

.....

## Reconfiguration d'arbre multicast dans les réseaux optiques WDM

Joël C. ADEPO, Souleymane OUMTANAGA

Laboratoire de Recherche en Informatique et Télécoms (LARIT)

INPHB-Yamoussoukro

COTE D'IVOIRE

jscadep@hotmail.fr; oumtana@yahoo.fr

.....

**RÉSUMÉ.** Les réseaux d'aujourd'hui font face à des phénomènes tels que les surcharges de trafic, les pannes du réseau ou les opérations de maintenance. Pour y faire face, les opérateurs de réseau sont amenés à calculer une nouvelle route. Ensuite basculer le flux de la route initiale vers la nouvelle route par la reconfiguration. La reconfiguration doit se faire rapidement afin de répondre efficacement aux phénomènes perturbateurs. Le trafic sur ces réseaux étant de plus en plus constitué de trafic multicast, et les applications de moins en moins tolérantes aux interruptions de connexions, nous proposons un algorithme de reconfiguration d'arbre multicast sans interruption de connexion dans les réseaux optiques WDM. Cet algorithme permet de faire la reconfiguration tout réduisant la durée du processus et le coût des ressources additionnelles utilisées.

**ABSTRACT.** Today, networks are facing many phenomena such as changes in traffic model, network failures or maintenance operations. To cope with these phenomena, network operators have to compute a new path. Then migrate from the initial path to the new one by reconfiguration. Reconfiguration must be done quickly to cope with these phenomena efficiently. Traffic on these networks is increasingly made up of multicast traffic, and applications less indulgent with connections interruption, we propose a multicast tree reconfiguration algorithm without connection interruption in optical WDM networks. This algorithm does reconfiguration by reducing the process setup time and the cost of additional resources used.

**MOTS-CLÉS :** reconfiguration, réseau optique WDM, arbre multicast, interruption de connexion.

**KEYWORDS:** reconfiguration, optical WDM networks, multicast tree, connection interruption.

## 1. Introduction

Les réseaux d'aujourd'hui sont fréquemment confrontés à des phénomènes tels que les surcharges de trafic, les pannes ou les opérations de maintenance qui détériorent la qualité de service. Pour faire face à ces phénomènes, les opérateurs de réseau utilisent des techniques comme la reconfiguration. Elle se déroule en trois étapes séquentielles et itératives [6]: décision de la politique de reconfiguration, calcul d'une nouvelle route et migration vers cette nouvelle route. La reconfiguration dans les réseaux optiques WDM [4] est très bénéfique à cause de leur large déploiement aujourd'hui. Mais elle constitue un challenge car elle est effectuée en ligne en libérant temporairement certaines ressources (émetteurs, récepteurs, longueurs d'onde). Alors une mauvaise gestion ou mauvaise planification peut causer la perte de beaucoup données dans ces réseaux à très haut débit [8]. La gestion des connexions des réseaux WDM peut être faite par le protocole GSMP [1]. La technique utilisée par ce protocole est d'interrompre le flux de l'ancienne connexion avant d'établir la nouvelle. Cela ne garantit toujours pas la continuité du trafic et n'est pas optimal en termes de durée. Le calcul de la nouvelle route étant basé sur l'état du réseau, la reconfiguration doit se faire rapidement afin de répondre efficacement aux phénomènes perturbateurs. De plus, certaines applications, telles qu'IPTV, sont des applications temps réel et peu tolérantes aux interruptions. Alors la reconfiguration doit se faire sans interruption de connexion. Le trafic des réseaux étant de plus en plus composé de trafic multicast, alors nous nous intéressons à la reconfiguration d'arbre multicast. Dans ce travail, nous supposons que la reconfiguration est contrôlée de manière centralisée. La fonction de reconfiguration peut être implémentée selon une architecture PCE comme une partie du système de gestion centralisée [2]. Le problème considéré est alors de basculer le flux de données d'un arbre multicast initial vers un arbre final, sans interrompre la continuité du flux vers toutes les destinations du groupe multicast, tout en minimisant la durée du processus de reconfiguration et le coût des ressources additionnelles utilisées.

Dans la suite, dans la section 2, nous présentons les travaux existants sur la reconfiguration multicast. La section 3 présente notre formulation du problème de reconfiguration d'arbre multicast et ses objectifs. Ensuite, nous présentons la reconfiguration multicast utilisant plusieurs longueurs d'onde et l'algorithme de reconfiguration proposé. Cet algorithme permet de faire la reconfiguration d'arbre multicast sans interruption de connexion, avec une durée réduite et une bonne utilisation des ressources du réseau. Dans la section 4, nous présentons les résultats des évaluations. Enfin nous concluons dans la section 5.

