

Entrées : T_0, T_f, // T_0 : Arbre mono-optique initial, T_f : Arbre mono-optique final
Sorties : SC // Séquence (ou liste) de configurations

1 $T_c \leftarrow T_0$; $Curr_conf \leftarrow T_0$; $SC \leftarrow [Curr_conf]$ // $Curr_conf$: Configuration actuelle
2 $Long_{T0,T_f} \leftarrow Long\,(T_0, T_f)$ // Longueur d'onde utilisée par la paire d'arbres mono-optiques
3 $Div_{T_c,T_f} \leftarrow$ Liste des nœuds divergents de la paire d'arbres (T_c, T_f)
4 **Tant que** $Div_{T_c,T_f} \neq null$ **faire**
5 $n \leftarrow$ Premier élément de Div_{T_c,T_f}
6 $(ST_c, ST_f) \leftarrow Select_PSLD\,(T_c, T_f, n)$ // Selectionner une paire de sous-arbres à liens disjoints
7 **Si** $ST_c \neq null$ ***Et*** $ST_f \neq null$ **alors**
8 $Ens_Canaux \leftarrow Canaux_preetablir\left(ST_c, ST_f, Long_{T0,T_f}\right)$
9 $Curr_conf \leftarrow Curr_conf \cup Ens_canaux$ // Créer la Config. pour préétablir ST_f
10 $SC.enqueue\,(Curr_conf)$ // Ajouter Config. pour préétablir ST_f
11 $Ens_Canaux \leftarrow Canaux_migrer\left(ST_c, ST_f, Long_{T0,T_f}, Long_{T0,T_f}\right)$
12 $Curr_conf \leftarrow Curr_conf \cup Ens_canaux$ // Créer la Config. du basculement
13 $SC.enqueue\,(Curr_conf)$ // Ajouter la config. requise pour basculement
14 $Ens_Canaux \leftarrow Canaux_supprimer\left(ST_c, ST_f, Long_{T0,T_f}\right)$
15 $Curr_conf \leftarrow Curr_conf \setminus Ens_canaux$ // //Créer la config. pour supprimer STc
16 $SC.enqueue\,(Curr_conf)$ // Ajouter la Config. pour supprimer ST_c
17 $T_c \leftarrow Curr_conf$
18 $Div_{T_c,T_f} \leftarrow$ Liste des nœuds divergents de la paire d'arbres (T_c, T_f)
19 **Sinon**
20 $Div_{T_c,T_f} \leftarrow$ Liste des nœuds divergents de la paire d'arbres (T_c, T_f) hormis n
21 **Fin Si**
22 **Fin Tant que**
23 $CG_{T_c,T_f} \leftarrow$ Liste des nœuds convergents de la paire d'arbres (T_c, T_f)
24 **Tant que** $CG_{T_c,T_f} \neq null$ **faire**
25 $m \leftarrow$ Premier élément de CG_{T_c,T_f}
26 $(ST_c, ST_f) \leftarrow Select_PSLP\,(T_c, T_f, m)$ // Selectionner une paire de sous-arbres à liens partagés
27 **Si** $ST_c \neq null$ ***Et*** $ST_f \neq null$ **alors**
28 $Long_add_{T_c,T_f} \leftarrow Long_add\,(T_c, T_f)$ // Longueur d'onde additionnelle de la paire (T_c, T_f)
29 $Ens_Canaux \leftarrow Canaux_preetablir\left(ST_c, ST_f, Long_add_{T_c,T_f}\right)$
30 $Curr_conf \leftarrow Curr_conf \cup Ens_canaux$
31 $SC.enqueue\,(Curr_conf)$
32 $Ens_Canaux \leftarrow Canaux_migrer\left(ST_c, ST_f, Long_{T_0,T_f}, Long_add_{T_c,T_f}\right)$
33 $Curr_conf \leftarrow Curr_conf \cup Ens_canaux$
34 $SC.enqueue\,(Curr_conf)$
35 $Ens_Canaux \leftarrow Canaux_supprimer\left(ST_c, ST_f, Long_{T0,T_f}\right)$
36 $Curr_conf \leftarrow Curr_conf \cup Ens_canaux$
37 $SC.enqueue\,(Curr_conf)$
38 Ré-exécutez à les instructions de la ligne 26 à la ligne 35 en remplaçant $Long_add_{T_c,T_f}$ par $Long_{T0,T_f}$ aux lignes 27 et 30 et en remplaçant $Long_{T0,T_f}$ par $Long_add_{T_c,T_f}$ à la ligne 33
39 $T_c \leftarrow Curr_conf$
40 **Fin Si**
41 $CG_{T_c,T_f} \leftarrow$ Liste des nœuds convergents de la paire d'arbres (T_c, T_f)
42 **Fin Tant que**
43 **Retourner** SC

l'Algorithme 4.6). Cette phase consiste à appliquer en deux tours les étapes (i) à (iv) aux paires de sous-arbres à liens partagés. Dans le premier tour, les étapes (ii) et (iii) utilisent une longueur d'onde additionnelle quelques fois. Dans le second tour, l'étape (iv) utilise également une longueur d'onde additionnelle quelques fois.

➢ Analyse de la correction de l'algorithme

Nous rappelons que l'Algorithme 4.6 prend en entrées une paire d'arbres (T_0, T_f) puis retourne une séquence de configurations qui n'interrompt pas le flux et qui utilise un faible nombre de fois des longueurs d'onde additionnelles. Nous montrons la correction totale de cet algorithme dans la suite de ce paragraphe. Les configurations créées pour la reconfiguration des paires de sous-arbres à liens disjoints n'utilisent pas de longueur d'onde additionnelle. L'Algorithme 4.6 utilise à certaines étapes une longueur d'onde additionnelle que pour les configurations nécessaires à la reconfiguration des paires de sous-arbres à liens partagés. Ainsi, l'Algorithme 4.6 utilise un faible nombre de fois des longueurs d'onde additionnelles (: résultat i). Les paires de sous-arbres à liens disjoints et les paires de sous-arbres à liens partagés sont sélectionnées (voir Algorithmes 4.1 & 4.2) de sorte que les nœuds destinations ne sont pas privés du flux lors de la reconfiguration de ces paires de sous-arbres. Ainsi, l'Algorithme 4.6 va retourner une séquence de configurations qui n'interrompt pas le flux (: résultat ii). Les résultats i et ii prouvent la correction partielle de l'Algorithme 4.6 (: résultat iii). La boucle «tant que» (voir ligne 4 à ligne 23) de recherche des configurations qui concernent les paires de sous-arbres à liens disjoints se termine (: résultat iv). En effet, à chaque itération de cette boucle un nœud divergent est traité et l'arbre actuel (ou courant) est mis à jour. Autrement dit, un nœud divergent traité à l'itération i n'est plus un nœud divergent à l'itération $i+1$. Ainsi l'ensemble des nœuds divergents se réduit après chaque itération. En clair, il existe une itération k où cette boucle se termine. De même, la boucle «tant que» (voir ligne 22 à 43) de recherche des configurations qui concernent les paires de sous-arbres à liens partagés se termine (: résultat v). En effet, à chaque itération de cette boucle, un nœud convergent est traité et l'arbre courant est mis à jour. Autrement dit, un nœud convergent traité à l'itération j n'est plus un nœud convergent à l'itération $j+1$. Ainsi, l'ensemble des nœuds convergents se réduit après chaque itération. En clair, il existe une itération p où cette boucle se termine. On déduit des résultats iv et v que l'Algorithme 4.6 se termine (: résultat vi). Les résultats iii et vi prouvent la correction totale de l'Algorithme 4.6.

➢ Analyse de la complexité de l'algorithme

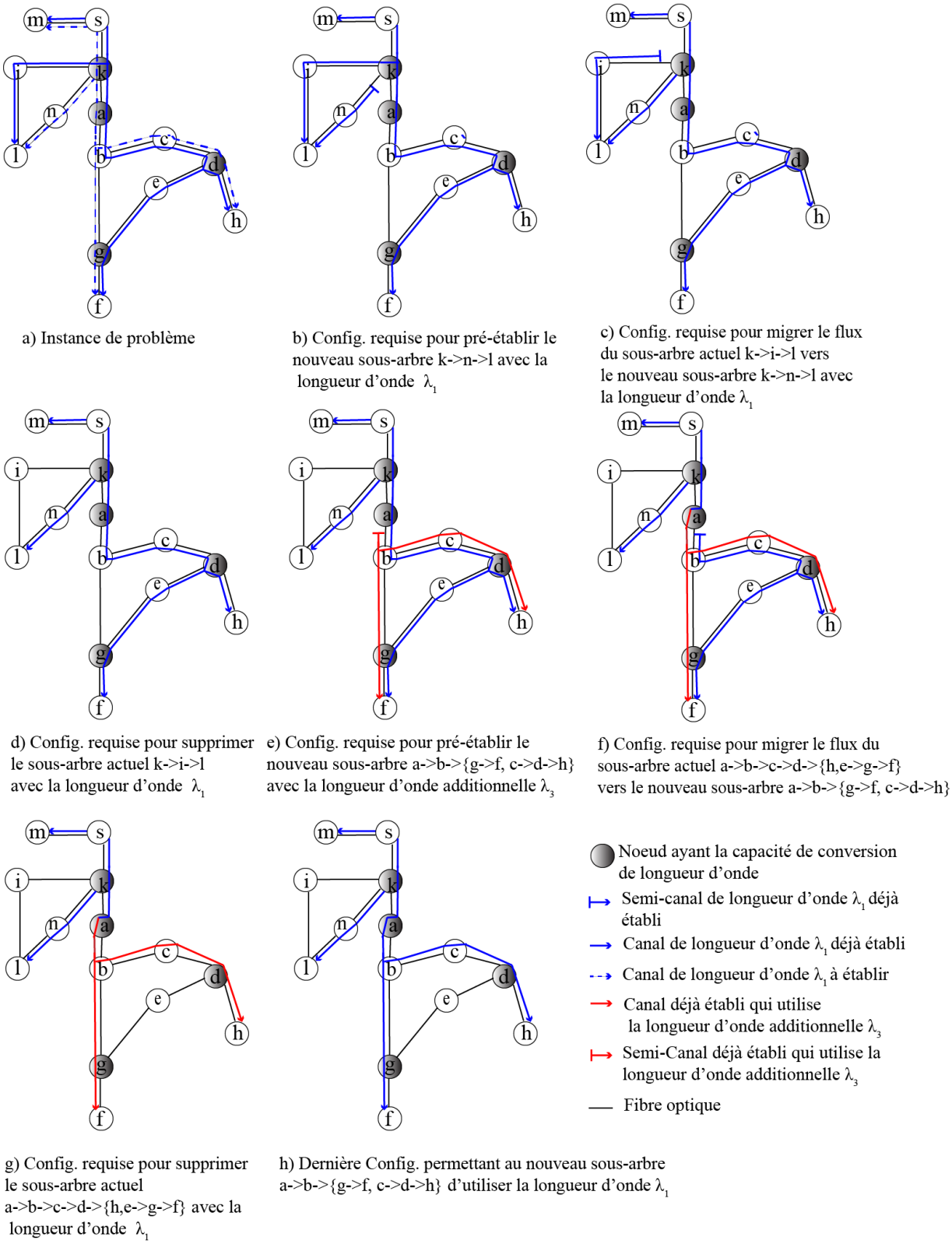
Lemme 4.6 *Le processus (voir Algorithme 4.6) de recherche d'une séquence de configurations pour une paire d'arbres mono-optiques à reconfigurer, a une complexité temporelle (dans le pire des cas) en* $O\left(q\left(|V_c|^4 + p.|V_c|^3 + |V_f|^3.|V_c|\right)\right)$ *où* $p = max\{|V_c|\,;|V_f|\}$ *et* $q = |V_c \bigcap V_f|$.

Preuve Les hypothèses et conventions mentionnées dans la preuve du lemme 4.1 sont également valables ici. Ainsi, $Comp\,(ligne\ 1-3)$ est en $O\left(|V_c|^2.|V_f|\right)$ (i) car $Comp\,(ligne\ 3)$ est en $O\left(|V_c|^2.|V_f|\right)$. Des lemmes 4.3, 4.4 et 4.5, on déduit respectivement que $Comp\,(ligne\ 8)$ est en $O\left(|V_f|^2 + p.|V_c|\right)$, $Comp\,(ligne\ 11)$ est en $O\left(|V_f|\right)$ et $Comp\,(ligne\ 14)$ est en $O\left(|V_c|^2 + p.|V_c|\right)$, où $p = max\{|V_c|\,;|V_f|\}$. Soit $V\,(Curr_conf)$, l'ensemble des nœuds appartenant à la configuration courante $Curr_conf$. Par construction, $V\,(Curr_conf) \subseteq V_c \bigcup V_f$. Par conséquent, on a $|V\,(Curr_conf)| \leq |V_c| + |V_f|$. Il s'ensuit que $Comp\,(ligne\ 9)$ et $Comp\,(ligne\ 12)$ sont en $O\left(|V_f|^2 + |V_f|.|V_c|\right)$. Par ailleurs, $Comp\,(ligne\ 15)$ est en $O\left(|V_c|^2 + |V_f|.|V_c|\right)$. Il résulte de tout ce qui précède que $Comp\,(ligne\ 7-18)$ est en $O\left(|V_c|^2 + |V_f|^2\right)$. Du lemme 4.1, on déduit que $Comp\,(ligne\ 6)$ est en $O\left(p.|V_c|^3 + p.|V_f|.|V_c|\right)$. Par conséquent, $Comp\,(ligne\ 6-18)$ est en $O\left(p.|V_c|^3 + p.|V_f|.|V_c|\right)$. Par construction, on a $Div_{T_c,T_f} \subseteq V_c \bigcap V_f$. Donc, $|Div_{T_c,T_f}| \leq q$, où $q = |V_c \bigcap V_f|$. Il s'ensuit que $Comp\,(ligne\ 4-20)$ est en $O\left(p.q\left(|V_c|^3 + |V_f|.|V_c|\right)\right)$ (ii). Du lemme 4.2, on obtient que $Comp\,(ligne\ 24)$ est en $O\left(|V_c|^4 + |V_f|^3.|V_c|\right)$. Par analogie à $Comp\,(ligne\ 6-18)$ qui est égale $Comp\,(ligne\ 6)$, on déduit que $Comp\,(ligne\ 24-38)$ est égale à $Comp\,(ligne\ 24)$, c'est-à-dire $O\left(|V_c|^4 + |V_f|^3.|V_c|\right)$. Il s'ensuit que $Comp\,(ligne\ 22-40)$ est en $O\left(q\left(|V_c|^4 + |V_f|^3.|V_c|\right)\right)$ (iii). De (i), (ii) et (iii), on conclut que l'Algorithme 4.6 a une complexité temporelle en $O\left(q\left(|V_c|^4 + p.|V_c|^3 + |V_f|^3.|V_c|\right)\right)$.

➢ Illustration de l'algorithme

La Figure 4.3.a présente une instance du problème de reconfiguration d'une paire d'arbres mono-optiques (arbre mono-optique initial, arbre mono-optique final) concernant la connexion multicast $< s, \{l, m, f, h\} >$. Sur cette figure, l'ensemble des traits continus et de couleur bleue forme l'ensemble des liens de l'arbre mono-optique initial T_0 enraciné au nœud s. L'ensemble des traits discontinus et de couleur bleue forme

l'ensemble des liens de l'arbre mono-optique final T_f enraciné au nœud s. Par ailleurs, au début du processus, l'arbre mono-optique actuel T_c équivaut à l'arbre mono-optique initial T_0. Cet arbre actuel est pris comme configuration actuelle. Le premier nœud divergent descendant de la racine s de la paire d'arbres mono-optiques (arbre mono-optique actuel, arbre mono-optique final) est le nœud k. La paire de sous-arbres à liens disjoints enracinée au nœud k $(k \rightarrow i \rightarrow l, k \rightarrow n \rightarrow l)$ est retournée par la fonction $Select_PSLD$. Le sous-arbre $k \rightarrow i \rightarrow l$ désigne le sous-arbre actuel et le sous-arbre $k \rightarrow n \rightarrow l$ désigne le nouveau sous-arbre. Les Figures 4.3.b à 4.3.d illustrent les résultats de la phase de recherche des configurations qui concernent la paire de sous-arbres à liens disjoints. En clair, la configuration requise pour le préétablissement du nouveau sous-arbre, la configuration pour la migration (ou basculement) du flux sur le nouveau sous-arbre et la configuration requise à la suppression du sous-arbre actuel sont illustrées par l'ensemble des traits continus de couleur bleue respectivement sur la Figure 4.3.b, sur la Figure 4.3.c et sur la Figure 4.3.d. A la fin de cette phase, l'arbre mono-optique actuel (voir Figure 4.3.d) contient au moins un nœud convergent. Le seul nœud convergent ici est le nœud g. Ainsi, la paire de sous-arbres à liens partagés enracinée au nœud a $(a \rightarrow b \rightarrow c \rightarrow d \rightarrow \{h, e \rightarrow f \rightarrow g\}, a \rightarrow b \rightarrow \{g \rightarrow f, c \rightarrow d \rightarrow h\})$ est retournée par la fonction $Select_PSLP$. Le sous-arbre $a \rightarrow b \rightarrow c \rightarrow d \rightarrow \{h, e \rightarrow f \rightarrow g\}$ désigne le sous-arbre actuel et le sous-arbre $a \rightarrow b \rightarrow \{g \rightarrow f, c \rightarrow d \rightarrow h\}$ désigne le nouveau sous-arbre. Les Figures 4.3.e à 4.3.g illustrent les résultats du premier tour de la phase de recherche des configurations qui concernent la paire de sous-arbres à liens partagés. En clair, la configuration requise pour le préétablissement du nouveau sous-arbre à l'aide d'une longueur d'onde additionnelle, la configuration requise à la migration du flux sur le nouveau sous-arbre à l'aide d'une nouvelle longueur d'onde additionnelle et la configuration requise à la suppression du sous-arbre actuel sont illustrées par l'ensemble des traits continus de couleur bleue et/ou rouge, respectivement sur les Figures 4.3.e, 4.3.f et 4.3.g. Comme on peut le voir, le nouveau sous-arbre utilise une longueur d'onde additionnelle au lieu de la longueur d'onde de la paire d'arbres (arbre mono-optique initial, arbre mono-optique final) à reconfigurer. C'est pourquoi un second tour vise à permettre au nouveau sous-arbre d'utiliser la même longueur d'onde que la paire d'arbres à reconfigurer. La Figure 4.3.h présente la configuration obtenue à la fin de ce second tour. Cette dernière configuration est bel et bien la configuration finale, car elle est le résultat de l'établissement des canaux optiques de l'arbre mono-optique final. Comme on peut le voir sur les différentes figures,



a) Instance de problème

b) Config. requise pour pré-établir le nouveau sous-arbre k->n->l avec la longueur d'onde λ_1

c) Config. requise pour migrer le flux du sous-arbre actuel k->i->l vers le nouveau sous-arbre k->n->l avec la longueur d'onde λ_1

d) Config. requise pour supprimer le sous-arbre actuel k->i->l avec la longueur d'onde λ_1

e) Config. requise pour pré-établir le nouveau sous-arbre a->b->{g->f, c->d->h} avec la longueur d'onde additionnelle λ_3

f) Config. requise pour migrer le flux du sous-arbre actuel a->b->c->d->{h,e->g->f} vers le nouveau sous-arbre a->b->{g->f, c->d->h}

g) Config. requise pour supprimer le sous-arbre actuel a->b->c->d->{h,e->g->f} avec la longueur d'onde λ_1

h) Dernière Config. permettant au nouveau sous-arbre a->b->{g->f, c->d->h} d'utiliser la longueur d'onde λ_1

FIGURE 4.3 – Illustration d'une séquence de configurations avec SbSRA

chacune des configurations contient un chemin entièrement configuré entre la source s et chaque nœud destination (à savoir les nœuds l, m, f et h). Ce qui signifie que la séquence de configurations n'interrompt pas le flux vers les nœuds destinations.

4.3.3 Méthode de parallèlisation de la reconfiguration Para_SbSRA

L'algorithme SbSRA permet de reconfigurer une paire d'arbres mono-optiques sans interruption de flux. SbSRA (voir Algorithme 4.6) requiert trois configurations pour reconfigurer une paire de sous-arbres à liens disjoints et six configurations sont nécessaires pour reconfigurer une paire de sous-arbres à liens partagés. Par ailleurs, SbSRA reconfigure une seule paire de sous-arbres à la fois. Par conséquent, plus il y a de paires de sous-arbres à reconfigurer dans une paire d'arbres mono-optiques plus la séquence de configurations retournée par SbSRA est longue. Pour éviter une longue séquence de configurations, il pourrait-être intéressant de reconfigurer en parallèle certaines paires de sous-arbres. Dans la section 4.3.3.1, on montre que les paires de sous-arbres à liens disjoints sont reconfigurables en parallèle. Puis dans la section 4.3.3.2, après avoir justifié que les paires de sous-arbres à liens partagés sont reconfigurables en parallèle, on présente un algorithme permettant de réduire la longueur d'une séquence de configurations.

4.3.3.1 Justification de la parallélisation de la reconfiguration de paires de sous-arbres

Soient $\left(ST_c^1, ST_f^1\right)$ et $\left(ST_c^2, ST_f^2\right)$ deux paires de sous-arbres à reconfigurer, où $ST_c^1 \subseteq T_c \setminus ST_c^2$, $ST_c^2 \subseteq T_c \setminus ST_c^1$, $ST_f^1 \subseteq T_f \setminus ST_f^2$ et $ST_f^2 \subseteq T_f \setminus ST_f^1$, T_c est l'arbre mono-optique actuel et T_f est l'arbre mono-optique final.

Définition 4.1 La paire de sous-arbres $\left(ST_c^1, ST_f^1\right)$ et la paire de sous-arbres $\left(ST_c^2, ST_f^2\right)$ sont dites indépendantes signifie que les sous-arbres ST_c^1 et ST_f^2 ne partagent aucun lien ainsi que les sous-arbres ST_c^2 et ST_f^1 ne partagent aucun lien.

Théorème 4.2: Théorème d'indépendance

Si la paire de sous-arbres $\left(ST_c^1, ST_f^1\right)$ et la paire de sous-arbres $\left(ST_c^2, ST_f^2\right)$ sont toutes les deux des paires de sous-arbres à liens disjoints alors les sous-arbres ST_c^1 et ST_f^2 ne partagent aucun lien ainsi que les sous-arbres ST_c^2 et ST_f^1 ne partagent aucun lien. Autrement dit, si deux paires sont des paires de sous-arbres à liens disjoints, alors elles sont des paires de sous-arbres indépendantes.

Preuve (Par implication) Nous supposons que $\left(ST_c^1, ST_f^1\right)$ et $\left(ST_c^2, ST_f^2\right)$ sont deux paires de sous-arbres appartenant à la catégorie des paires de sous-arbres à liens disjoints. Montrons que les sous-arbres ST_c^1 et ST_f^2 ne partagent aucun lien ainsi que les sous-arbres ST_c^2 et ST_f^1 ne partagent aucun lien.

La paire de sous-arbres $\left(ST_c^1, ST_f^1\right)$ est une paire de sous-arbres à liens disjoints. Cela implique que le sous-arbre ST_f^1 est à liens disjoints de T_c. Or on sait que $ST_c^2 \subseteq T_c$. Par conséquent, les sous-arbres ST_c^2 et ST_f^1 ne partagent aucun lien (i).

La paire de sous-arbres $\left(ST_c^2, ST_f^2\right)$ est une paire de sous-arbres à liens disjoints. Cela implique que le sous-arbre ST_f^2 est à liens disjoints de T_c. Or, on sait que $ST_c^1 \subseteq T_c$. Par conséquent, les sous-arbres ST_c^1 et ST_f^2 ne partagent aucun lien (ii).

De (i) et (ii), on déduit que le théorème d'indépendance est prouvé.

Corollaire 4.1: Du théorème d'indépendance

Si deux paires de sous-arbres sont toutes les deux des paires de sous-arbres à liens disjoints, alors ces deux paires de sous-arbres sont reconfigurables en parallèle sans utilisation de longueur d'onde additionnelle.

4.3.3.2 Algorithme de reconfiguration Para_SbSRA

La section 4.3.3.1 (voir Corollaire du théorème d'indépendance) nous apprend que les paires de sous-arbres à liens disjoints sont des paires de sous-arbres dites indépendantes qui sont donc reconfigurables en parallèle sans longueur d'onde additionnelle. Par ailleurs, les paires de sous-arbres à liens partagés utilisent une longueur d'onde additionnelle pour leur reconfiguration. Par conséquent, même si elles ne sont pas toujours indépendantes, elles sont reconfigurables en parallèle sans interruption

de flux en utilisant une longueur d'onde additionnelle. En effet, l'utilisation de la longueur d'onde additionnelle permet de contourner le blocage dû au partage de liens (canaux) entre le nouveau sous-arbre d'une paire de sous-arbres à liens partagés donnée et le sous-arbre actuel d'une autre paire. Ainsi, l'idée de l'algorithme Para_SbSRA est donc de reconfigurer en parallèle les paires de sous-arbres à liens disjoints et ensuite de reconfigurer en parallèle les paires de sous-arbres à liens partagés. Notons que les notations utilisées pour la présentation de l'algorithme SbSRA (voir Algorithme 4.6) sont reprises pour présenter l'algorithme Para_SbSRA.

