Sub-tree Pair Selection for Reconfiguration of a Light-tree Pair

Amanvon Ferdinand Atta, Bernard Cousin, Joel Christian Adepo, Souleymane Oumtanaga

Abstract

Reconfiguration of unicast or multicast connections in an optical network is a critical task. Indeed, if it is not carried out correctly, it can lead to optical flow (also called flow) interruptions that can cause damage to the network operator. It is therefore common to perform the reconfiguration in several steps. In this study, we focused on multicast connection reconfiguration because multicast connection become more attractive and efficient technique to transmit flow of multicast applications. A Multicast connection in an optical network can be represented by a point-to-multipoint all-optical path called light-tree. In short, we explain how to select a pair of sub-trees (current sub-tree, new sub-tree) to be reconfigured at a given step of the reconfiguration process.

Keywords— Light-tree; Sub-tree Pair Selection; Flow Migration; Routing Reconfiguration; Optical Network

1 Introduction

Light-tree [1] reconfiguration problem consists in migrating an optical flow from the initial light-tree T_0 to the final light-tree T_f without interrupting the flow [2]. The initial light-tree and the final light-tree share wavelength channels. Therefore, this reconfiguration requires intermediate steps [3]. An intermediate step is the result of reconfiguring (which can use a spare wavelength) a pair of sub-trees (current sub-tree, new sub-tree). A spare wavelength [4] is a wavelength not required by the current light-tree T_c and the final light-tree T_f. Note that initially, the initial light-tree refers to the current light-tree. Since the spare wavelengths are scarce optical resources, these optical resources must be used parsimoniously. In the rest of this paper, the current sub-tree is denoted by ST_c (with $ST_c \subseteq T_c$) and the new sub-tree is denoted by ST_f (with $ST_f \subseteq T_f$).

Depending on whether or not a reconfigurable pair of sub-trees requires the use of spare wavelengths, there are two categories of reconfigurable pairs of sub-trees to consider here:

1. The category of sub-tree pairs with disjointed links: A pair of sub-trees (ST_c, ST_f) belongs to this category if ST_f does not share any link with ST_c. The establishment of ST_f does not require a spare wavelength [2].

2. The category of sub-tree pairs with shared links: A pair of sub-trees (ST_c, ST_f) belongs to this category if ST_f shares links (but not all links) with ST_c. The establishment of ST_f requires a spare wavelength [2].

In the following section, we describe how to select a sub-tree pair of each category.

2 Sub-tree pair with disjointed links

2.1 Selection of a sub-tree pair with disjointed links

A Sub-tree pair (a current sub-tree, a new sub-tree) with Disjointed Links (SDL) to be selected must be reconfigurable without flow interruption. Note that for such a sub-tree pair, the new sub-tree does not share links with the current sub -tree. This implies that the root node of such pair must be a divergent node. Also, all leaf nodes of each sub-tree constituting the pair to be selected must be convergent nodes. A divergent node belongs to the current light-tree T_c and the final light-tree

Algorithm 1 Select_SDL

Input: T_c, T_f, n // T_c : The current light-tree; T_f : The new light-tree; n : a divergent node of (T_c, T_f)
Output: ST_c, ST_f // ST_c : The current sub-tree; ST_f : The new sub-tree

1: CG_n = set of convergent nodes which are descendant nodes of n on T_f and T_c ;
2: $Dis\_CG_n = \{x \setminus x \in CG_n$ and the segment from n to x on T_c is with disjointed links to the segment from n to x on $T_f\}$;
3: $(ST_c, ST_f) = (null, null)$;
4: $CG(ST_c, ST_f) = null$; // The set of convergent nodes of sub-tree pair (ST_c, ST_f) to be selected
5: $OUT\_LINKS(T_c, n)$ = list of outgoing links emanating from n on T_c ;
6: **while** $CG(ST_c, ST_f) == null$ **and** $OUT\_LINKS(T_c, n))$ is not empty **do**
7:     out_link = first element of $OUT\_LINKS(T_c, n)$;
8:     $CG(ST_c, ST_f)$ = subset of Dis_CG_n such that the paths (on T_c) from n to the different elements of C share out_link ;
9:     **if** $\exists\ x \in CG(ST_c, ST_f)$ **and** $is\_interrupt(T_c, T_f, n, x)$ **then**
10:         $CG(ST_c, ST_f) = null$;
11:     **end if**
12:     $OUT\_LINKS(T_c, n) = OUT\_LINKS(T_c, n) \setminus \{out\_link\}$;
13: **end while**
14: **if** $CG(ST_c, ST_f)$!= $null$ **then**
15:     D_c = subset of $CG(ST_c, ST_f)$ such as each element of this subset is not ancestor of any element of $CG(ST_c, ST_f)$ on T_c;
16:     ST_c = sub-tree of T_c rooted at n and having D_c as set of leaf nodes ;
17:     D_f = subset of $CG(ST_c, ST_f)$ such as each element of this subset is not ancestor of any element of $CG(ST_c, ST_f)$ on T_f;
18:     ST_f = sub-tree of T_f rooted at n and having D_f as set of leaf nodes ;
19: **end if**
20: Return (ST_c, ST_f) // The sub-tree pair with disjointed links;

T_f and has at least one child node on T_c that is different from its child nodes on T_f. A convergent node belongs to the current light-tree T_c and the final light-tree T_f and its parent node on the current light-tree is different from its parent node on the final light-tree. Given a pair of light-trees (T_c, T_f) and a divergent node n, $Select\_SDL$ (Algorithm 1) returns a sub-tree pair with disjointed links which can be reconfigurable without flow interruption.