---

## 2. Etat de l'art

Les travaux existants sur la reconfiguration s'intéressent au calcul ou au re-calcul de nouvelles routes, à la fréquence d'exécution de la reconfiguration tout en minimisant les interruptions de connexions et aux événements déclencheurs de la reconfiguration [5,9]. [3] propose un algorithme de reconfiguration pour le trafic unicast dans les réseaux à connexion. L'algorithme proposé fait la reconfiguration d'un chemin initial à un autre final sans interrompre la connexion et en une durée réduite. Peu de papiers se sont intéressés à la reconfiguration d'arbre multicast [7,10]. Selon nos connaissances, il n'existe pas de travaux dans la littérature sur la reconfiguration d'un arbre multicast vers un autre arbre sans interruption de connexion dans les réseaux optiques. La reconfiguration qui consiste à basculer le flux d'un arbre multicast vers un autre sans interruption de connexion est l'objectif de notre travail.

---

## 3. Processus de reconfiguration multicast

### 3.1. Formulation du problème

Soient  $T_0 = (E_0, V_0)$  et  $T_z = (E_z, V_z)$  deux arbres ayant la même source et les mêmes destinations.  $E_0$  et  $V_0$  représentent respectivement l'ensemble des nœuds (commutateurs) et des liens (fibres optiques) de l'arbre initial.  $E_z$  et  $V_z$  représentent respectivement l'ensemble des nœuds et des liens de l'arbre final. On veut passer de l'arbre  $T_0$  à l'arbre  $T_z$ . On suppose qu'à chaque étape de reconfiguration, on ne peut configurer simultanément qu'un sous-ensemble de nœuds les uns après les autres. Chaque étape doit assurer la continuité du trafic. On produit alors une série d'arbres  $T_0, T_1, \dots, T_{z-1}, T_z$ . On définit quatre types d'opérations de configuration possibles sur un commutateur : l'ajout de la commutation d'une longueur d'onde, la suppression de la commutation d'une longueur d'onde, le basculement du flux et la conversion de longueur d'onde. Enfin, Reconfigurer un nœud à une étape revient à reconfigurer une seule interface de sortie pour ce nœud. Il est supposé que l'opération de basculement qui permet de basculer le flux d'un port de sortie à un autre port de sortie d'un même commutateur est suffisamment rapide pour quelle puisse être effectuée sans interrompre le flux.

### 3.2. Objectifs

Le procédé de reconfiguration multicast vise trois objectifs. D'abord la reconfiguration doit s'exécuter sans interruption de connexion. Ensuite, elle doit réduire la durée du processus et enfin réduire le coût d'utilisation des longueurs d'onde du réseau. On veut étudier ce problème selon deux contextes. Dans le premier contexte on

suppose que les nœuds ne sont pas équipés de convertisseurs de longueurs d'onde : une seule longueur d'onde est utilisée dans ce cas (pour tous les arbres : initial, intermédiaires et final). Dans le deuxième contexte on suppose que les nœuds sont équipés de convertisseurs de longueurs d'onde : on s'autorise alors à utiliser plusieurs longueurs d'onde (pour passer de l'arbre initial, à l'arbre final), toutefois l'arbre initial et l'arbre final n'utilise qu'une longueur d'onde (la même longueur d'onde). Dans ce papier, nous présentons le deuxième contexte d'étude.

### 3.3. Reconfiguration utilisant plusieurs longueurs d'onde

Dans ce contexte, l'arbre initial et l'arbre final n'utilise qu'une longueur d'onde (la même longueur d'onde). Utilisons un procédé de reconfiguration branche par branche aller-retour (BpBAR). Supposons que la longueur d'onde utilisée par l'arbre initial soit  $a$ . Une branche de l'arbre initial sera appelée  $Br_0(S, i)$  et celle de l'arbre final  $Br_z(S, i)$ , avec  $i$  la destination. On établit  $Br_z(S, i)$  pour une destination  $i$  donnée en utilisant une autre longueur d'onde  $b$ . Puis on bascule à la source de  $a$  vers  $b$  pour uniquement cette destination. Puis on supprime la partie exclusive de  $Br_0(S, i)$  utilisant la longueur d'onde  $a$ . On répète pour toutes les destinations. On obtient le nouvel arbre (arbre suivant la nouvelle topologie) alimenté par  $b$ . Il faut alors établir un nouvel arbre non-alimenté avec la longueur d'onde  $a$  en utilisant le procédé précédent, puis on commute à la source de  $b$  vers  $a$ . On fait alors le passage de l'arbre initial  $T_0$  à l'arbre final  $T_z$  en produisant une succession d'arbres. Considérons maintenant le problème de reconfiguration formé par les arbres de la Figure 1 : arbre initial  $T_0$  (en rouge) et final  $T_z$  (en bleu).