L'algorithme Para_SbSRA (voir Algorithme 4.7) se compose des trois mêmes phases que l'algorithme SbSRA à savoir la phase d'initialisation, la phase de recherche des configurations nécessaires à la reconfiguration des paires des sous-arbres à liens disjoints et la phase de recherche des configurations nécessaires à la reconfiguration des paires de sous-arbres à liens partagés. Le principe de la phase d'initialisation (voir lignes 1 et 2 de Algorithme 4.7) est identique au principe de la phase d'initialisation de l'algorithme SbSRA. Quant à la phase de recherche des configurations requises pour la reconfiguration des paires de sous-arbres à liens disjoints, le principe est différent de celui de l'algorithme SbSRA. L'algorithme SbSRA recherche chacune des trois configurations nécessaires à la reconfiguration d'une seule paire de sous-arbres à liens disjoints à la fois alors que l'algorithme Para_SbSRA recherche chacune des trois configurations nécessaires à toutes les paires de sous-arbres à liens disjoints à la fois (voir lignes 3 à 15 de Algorithme 4.7). Pour ce qui est de la phase de recherche des configurations nécessaires à la reconfiguration des paires de sous-arbres à liens partagés, le principe est aussi différent de celui de l'algorithme SbSRA, car l'algorithme SbSRA recherche chacune des six configurations nécessaires à la reconfiguration d'une seule paire de sous-arbres à la fois. Dans cette dernière phase (voir lignes 16 à 29 de Algorithme 4.7), contrairement à l'algorithme SbSRA, l'algorithme Para_SbSRA recherche chacune des six configurations nécessaires à toutes les paires de sous-arbres à liens partagés à la fois. Ainsi, l'algorithme Para_SbSRA permet d'avoir une séquence de configurations plus courte que celle retournée par l'algorithme SbSRA sans pour autant utiliser un nombre supérieur de fois des longueurs d'onde additionnelles.

➢ Analyse de la correction de l'algorithme

Nous rappelons que l'Algorithme 4.7 prend en entrées une paire d'arbres (T_0, T_f) puis retourne une plus courte séquence (par rapport à celle retournée par SbSRA) de configurations qui n'interrompt pas le flux et qui utilise un faible nombre de fois des

Algorithme 4.7 : Para_SbSRA

Entrées : T_0, T_f, // T_0 : Arbre mono-optique initial, T_f : Arbre mono-optique final
Sorties : SC // Séquence (ou liste) de configurations

1 $T_c \leftarrow T_0$; $Curr_conf \leftarrow T_0$; $SC \leftarrow [Curr_conf]$ // $Curr_conf$: Configuration actuelle
2 $Long_{T0,T_f} \leftarrow Long\,(T_0, T_f)$ // Longueur d'onde utilisée par la paire d'arbres mono-optiques
// $Ens_Paires_SA = \{(ST_c^1, ST_f^1), ..., ST_c^k, ST_f^k)\}$
3 $Ens_Paires_SA \leftarrow$ Ensemble des paires de sous-arbres à liens disjoints de la paire (T_c, T_f)
4 **Si** Ens_Paires_SA *est non vide* **alors**
// Reconfiguration en parallèle des paires à liens disjoints
5 $Ens_Canaux \leftarrow$ Union des ensembles obtenus par application en parallèle de $Canaux_preetablir\left(ST_c, ST_f, Long_{T0,T_f}\right)$ sur les paires (ST_c, ST_f) contenus dans Ens_Paires_SA
6 $Curr_conf \leftarrow Curr_conf \cup Ens_canaux$ // Config. pour préétablir les ST_f de Ens_Paires_SA
7 $SC.enqueue\,(Curr_conf)$ // Ajout de la Config. pour préétablir les ST_f de Ens_Paires_SA
8 $Ens_Canaux \leftarrow$ Union des ensembles obtenus par application en parallèle de $Canaux_migrer\left(ST_c, ST_f, Long_{T0,T_f}, Long_{T0,T_f}\right)$ sur les paires (ST_c, ST_f) contenus dans Ens_Paires_SA
9 $Curr_conf \leftarrow Curr_conf \cup Ens_canaux$ // Config. pour basculer sur les ST_f de Ens_Paires_SA
10 $SC.enqueue\,(Curr_conf)$ // Ajout de Config. pour basculer sur les ST_f de Ens_Paires_SA
11 $Ens_Canaux \leftarrow$ Union des ensembles obtenus par application en parallèle de $Canaux_supprimer\left(ST_c, ST_f, Long_{T0,T_f}\right)$ sur les paires (ST_c, ST_f) contenus dans Ens_Paires_SA
12 $Curr_conf \leftarrow Curr_conf \setminus Ens_canaux$ // Config. pour supprimer les ST_c de Ens_Paires_SA
13 $SC.enqueue\,(Curr_conf)$ // Ajout de la Config. pour supprimer les ST_c de Ens_Paires_SA
14 $T_c \leftarrow Curr_conf$ // T_c : Arbre mono-optique courant
15 **Fin Si**
// $Ens_Paires_SA = \{(ST_c^1, ST_f^1), ..., ST_c^i, ST_f^i)\}$
16 $Ens_Paires_SA \leftarrow$ Ensemble des paires de sous-arbres à liens partagés de la paire (T_c, T_f)
17 **Si** Ens_Paires_SA *est non vide* **alors**
// Reconfiguration en parallèle des paires à liens partagés
18 $Long_add_{T_c,T_f} \leftarrow Long_add\,(T_c, T_f)$ // Longueur d'onde additionnelle de la paire (T_c, T_f)
19 $Ens_Canaux \leftarrow$ Union des ensembles obtenus par application en parallèle de $Canaux_preetablir\left(ST_c, ST_f, Long_add_{T_c,T_f}\right)$ sur les paires (ST_c, ST_f) contenus dans Ens_Paires_SA
20 $Curr_conf \leftarrow Curr_conf \cup Ens_canaux$ // Config. pour préétablir les ST_f de Ens_Paires_SA
21 $SC.enqueue\,(Curr_conf)$ // Ajout de la Config. pour préétablir les ST_f de Ens_Paires_SA
22 $Ens_Canaux \leftarrow$ Union des ensembles obtenus par application en parallèle de $Canaux_migrer\left(ST_c, ST_f, Long_{T_0,T_f}, Long_add_{T_c,T_f}\right)$ sur les paires de sous-arbres (ST_c, ST_f) contenus dans Ens_Paires_SA
23 $Curr_conf \leftarrow Curr_conf \cup Ens_canaux$ // Config. pour basculer sur les ST_f de Ens_Paires_SA
24 $SC.enqueue\,(Curr_conf)$ // Ajout de Config. pour basculer sur les ST_f de Ens_Paires_SA
25 $Ens_Canaux \leftarrow$ Union des ensembles obtenus par application en parallèle de $Canaux_supprimer\left(ST_c, ST_f, Long_{T0,T_f}\right)$ sur les paires (ST_c, ST_f) contenus dans Ens_Paires_SA
26 $Curr_conf \leftarrow Curr_conf \setminus Ens_canaux$ // Config. pour supprimer les ST_c de Ens_Paires_SA
27 $SC.enqueue\,(Curr_conf)$ // Ajout config. pour supprimer les ST_c de Ens_Paires_SA
28 Ré-exécutez les instructions de la ligne 19 à la ligne 27 en remplaçant $Long_add_{T_c,T_f}$ par $Long_{T0,T_f}$ aux lignes 19 et 22 et en remplaçant $Long_{T0,T_f}$ par $Long_add_{T_c,T_f}$ à la ligne 25
29 **Fin Si**
30 **Retourner** SC

longueurs d'onde additionnelles. Nous montrons la correction totale de cet algorithme dans la suite de ce paragraphe. Les configurations créées pour la reconfiguration des paires de sous-arbres à liens disjoints n'utilisent pas de longueur d'onde additionnelle. L'Algorithme 4.7 utilise à certaines étapes une longueur d'onde additionnelle, c'est-à-dire lors de la reconfiguration des paires de sous-arbres à liens partagés. Ainsi, l'Algorithme 4.7 utilise un faible nombre de fois des longueurs d'onde additionnelle (résultat i). Les paires de sous-arbres à liens disjoints et les paires de sous-arbres à liens partagés sont sélectionnées (voir Algorithmes 4.1& 4.2) de sorte que les nœuds destinations ne sont pas privés du flux lors de leurs reconfigurations. Ainsi, l'Algorithme 4.7 va retourner une séquence de configurations qui n'interrompt pas le flux (résultat ii). Les configurations (créées à chaque étape) contribuent à reconfigurer en parallèle soient les paires de sous-arbres à liens disjoints soient les paires de sous-arbres à liens partagés. Ainsi, l'Algorithme 4.7 retourne une courte séquence de configurations (iii). Par conséquent, les résultats i, ii et iii prouvent la correction partielle de l'Algorithme 4.7 (résultat iv). L'instruction de la ligne 3 de l'Algorithme 4.7 se termine, car l'Algorithme 4.1 se termine. De même, l'instruction 16 de l'Algorithme 4.7 se termine, car l'Algorithme 4.2 se termine. Par ailleurs, les deux blocs d'instructions « Si ... Alors » (voir ligne 4 à ligne 15 et ligne 17 à ligne 29) se termine, car les Algorithmes 4.3, 4.4 et Algorithme 4.5 se terminent. Il en ressort que l'Algorithme 4.7 se termine (résultat v). Les résultats iv et v prouvent la correction totale de l'Algorithme 4.7.

➢ Analyse de la complexité de l'algorithme

Lemme 4.7 *Le processus (voir Algorithme 4.7) de recherche d'une séquence de configurations pour une paire d'arbres mono-optiques à reconfigurer, a une complexité temporelle (dans le pire des cas) en* $O\left(p.q\left(|V_c|^3 + |V_f|.|V_c|\right) + q.|V_f|^3.|V_c|\right)$*, où* $p = max\{|V_c|;|V_f|\}$ *et* $q = |V_c \bigcap V_f|$.

Preuve Les hypothèses et conventions mentionnées dans la preuve du lemme 4.1 sont également valables ici. Soit Div_{T_c,T_f}, l'ensemble des nœuds divergents de la paire d'arbres (T_c, T_f). On a $Div_{T_c,T_f} \subseteq V_c \bigcap V_f$. Autrement dit, il s'ensuit que $|Div_{T_c,T_f}| \leq q$, où $q = |V_c \bigcap V_f|$. Selon le lemme 4.1, la sélection d'une paire de sous-arbres à liens disjoints a une complexité temporelle en $O\left(p\left(|V_c|^3 + .|V_f|.|V_c|\right)\right)$. Ainsi, $Comp\,(ligne\ 3)$ est en $O\left(p.q\left(|V_c|^3 + .|V_f|.|V_c|\right)\right)$. Il s'ensuit que $Comp\,(ligne\ 1-3)$ est en $O\left(p.q\left(|V_c|^3 + .|V_f|.|V_c|\right)\right)$ (i), où $p = max\{|V_c|;|V_f|\}$. Des lemmes 4.3, 4.4 et 4.5, on déduit respectivement

que $Comp\,(ligne\ 5)$ est en $O\left(\left|V_f\right|^2 + p.\left|V_c\right|\right), Comp\,(ligne\ 8)$ est en $O\left(\left|V_f\right|\right)$ et $Comp\,(ligne\ 11)$ est en $O\left(\left|V_c\right|^2 + p.\left|V_c\right|\right)$, où $p = max\{|V_c|\,;\left|V_f\right|\}$. Soit $V\,(Curr_conf)$, l'ensemble des nœuds appartenant à la configuration courante $Curr_conf$. Par construction, $V\,(Curr_conf) \subseteq V_c \bigcup V_f$. Par conséquent, on a $|V\,(Curr_conf)| \leq |V_c| + \left|V_f\right|$. Il s'ensuit que $Comp\,(ligne\ 6)$ et $Comp\,(ligne\ 9)$ sont en $O\left(\left|V_f\right|^2 + \left|V_f\right|.\left|V_c\right|\right)$. De ce qui précède, on déduit que $Comp\,(ligne\ 5-14)$ est en $O\left(\left|V_f\right|^2 + |V_c|^2 + p.\left|V_c\right|\right)$. Par conséquent, $Comp\,(ligne\ 4-15)$ est en $O\left(\left|V_f\right|^2 + |V_c|^2 + p.\left|V_c\right|\right)$ (ii). Soit Cg_{T_c,T_f}, l'ensemble des nœuds convergents de la paire d'arbres (T_c, T_f). On a $Cg_{T_c,T_f} \subseteq V_c \bigcap V_f$. Autrement dit, il s'ensuit que $\left|Cg_{T_c,T_f}\right| \leq q$. Selon le lemme 4.2, la sélection d'une paire de sous-arbres à liens partagés a une complexité temporelle en $O\left(|V_c|^4 + \left|V_f\right|^3.\left|V_c\right|\right)$. Ainsi, $Comp\,(ligne\ 16)$ est en $O\left(q\left(|V_c|^4 + \left|V_f\right|^3.\left|V_c\right|\right)\right)$ (iii). Par analogie à $Comp\,(ligne\ 5-14)$, on a $Comp\,(ligne\ 17-29)$ est en $O\left(\left|V_f\right|^2 + |V_c|^2 + p.\left|V_c\right|\right)$ (iv). De (i), (ii), (iii) et (iv), on conclut que l'Algorithme 4.7 a une complexité en $O\left(p.q\left(|V_c|^3 + \left|V_f\right|.\left|V_c\right|\right) + q.\left|V_f\right|^3.\left|V_c\right|\right)$.

➢ Illustration de l'algorithme

La Figure 4.4 présente une instance du problème de reconfiguration d'une paire d'arbres mono-optiques (arbre mono-optique initial, arbre mono-optique final) concernant la connexion multicast $< s, \{l, m, n, o, p, k\} >$. Sur cette figure, l'ensemble des traits continus et de couleur bleue forme l'ensemble des liens de l'arbre mono-optique initial T_0 enraciné au nœud s. L'ensemble des traits discontinus et de couleur bleue forme l'ensemble des liens de l'arbre mono-optique final T_f enraciné au nœud s. Par ailleurs, au début du processus, l'arbre mono-optique actuel équivaut à T_0. Cet arbre actuel (voir l'ensemble des traits continus et de couleur bleue de la Figure 4.5.a) est pris comme configuration actuelle lors de la phase d'initialisation. L'ensemble des paires de sous-arbres à liens disjoints est formé par la paire de sous-arbres à liens disjoints $(a \rightarrow f \rightarrow g, a \rightarrow g)$ qui est enraciné au nœud a et la paire de sous-arbres à liens disjoints $(b \rightarrow e \rightarrow h, b \rightarrow h)$ qui est enraciné au nœud b. Ainsi, l'ensemble des nouveaux sous-arbres est formé par le sous-arbre $a \rightarrow g$ et le sous-arbre $b \rightarrow h$. Les Figures 4.5.b à 4.5.d montrent les trois configurations (déterminées par la deuxième phase de l'algorithme Para_SbSRA) qui concernent les paires de sous-arbres à liens disjoints. L'algorithme SbSRA aurait fourni six configurations pour l'ensemble des paires de sous-arbres à liens disjoints (contenant deux paires de sous-arbres dans cet exemple),

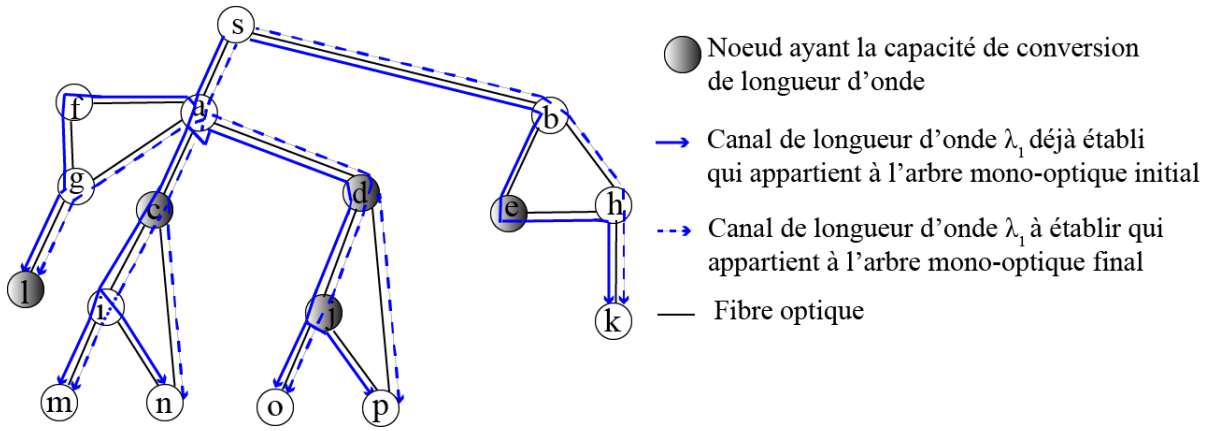


FIGURE 4.4 – Instance de problème à reconfigurer avec Para_SbSRA

à raison de trois configurations par paire de sous-arbres. L'ensemble des paires de sous-arbres à liens partagés est formé par les paires $(c \rightarrow i \rightarrow \{m, n\}, c \rightarrow \{n,\ i \rightarrow m\})$ et $(d \rightarrow j \rightarrow \{o, p\}, d \rightarrow \{p, j \rightarrow o\})$. Les Figures 4.5.e à 4.5.g montrent les trois premières configurations (sur un total de six) qui concernent les paires de sous-arbres à liens partagés. Par souci de concision, nous présentons sur la Figure 4.5.h la dernière configuration qui concernent les paires de sous-arbres à liens partagés. Cette dernière configuration est identique à la configuration finale C_f, c'est-à-dire à l'arbre mono-optique final T_f. Ainsi, la séquence de configurations est complète. L'algorithme SbSRA aurait fourni douze configurations dans cette dernière phase pour l'ensemble des paires de sous-arbres à liens partagés (contenant deux paires de sous-arbres dans cet exemple), à raison de six configurations par paire de sous-arbres au lieu de six au total pour l'algorithme Para_SbSRA.

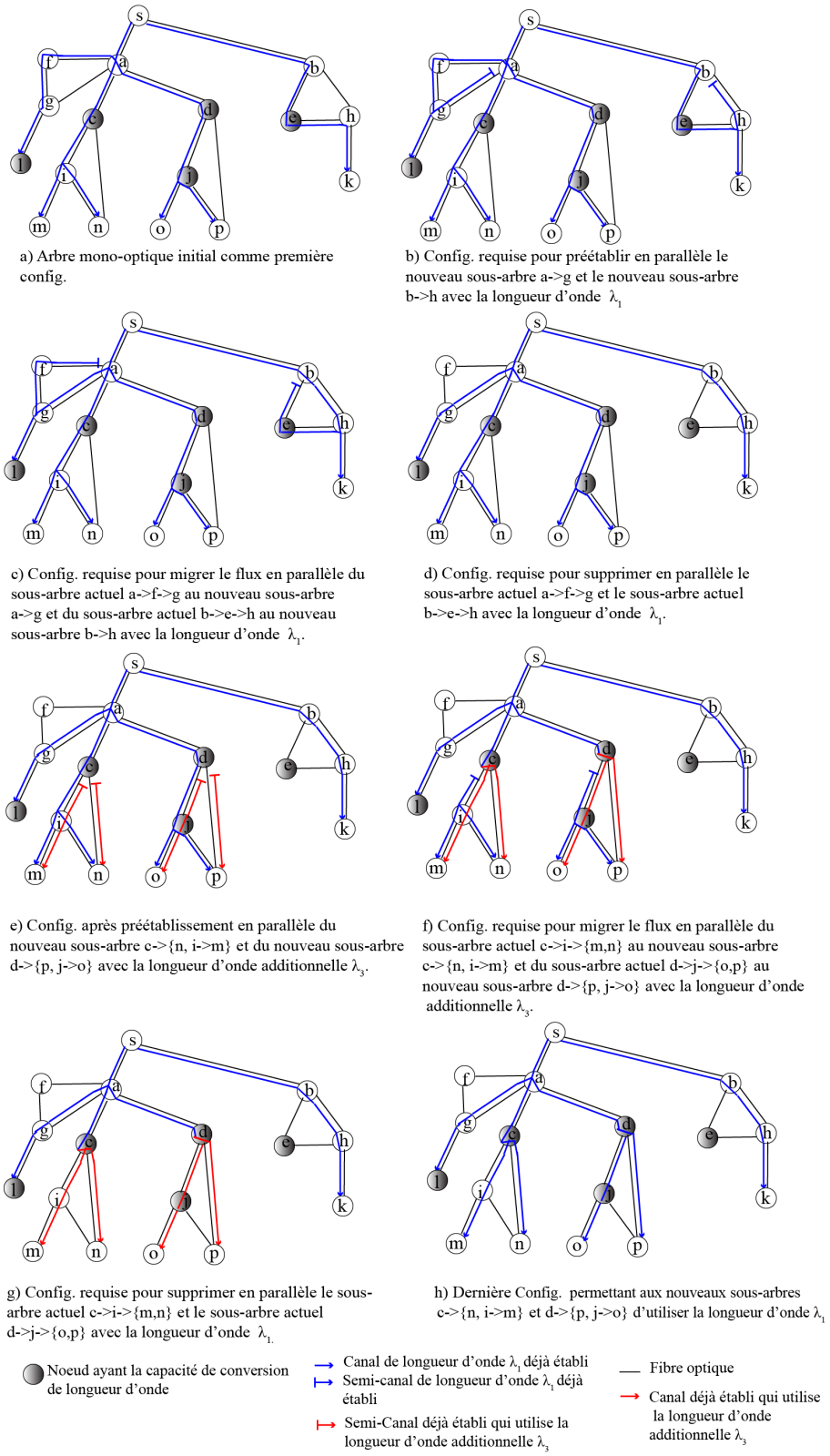


FIGURE 4.5 – Illustration d'une séquence de configurations avec Para_SbSRA

4.4 Evaluation de performance

Des simulations ont été menées pour étudier l'efficacité de la méthode Para_SbSRA lorsque l'on fait varier divers paramètres du système. Le langage Python est utilisé pour implémenter notre méthode. Toutes les expériences ont été réalisées sur un ordinateur portable équipé d'un processeur Intel Core i5-9300H à 2,40 GHz, de 8 Go de RAM et d'un système d'exploitation Windows 10. Les critères évalués au cours des expériences et le cadre des simulations sont présentés dans les sous-sections suivantes.

4.4.1 Critères d'évaluation

Pour déterminer l'efficacité de la méthode proposée, nous comparons notre méthode (c'est-à-dire Para_SbSRA) avec RCBRwPR et MBB_1. Pour ce faire, trois critères ont été retenus : le taux moyen d'interruptions du flux (voir Equation (4.2)), le coût des longueurs d'onde additionnelles (voir Equation (4.4)) et la longueur du processus de reconfiguration.

La longueur du processus de reconfiguration désignée par $nb_steps\,(SC)$ est le nombre d'étapes requises par la séquence de configurations SC. Une étape permet de passer d'une configuration à une autre. Ainsi, ce critère est défini par l'équation (4.5) :

$$nb_steps\,(SC) = |SC| - 1 \tag{4.5}$$

4.4.2 Description des simulations

Les expériences sont basées sur trois topologies de réseau (voir Tableau 4.1 et Figure 4.6) largement utilisées dans le domaine de la recherche sur les réseaux optiques [116]. Ces trois topologies sont NSFNET (petite taille), GEANT [2] (taille moyenne) et CORONET [3] (grande taille) [117].

2. http://www.topology-zoo.org/dataset.html
3. http://monarchna.com/topology.html

Tableau 4.1 – Informations sur les topologies

Topologies	Nb. de nœuds	Nb. de liens
NSFNET	14	21
GEANT	40	75
CORONET	75	99

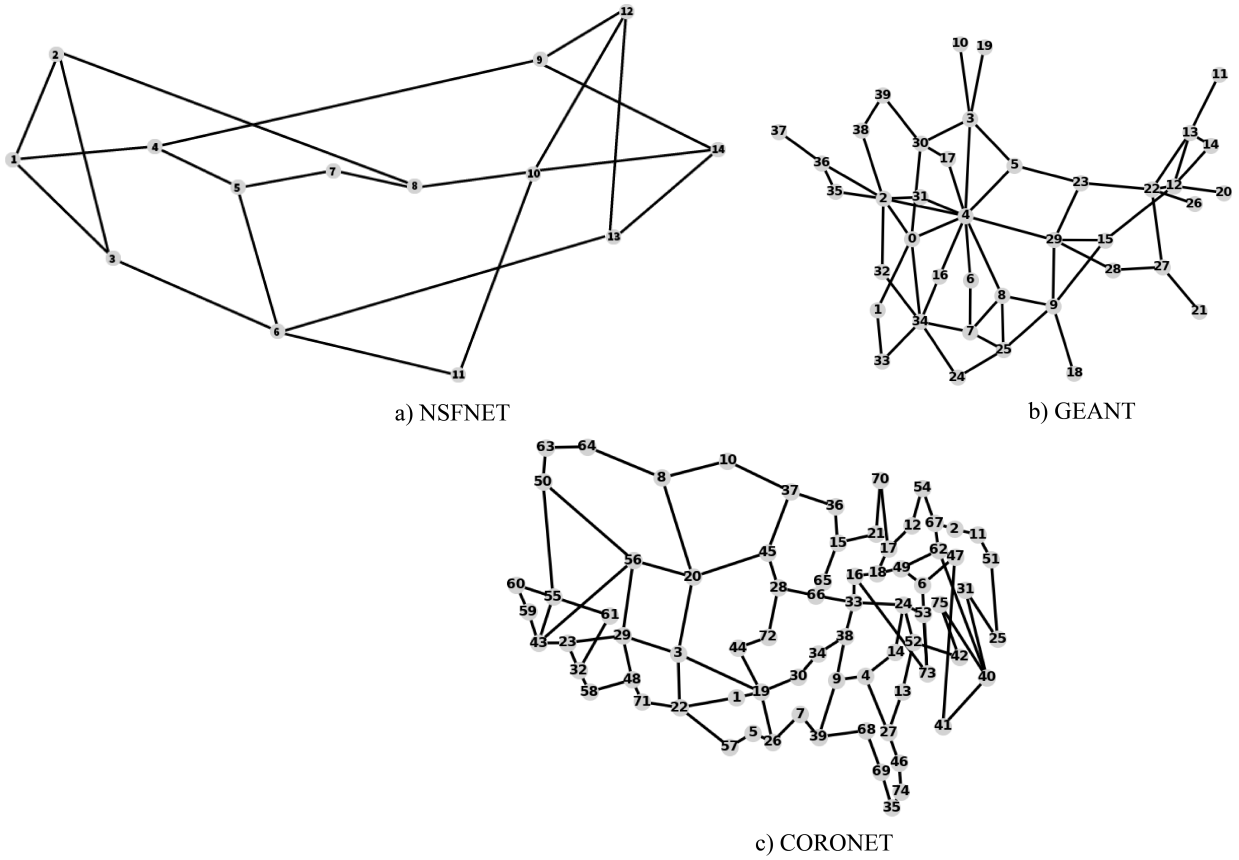


FIGURE 4.6 – Topologies réseaux

Chaque topologie est représentée par un graphe $G\ (V, E)$ qui est implémenté en utilisant Networkx [118], une librairie du langage Python destinée à l'analyse des graphes. V est l'ensemble des nœuds et E est l'ensemble des liens. Notez que pour chaque topologie, le nombre de nœuds ayant la capacité de conversion de longueur d'onde $|V_{cl}|$ est égale à l'arrondi à l'entier de $\frac{2}{5} * |V|$. Cette valeur est celle recommandée dans [14]. Par ailleurs, pour un réseau optique WDM ayant la propriété de conversion partielle de longueur d'onde, toute valeur dans l'intervalle $[\frac{|V|}{4}; \frac{|V|}{2}]$ est considérée comme un bon compris coût-performance pour l'opérateur de réseau [119].

Pour chaque topologie, une simulation a été réalisée. Nous supposons sans perte de généralité que chaque lien (fibre) a une capacité de $|\Omega|$ longueurs d'onde $(|\Omega| = 16)$. Une simulation consiste à générer un nombre d'instances NI (avec $NI = 10000$) du problème de reconfiguration. Une instance de problème est générée suivant le processus décrit ci-dessous :

— Sélectionnez aléatoirement un nœud de la topologie qui est pris comme nœud source de la connexion multicast ;

— Sélectionnez aléatoirement un nombre de nœuds qui sont utilisés comme nœuds destinations de la connexion multicast, parmi les nœuds de la topologie hormis le nœud source ;

— L'arbre initial de la connexion de multicast est construit en générant l'arbre des chemins les plus courts qui couvre le nœud source et les nœuds destinations à l'aide de l'algorithme de Dijkstra [120] ;

— L'arbre final est construit en générant l'arbre de poids minimum qui couvre le nœud source et les nœuds destinations à l'aide de l'algorithme Prim [121] tronqué autour des membres ;

— Les deux arbres se voient attribuer la même longueur d'onde choisie aléatoirement (loi uniforme) dans l'ensemble des longueurs d'onde disponibles.

— Cette paire est utilisée en entrée des trois méthodes : MBB_1, RCBRwPR et notre méthode (c'est-à-dire Para_SbSRA).

Le choix de $NI = 10000$ a été inspirée de la méthode de simulations successives décrite dans [122] en prenant $NI_1 = 1000$, $NI_2 = 2000$, $NI_3 = 3000$ jusqu'à $NI_{10} = 10000$ instances. En clair, nous avons calculé la moyenne (respectivement l'écart-type) pour un nombre varié d'exécutions d'une même simulation et constaté que l'écart entre deux moyennes (respectivement deux écarts-types) pour chaque critère d'évaluation devient négligeable lorsqu'on passe de $NI_9 = 9000$ à $NI_{10} = 10000$. Soient $X(NI_k)$ la valeur d'une mesure X pour un nombre d'instances NI_k et $X(NI_{k+1})$ la valeur de la mesure X pour un nombre d'instances NI_{k+1}. Nous admettons que l'écart relatif entre $X(NI_k)$ et $X(NI_{k+1})$, définie par la quantité $\frac{|X(NI_{k+1})-X(NI_k)|}{X(NI_{k+1})} * 100$ est dit négligeable si cette quantité est inférieur à 10%.

Notons qu'au début de chaque exécution du processus de création d'une instance de problème précédemment décrit, nous choisissons aléatoirement les $|V_{cl}|$ nœuds équipés de convertisseurs de longueur d'onde afin d'éviter que nos résultats ne soient

pas fortement dépendants d'un choix donné. À la fin de chaque simulation, nous évaluons statistiquement (moyenne, maximum, minimum, écart type) les critères de performance des trois méthodes.

4.4.3 Analyse des résultats

Les résultats (moyenne (AVG), écart type (SD), maximum (M) et minimum (m)) des trois critères de performance sont reportés dans les Tableaux 4.2 à 4.3. Le Tableau 4.2 montre que notre méthode (c'est-à-dire Para_SbSRA) fournit un taux moyen d'interruptions de flux égal à zéro, quelle que soit la topologie considérée (NSFNET, GEANT, CORONET). Ce résultat confirme que Para_SbSRA permet la reconfiguration d'une paire d'arbres mono-optiques sans interruption de flux dans un réseau optique WDM ayant la propriété de conversion partielle de longueur d'onde. Mais, le Tableau 4.2 et la Figure 4.7 montrent que MBB_1 et RCBRwPR fournissent un taux d'interruptions non nul (en pourcentage) quelle que soit la topologie de réseau utilisée. Cela s'explique par le fait que certaines instances (du problème) générées lors des simulations contenaient forcément des paires de branches qui partageaient des liens et que certains nœuds de basculement (différent de la racine de la paire d'arbres mono-optiques) n'avaient pas la capacité de conversion de longueur d'onde. Ainsi, ces deux méthodes provoquent des interruptions de flux. Notons que sur la Figure 4.7, la couleur de notre méthode (c'est-à-dire le bleu) n'apparaît pas, car selon le Tableau 4.2, notre méthode ne génère pas d'interruptions de flux. En bref, les expériences confirment que le MBB_1 et le RCBRwPR sont peu efficaces pour la reconfiguration des arbres mono-optiques dans un réseau optique WDM ayant la propriété de conversion partielle de longueur d'onde.

Tableau 4.2 – Taux d'interruptions moyen (%) causé par les trois méthodes

	NSFNET			**GEANT**			**CORONET**		
Methodes	**AVG**	**SD**	**m/M**	**AVG**	**SD**	**m/M**	**AVG**	**SD**	**m/M**
Para_SbSRA	0	0	0/0	0	0	0/0	0	0	0/0
RCBRwPR	24.06	22.86	0/66	16.63	14.12	0/88	12.98	12.95	0/78
MBB_1	26.02	16.11	2/83	32.55	19.42	1/95	57.06	21.02	6/96

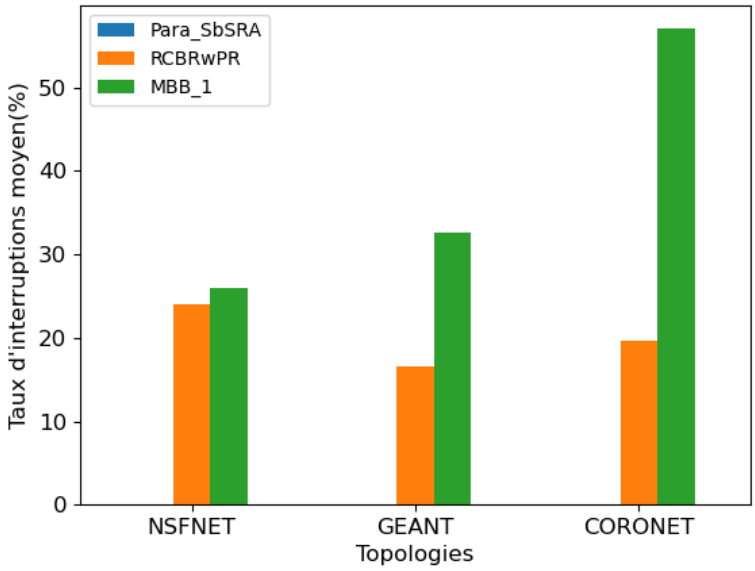


FIGURE 4.7 – Comparaison des taux d'interruptions moyen causées par les trois méthodes

Le Tableau 4.3 et la Figure 4.8 montrent que RCBRwPR fournit le coût des longueurs d'onde additionnelles le plus élevé. Cela s'explique par le fait que cette méthode utilise systématiquement une longueur d'onde additionnelle pour toute paire de branches donnée en dépit d'une réelle nécessité. Autrement dit, même si une paire de branches est à liens disjoints, une longueur d'onde additionnelle est requise par cette méthode. Cependant, Para_SbSRA (notre méthode) nécessite l'utilisation d'une seule longueur d'onde additionnelle uniquement pour reconfigurer des paires de sous-arbres à liens partagés et MBB_1 ne nécessite pas de longueur d'onde additionnelle. Notez que la couleur qui représente MBB_1 (c'est-à-dire le vert) n'apparaît pas dans la Figure 4.8, car MBB_1 ne nécessite pas l'utilisation d'une longueur d'onde additionnelle. Il est important de préciser également que puisque Para_SbSRA (notre méthode) nécessite l'utilisation d'une seule longueur d'onde additionnelle uniquement pour reconfigurer des paires de sous-arbres à liens partagés, si après la reconfiguration des paires de sous-arbres à liens disjoints l'arbre actuel obtenu est identique à l'arbre mono-optique alors la reconfiguration de la paire d'arbres n'utilisera pas de longueurs d'onde additionnelles. Cela explique le fait que la valeur minimale du coût des longueurs d'onde additionnelles soit égale à zéro pour notre méthode telle que confirmée par le Tableau

4.3.

Tableau 4.3 – Coût des longueurs d'onde additionnelles causé par les trois méthodes

	NSFNET			GEANT			CORONET		
Methodes	**AVG**	**SD**	**m/M**	**AVG**	**SD**	**m/M**	**AVG**	**SD**	**m/M**
Para_SbSRA	6.12	4.48	0/20	30.8	18.37	0/77	51.03	23.75	0/116
RCBRwPR	24.3	10.47	6/48	116.66	65.15	4/292	203.69	100.3	8/496
MBB_1	0	0	0/0	0	0	0/0	0	0	0/0

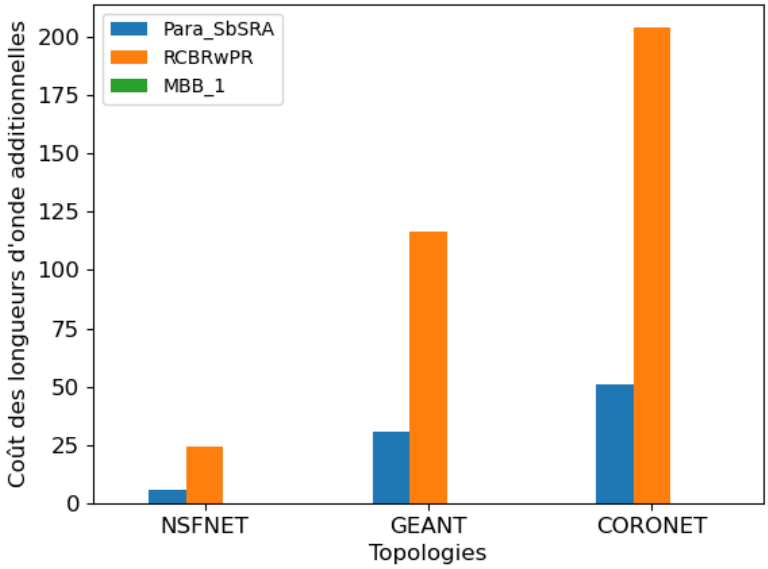


FIGURE 4.8 – Comparaison du coût des longueurs d'onde additionnelles causé par les trois méthodes

Le Tableau 4.4 et la Figure 4.9 montrent que MBB_1 induit un plus long processus de reconfiguration contrairement à Para_SbSRA qui induit le plus court processus de reconfiguration. Ces résultats s'expliquent par le fait qu'à chaque étape de reconfiguration, MBB_1, RCBRwPR et Para_SbSRA permettent respectivement de reconfigurer une seule paire de branches, de reconfigurer en parallèle un ensemble de paires de branches et de reconfigurer en parallèle un ensemble de paires de sous-arbres. Il est également important de noter que les résultats de l'expérience dans le Tableau 4.4 (voir les trois valeurs m/M) confirment que notre méthode (c'est-à-dire Para_SbSRA)

utilise au moins trois étapes et aux plus neuf étapes pour résoudre n'importe quelle instance du problème.

Tableau 4.4 – Durée (nombre d'étapes de reconfiguration) causée par les trois méthodes

	NSFNET			GEANT			CORONET		
Methodes	**AVG**	**SD**	**m/M**	**AVG**	**SD**	**m/M**	**AVG**	**SD**	**m/M**
Para_SbSRA	6.15	2.13	3/9	6.82	1.79	3/9	6.5	1.25	3/9
RCBRwPR	7.31	3.06	3/14	23.46	12	3/52	20.81	9.06	3/42
MBB_1	9.68	3.54	6/21	30.01	14.04	6/57	19.54	8.87	6/45

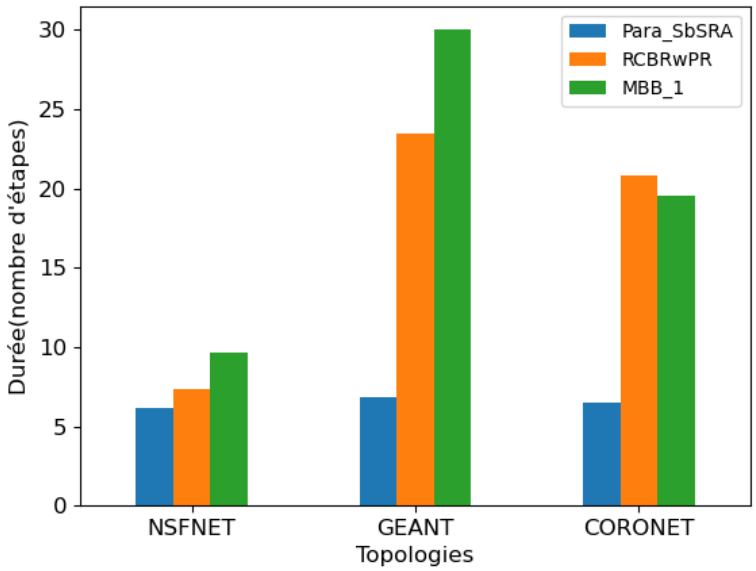


FIGURE 4.9 – Comparaison des durées de reconfiguration causées par les trois méthodes

4.5 Conclusion

Dans ce chapitre, nous avons abordé un problème relatif à la mono-reconfiguration du routage multicast à savoir le problème de reconfiguration d'une paire d'arbres mono-optiques. Il s'agit ici de trouver la séquence de configurations qui permet de migrer

le flux optique d'un arbre mono-optique initial vers un arbre mono-optique final sans interruption de flux tout en utilisant un faible nombre de fois des longueurs d'onde additionnelles. La recherche de la solution optimale étant fastidieuse, les travaux de la littérature (vus au chapitre précédent) ont proposé des méthodes (ou heuristiques) basées sur une approche de reconfiguration dite par branches. En clair, la paire d'arbres mono-optiques est reconfigurée suivant les paires de branches extraites de cette paire d'arbres mono-optiques. En outre, les méthodes (par exemple RCBRwPR) qui peuvent effectuer la reconfiguration sans interruption de flux s'appuient sur un nœud de basculement qui doit absolument avoir la capacité de conversion de longueur d'onde pour reconfigurer une paire de branches. Cette forte exigence sur le nœud de basculement ne peut être toujours honorée dans le cas d'un réseau optique WDM ayant la propriété de conversion de longueur d'onde. Afin d'y remédier, nous avons proposé des méthodes (ou heuristiques) de reconfiguration à savoir SbSRA et Para_SbSRA qui sont basées plutôt sur une approche par sous-arbre.

Cette approche par sous-arbres se base tout d'abord sur la catégorisation des paires de sous-arbres que l'on peut sélectionner dans la paire d'arbres mono-optiques à reconfigurer. Ainsi, il y a la catégorie des paires de sous-arbres à liens disjoints qui sont reconfigurables sans utilisation de longueur d'onde additionnelle et la catégorie des paires de sous-arbres à liens partagés qui sont reconfigurables en utilisant une longueur d'onde additionnelle. Nous avons aussi montré que pour résoudre le problème de reconfiguration étudié dans ce chapitre, il fallait reconfigurer les paires de sous-arbres dans l'ordre suivant : les paires de sous-arbres à liens disjoints suivis des paires de sous-arbres à liens partagés. La reconfiguration d'une paire de sous-arbres (sous-arbre actuel, nouveau sous-arbre) consiste à préétablir le nouveau sous-arbre, à migrer le flux sur le nouveau sous-arbre et à supprimer le sous-arbre actuel. Il est important de noter que le nœud utilisé comme nœud de basculement pour reconfigurer une paire de sous-arbres est le nœud racine de la paire de sous-arbres. Ainsi, la méthode SbSRA procède par reconfiguration des paires de sous-arbres à liens disjoints (une paire à la fois) suivi de la reconfiguration des paires de sous-arbres à liens partagés (une paire à la fois). Autrement dit, la séquence de configurations retournée par SbSRA se compose de la séquence de configurations qui concerne les paires de sous-arbres à liens disjoints, suivie de la séquence de configurations qui concerne les paires de sous-arbres à liens partagés. Et chaque configuration de la séquence de configurations concerne une seule paire de sous-arbres. Quant à la méthode Para_SbSRA,

elle se base sur le théorème d'indépendance et son corollaire proposé. Ainsi, contrairement à SbSRA, plusieurs paires de sous-arbres sont reconfigurées à la fois avec la méthode Para_SbSRA. Autrement dit, une configuration contenue dans la séquence de configurations retournée par Para_SbSRA concerne au moins une paire de sous-arbres.

Les simulations menées sur trois topologies différentes bien connues ont confirmé le fait que nos méthodes permettent de reconfigurer toute paire d'arbres mono-optiques et ceci sans interruption de flux tout en réduisant l'utilisation de longueurs d'onde additionnelles dans un réseau optique WDM ayant la propriété de conversion partielle de longueur d'onde ce qui n'est pas le cas des méthodes de reconfiguration de la littérature qui sont basées sur une approche par branches. Il important de noter qu'il a été montré dans [90] que s'il n'y a pas de longueur d'onde additionnelle disponible à une étape de tout processus de reconfiguration qui en requiert alors il y a interruption de flux vers au moins un des nœuds destinations. Par conséquent, aucune méthode ne peut effectuer une reconfiguration pour n'importe quelle instance du problème de reconfiguration sans interruption de flux et sans utiliser au moins une fois une longueur d'onde additionnelle .

Dans de nombreux cas, la reconfiguration du routage peut concerner plusieurs connexions à la fois. Par conséquent, le chapitre suivant aborde le problème de reconfiguration d'un ensemble de paires d'arbres mono-optiques.

CHAPITRE 5

RECONFIGURATION D'UN ENSEMBLE DE PAIRES D'ARBRES MONO-OPTIQUES

Sommaire

5.1 Introduction **110**

5.2 Spécification du problème **112**

5.3 Méthodes de reconfiguration pour un ensemble de plusieurs paires d'arbres mono-optiques **114**

5.3.1 Méthode de reconfiguration réduisant le coût d'interruptions de flux 115

5.3.2 Méthode de reconfiguration réduisant les pénalités financières 123

5.4 Evaluation de performance **126**

5.4.1 Critères d'évaluation 127

5.4.2 Description des simulations 127

5.4.3 Analyse des résultats 131

5.5 Conclusion **140**

5.1 Introduction

Dans le chapitre précédent, il a été question de résoudre un problème lié à la mono-reconfiguration du routage multicast dans les réseaux WDM ayant la propriété de conversion partielle de longueur d'onde. C'est le problème qui consiste à migrer le flux d'un arbre mono-optique initial d'une connexion multicast donnée vers un arbre mono-optique final de cette même connexion sans interruption de flux. Cependant, selon le chapitre 1, dans son activité d'administration du réseau, l'opérateur doit également faire face à des problématiques liées à la multi-reconfiguration du routage multicast. On parle de multi-reconfiguration du routage multicast lorsqu'un opérateur réseau

applique le mécanisme de reconfiguration du routage à au moins deux connexions multicasts. Nous rappelons que l'arbre mono-optique (voir chapitre 2) est la solution de routage multicast considéré dans cette thèse. Ainsi, dans ce chapitre, il est question de multi-reconfiguration du routage multicast, et plus précisément de reconfiguration d'un ensemble de paires d'arbres mono-optiques. Chaque paire d'arbres mono-optiques concerne une connexion multicast donnée. L'arbre mono-optique final d'une connexion multicast m_1 peut nécessiter des canaux optiques déjà utilisés par l'arbre mono-optique initial d'une connexion multicast m_2. Par conséquent, les solutions évoquées au chapitre 4, ne sont pas toujours utilisables pour reconfigurer en parallèle chaque paire d'arbres mono-optiques dans le cas d'une multi-reconfiguration du routage multicast.

Le problème étudié ici consiste à faire migrer le flux optique d'un ensemble d'arbres mono-optiques initialement établis (appelé configuration initiale) vers un ensemble d'arbres mono-optiques finaux (appelé configuration finale) à établir. Deux objectifs différents sont considérés lors de la migration du flux :

— **Objectif n° 1** : réduire le nombre d'interruptions de flux ;

— **Objectif n° 2** : réduire les pénalités financières.

En effet, l'objectif n° 1 est important si l'on ne veut pas dégrader la qualité du service fourni aux utilisateurs finaux. Quant à l'objectif n° 2, il est pris en compte, car les interruptions du flux peuvent entraîner des pénalités financières selon les contrats clients-fournisseurs ou *Service Level Agreements* (SLAs). Il convient donc d'éviter des pertes financières considérables à l'opérateur réseau. Le problème étudié est délicat, car il y a très souvent des dépendances entre les connexions concernées par la reconfiguration. Ainsi, les solutions proposées pour la reconfiguration d'une paire d'arbres mono-optiques (voir chapitre précédent) ne peuvent pas être utilisées pour reconfigurer en parallèle un ensemble de paires d'arbres mono-optiques.

Les principales contributions évoquées dans ce chapitre sont :

— La modélisation des dépendances entre les connexions à l'aide d'un graphe orienté ayant une pondération sur chacun de ses nœuds ;

— Un algorithme de reconfiguration noté LSRA induisant un nombre réduit d'interruptions de flux ;

— Un algorithme de reconfiguration noté LSRA_2 induisant un nombre réduit de pénalités financières.

Dans ce qui suit, il est question de faire une présentation détaillée du problème étudié (voir la section 5.2). Puis, nos méthodes sont exposées dans la section 5.3 avant d'être évaluées dans la section 5.4. Une conclusion (voir section 5.5) met fin à ce chapitre.