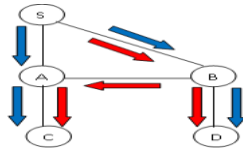


Figure 1. Arbre initial  $T_0$  et final  $T_z$

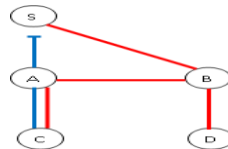


Figure 2. Branche non alimentée  $Br_z(S, C)$

Supposons qu'on crée d'abord la branche non alimentée  $Br_z(S, C)$  après une suite d'étapes. On obtient l'arbre de la figure 2. L'étape suivante est l'alimentation de  $Br_z(S, C)$  et la rupture de la connexion de l'ancienne branche. Mais sur cet arbre (Figure 2), reconfigurer la source  $S$  va certes alimenter la nouvelle branche mais couper les connexions vers toutes les destinations (ici  $D$ ). Alors, pour ce problème, on doit alimenter la nouvelle branche  $Br_z(S, C)$  à partir du nœud  $B$  (Figures 3 et 4) car il permet en plus d'interrompre le flux uniquement sur l'ancienne branche  $Br_0(S, C)$ .

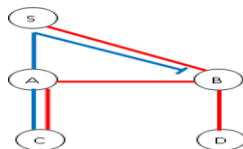


Figure 3. Branche non alimenté jusqu'à  $B$

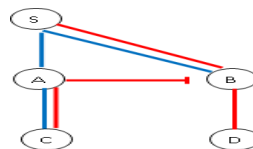


Figure 4. Basculement en  $B$

Le nœud par lequel on fait le basculement de flux sur une nouvelle branche dans la phase aller du procédé sera appelé *Com1* et *Com2* celui de la phase retour. *Com1* doit être choisi de telle sorte que sa reconfiguration ne rompt pas la connexion vers les autres destinations et permette aussi d'alimenter la nouvelle branche. Alors *Com1* doit être un nœud de  $Br_0(S, i)$  pour permettre l'alimentation de la nouvelle branche  $Br_z(S, i)$ . Ensuite, il doit être un nœud de branchement de l'arbre courant pour les canaux optiques de l'arbre initial et être le plus proche de la destination *i*. Ce critère assure que le basculement ne coupe pas la connexion vers les autres destinations. Si aucun *Com1* n'existe, alors on fait la commutation à la source *S*. Le nœud *Com2* est le nœud de branchement de l'arbre courant appartenant à la branche choisie  $Br_z(S, i)$ , utilisant les mêmes longueurs d'onde que l'arbre final et le plus proche de la destination *i*. Si aucun *Com2* n'existe, on fait le basculement à la source *S*. Nous définissons un premier procédé de reconfiguration multicast utilisant le procédé BpBAR appelé BpBAR\_1.

### 3.3.1. Procédé de reconfiguration BpBAR\_1

Le procédé BpBAR\_1 se décrit en ces étapes suivantes :

- 1) Choisir une branche  $Br_z(S, i)$  suivant la nouvelle topologie. Au départ  $Com1=S$  ; rechercher *Com1* et configurer un à un les ports des nœuds de la branche compris entre la destination *i* et *S*, et ceux compris entre *S* et *Com1*.
- 2) Configurer *Com1* pour basculer le flux sur  $Br_z(S, i)$  et le couper sur l'ancienne branche  $Br_0(S, i)$ .
- 3) Configurer un à un les ports des nœuds compris entre *Com1* et *i* sur  $Br_0(S, i)$  pour supprimer l'ancienne branche.
- 4) Si *Com1* est différent de *S*, alors configurer *S* pour couper le flux de la partie entre *S* et *Com1*.
- Si l'étape 4) est réalisée alors étape 5) Configurer un à un les ports des nœuds entre *S* et *Com1* pour supprimer cette partie.
- 6) Répéter les étapes 1) à 4) jusqu'à l'établissement de toutes les nouvelles branches.
- 7) Choisir une branche  $Br_z(S, i)$  sur la nouvelle topologie. Au départ  $Com2=S$  ; rechercher *Com2* et configurer un à un les ports des nœuds compris entre *i* et *Com2* sur  $Br_z(S, i)$  pour établir la branche.
- 8) Configurer *Com2* pour basculer le flux sur  $Br_z(S, i)$  et couper le flux de l'ancienne branche.
- 9) Supprimer l'ancienne branche en configurant un à un les ports des nœuds entre *S* et la destination *i*.
- 10) Répéter 7) à 9) jusqu'à l'établissement de toutes les nouvelles branches.