5.2 Spécification du problème

Le réseau est modélisé comme un graphe non-orienté $G = (V, E)$ où V est l'ensemble des nœuds optiques et E est l'ensemble des liens. Notons qu'on suppose que chaque lien est constitué d'une seule fibre. Soit $M = \{m_1, m_2, \ldots, m_n\}$ un ensemble de connexions multicasts établies dans le réseau WDM où $n = |M|$. Chaque connexion multicast $m_k \in M$ est également dénotée par $m_k =< s_k, D_k, PF_k >$ où s_k est la source de la connexion multicast m_k, $D_k = \{d_k^1, \ldots, d_k^{i_k}\}$ est l'ensemble des nœuds destinations de m_k, et PF_k la pénalité financière (en Unité Monétaire (UM)) par unité de temps pour chaque interruption du flux vers un nœud destination de la connexion m_k conformément au *Service Level Agreement* (SLA) entre l'opérateur et le client concerné par la connexion m_k. En clair, nous supposons ici que le coût en pénalité financière associé à l'interruption du flux vers tout nœud destination d'une connexion multicast donnée n'est pas fonction de ce nœud, mais est fixe. Le flux optique de la connexion multicast m_k est initialement transporté par un arbre mono-optique, dit initial, désigné par $T_k^0 = \left(V_k^0, E_k^0\left(\lambda_k^0\right)\right)$. Il convient de noter que $V_k^0 \subseteq V$, $s_k \in V_k^0, D_k \subseteq V_k^0 \setminus \{s_k\}$, $E_k^0 \subseteq E$. λ_k^0 est la longueur d'onde utilisée par tout canal optique établi appartenant à $E_k^0\left(\lambda_k^0\right)$. À un moment donné, l'opérateur de réseau décide de procéder à une multi-reconfiguration du routage multicast. Par conséquent, pour chaque connexion multicast m_k de M, il calcule un arbre mono-optique appelé arbre mono-optique final $T_k^f = \left(V_k^f, E_k^f\left(\lambda_k^f\right)\right)$ à établir afin d'être utilisé pour transporter le flux optique de m_k à la fin du processus de reconfiguration. Il convient de noter que $V_k^f \subseteq V$, $s_k \in V_k^f$, $D_k \subseteq V_k^f \setminus \{s_k\}$, $E_k^f \subseteq E$. λ_k^f est la longueur d'onde à utiliser pour établir tout canal optique non établi appartenant à $E_k^f\left(\lambda_k^f\right)$. Nous supposons qu'aucune longueur d'onde additionnelle n'est disponible lors de la multi-reconfiguration du routage multicast. Cela signifie que chaque connexion ne peut pas utiliser temporairement un arbre mono-optique utilisant une autre longueur d'onde. Par conséquent, les états de blocage vont entraîner des interruptions qu'il faut réduire et éventuellement des pénalités financières à réduire

également. Par ailleurs, nous supposons que si l'arbre mono-optique final T_k^f d'une connexion multicast m_k partage des canaux optiques déjà utilisés par l'arbre mono-optique initial T_{k-1}^0 d'une connexion multicast m_{k-1} alors au moins un de ces canaux partagés appartient à chaque chemin entre la source de m_{k-1} et toute destination de m_{k-1}. En clair, établir l'arbre mono-optique final de T_k^f de m_k ne peut se faire sans créer une interruption de flux [4] vers chaque nœud destination de m_{k-1}.

Le problème est formulé comme suit :

— **Données** : Un ensemble de connexions multicasts $M = \{m_1, m_2, \ldots, m_n\}$, la configuration initiale d'arbres mono-optiques $C_0 = \{T_1^0, \ldots, T_n^0\}$ et la configuration finale d'arbres mono-optiques $\{T_1^f, \ldots, T_n^f\}$ à établir où T_k^0 et T_k^f désigne respectivement l'arbre mono-optique initial déjà établi de la connexion multicast m_k et l'arbre mono-optique final à établir de la connexion multicast m_k.

— **Solution** : la séquence de configurations désignée par $SC =< C_0, C_1, \ldots, C_{f-1}, C_f >$ pour migrer le flux optique de C_0 à C_f, où $C_i = \{T_1^i, \ldots, T_n^i\}$ est la configuration d'arbres mono-optiques requise par l'ensemble des connexions multicasts M à l'étape i du processus de reconfiguration. Plus précisément, T_k^i est l'arbre mono-optique qui est utilisé pour transporter le flux optique de la connexion multicast m_k à l'étape i du processus de reconfiguration.

— **Objectifs** : La séquence de configurations SC doit impliquer soit un faible coût d'interruptions (ou nombre total d'interruptions de flux) soit un faible coût en pénalités financières.

Le coût d'interruptions désigné par $FLUX_INT\,(SC)$, est le nombre total d'interruptions de flux générées par le processus de reconfiguration. Soit $FLUX_INT\,(C_i)$, le nombre d'interruptions de flux causées par la configuration C_i. En d'autres termes, $FLUX_INT\,(C_i)$ est le nombre total d'interruptions de flux causées par l'ensemble des connexions multicasts M à l'étape i du processus de reconfiguration. En fait, il s'agit de la somme des destinations des connexions de M ne possédant pas un arbre mono-optique (c'est-à-dire un graphe nul [123]) dans la configuration C_i. $FLUX_INT\,(C_i)$ est obtenu grâce à l'Equation (5.1) :

$$FLUX_INT\,(C_i) = \sum_{m_k \in M} INT\,(C_i, m_k) * |D_k| \tag{5.1}$$

4. Cette contrainte permet d'avoir une version de base du problème de multi-RRM (étant plus proche de celle du cas unicast : une suppression de chemin interrompt le flux vers un seul nœud destination) à améliorer comme indiqué dans la section 6.2

où $INT(C_i, m_k) = \begin{cases} 1 & \text{si l'arbre de } m_k =< s_k, D_k, PF_k > \text{ dans } C_i \text{ est un graphe nul} \\ 0 & \text{Sinon.} \end{cases}$

Par conséquent, le coût d'interruptions (ou nombre total d'interruptions générées) $FLUX_INT(SC)$ est obtenu par l'Equation (5.2) :

$$FLUX_INT(SC) = \sum_{C_i \in SC} FLUX_INT(C_i) \tag{5.2}$$

Le coût en pénalités financières désigné par $PEN_FIN(SC)$, est le nombre total de pénalités financières générées par le processus de reconfiguration. Soit $PEN_FIN(C_i)$, la somme des pénalités financières de chacune des connexions de M dont l'arbre mono-optique dans la configuration C_i est un graphe nul. Nous supposons qu'il faut une unité de temps pour passer d'une configuration à une autre de la séquence de configurations SC. Ainsi, $PEN_FIN(C_i)$ est obtenu grâce à l'Equation (5.3) :

$$PEN_FIN(C_i) = \sum_{m_k \in M} INT(C_i, m_k) * PF_k * |D_k| \tag{5.3}$$

Par conséquent, Le coût en pénalités financières $PEN_FIN(SC)$ du processus de reconfiguration est donné par l'Equation (5.4) :

$$PEN_FIN(SC) = \sum_{C_i \in SC} PEN_FIN(C_i) \tag{5.4}$$

5.3 Méthodes de reconfiguration pour un ensemble de plusieurs paires d'arbres mono-optiques

Dans le chapitre 3, nous avons vu que le problème de reconfiguration d'un ensemble de paires de chemins (unicasts) mono-optiques est un cas spécifique du problème de reconfiguration d'un ensemble de paires d'arbres mono-optiques (voir section 3.4.3). Les solutions connues au problème de reconfiguration d'un ensemble de paires de chemins mono-optiques tiennent compte uniquement de l'objectif de réduction du nombre total d'interruptions de flux. Ces solutions sont basées sur l'approche MFVSA. Cependant, MFVSA (proposé dans la littérature) ne renvoie pas une bonne solution pour le problème de reconfiguration d'un ensemble de paires d'arbres mono-optiques étudié dans ce chapitre.

Dans cette section, nous présentons dans un premier temps (voir section 5.3.1) une méthode de reconfiguration d'un ensemble de paires d'arbres mono-optiques avec pour objectif la réduction du coût d'interruptions (ou nombre total d'interruptions de flux). Puis dans un second temps (voir section 5.3.2), nous présentons une méthode de reconfiguration d'un ensemble de paires d'arbres mono-optiques avec pour objectif la réduction du coût en pénalités financières (ou nombre total de pénalités financières).

5.3.1 Méthode de reconfiguration réduisant le coût d'interruptions de flux

L'approche MFVSA s'appuie sur un ensemble de nœuds de rétroaction de cardinalité minimum. Ce qui fait que les méthodes basées sur cette approche ne renvoient pas toujours une bonne solution au problème de reconfiguration d'un ensemble de paires d'arbres mono-optiques. Par conséquent, nous proposons de nous appuyer sur un ensemble de nœuds de rétroaction de faible coût d'interruptions. Afin de déterminer un ensemble de nœuds de rétroaction de faible coût d'interruptions, nous faisons recours à un graphe orienté de dépendances qui est pondéré par les nœuds. Autrement dit, un poids est associé à chaque nœud du graphe. Dans les sous-sections suivantes, nous décrivons dans un premier temps comment construire le graphe orienté des dépendances qui est pondéré par les nœuds et dans un second temps nous présentons en détail l'algorithme proposé.

5.3.1.1 Graphe orienté pondéré de dépendances

Un graphe orienté pondéré par les nœuds ou *vertex-weighted graph* est un outil classique de la théorie des graphes qui a plusieurs applications [124] dont la modélisation de certains problèmes d'ordonnancement. Nous utilisons cet outil pour modéliser les dépendances entre les connexions. Ainsi, le graphe orienté pondéré de dépendances par les nœuds se construit comme suit :

— Les nœuds et les arcs de ce graphe sont créés suivant le procédé de construction du graphe orienté non pondéré de dépendances décrit au chapitre 3 (voir section 3.4.2)

— Un poids est attribué à chaque nœud selon une fonction de pondération. Formellement, le graphe orienté de dépendances qui est pondéré par les nœuds

est désigné par $G_p\left(V_{G_p}, E_{G_p}, p\right)$ où V_{G_p} est l'ensemble des nœuds, E_{G_p} est l'ensemble des arcs et p est la fonction de pondération qui est définie comme suit :

$$\begin{aligned} p : V_{G_p} &\longrightarrow N^* \\ v &\mapsto p(v), \end{aligned}$$

où $p(v)$ est le coût de la suppression de l'arbre mono-optique initial de la connexion correspondant au nœud v dans la configuration initiale. Notre algorithme doit provoquer un nombre réduit d'interruptions de flux vers les nœuds destinations des connexions multicasts. Par conséquent, intuitivement, $p(v)$ indique le nombre de nœuds destinations de la connexion que représente le nœud v.

5.3.1.2 Algorithme proposé

❖ **Présentation de l'algorithme**

Dans cette sous-section, nous proposons l'algorithme de reconfiguration d'un ensemble de paires d'arbres mono-optiques (ou Light-tree pairs Set Reconfiguration Algorithm) noté LSRA afin de résoudre le problème de multi-reconfiguration du routage multicast. En d'autres termes, nous proposons l'algorithme LSRA afin de reconfigurer un ensemble de paires d'arbres mono-optiques. Cet algorithme (voir Algorithme 5.1) prend en entrées la configuration initiale C_0, la configuration finale C_f et l'ensemble des connexions multicasts M. Il renvoie une séquence de configurations désignée par SC qui provoque un petit nombre d'interruptions de flux vers les nœuds destinations des connexions multicasts appartenant à M. Pour ce faire, notre algorithme peut se résumer aux points suivants :

(i) Construire un graphe orienté de dépendances qui est pondéré par les nœuds (voir la ligne 1 de l'Algorithme 5.1).

(ii) Supprimer temporairement l'arbre mono-optique initial de chaque connexion appartenant à un ensemble de connexions de faible coût d'interruptions noté MCFVS (voir ligne 5 de l'Algorithme 5.1).

(iii) Supprimer l'arbre initial et établir l'arbre final de chaque connexion indépendante qui n'appartient pas à MCFVS (voir ligne 8 à ligne 13 de l'Algorithme 5.1) puis terminer éventuellement la reconfiguration (c'est-à-dire établir l'arbre final) de

chaque connexion devenue indépendante qui appartient à MCFVS (voir de la ligne 17 à la ligne 21 de l'Algorithme 5.1). Répéter le procédé décrit dans ce point jusqu'à ce que toutes les connexions n'appartenant pas à MCFVS soient entièrement reconfigurées. Une connexion est entièrement reconfigurée si son arbre initial a été supprimé et que son arbre final a été établi. Une connexion est indépendante si son arbre final ne contient aucun canal optique actuellement requis par l'arbre utilisé par une autre connexion.

(iv) Si certaines connexions de MCFVS ne sont pas entièrement reconfigurées alors terminer leurs reconfigurations en établissant leurs arbres finaux (voir de la ligne 23 à 26 de l'Algorithme 5.1).

Algorithme 5.1 : LSRA

Entrées : $C_0 = \{T_1^0, \ldots, T_n^0\}$, $C_f = \{T_1^f, \ldots, T_n^f\}$,
$M = \{m_1, \ldots, m_n\}$ `// M : Ensemble de connexions multicasts`
Sorties : $SC =< C_0, C_1, \ldots, C_{f-1}, C_f >$ `// Séquence (ou liste) de configurations`

1 $G_p \leftarrow$ Graphe orienté pondéré de dépendances correspondant à C_0, C_f et M
2 $MCFVS \leftarrow$ Ensemble de nœuds de rétroaction de faible coût issu de G_p
3 $SC.inserer\,(C_0)\,;\, Curr_Conf \leftarrow C_0$
4 $Ens_arbres_act \leftarrow$ Ensemble d'arbres initiaux dans C_0 des connexions correspondantes aux nœuds de $MCFVS$
5 $Curr_Conf \leftarrow Curr_Conf \setminus Ens_arbres_act\,;\, SC.inserer\,(Curr_Conf)$
6 $V^* \leftarrow V_{G_p} \setminus MCFVS$ `//` V_{G_p} `: Ensemble des nœuds de` G_p
7 $V^*_reconfigurer \leftarrow \emptyset\,;\, MCFVS_reconfigurer \leftarrow \emptyset$
8 **Répéter**
9 $\quad V^*_nonReconf_graph \leftarrow$ Sous-graphe de G_p induit par $V^* \setminus V^*_reconfigurer$
10 $\quad V^*_Indep \leftarrow$ Ensemble de nœuds indépendants issu de $V^*_nonReconf_graph$
11 $\quad Ens_arbres_act \leftarrow$ Ensemble d'arbres initiaux dans C_0 des connexions correspondantes aux nœuds de V^*_Indep
12 $\quad Ens_arbres_fin \leftarrow$ Ensemble des configurations correspondantes aux arbres finaux dans C_f des connexions associées aux nœuds de V^*_Indep
13 $\quad Curr_Conf \leftarrow Curr_Conf \cup (Ens_arbres_fin) \setminus (Ens_arbres_act)\,;\, SC.inserer\,(Curr_Conf)$
14 $\quad V^*_reconfigurer \leftarrow V^*_reconfigurer \cup V^*_Indep$
15 $\quad V_nonReconf_graph \leftarrow$ Sous-graphe de G_p induit par $V \setminus (V^*_reconfigurer \cup MCFVS_reconfigurer)$
16 $\quad MCFVS_Indep \leftarrow$ Ensemble de nœuds indépendants issu de $V_nonReconf_graph$ tel que les nœuds de cet ensemble appartiennent à $MCFVS$
17 $\quad$ **Si** $MCFVS_Indep$ *est non vide* **alors**
18 $\quad\quad Ens_arbres_fin \leftarrow$ Ensemble des configurations correspondantes aux arbres finaux dans C_f des connexions associées aux nœuds de $MCFVS_Indep$
19 $\quad\quad Curr_Conf \leftarrow Curr_Conf \cup Ens_arbres_fin\,;\, SC.inserer\,(Curr_Conf)$
20 $\quad\quad MCFVS_reconfigurer \leftarrow MCFVS_reconfigurer \cup MCFVS_Indep$
21 $\quad$ **Fin Si**
22 **Jusqu'à** $V^*reconfigurer = V^*$;
23 **Si** $MCFVS_reconfigurer \neq MCFVS$ **alors**
24 $\quad Ens_arbres_fin \leftarrow$ Ensemble de configurations correspondantes aux arbres finaux dans C_f des connexions associées aux nœuds de $MCFVS \setminus MCFVS_reconfigurer$
25 $\quad Curr_Conf \leftarrow Curr_Conf \cup Ens_arbres_fin\,;\, SC.inserer\,(Curr_Conf)$
26 **Fin Si**
27 **Retourner** SC

En clair, le premier élément de la séquence (de configurations) renvoyée par LSRA

est C_0 (voir ligne 3 de l'Algorithme 5.1). L'élément suivant de cette séquence est obtenu (voir lignes 4 et 5 de l'Algorithme 5.1) en supprimant de la configuration courante $Curr_Conf$, les arbres mono-optiques initiaux des connexions dont les nœuds forment un ensemble des nœuds de rétroaction de faible coût. Notez que le problème du calcul d'un ensemble de nœuds de rétroaction à coût minimal (ou *Minimum Cost Feedback Vertex Set*) est NP-difficile [125]. De nombreuses heuristiques (dont l'heuristique de Demetrescu and Finocchi [126]) existent pour calculer un ensemble de nœuds de rétroaction de faible coût. Ensuite, d'autres éléments de la séquence de configurations sont obtenus en effectuant une opération qui concerne des nœuds indépendants (voir la ligne 13 ou la ligne 19 de l'Algorithme 5.1). Ces nœuds indépendants appartiennent soit à V^* $\left(V^* = V_{G_p} \setminus MCFVS\right)$ soit à MCFVS, où V_{G_p} est l'ensemble des nœuds du graphe orienté de dépendances G_p qui est pondéré par les nœuds. Notez qu'on dit d'un nœud n d'un graphe orienté G qu'il est un nœud indépendant si et seulement si son degré sortant est égal à zéro. En d'autres termes, un nœud indépendant représente une connexion qui ne dépend pas d'autres connexions à une étape du processus de reconfiguration. Si après le traitement des sommets appartenant à V^*, certains éléments de MCFVS ne sont pas entièrement traités, alors une configuration est ajoutée à la séquence de configurations pour compléter cette séquence (voir de la ligne 23 à la ligne 26 de l'Algorithme 5.1).

➢ Analyse de la correction de l'algorithme

L'ensemble des nœuds V_{G_p} du graphe orienté de dépendances G_p qui est pondéré par les nœuds, est tel que :

$$V_{G_p} = MCFVS \cup V^* \tag{5.5}$$

Le premier élément de la séquence de configurations SC est C_0. Les lignes 4 à 5, les lignes 11 et 13 de l'Algorithme 5.1, et l'Equation(5.5) montrent que le dernier élément de la séquence ne contient aucun arbre mono-optique appartenant à la configuration C_0. De plus, les lignes 12 et 13, les lignes 18 et 19, les lignes 24 et 25, et l'Equation (5.5) montrent que la dernière configuration contient chaque arbre mono-optique de la configuration finale. Ainsi, la dernière configuration est la configuration finale.

Il en résulte que SC est une séquence de configurations pour migrer de la configuration initiale à la configuration finale (: résultat i). MCFVS est un ensemble de nœuds de rétroaction de faible coût. Par conséquent, l'Algorithme 5.1 provoque un faible nombre d'interruptions de flux vers les nœuds destinations pendant le processus de reconfigu-

ration (: résultat ii). Les résultats i et ii prouvent la correction partielle de l'Algorithme 5.1 (: résultat iii). $V^*_reconfigurer$ est initialement vide (voir ligne 7 de l'Algorithme 5.1). De plus, à chaque itération de la boucle "Répéter", les éléments de V^*_Indep sont ajoutés à $V^*_reconfigurer$ (voir ligne 14 de l'Algorithme 5.1). Notons que V^*_Indep est un sous-ensemble de V^* (voir lignes 9 et 10). Ainsi, à une certaine itération de la boucle "Répéter", $V^*_reconfigurer$ devient égal à V^* : la boucle "Répéter" (voir de la ligne 8 à la ligne 20) se termine. Par conséquent, la terminaison de l'Algorithme 5.1 est prouvée (résultat iv). Le résultat iii et le résultat iv prouvent la correction totale de l'Algorithme 5.1. Cependant, l'Algorithme 5.1 fournit une bonne solution au problème de reconfiguration d'un ensemble de paires d'arbres mono-optiques, mais pas toujours la solution optimale. Comme mentionné dans l'état de l'art (voir section 3.4.2), la recherche de la solution optimale passe par une recherche exhaustive qui implique que pour chaque ensemble de nœuds de rétroaction FVS, il faut rechercher la séquence de configurations en évaluant le coût d'interruptions de flux qui est associée à cette séquence. Cette recherche exhaustive coûteuse en temps n'est pas souhaitable car elle peut laisser le réseau dans un état sous-optimale pendant longtemps.

➢ Analyse de la complexité de l'algorithme

L'analyse de la complexité temporelle repose sur les notations suivantes :

— n est le nombre de connexions à reconfigurer, avec $n = |M|$.

— $\left|\overline{E^f}\right|$ est le cardinal de l'arbre de plus grand ensemble de liens dans C_f.

— $\left|\overline{E^0}\right|$ est le cardinal de l'arbre de plus grand ensemble de liens dans C_0.

— $\left|E_{G_p}\right|$ est le nombre de liens du graphe de dépendances pondéré $G_p\left(V_{G_p}, E_{G_p}, p\right)$.

Rappelons que par construction, on a le cardinal de l'ensemble des nœuds $\left|V_{G_p}\right|$ qui vaut n.

Lemme 5.1 *Le processus (voir l'Algorithme 5.1) de reconfiguration d'un ensemble de plusieurs paires d'arbres induit une complexité temporelle (dans le pire des cas) en* $O\left(n^3.\left|E_{G_p}\right|\right)$.

Preuve On convient que $Comp\,(ligne\ x)$ désigne la complexité temporelle de l'instruction indiquée à la ligne x et $Comp\,(ligne\ x-y)$ désigne la complexité temporelle du bloc d'instructions commençant à la ligne x et se terminant à la ligne y. On convient

par souci de concision de ne pas mentionner tout $Comp\,(ligne\ x)$ en $O(1)$. Du principe de construction du graphe de dépendances (voir section 5.3.1.1), on déduit que $Comp\,(ligne\ 1)$ est en $O\left(n^2.\left|\overline{E^0}\right|.\left|\overline{E^f}\right|\right)$. Nous supposons que le calcul de MCFVS est obtenu grâce à l'heuristique de Demetrescu and Finocchi [126]. Par conséquent, $Comp\,(ligne\ 2)$ est en $O\left(n.\left|E_{G_p}\right|\right)$. $MCFVS < \left|V_{G_p}\right| \leq n-1$. Donc, on a $Comp\,(ligne\ 4)$ en $O\left(n^2\right)$. Par ailleurs, on a également $Comp\,(ligne\ 5)$ et $Comp\,(ligne\ 6)$ en $O\left(n^2\right)$. De ce qui précède, on déduit que $Comp\,(ligne\ 1-7)$ est en $O\left(n^2.\left|\overline{E^0}\right|.\left|\overline{E^f}\right|+n.\left|E_{G_p}\right|\right)$ (ii). On a $Comp\,(ligne\ 9)$ et $Comp\,(ligne\ 15)$ en $O\left(n^2.\left|E_{G_p}\right|\right)$. Puis, $Comp\,(ligne\ 10)$ est en $O\,(n)$ et $Comp\,(ligne\ 16)$ est en $O\left(n^2\right)$. On a $Comp\,(ligne\ 11)$ et $Comp\,(ligne\ 12)$ en $O\,(n)$. Puis, $Comp\,(ligne\ 13)$ et $Comp\,(ligne\ 14)$ sont en $O\left(n^2\right)$. Par conséquent, $Comp\,(ligne\ 9-16)$ est en $O\left(n^2.\left|E_{G_p}\right|\right)$. $Comp\,(ligne\ 17-21)$ est en $O\left(n^2\right)$. Donc, $Comp\,(ligne\ 9-21)$ est en $O\left(n^2.\left|E_{G_p}\right|\right)$. $|V^*| \leq n-1$. Par conséquent, $Comp\,(ligne\ 8-22)$ est $O\left(n^3.\left|E_{G_p}\right|\right)$ (ii). $Comp\,(ligne\ 23-26)$ est en $O\left(n^2\right)$ (iii). De (i), (ii) et (iii), on déduit que la complexité de l'Algorithme 5.1 est en $O\left(n^3.\left|E_{G_p}\right|\right)$.

➢ Illustration de l'algorithme

Nous illustrons l'Algorithme 5.1 à travers l'instance de problème décrit au chapitre 3 (voir Figure 3.6). Nous rappelons que cette instance concerne trois connexions multicasts $m_1 =< s_1, \{d_1, d_2\} >$, $m_2 =< s_2, \{d_3, d_4, d_5, d_6\} >$ et $m_3 =< s_2, \{d_7, d_8\} >$. La Figure 5.1 montre le graphe orienté pondéré qui est renvoyé par la ligne 1 de Algorithme 5.1.