Ce procédé de reconfiguration multicast assure l'existence d'un arbre enraciné à la source du trafic à chaque étape, d'où la continuité du trafic. Mais il produit une très longue séquence d'étapes. La solution est de reconfigurer en parallèle, donc en une

étape, les opérations sur les nœuds dont les ports de sortie ne sont pas utilisés pour le trafic de l'étape courante. Ainsi donc la détermination des actions à réaliser en parallèle et les branches à établir effectivement (celles qui sont différentes des branches de l'arbre initial) dans le procédé de reconfiguration permet de réduire le nombre d'étapes de reconfigurations. Nous proposons le procédé BpBAR\_2 ci-dessous.

### 3.3.2. Notre proposition : Procédé de reconfiguration BpBAR\_2

Le procédé BpBAR\_2 se décrit en ces étapes suivantes :

- 1) Choisir une nouvelle branche  $Br_z(S, i)$  différente des branches de l'arbre initial. Au départ  $Com1=S$  ; rechercher  $Com1$  et configurer en parallèle les ports des nœuds de la branche compris entre la destination  $i$  et la source  $S$  et ceux compris entre  $S$  et  $Com1$ .
- 2) Configurer  $Com1$  pour basculer le flux sur la nouvelle branche et le couper sur l'ancienne branche.
- 3) Configurer en parallèle les ports des nœuds compris entre  $Com1$  et  $i$  sur  $Br_0(S, i)$  pour la supprimer.
- 4) Si  $Com1$  est différent de  $S$ , alors configurer  $S$  pour couper le flux de la partie entre  $S$  et  $Com1$ .  
Si l'étape 4) est réalisée alors étape 5) Configurer en parallèle les ports des nœuds entre  $S$  et  $Com1$  pour supprimer cette partie.
- 6) Répéter les étapes 1) à 4) jusqu'à l'établissement de toutes les nouvelles branches.
- 7) Choisir une nouvelle branche  $Br_z(S, i)$  sur la nouvelle topologie. Au départ  $Com2=S$  ; rechercher  $Com2$  et configurer en parallèle les ports des nœuds compris entre  $i$  et  $Com2$  sur  $Br_z(S, i)$  pour établir la branche.
- 8) Configurer  $Com2$  pour alimenter la nouvelle branche et couper le flux de l'ancienne branche.
- 9) Supprimer l'ancienne branche en configurant en parallèle les ports des nœuds entre  $S$  et la destination  $i$ .
- 10) Répéter 7) à 9) jusqu'à l'établissement de toutes les nouvelles branches.

---

## 4. Evaluation

Les topologies de réseau utilisées pour évaluer notre procédé sont COST 239 et NSF Network. Ce sont deux topologies fréquemment utilisées dans les travaux scientifiques du domaine. L'arbre initial est déterminé avec l'algorithme de Dijkstra entre la source et chaque destination. On utilise l'algorithme de Prim pour déterminer l'arbre final. L'expression du cout additionnel  $Cout\_add(T_k)$  de l'arbre  $T_k$  à l'étape  $k$  est définie ci-dessous. Notons :

$$T_{i,j}^{k,m} = \begin{cases} 1 & \text{si l'arbre } T_k \text{ utilise la longueur} \\ & \text{d'onde } m \text{ sur le lien } (i,j) \\ 0 & \text{sinon} \end{cases} \quad \alpha_{i,j} = \begin{cases} 1 & \text{si } (i,j) \text{ est un lien de l'arbre} \\ & \text{initial ou final} \\ 0 & \text{sinon} \end{cases}$$

$$Cost\_add(T_k) = \sum_{i=1}^n \sum_{j=1}^n \sum_{m=0}^N T_{i,j}^{k,m} - \sum_{i=1}^n \sum_{j=1}^n \alpha_{i,j} T_{i,j}^{k,0} \quad (1)$$

$n$  est le nombre de nœuds et  $N+1$  le nombre de longueurs d'onde. La durée d'une étape de reconfiguration est supposée unitaire et les longueurs d'onde ont le même cout d'utilisation sur les liens. Le cout additionnel d'une séquence de reconfiguration  $Seq$ , avec  $|Seq|$  le nombre d'arbres de la séquence est :

$$Cost\_add(Seq) = \sum_{k=0}^{|Seq|-1} cost\_add(T_k) \quad (2)$$

On utilise le procédé de calcul, dit Méthode de Monte Carlo pour le calcul de la moyenne des étapes et du cout additionnel. Les résultats sont présentés ci-dessous :