FIGURE 5.1 – Graphe orienté de dépendances pondéré par les nœuds, associé à l'instance de problème

Notons que le poids de chaque nœud est signifié entre les parenthèses suivant le nœud. Par exemple, le poids du nœud m_2 est de quatre. Ce poids signifie que si l'arbre mono-optique de m_2 dans la configuration d'une étape de reconfiguration est un graphe nul alors m_2 provoque l'interruption du flux vers quatre nœuds destinations (c'est-à-dire quatre interruptions de flux) à cette étape. La Figure 5.2 montre un état de blocage parce que m_1 dépend de m_2 et m_2 dépend de m_1. Pour sortir de cet état de blocage, la ligne 2 de l'Algorithme 5.1 calcule l'ensemble des nœuds de rétroaction de faible coût

MCFVS. Il existe deux ensembles (utiles) de nœuds de rétroaction issus du graphe orienté de dépendances pondéré par les nœuds : $FVS1 = \{m_1\}$ et $FVS2 = \{m_2\}$. Le coût de $FVS1$ est égal au poids de m_1 et le coût de $FVS2$ est égal au poids de m_2. Par conséquent, l'ensemble de nœuds de rétroaction de faible coût retourné par l'Algorithme 5.1 est $FVS1$. Cela signifie que la configuration suivant la configuration C_0 dans la séquence de configurations SC est la configuration obtenue en supprimant de la configuration courante (c'est-à-dire C_0), l'arbre mono-optique initial de la connexion m_1. Ainsi, Il n'y a pas d'arbre mono-optique pour transmettre le flux de la source s_1 vers les nœuds destinations d_1 et d_2 de m_1. Par conséquent, cette nouvelle configuration provoque deux interruptions de flux équivalant au poids de m_1. La Figure 5.2 montre le graphe obtenu à la ligne 9 de l'Algorithme 5.1 (c'est-à-dire $V^*_nonReconf_graph$) de l'Algorithme 5.1. De ce graphe, il résulte que la première valeur de V^*_Indep (voir ligne 10 de l'Algorithme 5.1) est $\{m_2\}$.

Ainsi, on obtient une nouvelle configuration en ajoutant l'arbre final de m_2 à la configuration courante et en supprimant de la configuration courante, l'arbre initial de m_2. En outre, cette nouvelle valeur de la configuration courante provoque deux interruptions de flux également puisque aucun arbre ne permet de transmettre le flux de s_1 vers d_1 et d_2. Puisque les canaux requis par l'arbre final de la connexion m_1 sont libres alors l'arbre final configuré de m_1 est ajouté à la configuration courante. Puis le nouveau graphe $V^*_nonReconf_graph$, ne contient plus que le nœud m_3. Par conséquent, la nouvelle valeur de V^*_Indep est $\{m_3\}$. On obtient donc une nouvelle configuration en ajoutant l'arbre final de m_3 à la configuration courante et en supprimant de la configuration courante, l'arbre initial de m_3. Ainsi, la séquence de configurations renvoyée par Algorithme 5.1 est complète. De plus, notre dernière valeur de la configuration courante ne provoque pas d'interruptions de flux. Par conséquent, la séquence de configurations (retournée par LSRA illustrée par la Figure 5.3) cause quatre interruptions de flux contre huit interruptions pour la séquence de configurations retournées par MFVSA (voir Figure 3.11) pour l'instance de problème de la Figure 3.6.

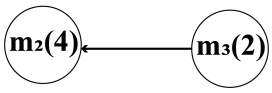
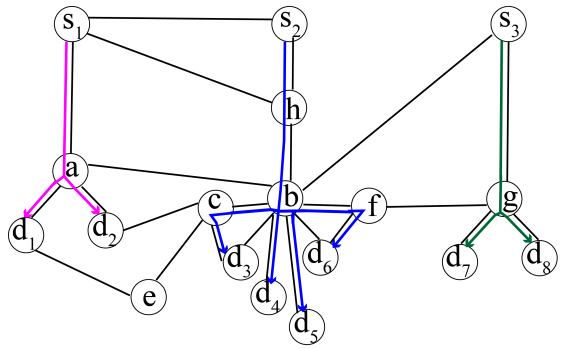
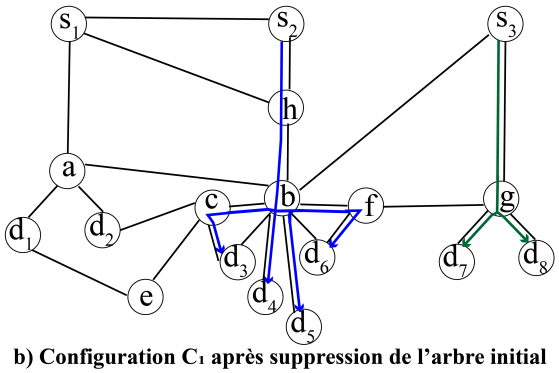


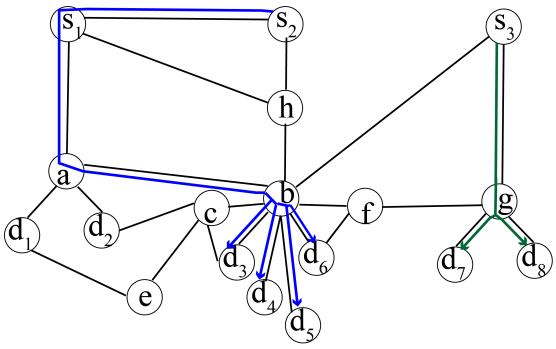
FIGURE 5.2 – Graphe obtenu après suppression de MCFVS



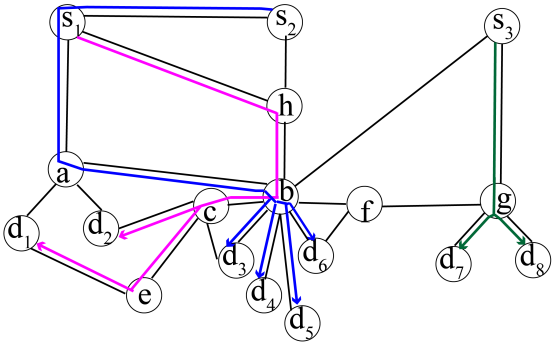
a) Configuration C_0 : pas d'interruption



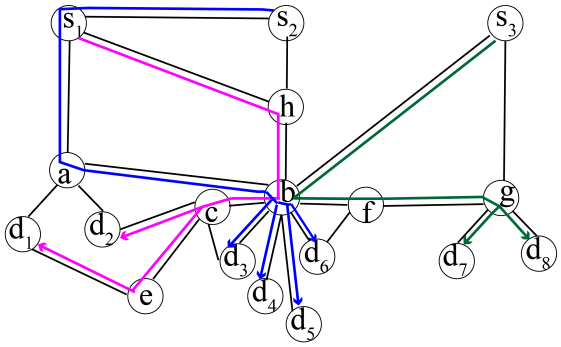
b) Configuration C_1 après suppression de l'arbre initial de m_1 : deux interruptions (d_1 et d_2)



c) Configuration C_2 après ajout de l'arbre final de m_2 et suppression de l'arbre initial m_2 : deux interruptions (d_1 et d_2)



d) Configuration C_3 après ajout de l'arbre final de m_1 : pas d'interruption



e) Configuration C_4 après ajout de l'arbre final de m_3 et suppression de l'arbre initial m_3 : pas d'interruption

FIGURE 5.3 – Séquence de configurations générée par LSRA

5.3.2 Méthode de reconfiguration réduisant les pénalités financières

5.3.2.1 Motivation

Dans la section 5.3.1, nous avons présenté une méthode de reconfiguration d'un ensemble de paires d'arbres mono-optiques ayant pour objectif la réduction du coût d'interruptions de flux (ou nombre total d'interruptions de flux). En réalité, l'opérateur réseau aura recours à une telle méthode dans l'un des deux cas ci-dessous :

— *Cas 1* : Aucun SLA n'implique des pénalités financières :
Autrement dit, les clients dont les connexions sont concernées par la reconfiguration n'induit aucune pénalité financière. Cependant, l'opérateur réalise la reconfiguration en réduisant le coût d'interruptions de flux pour ne pas trop impacter négativement l'activité de ses clients et les fidéliser. Il est évident que l'opérateur n'a pas besoin de réduire les pénalités financières puisqu'elles sont toutes nulles.

— *Cas 2* : Tous les SLAs impliquent des pénalités financières non nulles qui sont identiques :
Autrement dit, les clients dont les connexions sont concernées par la reconfiguration requiert la même pénalité financière par destination concernée par une interruption de flux. Dans ce cas, on peut se satisfaire uniquement d'une méthode de reconfiguration réduisant le coût d'interruptions de flux. En effet, comme la pénalité financière est la même pour les différentes connexions alors des Equations (5.1) et (5.3), il résulte que la pénalité financière totale causée par une configuration donnée est le produit de la pénalité financière par destination concernée par une interruption de flux et du nombre d'interruptions causées par la même configuration.

Dans les deux cas ci-dessus, les pénalités financières sont égales (nulles ou non nulles). Cependant, dans un troisième cas, la reconfiguration concerne des connexions auxquelles sont associées des pénalités différentes. En effet, certaines connexions clients peuvent être moins tolérantes aux interruptions que d'autres. Ainsi, aux connexions moins tolérantes sont associées des pénalités plus élevées que les connexions plus tolérantes aux interruptions de flux. En conséquence, les connexions clients ont des niveaux de priorité différents dans la politique de gestion du réseau. L'opérateur réseau menant une activité à but lucratif, ce dernier aura tendance dans

ce dernier cas de figure à réduire le coût en pénalités financières. Par ailleurs, dans le cas où les pénalités financières ne sont pas identiques, une méthode de reconfiguration réduisant les interruptions de flux ne réduit pas forcément les pénalités financières. Soit l'instance de problème de reconfiguration dans le cas où les pénalités financières sont différentes comme illustrée par la Figure 5.4. LSRA (voir Algorithme 5.1) qui est

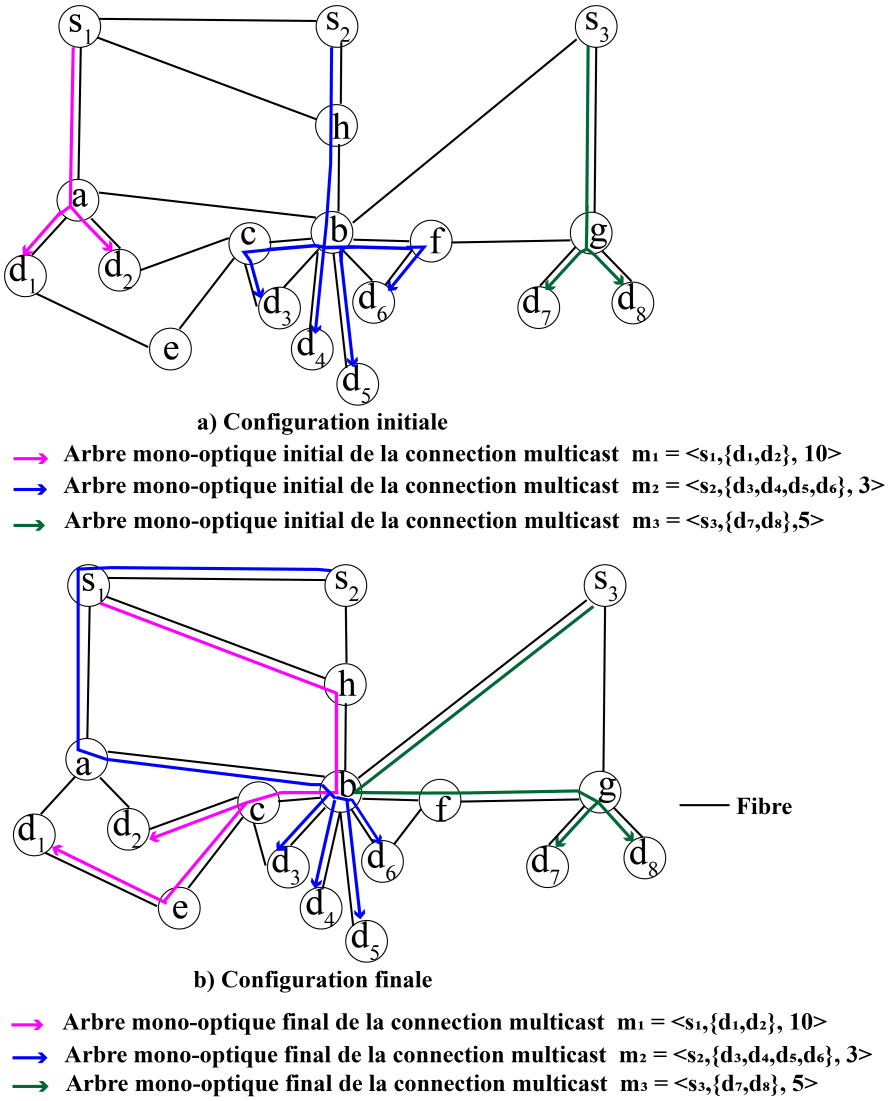


FIGURE 5.4 – Instance de problème avec des pénalités différentes

basé sur l'utilisation d'un ensemble de nœuds de rétroaction de faible coût d'interruptions va supprimer temporairement l'arbre de la connexion m_1. Cela permet de mieux réduire les interruptions de flux, car cela ne cause que deux interruptions de flux si

l'arbre de la connexion m_1 (car cette connexion contient deux nœuds destinations) est supprimé contre quatre interruptions de flux si l'arbre de la connexion m_2 (car cette connexion contient quatre nœuds destinations) est supprimé. Cependant, le coût en pénalités financières si l'arbre de la connexion m_1 (car cette connexion contient deux nœuds destinations) est supprimé vaut 2*10 UM soit 20 UM contre 4*3 UM soit 12 UM si l'arbre de la connexion m_2 (car cette connexion contient quatre nœuds destinations) est supprimé. Ainsi, un ensemble de faible coût d'interruptions n'est pas forcément un ensemble de faible coût en pénalités financières. Par conséquent, Bien que LSRA réduit mieux les interruptions de flux, il peut causer beaucoup de pénalités financières.

5.3.2.2 Présentation de la méthode

Il a été précédemment prouvé que la méthode de reconfiguration basée sur l'utilisation d'un ensemble de nœuds de rétroaction de faible coût d'interruptions ne permet pas de mieux réduire les pénalités financières. Par conséquent, nous utilisons un nouveau graphe orienté pondéré de dépendances qui diffère de la précédente par sa fonction poids. Ainsi, le nouveau graphe orienté pondéré de dépendances se construit comme suit :

— Les nœuds et les arcs de ce graphe sont créés suivant le procédé de construction du graphe orienté non pondéré de dépendances décrit au chapitre 3 (voir section 3.4.2).

— Un poids est attribué à chaque nœud selon une fonction de pondération. Formellement, le nouveau graphe orienté de dépendances qui est pondéré par les nœuds est désigné par $G_{p\prime}\left(V_{G_{p\prime}}, E_{G_{p\prime}}, p\prime\right)$ où $V_{G_{p\prime}}$ est l'ensemble des nœuds, $E_{G_{p\prime}}$ est l'ensemble des arcs et $p\prime$ est la fonction de pondération qui est définie comme suit :

$$\begin{aligned} p\prime : V_{G_{p\prime}} &\longrightarrow N^* \\ v &\mapsto p\prime(v) \end{aligned}$$

Soit $m_k =< s_k, D_k, PF_k >$ la connexion correspondante au nœud v du graphe $G_{p\prime}\left(V_{G_{p\prime}}, E_{G_{p\prime}}, p\prime\right)$. L'Equation (5.6) permet de calculer la valeur du poids de chaque sommet :

$$p'(v) = PF_k * |D_k| \tag{5.6}$$

où, v représente la connexion m_k dans le graphe $G_{p'}$.

Ainsi, la Figure 5.5 présente le nouveau graphe orienté pondéré de dépendances pour l'instance de problème de la Figure 5.4. L'algorithme de reconfiguration visant à réduire le coût en pénalités financières noté LSRA_2 diffère uniquement dans sa capacité à se baser sur un ensemble de noeuds de rétroaction de faible coût en pénalités financières qui réduit au mieux le coût d'interruptions de flux. Autrement dit, s'il faut choisir un ensemble parmi plusieurs ensembles de rétroaction de faible coût (identique) en pénalités financières alors LSRA_2 retient celui qui a le plus faible coût d'interruptions de flux. Ainsi, conformément à la Figure 5.5, l'ensemble de rétroaction de faible coût en pénalités financières correspond au singleton $\{m_2\}$.

FIGURE 5.5 – Graphe orienté pondéré de dépendances si les pénalités financières sont différentes

5.4 Evaluation de performance

Des simulations ont été menées pour comparer l'efficacité de l'algorithme LSRA qui reconfigure en réduisant les interruptions de flux et l'éfficacité de l'algorithme LSRA_2 qui reconfigure en réduisant les pénalités financières. Le langage Python est utilisé pour mener ces simulations sur un ordinateur portable équipé d'un processeur Intel Core i5-9300H ayant 2,40 GHz, 8 Go de RAM et muni d'un système d'exploitation Windows 10. La reconfiguration étant étudiée dans ce chapitre suivant deux objectifs différents, nous menons des simulations pour évaluer l'algorithme proposé pour chaque objectif de reconfiguration. Autrement dit, nous menons deux types de simulations : des simulations pour évaluer LSRA suivant l'objectif de réduction des interruptions de flux et des simulations pour évaluer LSRA_2 suivant l'objectif de réduction des pénalités financières. Ainsi, les critères d'évaluation lors de chaque type de simulations, les paramètres et les résultats de chaque type de simulations sont présentés dans les sous-sections suivantes.

5.4.1 Critères d'évaluation

5.4.1.1 Reconfiguration visant à réduire les interruptions de flux

Le premier critère (ou le critère principal) utilisé ici pour comparer notre algorithme (c'est-à-dire LSRA) avec l'algorithme (noté MFVSA) est le nombre total d'interruptions de flux vers les nœuds destinations de l'ensemble des connexions multicasts. Notez que ce critère est spécifié par l'Equation (5.2) lors de la formulation du problème de reconfiguration d'un ensemble de paires d'arbres mono-optiques (voir la section 5.2). Le second critère est le nombre d'étapes nécessaire pour migrer de la configuration initiale à la configuration finale. Dans ce qui suit, $SC =< C_0, C_1, \ldots, C_{f-1}, C_f >$ indique la séquence de configurations nécessaires pour faire migrer un flux optique de C_0 à C_f. Soit $nb_etapes\,(SC)$ le nombre d'étapes à rechercher. Ce second critère est exprimé par l'Equation (5.7) :

$$nb_etapes\,(SC) = |SC| - 1 \tag{5.7}$$

5.4.1.2 Reconfiguration visant à réduire les pénalités financières

Le premier critère (ou le critère principal) utilisé ici pour comparer l'algorithme LSRA avec LSRA_2 est le coût total des pénalités financières à payer par l'opérateur réseau. Notez que ce critère est spécifié par l'Equation (5.4) lors de la spécification du problème de reconfiguration d'un ensemble de paires d'arbres mono-optiques (voir la section (5.2). Un critère secondaire à savoir le nombre d'interruptions de flux qui est pris en compte lors des simulations qui concernent la reconfiguration visant à réduire les interruptions de flux est également analysé ici.

5.4.2 Description des simulations

5.4.2.1 Reconfiguration visant à réduire les interruptions de flux

Pour mener les simulations, nous avons utilisé des réseaux générés de façon aléatoire. Cela garantit que les résultats des simulations sont indépendants des caractéristiques de toute topologie de réseau. Nous utilisons le LSRA et le MFVSA sur un réseau test qui est généré de manière aléatoire et qui comporte 220 nœuds.

Pour générer aléatoirement le graphe du réseau, nous adoptons l'approche bien connue de Waxman [127] qui est appliquée au réseau WDM et expliquée de façon

détaillée dans [128]. Dans cette approche, les nœuds sont placés aléatoirement dans une grille 2D en générant des valeurs uniformément distribuées pour leurs coordonnées x et y, deux nœuds ne peuvent pas être placés sur le même point de la grille. Une arête entre deux nœuds u et v est ajoutée en utilisant la fonction de probabilité :

$$F = \lambda \, exp\left(-\frac{d\,(u,v)}{\gamma\delta}\right) \tag{5.8}$$

où $d\,(u,v)$ est la distance euclidienne entre les nœuds qui forment les nœuds d'extrémités de l'arête (u,v), δ est la distance euclidienne maximale entre deux nœuds quelconques, $0 \leq \lambda, \gamma \leq 1$. γ est la longueur moyenne d'une arête. λ est le degré de connexion moyen d'un nœud. Ainsi, plus λ augmente plus le nombre d'arêtes augmente, et plus γ augmente plus le rapport entre arêtes longues et arêtes courtes augmente. Dans les expériences, λ et γ sont fixés respectivement à 0.7 et 0.9 afin d'obtenir un réseau [5] suffisamment maillé pour nos expériences.

Le réseau WDM représenté par le graphe généré aléatoirement a été utilisé pour générer des exemples du problème de reconfiguration d'un ensemble de paires d'arbres mono-optiques. Nous considérons deux paramètres de simulations : le nombre de connexions multicasts $|M| = n(n_1 = 5, n_2 = 15, n_3 = 25)$ et la plage sur laquelle est tirée aléatoirement (loi uniforme) le nombre de nœuds destinations de toute connexion multicast : $\Delta D(\Delta D_1 = [2;8],\ \Delta D_2 = [9;15],\ \Delta D_3 = [16;22])$. En d'autres termes, une simulation consiste à prendre une valeur de la paire de paramètres $(n_i, \Delta D_i)$, à générer un nombre d'instances $NI(NI = 300)$ [6] du problème de reconfiguration d'un ensemble de paires d'arbres mono-optiques, à exécuter les algorithmes sur chacune de ces instances et à évaluer la moyenne de chacun des critères d'évaluation de la simulation. Le choix de $NI = 300$ a été inspirée de la méthode de simulations successives décrite dans [122] en prenant $NI_1 = 50$, $NI_2 = 100$, $NI_3 = 150$ jusqu'à $NI_6 = 300$ instances. En clair, nous avons calculé la moyenne (respectivement l'écart-type) pour un nombre varié d'exécutions d'une même simulation et constaté que l'écart entre deux moyennes (respectivement deux écarts-types) pour chaque critère d'évaluation devient négligeable lorsqu'on passe de $NI_5 = 250$ à $NI_6 = 300$. Soient

5. Le nombre élevé de nœuds (220) et d'arêtes (selon la valeur de λ) font qu'il n'est pas opportun de présenter le graphe du réseau ici.

6. Avec 300, les simulations prennent un peu plus d'une semaine sur notre machine et nous obtenons des résultats probants déjà avec ce nombre. Par conséquent, nous avons jugé bon de ne pas augmenter ce nombre.

$X(NI_k)$ la valeur d'une mesure X pour un nombre d'instances NI_k et $X(NI_{k+1})$ la valeur de la mesure X pour un nombre d'instances NI_{k+1}. Nous admettons que l'écart relatif entre $X(NI_k)$ et $X(NI_{k+1})$, définie par la quantité $\frac{|X(NI_{k+1})-X(NI_k)|}{X(NI_{k+1})} * 100$ est dit négligeable, si cette quantité est inférieure à 10%. Chaque instance du problème est générée comme suit :

— n_i connexions multicasts sont générées : Pour chaque connexion multicast un nœud source est choisi uniformément parmi l'ensemble des nœuds (c'est-à-dire dans [1 ;220]) et pour chaque connexion multicast, k (avec k choisi uniformément dans ΔD_i) destinations sont choisis uniformément dans l'ensemble des nœuds hormis la source ;

— La configuration initiale est construite : Pour chaque connexion multicast, l'arbre mono-optique initial est obtenu en utilisant l'algorithme de Dijkstra [120] et une longueur d'onde ;

— La configuration finale est construite : Pour chaque connexion multicast, l'arbre mono-optique final est obtenu en utilisant l'algorithme Prim [121] et une longueur d'onde telle que chaque connexion multicast dépend de deux autres connexions multicasts (soient $m_0 < s_0, D_0 >$ et $m_1 < s_1, D_1 >$) qui sont choisies aléatoirement selon le processus [7] présenté comme suit :

1. Rechercher un chemin (selon Dijkstra) seg_0 entre la source s_k de la connexion $m_k < s_k, D_k >$ et la source s_0 de la connexion m_0 dans le réseau tel que seg_0 ne partage aucun lien (canal) avec l'arbre initial de m_0 ;
2. Choisir aléatoirement un nœud n_0 de l'arbre initial de $m_0 < s_0, D_0 >$ et sélectionner le chemin seg_1 sur l'arbre initial qui relie s_0 à n_0 ;
3. Rechercher un chemin (selon Dijkstra) noté seg_2 reliant n_0 à la source s_1 de m_1 dans le réseau qui ne contient pas les antécédents de n_0 sur l'arbre initial de m_0 ;
4. Choisir aléatoirement un nœud n_1 de l'arbre initial de $m_1 < s_1, D_1 >$ et extraire le chemin seg_3 sur l'arbre initial qui relie s_1 à n_1 ;
5. Calculer l'arbre $T_f^{'}$ enraciné en n_1 et couvrant D_k à l'aide de l'algorithme de Prim tronqué autour de n_1 et des éléments de D_k.

7. Ce processus permet d'obtenir un arbre final approprié pour chaque connexion à condition de régler expérimentalement les paramètres (de Waxman) de construction du graphe afin que ce dernier soit suffisamment maillé : ce qui est notre cas ici.

L'arbre final de m_k est formé par l'union des canaux des segments (c'est-à-dire seg_0, seg_1, seg_2 et seg_3) obtenus aux étapes 1 à 4 puis de l'arbre obtenu (c'est-à-dire $T_f^{'}$) à l'étape 5 du processus précédemment présenté. En clair, pour toute connexion $m_k < s_k, D_k >$ on choisit aléatoirement deux connexions $m_0 < s_0, D_0 >$ et $m_1(s_1, D_1)$ dont l'arbre final doit dépendre. Puis on construit l'arbre final de m_k de sorte qu'elle inclut un segment (séquence de canaux) de l'arbre initial de m_0 et un segment de l'arbre initial de m_1. L'utilisation de l'algorithme Prim tronqué autour des nœuds destinations intervient pour compléter la construction de l'arbre final.

Les deux configurations (initiale et finale) sont prises comme entrée par notre algorithme (c'est-à-dire LSRA) et MFVSA.

5.4.2.2 Reconfiguration visant à réduire les pénalités financières

Lors des simulations relatives à la reconfiguration visant à réduire les pénalités financières, le procédé de génération de graphe aléatoire utilisé est celui utilisé par les simulations relatives à la reconfiguration visant à réduire les interruptions de flux.

Le réseau WDM représenté par le graphe généré aléatoirement est utilisé pour générer des instances du problème de reconfiguration d'un ensemble de paires d'arbres mono-optiques. Nous supposons ici que le nombre $|M|$ de connexions multicasts concernées par la reconfiguration est égal à 25. La plage du nombre de nœuds destinations ΔD de toute connexion multicast prend la même valeur que dans le cas de la reconfiguration visant à réduire les interruptions de flux (c'est-à-dire $\Delta D_1 = [2;8]$, $\Delta D_2 = [9;15]$, $\Delta D_3 = [16;22]$). Un autre paramètre de simulation est la pénalité financière PF associée à l'interruption du flux vers un nœud destination d'une connexion donnée. On admet que plus PF est élevée plus la connexion associée à PF a un niveau de priorité élevé du point de vue de l'opérateur réseau. Nous fixons deux valeurs possibles pour PF à savoir 1 UM, 5 UM. En clair, si la pénalité financière associée à l'interruption du flux vers un nœud destination d'une connexion donnée vaut 1 UM alors cette connexion est qualifiée de faiblement prioritaire. Dans le cas où la pénalité est de 5 UM, on parle de connexion fortement prioritaire. On choisit deux distributions possibles des différents types de connexions dans l'ensemble des connexions :

— **Distribution à connexions fortement prioritaires majoritaires** : 80% du

nombre total des connexions (soit 20 connexions ici) sont des connexions fortement prioritaires tandis que 20% du nombre total des connexions (soit 5 connexions ici) sont des connexions faiblement prioritaires.

— **Distribution à connexions fortement prioritaires minoritaires** : 20% du nombre total de connexions (soit 5 connexions ici) sont des connexions fortement prioritaires tandis que 80% du nombre total des connexions (soit 20 connexions ici) sont des connexions faiblement prioritaires.

Pour chaque type de distribution, on mène une simulation pour chaque valeur de plage du nombre de nœuds destinations ΔD_i en générant 300 instances[8] (comme dans la section 5.4.2.1) du problème de reconfiguration d'un ensemble de paires d'arbres mono-optiques. Chaque instance du problème est générée comme suit :

— Vingt-cinq (25) connexions multicasts sont générées : 25 nœuds sources sont choisis uniformément dans [1 ;220] et pour chaque connexion multicast, le nombre de destinations est choisi aléatoirement dans ΔD_i.

— Le type de distribution considéré (c'est-à-dire à connexions fortement prioritaires minoritaires ou à connexions fortement prioritaires majoritaires) est créé en affectant aléatoirement chacune des 25 connexions à un type de connexion.

— La configuration initiale et la configuration finale sont construites comme dans le cas de la reconfiguration visant à réduire les interruptions de flux (voir section 5.4.2.1). Ces deux éléments sont pris comme entrée par LSRA et LSRA_2.

A la fin de chaque simulation, la moyenne des valeurs des différents critères d'évaluation pour chacun des deux algorithmes est retenue.

5.4.3 Analyse des résultats

5.4.3.1 Reconfiguration visant à réduire les interruptions de flux

Dans cette sous-section, nous comparons les performances de l'approche LSRA que nous proposons aux performances de l'approche MFVSA. Nous rappelons que, pour chaque valeur de la plage du nombre de nœuds destinations : $\Delta D(\Delta D_1 = [2;8], \Delta D_2 = [9;15], \Delta D_3 = [16;22])$ de toute connexion multicast, le nombre de

8. Avec 300, les simulations prennent un peu plus d'une semaine sur notre machine et nous obtenons des résultats probants déjà avec ce nombre. Par conséquent, nous avons jugé bon de ne pas augmenter ce nombre.

connexions multicasts $|M|$ est successivement fixé à 5, 15 et 25. Pour obtenir des résultats représentatifs, pour chaque valeur de $|M|$, 300 instances de problème de reconfiguration ont été générées, puis la moyenne des différents critères a été calculée. Ainsi, les Figures 5.6 à 5.8 montrent la moyenne du nombre d'interruptions de flux en fonction des différentes valeurs du nombre de connexions multicasts $|M|$ (5 , 15 et 25) lorsque ΔD prend respectivement les valeurs $\Delta D_1 = [2;8], \Delta D_2 = [9;15], \Delta D_3 = [16;22]$.

Des Figures 5.6 à 5.8, les observations suivantes sont faites :

— Quelle que soit la valeur de ΔD, notre algorithme (c'est-à-dire LSRA) produit moins d'interruptions de flux que MFVSA. La raison est que LSRA supprime temporairement les arbres mono-optiques d'un ensemble de nœuds de rétroaction de faible coût d'interruptions (ou nombre d'interruptions) tandis que MFVSA supprime temporairement les arbres d'un ensemble de nœuds de rétroaction de cardinalité minimum. Ainsi, le fait d'être un ensemble de cardinalité minimum ne signifie pas que cet ensemble est un ensemble de faible coût d'interruptions de flux dans le cas de la multi-reconfiguration du routage multicast.

— Pour chaque valeur de ΔD, le nombre d'interruptions de flux augmente avec le nombre de connexions $|M|$. Cette observation s'explique par deux raisons : a) Un nombre plus élevé de connexions implique une valeur élevée du coût de l'ensemble de nœuds de rétroaction choisi. En effet, plus de connexions signifie qu'il y a plus de dépendances et donc potentiellement plus de cycles dans le graphe de dépendances ; b) L'Equation (5.2) montre que le nombre d'interruptions de flux d'une configuration est la somme des nombres d'interruptions de flux pour ces différentes connexions. Par conséquent, plus de connexions impliquent logiquement plus d'interruptions de flux.

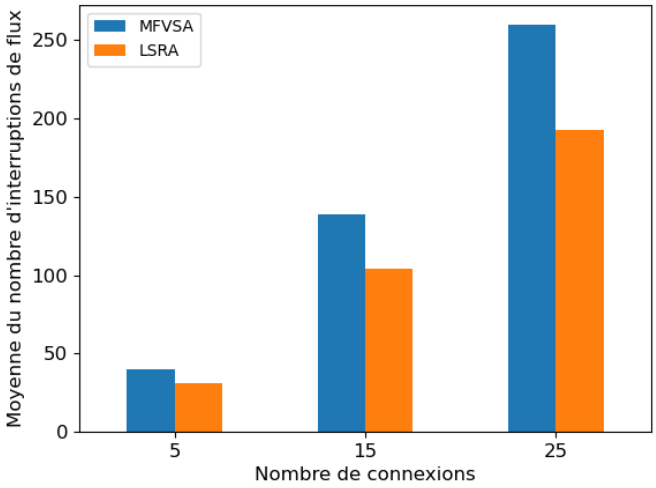


FIGURE 5.6 – Moyenne du nombre d'interruptions produite par les deux approches lorsque la plage du nombre de nœuds destinations est [2 ;8]

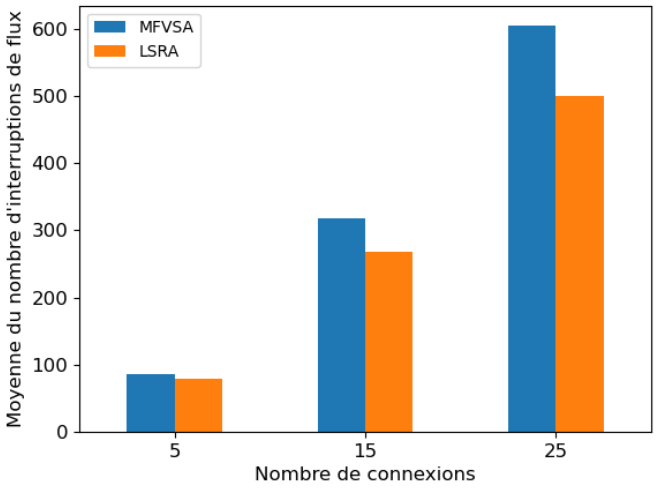


FIGURE 5.7 – Moyenne du nombre d'interruptions produite par les deux approches lorsque la plage du nombre de nœuds destinations est [9 ;15]

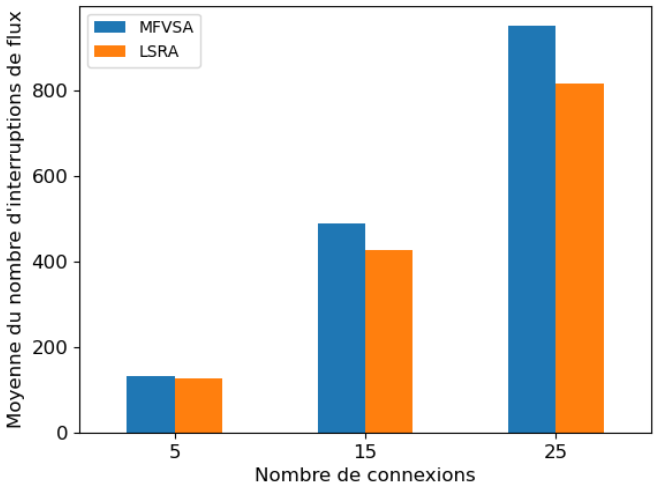


FIGURE 5.8 – Moyenne du nombre d'interruptions produite par les deux approches lorsque la plage du nombre de nœuds destinations est [16 ;22]

— Pour un nombre donné de connexions multicasts, le nombre d'interruptions de flux augmente au fur et à mesure que le nombre de nœuds destinations augmente. Ceci est cohérent avec les Equations 5.1, et 5.2, qui sont utilisées pour déterminer le nombre d'interruptions de flux.

Pour ce qui est du nombre d'étapes nécessaires pour migrer le flux optique de la configuration initiale à la configuration finale, les Figures 5.9 à 5.11 montrent que quelle que soit la valeur de la paire de paramètres de simulation utilisée, notre approche LSRA nécessite moins d'étapes que l'approche MFVSA. Cela s'explique par le fait que LSRA établit l'arbre final de chaque connexion de MCFVS devenu indépendante juste après la reconfiguration de connexions n'appartenant à MCFVS (voir de la ligne 17 à la ligne 21 de Algorithme 5.1) tandis que MFVSA reconfigure complètement (c'est-à-dire supprime l'arbre initial et établit l'arbre final) de toutes les connexions qui n'appartiennent pas à MFVS avant d'établir l'arbre final de chaque connexion appartenant à MFVS.

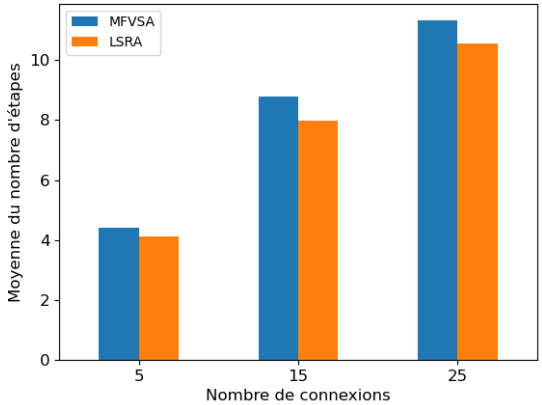


FIGURE 5.9 – Moyenne du nombre d'étapes produite par les deux approches lorsque la plage du nombre de nœuds destinations est [2 ;8]

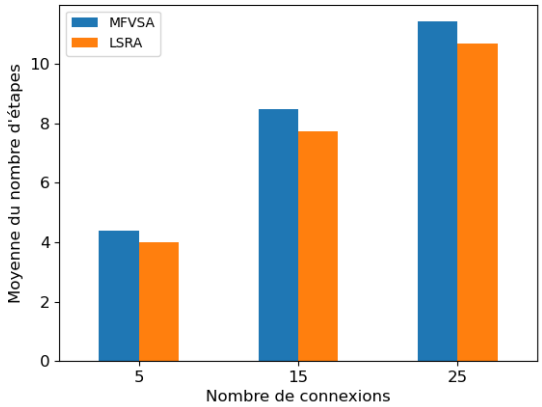


FIGURE 5.10 – Moyenne du nombre d'étapes produite par les deux approches lorsque la plage du nombre de nœuds destinations est [9 ;15]

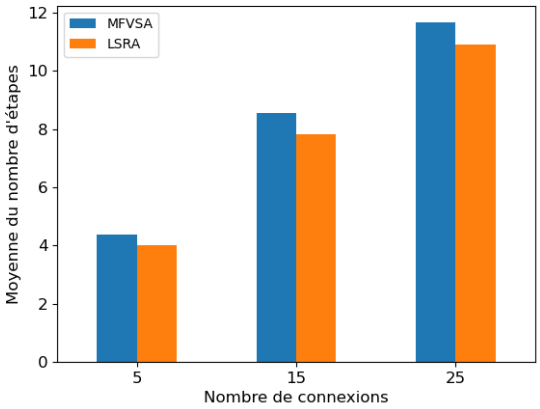


FIGURE 5.11 – Moyenne du nombre d'étapes produite par les deux approches lorsque la plage du nombre de nœuds destinations est [16 ;22]

5.4.3.2 Reconfiguration visant à réduire les pénalités financières

Dans cette sous-section, nous comparons les performances de l'approche LSRA (ayant pour objectif principal de réduire les interruptions de flux) et l'approche LSRA_2 (ayant pour objectif principal de réduire les pénalités financières). Nous rappelons que, nous avons établi vingt-cinq (25) connexions et se basant sur deux niveaux de priorités (priorité faible et prioritaire forte), nous les avons distribués suivant deux types de distributions (à connexions fortement prioritaires majoritaires et a connexions fortement prioritaires minoritaires). Pour chaque type de distribution, nous avons successivement considéré $\Delta D_1 = [2;8]$, $\Delta D_2 = [9;15]$ et $\Delta D_3 = [16;22]$ comme la plage du nombre de nœuds destinations de toute connexion multicast. Pour obtenir des résultats représentatifs, 300 instances de problème de reconfiguration ont été générées, puis la moyenne du nombre total d'interruptions de flux et la moyenne des pénalités financières ont été calculées. Dans un premier temps, nous analysons les résultats qui concernent les pénalités financières et puis dans un second temps ceux qui concernent le nombre total d'interruptions de flux.

Les Figures 5.12 et 5.13 montrent pour chaque type de distribution considéré, la moyenne des pénalités financières lorsque ΔD prend respectivement les valeurs $\Delta D_1 = [2;8]$, $\Delta D_2 = [9;15]$ et $\Delta D_3 = [16;22]$. Des Figures 5.12 et 5.13, les observations suivantes sont faites :

— Quel que soit le type de distribution considéré, l'algorithme LSRA_2 produit

moins de pénalités financières que LSRA pour chacune des plages de nombre de nœuds destinations. La raison est que LSRA supprime temporairement les arbres mono-optiques d'un ensemble de nœuds de rétroaction de faible coût d'interruptions tandis que LSRA_2 supprime temporairement les arbres mono-optiques d'un ensemble de nœuds de rétroaction de faible coût en pénalités financières. En effet, plusieurs nœuds appartenant à un ensemble de nœuds de rétroaction de faible coût d'interruptions peuvent être des connexions fortement prioritaires. Ainsi, le coût en pénalités financières de ce dernier ensemble est supérieur ou égal à celui d'un ensemble de nœuds de rétroaction de faible coût en pénalités financières. Autrement dit, le fait d'être un ensemble de nœuds de rétroaction de faible coût d'interruptions de flux ne signifie pas que cet ensemble est un ensemble de nœuds de rétroaction de faible coût en pénalités financières.

— Pour chaque type de distribution, les pénalités financières augmentent avec la plage du nombre de nœuds destinations considérée. Autrement dit, plus il y a de nombre de nœuds destinations plus les pénalités financières sont importantes. Cela est cohérent avec l'Equation (5.4).

— Les pénalités financières sont plus grandes dans le cas d'une distribution à connexions fortement prioritaires majoritaires. Cette observation s'explique par le fait que dans le cadre d'une distribution à connexions fortement prioritaires majoritaires, on a un nombre plus élevé de connexions à forte priorité que de connexions à faible priorité. Ainsi, l'ensemble de nœuds de rétroaction de faible coût en pénalités financières est susceptible de contenir des connexions à forte priorité auxquelles sont associées les pénalités les plus importantes.

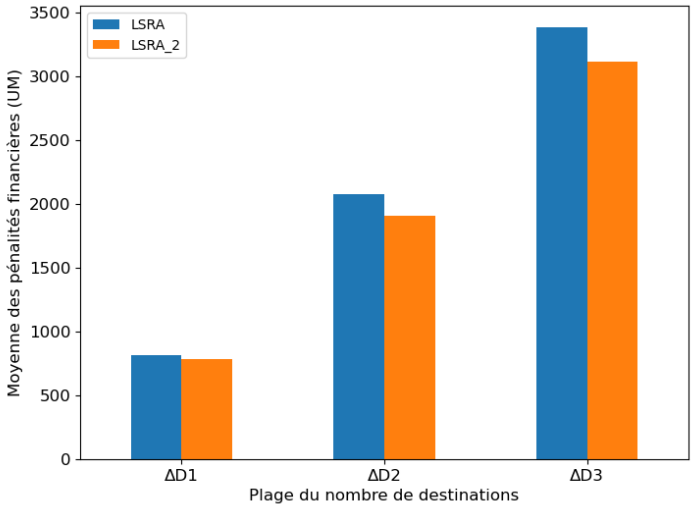


FIGURE 5.12 – Moyenne des pénalités financières produite par les deux approches dans le cadre d'une distribution à connexions fortement prioritaires majoritaires

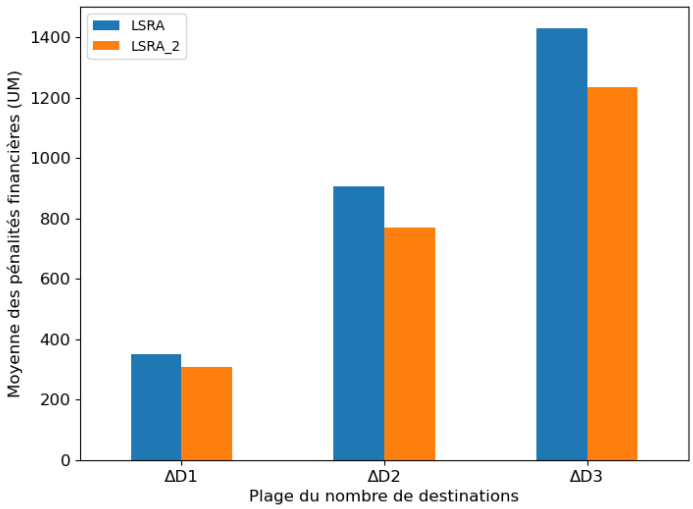


FIGURE 5.13 – Moyenne des pénalités financières produite par les deux approches dans le cadre d'une distribution à connexions fortement prioritaires minoritaires

Les Figures 5.14 et 5.15 montrent pour chaque type de distribution considéré, la moyenne des interruptions de flux lorsque ΔD prend respectivement les valeurs $\Delta D_1 = [2;8]$, $\Delta D_2 = [9;15]$ et $\Delta D_3 = [16;22]$. Des Figures 5.14 et 5.15, les observations suivantes sont faites :

— Quelle que soit le type de distribution considéré, l'algorithme LSRA_2 produit plus d'interruptions de flux que LSRA. La raison est que LSRA supprime temporairement les arbres mono-optiques d'un ensemble de nœuds de rétroaction de faible coût d'interruptions tandis que LSRA_2 supprime temporairement un ensemble de nœuds de rétroaction de faible coût en pénalités financières.

— L'écart entre les interruptions de flux causées par LSRA_2 celles causées par LSRA est moins important que l'écart entre les pénalités financières (voir Figures 5.12 et 5.13) causées par LSRA_2 et celles causées par LSRA. Cela s'explique par le fait que lorsqu'on a plusieurs ensembles de rétroaction de faible coût en pénalités financières c'est celui qui a le plus petit nombre d'interruptions de flux qui est retenu.

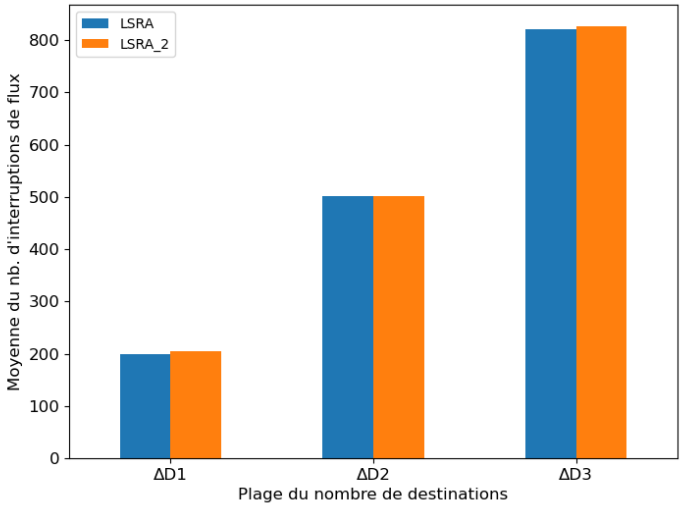


FIGURE 5.14 – Moyenne du nombre d'interruptions produite par les deux approches dans le cadre d'une distribution à connexions fortement prioritaires majoritaires

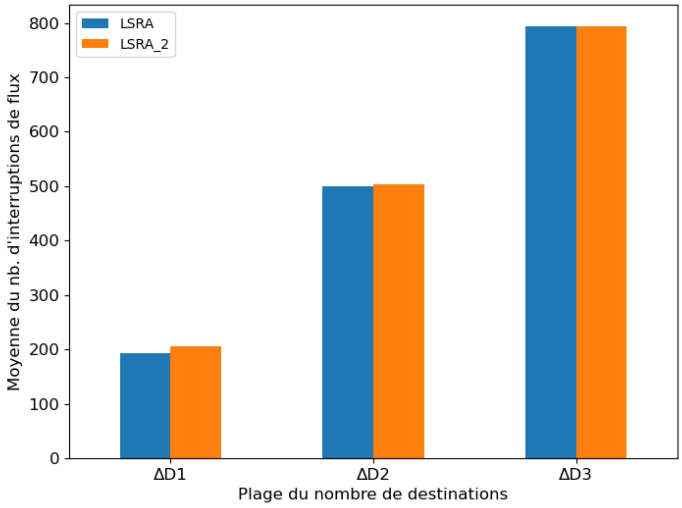


FIGURE 5.15 – Moyenne du nombre d'interruptions produite par les deux approches dans le cadre d'une distribution à connexions fortement prioritaires majoritaires

5.5 Conclusion

Dans ce chapitre, nous avons abordé le problème de reconfiguration d'un ensemble de paires d'arbres mono-optiques qui est relatif à un ensemble de connexions multicasts. Il s'agit ici de trouver la séquence de configurations qui permet de migrer le flux optique de la configuration initiale vers la configuration finale. Cette séquence de configurations doit soit causer un faible nombre d'interruptions de flux soit causer de faibles pénalités financières.