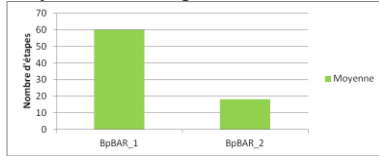


Figure 5. Moyenne du nombre d'étapes par procédé de reconfiguration dans COST 239

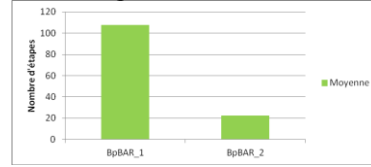


Figure 6. Moyenne du nombre d'étapes par procédé de reconfiguration dans NSF Net

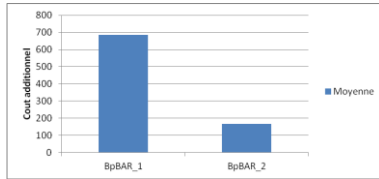


Figure 7. Moyenne du cout additionnel par procédé de reconfiguration dans COST 239



Figure 8. Moyenne du cout additionnel par procédé de reconfiguration dans NSF Net

BpBAR\_1 et BpBAR\_2 permettent de faire la reconfiguration d'arbre sans interruption de connexion en utilisant les convertisseurs de longueur d'onde et la propriété de basculement rapide des commutateurs considérés. L'algorithme BpBAR\_2 proposé réduit la durée de reconfiguration (Figures 5 et 6) et le cout des longueurs d'onde utilisées (Figures 7 et 8). Ces gains peuvent être réalisés sur tout réseau car les topologies utilisées pour l'évaluation sont fréquemment utilisées dans ce domaine.

## 5. Conclusion

La reconfiguration d'arbre multicast est étudiée dans cet article. Le problème qui est considéré est la détermination d'un algorithme qui fait la reconfiguration d'un arbre

initial vers un autre arbre final dans les réseaux optiques WDM. La reconfiguration doit se faire sans interruption de connexion, tout en réduisant la durée du processus et le coût des ressources additionnelles utilisées. Selon nos connaissances, les travaux existants dans la littérature ne parlent pas de reconfiguration d'arbre sans interruption de connexion. Une approche de résolution est d'utiliser une seule et même longueur d'onde durant toutes les étapes. Dans la deuxième approche, plusieurs longueurs d'onde sont utilisées pour la reconfiguration d'un arbre. Pour cette approche, l'algorithme BpBAR\_2 est proposé. Une étude de la qualité de l'opération de basculement sur plusieurs ports de sortie pourrait améliorer les performances de BpBAR\_2.

---

### 13. Bibliographie

- [1] A. Doria, F. Hellstrand, K. Sundell, T. Worster, "General Switch Management Protocol (GSMP) V3," *Internet RFC 3292*, 2002.
- [2] A. Farrel, J.-P. Vasseur, J. Ash, A Path Computation Element (PCE)-Based Architecture, *Request for Comments 4655 Old Dog Consulting*, 2006.
- [3]. B. Cousin, M. Molnar, "Fast reconfiguration of dynamic networks," *International Workshop on Dynamic Networks*, 2007.
- [4] B. Mukherje, "Wdm optical communication network: progress and challenges," *IEEE Journal on Selected Areas in Communication*, 2000.
- [5] I. Baldine, G.N. Rouskas, "Traffic adaptive WDM networks: a study of reconfiguration issues," *IEEE/OSA Journal of Lightwave Technology*, 2001.
- [6] J. Wu, "A Survey of WDM Network Reconfiguration: Strategies and Triggering Methods," *Computer Networks*, 2011.
- [7] M. Perényi, P. Soproni, T. Cinkler, D. Larrabei, "Regular Reconfiguration of Light-Trees in Multilayer Optical Networks," *12<sup>th</sup> Int. Conference on Optical Networking Design and Modeling (ONDM 2008)*, 2008.
- [8] N. Geary, N. Parnis, A. Antonopoulos, E. Drakopoulos, John O'Reilly, "The Benefits of Reconfiguration in Optical Networks," *10<sup>th</sup> International Telecommunication Network Strategy and Planning Symposium*, 2002.
- [9] T. Kárász, Z. Pándi, "Optimal reconfiguration of provisioning oriented optical networks" *proceedings of the 3<sup>rd</sup> International Working Conference on Performance Modeling and Evaluation of Heterogeneous Networks (HET-NETs)*, 2005.
- [10] X. Huang, F. Farahmand, T. Zhang, JP Jue, "Dynamic Multicast Traffic Grooming in WDM Networks with Reconfigurable Light-Trees," *Optical Society of America*, 2004.