Lorsque la même pénalité financière (nulle ou non nulle) est associée à toutes les connexions concernées par la reconfiguration, alors il convient de s'intéresser uniquement à la réduction des interruptions de flux, car l'opérateur associe la même priorité à toutes ces connexions dans la gestion du réseau. L'examen de la littérature nous a permis de savoir que l'approche MFVSA suivie par les solutions proposées est adaptée au problème de reconfiguration d'un ensemble de paires de chemins (unicasts) mono-optiques. Cependant cette approche ne garantit pas une solution efficace au problème de reconfiguration d'un ensemble de paires d'arbres mono-optiques. En effet, cette approche s'appuie sur la recherche d'un ensemble de nœuds de rétroaction de cardinalité minimum afin de surmonter les situations de blocage dans le proces-

sus de reconfiguration. Cependant, un tel ensemble n'implique pas forcément une séquence de configurations causant un faible nombre d'interruptions de flux. Par conséquent, nous avons proposé une nouvelle approche appelée LSRA qui s'appuie sur la recherche d'un ensemble de nœuds (ou connexions) de rétroaction de faible coût (ou nombre) d'interruptions de flux. Pour ce faire, nous avons modélisé à l'aide d'un graphe orienté pondéré par les nœuds, les dépendances entre les connexions concernées par le problème de reconfiguration d'un ensemble de paires d'arbres mono-optiques ayant pour objectif la réduction des interruptions de flux. LSRA supprime temporairement les arbres des connexions appartenant à l'ensemble de nœuds de rétroaction de faible coût d'interruptions afin de surmonter les situations de blocage déduites du graphe de dépendances. Les simulations menées ont confirmé le fait que l'approche LSRA (notre approche) cause moins d'interruptions de flux que l'approche MFVSA. De plus, LSRA nécessite moins d'étapes pour reconfigurer un ensemble de paires d'arbres mono-optiques que MFVSA. Ainsi, LSRA est plus adaptée pour reconfigurer un ensemble de paires d'arbres mono-optiques que l'approche MFVSA.

Lorsque différentes pénalités financières sont associées aux connexions concernées par la reconfiguration, alors l'opérateur réseau a tendance à perturber le moins possibles les connexions engendrant des pénalités financières élevées afin de réduire ses pénalités financières. Ainsi, il convient de s'intéresser prioritairement à la réduction des pénalités financières. LSRA supprime temporairement les arbres des connexions appartenant à l'ensemble de nœuds (ou de connexions) de rétroaction de faible coût d'interruptions afin de surmonter les situations de blocage. Cependant, les connexions de cet ensemble peuvent être associées à des pénalités financières élevées. Par conséquent, nous avons proposée LSRA_2 qui permet de supprimer temporairement les arbres dont les connexions appartiennent à un ensemble de connexions de faible coût en pénalités afin de surmonter les situations de blocage.

CHAPITRE 6

CONCLUSION GÉNÉRALE

Sommaire

6.1 Resumé des travaux 142
6.2 Perspectives 143

6.1 Resumé des travaux

La reconfiguration du routage est une tâche importante et critique dans les réseaux optiques WDM. En effet, la reconfiguration permet à l'opérateur réseau d'optimiser l'utilisation des ressources du réseau. Cependant, si elle n'est pas effectuée avec beaucoup de précautions alors elle peut entraîner de nombreuses (et longues) interruptions de flux qui ne sont pas sans conséquence sur la qualité du service fourni par l'opérateur réseau à ses clients. Ces interruptions de flux sont à la fois dommageables pour le client vu que la dégradation de la qualité du service peut avoir de graves répercussions sur l'activité du client et pour l'opérateur vu qu'elles peuvent entraîner (selon les SLAs) des contreparties (ou pénalités) financières que l'opérateur de réseau doit souvent verser aux clients sinistrés.

Dans cette thèse, il a été question de traiter certains problèmes liés à la reconfiguration du routage multicast dans les réseaux optiques WDM ayant la propriété de conversion partielle de longueur d'onde. En clair, nous avons traité d'un premier problème (ou problème de reconfiguration d'une paire d'arbres mono-optiques) qui consiste à migrer le flux optique d'un arbre mono-optique initial vers un arbre mono-optique final sans interruption du flux tout en réduisant le nombre de fois où des longueurs d'onde additionnelles sont utilisées. Le deuxième problème (ou problème de reconfiguration d'un ensemble de plusieurs paires d'arbres mono-optiques) traité ici consiste à migrer le flux optique d'un ensemble d'arbres mono-optiques vers un autre ensemble d'arbres mono-optiques soit en réduisant les interruptions de flux soit en réduisant les pénalités

financières.

Les méthodes que nous avons proposées pour résoudre le premier problème procèdent par une reconfiguration par paire de sous sous-arbres en distinguant la reconfiguration des paires de sous-arbres à liens disjoints de la reconfiguration des paires de sous-arbres à liens partagés. En effet, les paires de sous-arbres à liens partagés nécessitent d'utiliser une longueur d'onde additionnelle ce qui n'est pas le cas des paires de sous-arbres à liens disjoints. Il a été montré que les paires de sous-arbres d'une même catégorie peuvent être reconfigurées en parallèle et que les différentes catégories possèdent la propriété de non-interruption du flux. Ainsi, nos méthodes permettent de résoudre le premier problème contrairement aux méthodes de la littérature qui peuvent entraîner des interruptions de flux. En effet, les méthodes de la littérature sont basées sur une approche de reconfiguration par paire de branches avec aucune garantie de capacité de conversion de longueur d'onde pour le nœud de basculement dans un réseau WDM ayant la propriété de conversion partielle de longueur d'onde.

Concernant le second problème, lorsque l'objectif visé est la réduction des interruptions de flux, nous proposons la solution LSRA qui supprime temporairement les arbres dont les connexions appartiennent à un ensemble de nœuds (ou de connexions) de rétroaction de faible coût d'interruptions. En outre, lorsque l'objectif visé est la réduction des pénalités financières, nous proposons la solution LSRA_2 qui supprime temporairement les arbres dont les connexions appartiennent à un ensemble de nœuds (ou de connexions) de rétroaction de faible coût en pénalités financières. La Figure 6.1 présente les spécificités des deux problèmes traités dans cette thèse ainsi que les solutions apportées pour chacun d'entre eux.

6.2 Perspectives

Le problème de mono-reconfiguration du routage multicast étudié dans cette thèse concerne la reconfiguration d'une paire d'arbres mono-optiques. Il s'agit de produire une séquence de configurations qui n'interrompt pas le flux tout en utilisant un faible nombre de fois des longueurs d'onde additionnelles. Nos méthodes proposées bien qu'efficaces sont toutes basées sur l'hypothèse qu'une longueur d'onde additionnelle est toujours disponible pour la reconfiguration éventuellement des paires de sous-arbres à liens partagés. Cette hypothèse ne peut pas être toujours garantie en pratique. Il convient donc de proposer des méthodes permettant un bon compromis entre

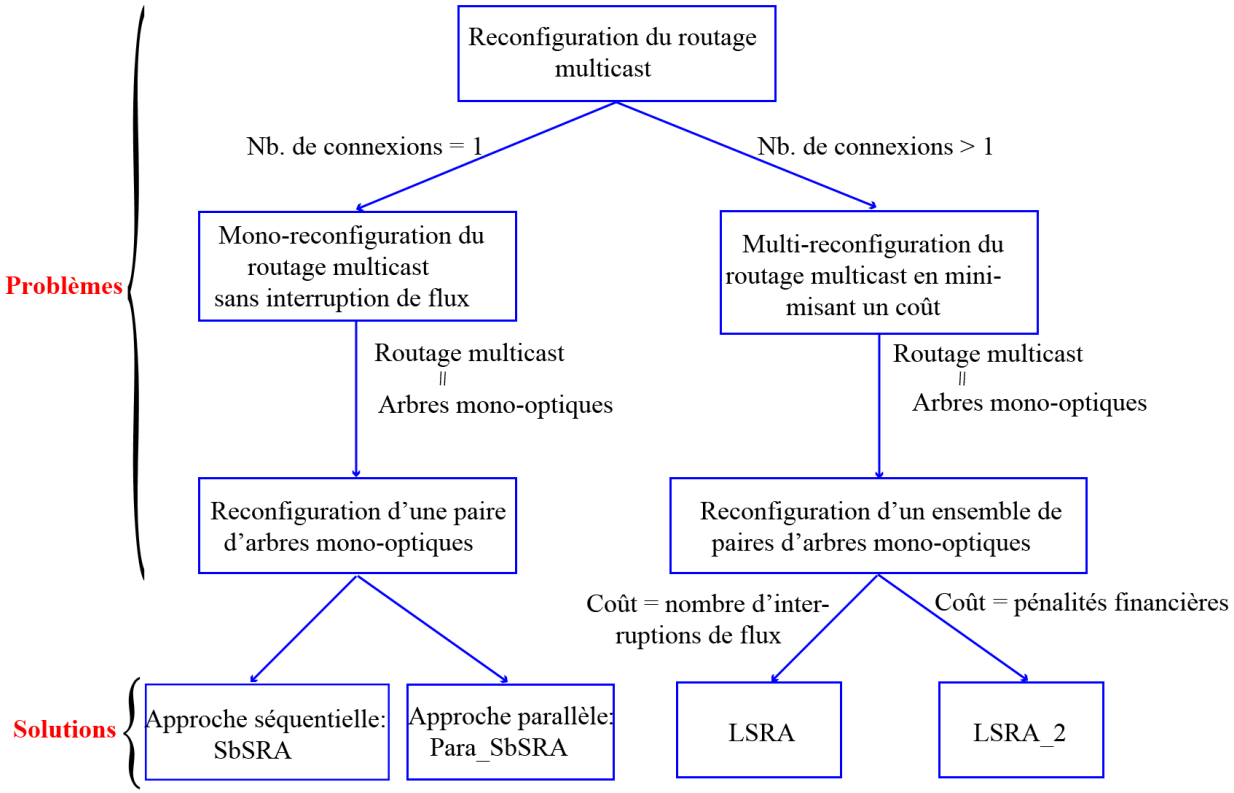


FIGURE 6.1 – Classification des problèmes abordés et les solutions proposées

les interruptions de flux et le coût des longueurs d'onde additionnelles. En outre, l'étude de l'impact du nombre de liens partagés sur la performance de nos méthodes et de celles de la littérature existante mérite d'être effectuée . Un autre problème de mono-reconfiguration mérite d'être étudié. Il s'agit de la reconfiguration d'une paire d'arbres multi-optiques (arbre multi-optique initial, arbre multi-optique final) qui peut également s'avérer très difficile dans certains cas. En effet, bien que certains nœuds du réseau optique WDM ayant la propriété de conversion partielle de longueur d'onde soient munis de convertisseurs de longueur d'onde, il faut savoir que le nombre de convertisseurs pour un nœud du réseau est limité en pratique. Il peut donc avoir un problème de concurrence aux convertisseurs de longueur d'onde entre l'arbre multi-optique initial et l'arbre multi-optique final qui n'est pas facile à modéliser. Ainsi, une première perspective de notre travail serait de proposer un outil de modélisation de la concurrence

d'accès aux convertisseurs de longueur d'onde entre les arbres multi-optiques. Cet outil de modélisation permettrait de réduire les interruptions lors de la reconfiguration d'une paire d'arbres multi-optiques.

Par ailleurs, le problème de multi-reconfiguration traité dans cette thèse est le problème de reconfiguration d'un ensemble de plusieurs paires d'arbres mono-optiques. L'objectif ici est soit de réduire les interruptions de flux soit les pénalités financières en priorité. Dans le dernier cas, les connexions avec de faibles pénalités pour l'opérateur réseau sont touchées par des interruptions de flux au détriment des connexions avec de fortes pénalités pour l'opérateur réseau. Ainsi, les clients ayant des connexions avec des faibles pénalités pour l'opérateur réseau pourraient être tentés de changer d'opérateur réseau. Pour pallier cela, il serait intéressant d'étudier la possibilité d'un compromis entre les interruptions de flux et les pénalités financières. En outre, les méthodes proposées dans cette thèse sont une extension de celles proposées dans la littérature pour traiter le problème de reconfiguration d'un ensemble de plusieurs paires de chemins. C'est pourquoi, nous avons supposé que tout canal optique à l'origine d'une dépendance (entre deux connexions) est un canal menant à toutes destinations de ces connexions. Il serait judicieux d'envisager le cas où cette hypothèse n'est pas vérifiée.

Dans cette thèse, nous supposons que tous les nœuds sont équipés de duplicateurs optiques ou *light splitter*. Cependant, bon nombre d'études apporte des solutions au problème de routage multicast en supposant qu'un nombre restreint de nœuds sont dotés de duplicateurs optiques. Ces solutions peuvent être soit des forêts optiques (ou *light-forests*) soit des hiérarchies optiques (ou *light-hierarchies*) comme mentionnées au chapitre 2 (voir section 2.6.3 et 2.6.4). Ainsi, il convient d'étudier les problèmes de mono-reconfiguration du routage et de multi-reconfiguration du routage en supposant que les entrées de ces problèmes sont des forêts optiques, hiérarchies optiques. De plus, des parcours optiques [129] (ou *light-trails*) peuvent être envisagée comme solution du routage multicast si on suppose qu'aucun nœud n'est muni de duplicateurs optiques. Ainsi, la mono/multi-reconfiguration du routage multicast pourrait être étudié en supposant que les entrées du problème sont des parcours optiques. Autrement dit, il convient également d'étudier les problèmes de mono-reconfiguration du routage et de multi-reconfiguration du routage en supposant que les entrées de ces problèmes sont des parcours optiques.

Malgré des études approfondies sur la reconfiguration du réseau WDM, très peu

d'études ont abordé les problèmes et les techniques liés aux ajustements des couches supérieures (par exemple IP) lors des reconfigurations (plus précisément des migrations) du routage (unicast et multicast) dans les réseaux optiques WDM. Autrement dit, les évaluations des performances de toutes les méthodes proposées ne sont pas effectuées directement du point de vue des utilisateurs finaux. La complexité des interactions entre les multiples couches de réseau rend des évaluations complètes de la reconfiguration du routage dans les réseaux WDM un véritable défi. L'avènement des réseaux optiques définis par logiciels ou Software-Defined Optical Networks [130] (SDON) n'est pas sans avantage. En effet, un SDON a comme avantage d'apporter plus de flexibilité dans la programmation des tâches d'administration des réseaux optiques en général. Elle permet de mieux prendre en charge l'administration des ressources du réseau. Une perspective de travail serait donc d'effectuer des évaluations de performances plus en profondeur de nos méthodes (et des méthodes antérieures aux nôtres) de reconfiguration dans un SDON.

RÉFÉRENCES

[1] C. Siva Ram MURTHY et Gurusamy MOHAN. *WDM Optical Networks : Concepts, Design and Algorithms.* PHI, 2003.

[2] Biswanath MUKHERJEE. *Optical WDM Networks*. Springer, 2006. DOI : 10.1007/0-387-29188-1.

[3] Zhi shu SHEN, Hiroshi HASEGAWA et Ken ichi SATO. « Effectiveness of Wavelength/Waveband Conversion and Its Allowable Cost Bound for Hierarchical Optical Path Networks ». In : *Journal of Optical Communications and Networking* 5.11 (2013), p. 1262. DOI : 10.1364/jocn.5.001262.

[4] Xi WANG, Inwoong KIM, Qiong ZHANG, Paparao PALACHARLA et Tadashi IKEUCHI. « Efficient All-Optical Wavelength Converter Placement and Wavelength Assignment in Optical Networks ». In : *Optical Fiber Communication Conference*. OSA, 2016. DOI : 10.1364/ofc.2016.w2a.52.

[5] Kiyo ISHII, Takashi INOUE, Inwoong KIM, Xi WANG, Hung Nguyen TAN, Qiong ZHANG, Tadashi IKEUCHI et Shu NAMIKI. « Analysis and Demonstration of Network Utilization Improvement Through Format-Agnostic Multi-Channel Wavelength Converters ». In : *Journal of Optical Communications and Networking* 10.2 (2018), A165. DOI : 10.1364/jocn.10.00a165.

[6] S. J. Ben YOO. « Wavelength conversion technologies for WDM network applications ». In : *Journal of Lightwave Technology* 14.6 (1996), p. 955-966. DOI : 10.1109/50.511595.

[7] Bijoy Chand CHATTERJEE et Eiji OKI. *Elastic Optical Networks : Fundamentals, Design, Control, and Management*. CRC Press, 2020. DOI : 10.1201/9780429465284.

[8] Shrinivas PETALE et T. JAISINGH. « Optimal of wavelength converter deployment in WDM optical networks ». In : *2018 3rd International Conference on Microwave and Photonics (ICMAP)*. IEEE, 2018. DOI : 10.1109/icmap.2018.8354558.

[9] Diego P. PINTO-ROA, Carlos A. BRIZUELA et Benjamín BARÁN. « Multi-objective routing and wavelength converter allocation under uncertain traffic ». In : *Optical Switching and Networking* 16 (2015), p. 1-20. DOI : 10.1016/j.osn.2014.10.001.

[10] Amiyne ZAKOUNI, Jiawei LUO et Fouad KHARROUBI. « Genetic algorithm and tabu search algorithm for solving the static manycast RWA problem in optical networks ». In : *Journal of Combinatorial Optimization* 33.2 (2016), p. 726-741. DOI : 10.1007/s10878-016-0002-3.

[11] S. SUBRAMANIAM, M. AZIZOGLU et A. K. SOMANI. « On optimal converter placement in wavelength-routed networks ». In : *IEEE/ACM Transactions on Networking* 7.5 (1999), p. 754-766. DOI : 10.1109/90.803388.

[12] Sumitha GEORGE et Vinod CHANDRA. « Algorithms for Placement of Wavelength Converters in Optical Networks ». In : *2008 International Conference on Signal Processing, Communications and Networking*. IEEE, 2008. DOI : 10.1109/icscn.2008.4447205.

[13] P. C. SHRUTHI et T. S. INDUMATHI. « Wavelength converter placement optimization in WDM network ». In : *2015 International Conference on Communications and Signal Processing (ICCSP)*. IEEE, 2015. DOI : 10.1109/iccsp.2015.7322843.

[14] Jaisingh THANGARAJ et Shrinivas PETALE. « RWA Based Optimal Placement of Wavelength Converters in WDM Optical Networks ». In : *2018 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. IEEE, 2018. DOI : 10.1109/wispnet.2018.8538653.

[15] Vidhi GUPTA et Rachna ASTHANA. « Study of Wavelength Converter Placement in p(pre-configured)-Cycle Protection ». In : *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE, 2020. DOI : 10.1109/iccmc48092.2020.iccmc-00086.

[16] Yun WU et Songming XU. « An Algorithm for Dynamic Routing and Wavelength Assignment in WDM Network ». In : *Proceedings of the 2016 Joint International Information Technology, Mechanical and Electronic Engineering*. Atlantis Press, 2016. DOI : 10.2991/jimec-16.2016.51.

[17] Elias DOUMITH. « Traffic grooming and rerouting in multi-layer WDM network ». Thèse de doct. Télécom ParisTech, 2007.

[18] Huawei LI et Jing WU. « Survey of WDM network reconfiguration : topology migrations and their impact on service disruptions ». In : *Telecommunication Systems* 60.3 (2015), p. 349-366. DOI : 10.1007/s11235-015-0050-5.

[19] David COUDERT, Florian HUC, Daurian MAZAURIC, Nicolas NISSE et Jean-Sebastien SERENI. « Reconfiguration of the routing in WDM networks with two classes of services ». In : *2009 International Conference on Optical Network Design and Modeling*. 2009, p. 1-6.

[20] Rie HAYASHI, Takashi MIYAMURA, Daisaku SHIMAZAKI, Eiji OKI et Kohei SHIOMOTO. « Latest Trends in Traffic Matrix Modeling and Its Application to Multilayer TE ». In : *IEICE Transactions on Communications* E90-B.8 (2007), p. 1912-1921. DOI : 10.1093/ietcom/e90-b.8.1912.

[21] Takashi MIYAMURA, Eiji OKI, Ichiro INOUE et Kohei SHIOMOTO. « Enhancing Bandwidth on Demand Service Based on Virtual Network Topology Control ». In : *NOMS Workshops 2008 - IEEE Network Operations and Management Symposium Workshops*. IEEE, 2008. DOI : 10.1109/nomsw.2007.34.

[22] W. GOLAB et R. BOUTABA. « Policy-driven automated reconfiguration for performance management in WDM optical networks ». In : *IEEE Communications Magazine* 42.1 (2004), p. 44-51. DOI : 10.1109/mcom.2004.1262161.

[23] Ibrahim ALFOUZAN et Anura P. JAYASUMANA. « Dynamic reconfiguration of wavelength-routed WDM networks ». In : *Proceedings LCN 2001. 26th Annual IEEE Conference on Local Computer Networks*. IEEE Comput. Soc, 2001, p. 477-485. DOI : 10.1109/lcn.2001.990825.

[24] Mauro BRUNATO, Roberto BATTITI et Elio SALVADORI. « Load Balancing in WDM Networks through Adaptive Routing Table Changes ». In : *NETWORKING 2002, Networking Technologies, Services, and Protocols ; Performance of Computer and Communication Networks ; and Mobile and Wireless Communications, Second International IFIP-TC6 Networking Conference*. T. 2345. Lecture Notes in Computer Science. Springer, 2002, p. 289-300. DOI : 10.1007/3-540-47906-6_23.

[25] Jawwad AHMED, Fernando SOLANO, Paolo MONTI et Lena WOSINSKA. « Traffic re-optimization strategies for dynamically provisioned WDM networks ». In : *15th International Conference on Optical Network Design and Modeling - ONDM 2011*. 2011, p. 1-6.

[26] T. KARASZ, B. KARASZ, P. WILCSINSZKY et T. JAKAB. « Optimal scheduling of network consolidation ». In : *2007 IEEE Symposium on Computers and Communications*. IEEE, 2007. DOI : 10.1109/iscc.2007.4381498.

[27] Ilia BALDINE et Georges N. ROUSKAS. « Dynamic reconfiguration policies for WDM networks ». In : *Proceedings IEEE INFOCOM '99, The Conference on Computer Communications, Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies*. IEEE, 1999. DOI : 10.1109/infcom.1999.749297.

[28] Neil GEARY, Noel PARNIS, Andreas ANTONOPOULOS, Elias DRAKOPOULOS et John O'REILLY. « The benefits of reconfiguration in optical networks ». In : *10th International Telecommunication Network Strategy and Planning Symposium*. 2002, p. 373-378.

[29] Aysçegül GENÇATA, Laxman SAHASRABUDDHE et Biswanath MUKHERJEE. « Virtual-topology adaptation with minimal lightpath change for dynamic traffic in WDM mesh networks ». In : *Optical Fiber Communication Conference and Exhibit*. Opt Soc. America, 2002. DOI : 10.1109/ofc.2002.1036715.

[30] Ilker AKGUN et Feza BUZLUCA. « Virtual topology reconfiguration on optical WDM networks considering traffic grooming ». In : *Optical Switching and Networking* 3.1 (2006), p. 11-23. DOI : 10.1016/j.osn.2005.11.001.

[31] Yingyu WAN et Weifa LIANG. « Wavelength rerouting for on-line multicast in WDM networks ». In : *29th Annual IEEE International Conference on Local Computer Networks*. IEEE (Comput. Soc.), 2004. DOI : 10.1109/lcn.2004.137.

[32] Mohamed KOUBÀA et Maurice GAGNAIRE. « Lightpath Rerouting Strategies in WDM All-Optical Networks Under Scheduled and Random Traffic ». In : *Journal of Optical Communications and Networking* 2.10 (2010), p. 859-871. DOI : 10.1364/jocn.2.000859.

[33] Punam R. PATIL et Bhushan V. PATIL. « Routing and Re-routing Scheme for Cost Effective Mechanism in WDM Network ». In : *Procedia Computer Science* 49 (2015), p. 155-161. DOI : 10.1016/j.procs.2015.04.239.

[34] Der-Rong DIN et Chen-Yu LIEN. « Delay-constrained survivable multicast routing problem on WDM networks for node failure case ». In : *Computer Communications* 103 (2017), p. 165-192. DOI : 10.1016/j.comcom.2017.01.018.

[35] Y. OHSITA, T. MIYAMURA, S. ARAKAWA, S. ATA, E. OKI, K. SHIOMOTO et M. MURATA. « Gradually Reconfiguring Virtual Network Topologies Based on Estimated Traffic Matrices ». In : *IEEE/ACM Transactions on Networking* 18.1 (2010), p. 177-189. DOI : 10.1109/tnet.2009.2022263.

[36] Jing WU. « A survey of WDM network reconfiguration : Strategies and triggering methods ». In : *Computer Networks* 55.11 (2011), p. 2622-2645. DOI : 10.1016/j.comnet.2011.05.012.

[37] Gigi KARMOUS-EDWARDS, Douglas S. REEVES, George N. ROUSKAS, Lina BATTESTILLI, Priyanka VEGESNA et Arun VISHWANATH. « Edge-Reconfigurable Optical Networks (ERONs) : Rationale, Network Design, and Evaluation ». In : *Journal of Lightwave Technology* 27.12 (2009), p. 1837-1845. DOI : 10.1109/jlt.2009.2021279.

[38] Phuong Nga TRAN et Ulrich KILLAT. « Dynamic reconfiguration of logical topology for WDM networks under traffic changes ». In : *NOMS 2008 - 2008 IEEE Network Operations and Management Symposium*. IEEE, 2008. DOI : 10.1109/noms.2008.4575145.

[39] Sonia BELHARETH, David COUDERT, Dorian MAZAURIC, Nicolas NISSE et Issam TAHIRI. « Reconfiguration with physical constraints in WDM networks ». In : *2012 IEEE International Conference on Communications (ICC)*. IEEE, 2012. DOI : 10.1109/icc.2012.6364833.

[40] Alessandro VALENTINI, Balazs VASS, Jorik OOSTENBRINK, Levente CSAK, Fernando KUIPERS, Bruno PACE, David HAY et Janos TAPOLCAI. « Network Resiliency Against Earthquakes ». In : *2019 11th International Workshop on Resilient Networks Design and Modeling (RNDM)*. IEEE, 2019. DOI : 10.1109/rndm48015.2019.8949088.

[41] Der-Rong DIN. « Virtual topology transition sequence problem on WDM networks with dedicated protection ». In : *Photonic Network Communications* 18.2 (2008), p. 174-182. DOI : 10.1007/s11107-008-0181-7.

[42] Der-Rong DIN, Chi-Yen HUNG, Yu-Cyuan CHEN, Hung-Yin WANG et Chung-Yang TU. « Virtual topology transition sequence problem in WDM networks with FIPP p-cycles protection ». In : *Photonic Network Communications* 23.1 (2011), p. 40-52. DOI : 10.1007/s11107-011-0334-y.

[43] Der-Rong DIN et Chih-Wei CHOU. « Virtual topology reconfiguration for mixed-line-rate optical WDM networks under dynamic traffic ». In : *Photonic Network Communications* 30.2 (2015), p. 290-308. DOI : 10.1007/s11107-015-0517-z.

[44] CISCO. *Cisco annual internet report (2018–2023) white paper*. https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internetreport/white-paper-c11-741490.html. Accessed : May 25, 2020. [Online]. 2020.

[45] Nicolas JARA, Hermann PEMPELFORT, Gerardo RUBINO et Reinaldo VALLEJOS. « A Fault-Tolerance Solution to Any Set of Failure Scenarios on Dynamic WDM Optical Networks With Wavelength Continuity Constraints ». In : *IEEE Access* 8 (2020), p. 21291-21301. DOI : 10.1109/access.2020.2967751.

[46] R. RAMASWAMI. « Multiwavelength lightwave networks for computer communication ». In : *IEEE Communications Magazine* 31.2 (1993), p. 78-88. DOI : 10.1109/35.186364.

[47] Devi CHADHA. *Optical WDM Networks : From Static to Elastic Networks*. John Wiley & Sons, 2019.

[48] Farzad ABTAHI, Cicek CAVDAR, Jiajia CHEN, Sahar KHANMOHAMADI, Lena WOSINSKA, Guillaume MANTELET, Émile ARCHAMBAULT, Christine TREMBLAY et Michel P. BÉLANGER. « Optimal Design of Cost- and Energy-Efficient Scalable Passive Optical Backbone Networks ». In : *Asia Communications and Photonics Conference*. OSA, 2012. DOI : 10.1364/acpc.2012.as3d.4.

[49] Dinh Danh LE. « All-Optical Multicast Routing under Optical Constraints ». Thèse de doct. University of Montpellier, France, 2015.

[50] Biao CHEN et Jianping WANG. « Efficient routing and wavelength assignment for multicast in WDM networks ». In : *IEEE Journal on Selected Areas in Communications* 20.1 (2002), p. 97-109. DOI : 10.1109/49.974665.

[51] Fangfang YAN, Weisheng HU, Weiqiang SUN, Wei GUO et Yaohui JIN. « Efficient Sharing of Fixed Wavelength Converters in Clos-Type Wavelength Interchanging Cross Connects ». In : *Journal of Lightwave Technology* 27.19 (2009), p. 4189-4197. DOI : 10.1109/jlt.2009.2023160.

[52] George N. ROUSKAS. « Routing and Wavelength Assignment in Optical WDM Networks ». In : *Wiley Encyclopedia of Telecommunications* 4 (2001), p. 2097-2105.

[53] Rajiv RAMASWAMI et Kumar N. SIVARAJAN. « Routing and wavelength assignment in all-optical networks ». In : *IEEE/ACM Transactions on Networking* 3.5 (1995), p. 489-500. DOI : 10.1109/90.469957.

[54] Jun ZHENG et Hussein T. MOUFTAH. *Optical WDM Networks : concepts and design principles.* John Wiley & Sons, 2004. DOI : 10.1109/9780470545447.

[55] Tingyi ZHOU, Yiming ZHOU, Tianwei JIANG et Bahram JALALI. « AI compensation of crosstalk in WDM communication ». In : *AI and Optical Data Sciences II*. Sous la dir. de Ken ichi KITAYAMA et Bahram JALALI. SPIE, 2021. DOI : 10.1117/12.2576983.

[56] S. L. DANIELSEN, B. MIKKELSEN, P. B. HANSEN, H. N. POULSEN, A. KLOCH et K. E. STUBKJAER. « Wavelength conversion techniques and devices ». In : *Conference Proceedings. LEOS '97. 10th Annual Meeting IEEE Lasers and Electro-Optics Society 1997 Annual Meeting*. IEEE, 1996, p. 94-95. DOI : 10.1109/leos.1997.630534.

[57] Kristian E. STUBKJAER, Allan KLOCH, Peter Bukhave HANSEN, Henrik N. POULSEN, David WOLFSON, Kim Stokholm JEPSEN, Anders Thomas CLAUSEN, Emmanuel LIMAL et Alvaro BUXENS. « Wavelength Converter Technology ». In : *IEICE Transactions on Electronics* 82.2 (1999), p. 338-348.

[58] R. RAMAMURTHY et B. MUKHERJEE. « Fixed-alternate routing and wavelength conversion in wavelength-routed optical networks ». In : *IEEE/ACM Transactions on Networking* 10.3 (2002), p. 351-367. DOI : 10.1109/tnet.2002.1012367.

[59] Partha Pratim SAHU. *Optical Networks and Components : Fundamentals and Advances*. CRC Press, 2020.

[60] Rajiv RAMASWAMI, Kumar SIVARAJAN et Galen SASAKI. *Optical networks : a practical perspective 3rd Edition Ed. 3*. Morgan Kaufmann, 2009.

[61] Jean-Michel MUR. *Les fibres optiques Notions fondamentales, 3ième édition*. Sous la dir. d'ENI. 2019.

[62] S. J. CHUA et B. J. LI. « Introduction to optical switches ». In : *Optical Switches*. Elsevier, 2010, p. 1-4. DOI : 10.1533/9780857090416.1.

[63] Yunfeng SHEN, Kejie LU et Wanyi GU. « Coherent and incoherent crosstalk in WDM optical networks ». In : *Journal of Lightwave Technology* 17.5 (1999), p. 759-764. DOI : 10.1109/50.762889.

[64] W. S. HU et Q. J. ZENG. « Multicasting optical cross connects employing splitter-and-delivery switch ». In : *IEEE Photonics Technology Letters* 10.7 (1998), p. 970-972. DOI : 10.1109/68.681287.

[65] Qiaoshan CHEN, Fanfan ZHANG, Ruiqiang JI, Lei ZHANG et Lin YANG. « Universal method for constructing N-port non-blocking optical router based on 2 × 2 optical switch for photonic networks-on-chip ». In : *Optics Express* 22.10 (2014), p. 12614. DOI : 10.1364/oe.22.012614.

[66] M. ALI et J. S. DEOGUN. « Power-efficient design of multicast wavelength-routed networks ». In : *IEEE Journal on Selected Areas in Communications* 18.10 (2000), p. 1852-1862. DOI : 10.1109/49.887907.

[67] L. H. SAHASRABUDDHE et B. MUKHERJEE. « Light trees : optical multicasting for improved performance in wavelength routed networks ». In : *IEEE Communications Magazine* 37.2 (1999), p. 67-73. DOI : 10.1109/35.747251.

[68] Kuo-Chun LEE et V. O. LI. « A wavelength-convertible optical network ». In : *Journal of Lightwave Technology* 11.5 (1993), p. 962-970. DOI : 10.1109/50.233260.

[69] Jingyi HE, S.-H. Gary CHAN et Danny H. K. TSANG. « Multicasting in WDM networks ». In : *IEEE Communications Surveys & Tutorials* 4.1 (2002), p. 2-20. DOI : 10.1109/comst.2002.5341330.

[70] Ashraf HAMAD, Tao WU, Ahmed E. KAMAL et Arun K. SOMANI. « On multicasting in wavelength-routing mesh networks ». In : *Computer Networks* 50.16 (2006), p. 3105-3164. DOI : 10.1016/j.comnet.2005.12.012.

[71] Chunming QIAO, Myoungki JEONG, A. GUHA, Xijun ZHANG et J. WEI. « WDM multicasting in IP over WDM networks ». In : *Proceedings. Seventh International Conference on Network Protocols*. IEEE Comput. Soc, 1999, p. 89-96. DOI : 10.1109/icnp.1999.801919.

[72] Rajkumar MALLI, Xijun ZHANG et Chunming QIAO. « Benefits of multicasting in all-optical networks ». In : *All-Optical Networking : Architecture, Control, and Management Issues*. Sous la dir. de John M. SENIOR et Chunming QIAO. T. 3531. SPIE, 1998. DOI : 10.1117/12.327060.

[73] Xijun ZHANG, J. WEI et Chunming QIAO. « On fundamental issues in IP over WDM multicast ». In : *Proceedings Eight International Conference on Computer Communications and Networks (Cat. No.99EX370)*. IEEE, 1999. DOI : 10.1109/icccn.1999.805500.

[74] F. K. HWANG et Dana S. RICHARDS. « Steiner tree problems ». In : *Networks* 22.1 (1992), p. 55-89. DOI : 10.1002/net.3230220105.

[75] S. RAMYA et T. S. INDUMATHI. « A Comparative Study on the Routing and Wavelength Assignment in WDM ». In : *2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*. Bangalore, India : IEEE, juil. 2020, p. 1-5. DOI : 10.1109/conecct50063.2020.9198388.

[76] Y. ZHOU et G. S. POO. « Multicast Wavelength Assignment for Sparse Wavelength Conversion in WDM Networks ». In : *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*. IEEE, 2006. DOI : 10.1109/infocom.2006.217.

[77] I-Hsuan PENG, Yen-Wen CHEN et Hsiang-Ru PENG. « A Systematic Wavelength Assign Algorithm for Multicast in WDM Networks with Sparse Conversion Nodes ». In : *Journal of Information Science & Engineering* 25.2 (2009), p. 559-574.

[78] Nina SKORIN-KAPOV. « Multicast routing and wavelength assignment in WDM networks : a bin packing approach ». In : *Journal of Optical Networking* 5.4 (2006), p. 266. DOI : 10.1364/jon.5.000266.

[79] Hui CHENG, Xingwei WANG, Shengxiang YANG, Min HUANG et Jiannong CAO. « QoS Multicast Routing and Wavelength Assignment in IP/DWDM Optical Internet by Bio-inspired Algorithms ». In : *Citeseer* (2010).

[80] I-Shyan HWANG, San-Nan LEE et Ying-Fung CHUANG. « Multicast wavelength assignment with sparse wavelength converters to maximize the network capacity using ILP formulation in WDM mesh networks ». In : *Photonic Network Communications* 12.2 (2006), p. 161-172. DOI : 10.1007/s11107-006-0009-2.

[81] Siamak AZODOLMOLKY, Marianna ANGELOU, Ioannis TOMKOS, Tania PANAYIOTOU, Georgios ELLINAS et Neophytos (Neo) ANTONIADES. « Impairment-Aware Optical Networking : A Survey ». In : *Optical Networks*. Springer New York, 2011, p. 443-479. DOI : 10.1007/978-1-4614-1093-5_12.

[82] Aijun DING, Gee-Swee POO et Sun-Teck TAN. « An expanded graph model for MCRWA problem in WDM networks ». In : *27th Annual IEEE Conference on Local Computer Networks, 2002. Proceedings. LCN 2002.* IEEE Comput. Soc, 2002, p. 557-564. DOI : 10.1109/lcn.2002.1181830.

[83] Der-Rong DIN. « Genetic Algorithm for Finding Minimal Cost Light Forest of Multicast Routing on WDM Networks ». In : *Journal of Artificial Evolution and Applications* 2008 (2008), p. 1-20. DOI : 10.1155/2008/536913.

[84] Fen ZHOU, Miklos MOLNAR et Bernard COUSIN. « Is Light-Tree Structure Optimal for Multicast Routing in Sparse Light Splitting WDM Networks ? » In : *2009 Proceedings of 18th International Conference on Computer Communications and Networks*. IEEE, 2009, p. 1-7. DOI : 10.1109/icccn.2009.5235386.

[85] George N. ROUSKAS. « Optical layer multicast : rationale, building blocks, and challenges ». In : *IEEE Network* 17.1 (2003), p. 60-65. DOI : 10.1109/mnet.2003.1174179.

[86] Tianping SHUAI et Wenbao AI. « New algorithms for multicast routing and wavelength assignment in multi-hop optical WDM networks ». In : *Photonic Network Communications* 23.1 (2011), p. 53-59. DOI : 10.1007/s11107-011-0335-x.

[87] Xijun ZHANG, J. Y. WEI et Chunming QIAO. « Constrained multicast routing in WDM networks with sparse light splitting ». In : *Journal of Lightwave Technology* 18.12 (2000), p. 1917-1927. DOI : 10.1109/50.908787.

[88] Fen ZHOU, Miklos MOLNAR et Bernard COUSIN. « Light-hierarchy : the optimal structure for multicast routing in WDM mesh networks ». In : *The IEEE symposium on Computers and Communications*. IEEE, 2010, p. 611-616. DOI : 10.1109/iscc.2010.5643912.

[89] Miklos MOLNAR. « Non-elementary routes for multicasting in Transparent Optical Networks ». In : *2017 19th International Conference on Transparent Optical Networks (ICTON)*. IEEE, 2017, p. 1-4. DOI : 10.1109/icton.2017.8025135.

[90] Bernard COUSIN, Joel Christian ADÉPO, Souleymane OUMTANAGA et Michel BABRI. « Tree reconfiguration without lightpath interruption in WDM optical networks ». In : *International Journal of Internet Protocol Technology* 7.2 (2012), p. 85-95. DOI : 10.1504/ijipt.2012.050217.

[91] Javier E. SIERRA, Luis F. CARO, Fernando SOLANO, Jose L. MARZO, Ramon FABREGAT et Yezid DONOSO. « Dynamic unicast/multicast traffic grooming using S/G light-tree in WDM networks ». In : *2008 International Symposium on Performance Evaluation of Computer and Telecommunication Systems*. IEEE. 2008, p. 263-268.

[92] Rongping LIN, Wen-De ZHONG, Sanjay Kumar BOSE et Moshe ZUKERMAN. « Light-tree configuration for multicast traffic grooming in WDM mesh networks ». In : *Photonic Network Communications* 20.2 (2010), p. 151-164. DOI : 10.1007/s11107-010-0255-1.

[93] Kunanan LUEKIJNA et Chaiyachet SAIVICHIT. « Multicast Traffic Reconfiguration in WDM Network for Single Node Failure Design ». In : *The 9th International Conference on Advanced Communication Technology*. IEEE, 2007. DOI : 10.1109/icact.2007.358727.

[94] Marcell PERENYI, Peter SOPRONI, Tibor CINKLER et David LARRABEITI. « Regular reconfiguration of light-trees in multilayer optical networks ». In : *2008 International Conference on Optical Network Design and Modeling*. IEEE, 2008, p. 1-6. DOI : 10.1109/ondm.2008.4578411.

[95] Akihiro KADOHATA, Akira HIRANO, Fumikazu INUZUKA, Atsushi WATANABE et Osamu ISHIDA. « Wavelength Path Reconfiguration Design in Transparent Optical WDM Networks ». In : *Journal of Optical Communications and Networking* 5.7 (2013), p. 751-761. DOI : 10.1364/jocn.5.000751.

[96] Qi wu WU, Hao CHEN, Ling zhi JIANG et Yang ZHOU. « Dynamic multicast sharing protection algorithm based on fuzzy game in multi-domain optical network ». In : *Optoelectronics Letters* 16.6 (2020), p. 455-461. DOI : 10.1007/s11801-020-0006-1.

[97] Brigitte JAUMARD, Huy Quang DUONG, Romualdas ARMOLAVICIUS, Todd MORRIS et Petar DJUKIC. « Efficient Real-Time Scalable Make-Before-Break Network Re-Routing ». In : *Journal of Optical Communications and Networking* 11.3 (2019), p. 52. DOI : 10.1364/jocn.11.000052.

[98] Brigitte JAUMARD, Hamed POUYA et David COUDERT. « Wavelength Defragmentation for Seamless Migration ». In : *Journal of Lightwave Technology* 37.17 (2019), p. 4382-4393. DOI : 10.1109/jlt.2019.2924914.

[99] DGTLVSD AWDUCHE, Lou BERGER, D. GAN, Tony LI, Vijay SRINIVASAN et George SWALLOW. *RSVP-TE : extensions to RSVP for LSP tunnels*. RFC 3209, December. 2001.

[100] Tatsumi TAKAGI, Hiroshi HASEGAWA, Ken ichi SATO, Yoshiaki SONE, Akira HIRANO et Masahiko JINNO. « Disruption Minimized Spectrum Defragmentation in Elastic Optical Path Networks that Adopt Distance Adaptive Modulation ». In : *37th European Conference and Exposition on Optical Communications*. OSA, 2011. DOI : 10.1364/ecoc.2011.mo.2.k.3.

[101] Joel Christian ADÉPO. « Reconfiguration du routage multicast dans les réseaux optiques WDM ». Thèses. Université Nangui Abrogoua (Côte d'Ivoire), sept. 2016. URL : https://tel.archives-ouvertes.fr/tel-01428414.

[102] Bernard COUSIN, Joël Christian ADÉPO, Michel BABRI et Souleymane OUMTANAGA. « Tree Reconfiguration without Lightpath Interruption in Wavelength Division Multiplexing Optical Networks with Limited Resources. » In : *International Journal of Computer Science Issues (IJCSI)* 11.2 (2014), p. 7-18.

[103] Joël Christian ADÉPO, Boko AKA et Michel BABRI. « Tree Reconfiguration with Network Resources Constraint ». In : *nternational Journal of Computer Science and Telecommunications* 7 (2016), p. 1-4.

[104] I. CHLAMTAC, A. FARAGO et Tao ZHANG. « Lightpath (wavelength) routing in large WDM networks ». In : *IEEE Journal on Selected Areas in Communications* 14.5 (1996), p. 909-913. DOI : 10.1109/49.510914.

[105] Yutaka TAKITA, Kazuyuki TAJIMA, Tomohiro HASHIGUCHI et Toru KATAGIRI. « Wavelength Defragmentation for Seamless Service Migration ». In : *Journal of Optical Communications and Networking* 9.2 (2017), A154. DOI : 10.1364/jocn.9.00a154.

[106] Huy DUONG, Brigitte JAUMARD, David COUDERT et Ron ARMOLAVICIUS. « Efficient Make Before Break Capacity Defragmentation ». In : *2018 IEEE 19th International Conference on High Performance Switching and Routing (HPSR)*. IEEE, 2018, p. 1-6. DOI : 10.1109/hpsr.2018.8850754.

[107] Brigitte JAUMARD, Hamed POUYA et David COUDERT. « Make-Before-Break Wavelength Defragmentation ». In : *2018 20th International Conference on Transparent Optical Networks (ICTON)*. IEEE, 2018, p. 1-5. DOI : 10.1109/icton.2018.8473893.

[108] Enrique J. DAVALOS, Matilde F. ROMERO, Sergio M. GALEANO, Diego A. BAEZ, Ariel LEIVA et Benjamin BARAN. « Spectrum Defragmentation in Elastic Optical Networks : Two Approaches With Metaheuristics ». In : *IEEE Access* 7 (2019), p. 119835-119843. DOI : 10.1109/access.2019.2937032.

[109] F. BALMAS. « Displaying dependence graphs : a hierarchical approach ». In : *Proceedings Eighth Working Conference on Reverse Engineering*. IEEE Comput. Soc, 2001. DOI : 10.1109/wcre.2001.957830.

[110] Fernando SOLANO. « Slick Lightpath Reconfiguration Using Spare Resources ». In : *Journal of Optical Communications and Networking* 5.9 (2013), p. 1021. DOI : 10.1364/jocn.5.001021.

[111] Fernando SOLANO. « Analyzing Two Conflicting Objectives of the WDM Lightpath Reconfiguration Problem ». In : *GLOBECOM 2009 - 2009 IEEE Global Telecommunications Conference*. IEEE, 2009, p. 1-7. DOI : 10.1109/glocom.2009.5426108.

[112] N. JOSE et A. K. SOMANI. « Connection rerouting/network reconfiguration ». In : *Fourth International Workshop on Design of Reliable Communication Networks, 2003. (DRCN 2003). Proceedings.* IEEE, 2003, p. 23-30. DOI : 10.1109/drcn.2003.1275334.

[113] Fernando SOLANO et Michał PIÓRO. « Lightpath Reconfiguration in WDM Networks ». In : *Journal of Optical Communications and Networking* 2.12 (2010), p. 1010-1021. DOI : 10.1364/jocn.2.001010.

[114] Y. Daniel LIANG et Maw-Shang CHANG. « Minimum feedback vertex sets in cocomparability graphs and convex bipartite graphs ». In : *Acta Informatica* 34.5 (1997), p. 337-346. DOI : 10.1007/s002360050088.

[115] Hen-Ming LIN et Jing-Yang JOU. « Computing minimum feedback vertex sets by contraction operations and its applications on CAD ». In : *Proceedings 1999 IEEE International Conference on Computer Design : VLSI in Computers and Processors (Cat. No.99CB37040)*. IEEE Comput. Soc, 1999, p. 364-369. DOI : 10.1109/iccd.1999.808567.

[116] Claunir PAVAN, Rui Manuel MORAIS, José R. Ferreira da ROCHA et Armando Nolasco PINTO. « Generating Realistic Optical Transport Network Topologies ». In : *Journal of Optical Communications and Networking* 2.1 (2009), p. 80. DOI : 10.1364/jocn.2.000080.

[117] *The Internet Topology Zoo*. http://www.topology-zoo.org/dataset.html. Accessed : 2019-06-14.

[118] Aric HAGBERG, Pieter SWART et Daniel S. CHULT. *Exploring network structure, dynamics, and function using networkx*. Rapp. tech. Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

[119] Xiaowen CHU, Bo LI et I. CHLAMTAC. « Wavelength converter placement under different RWA algorithms in wavelength-routed all-optical networks ». In : *IEEE Transactions on Communications* 51.4 (2003), p. 607-617. DOI : 10.1109/tcomm.2003.810834.

[120] E. W. DIJKSTRA. « A note on two problems in connexion with graphs ». In : *Numerische Mathematik* 1.1 (1959), p. 269-271. DOI : 10.1007/bf01386390.

[121] R. C. PRIM. « Shortest Connection Networks And Some Generalizations ». In : *Bell System Technical Journal* 36.6 (1957), p. 1389-1401. DOI : 10.1002/j.1538-7305.1957.tb01515.x.

[122] Frank E. RITTER, Michael J. SCHOELLES, Karen S. QUIGLEY et Laura Cousino KLEIN. « Determining the Number of Simulation Runs : Treating Simulations as Theories by Not Sampling Their Behavior ». In : *Human-in-the-Loop Simulations*. Springer London, 2011, p. 97-116. DOI : 10.1007/978-0-85729-883-6_5.

[123] Frank HARARY et Ronald C. READ. « Is the null-graph a pointless concept ? » In : *Lecture Notes in Mathematics*. Springer Berlin Heidelberg, 1974, p. 37-44. DOI : 10.1007/bfb0066433.

[124] Jeff R. KNISLEY et Debra J. KNISLEY. « Vertex-weighted graphs and their applications ». In : *Utilitas Mathematica* 94 (2014), p. 237-249.

[125] Paola FESTA, Panos M. PARDALOS et Mauricio G. C. RESENDE. « Feedback Set Problems ». In : *Handbook of Combinatorial Optimization*. Springer US, 1999, p. 209-258. DOI : 10.1007/978-1-4757-3023-4_4.

[126] Camil DEMETRESCU et Irene FINOCCHI. « Combinatorial algorithms for feedback problems in directed graphs ». In : *Information Processing Letters* 86.3 (2003), p. 129-136. DOI : 10.1016/s0020-0190(02)00491-x.

[127] B. M. WAXMAN. « Routing of multipoint connections ». In : *IEEE Journal on Selected Areas in Communications* 6.9 (1988), p. 1617-1622. DOI : 10.1109/49.12889.

[128] Xiao-Hua JIA, Ding-Zhu DU, Xiao-Dong HU, Man-Kei LEE et Jun GU. « Optimization of wavelength assignment for QoS multicast in WDM networks ». In : *IEEE Transactions on Communications* 49.2 (2001), p. 341-350. DOI : 10.1109/26.905896.

[129] D.D. LE, M. MOLNAR et J. PALAYSI. « Multicast routing in WDM networks without splitters ». In : *IEEE Communications Magazine* 52.7 (2014), p. 158-167. DOI : 10.1109/mcom.2014.6852098.

[130] Yu XIONG, Zhiqiang LI, Bin ZHOU et Xiancun DONG. « Cross-layer shared protection strategy towards data plane in software defined optical networks ». In : *Optics Communications* 412 (2018), p. 66-73. DOI : 10.1016/j.optcom.2017.11.085.