The process of selecting such a sub-tree pair (ST_c, ST_f) starts by searching the set of convergent nodes $CG(ST_c, ST_f)$ which must belongs to the sub-tree pair (refer from line 5 to line 13). If $CG(ST_c, ST_f)$ is empty then Algorithm 1 returns the empty pair of sub-trees. Otherwise, the current sub-tree ST_c is the sub-tree of the light-tree T_c which is rooted at n and covering $CG(ST_c, ST_f)$ (refer from line 15 to line 16). Also, the new sub-tree ST_f is the sub-tree of the light-tree T_f which is rooted at n and covering $CG(ST_c, ST_f)$ (refer from line 17 to line 18). When reconfiguring the sub-tree pair (ST_c, ST_f), the new sub-tree ST_f is fed by the optical flow and the current sub-tree ST_c is deleted. If this pair is not properly selected then a descendant node of n on T_c is deprived of the flow after this reconfiguration: i.e. the reconfiguration of a sub-tree pair with disjointed links (ST_c, ST_f) causes the flow interruption. To select properly the set of convergent nodes (ST_c, ST_f), the function $is\_interrupt(T_c, T_f, n, x)$ (refer to line 9) returns $True$ if one of the following conditions is fulfilled :

- Condition 1: The convergent node x has at least one ancestor y on T_c which is not belonging to T_f and y is a descendant of the divergent node n on T_c. Morever, y has a descendant n_y which is not an ancestor of x and which is not a descendant of n on T_f.

- Condition 2: The convergent node x has at least one ancestor y on T_c which is also belonging to T_f and y is a descendant of the divergent node n on T_c but it is not descendant of n on T_f.

- Condition 3: There is an ancestor y of convergent node x on the segment of T_f which connects the divergent node n to x such that y is not an ancestor of x on the segment of T_c which connects n to x.

2.2 Illustration of the selection of a sub-tree pair with disjointed links

We illustrate here, the selection of a sub-tree pair with disjointed links in an instance of pair of light-trees to be reconfigured. Figure 1 shows a problem instance of a pair of light-trees to be reconfigured from a multicast connection $(s, \{g, h\}, \lambda_0)$. The set of solid red links forms the current light-tree T_c rooted at the node s. The set of solid blue links forms the final light-tree T_f rooted at the node s. For this instance, we have only one divergent node denoted by a. Indeed, the node a belongs to the current light-tree and the final light-tree and b is a child node of the node a on the current light-tree but b is not a child node of the node a on the final light-tree. Let's apply Algorithm 1 with this pair of light-trees (T_c, T_f) and the divergent node a as input. e and f are convergent nodes descending from the divergent node a on both light-trees, i.e. $CG\_a = \{e, f\}$ (refer to line 1). The path from a to e on current light-tree is with disjointed links to the path which connects a to e on final light-tree. It is the same for node f. Therefore $Dis\_CG_a = CG\_a$. Link $a- > b$ is the single outgoing link emanating from a on T_c. Path on T_c from a to e and path on T_c from a to f share link $a- > b$. Also e and f do not fulfil any condition 1, 2 and 3. In fact, the set of ancestors (on current light-tree) of e which are descendant nodes of a are $\{b, d\}$. Also, the nodes b and d are not belonging to the final light-tree. But each of these nodes do not have descendant node which is not ancestor of e and do not a descendant node of node a on final light-tree. Therefore, node e does not fulfil the condition 1. In addition, $\{b, d\}$ are only on the current light-tree. Therefore, node e does not fulfil the condition 2. Similarly, f does not fulfil condition 1 and condition 2. The node e has no ancestor (belonging to both trees) on the segment (of T_f) from a to e which is not its ancestor on the segment (of T_c) from a to e: e does not fulfil condition 3. Similarly, f does not fulfil condition 3. So, at the end of the set of convergent nodes searching step (refer to line 13), $CG(ST_c, ST_f) = CG\_a = \{e, f\}$. Therefore, according to line 15 and line 16, the current sub-tree ST_c is equal to $\{a\{b\{d\{e, f\}\}\}\}$. The set of red dotted links forms ST_c. Also, according to line 17 and line 18, the new sub-tree ST_f is equal to $\{a\{k\{f\{e\}\}\}\}$. The set of blue dotted links forms ST_f. $\{a\{k\{f\{e\}\}\}\}$ does not share links with the current light-tree. Algorithm 1 ends and returns as pair of sub-trees $(ST_c, ST_f) = (\{a\{b\{d\{e, f\}\}\}\}, \{a\{k\{f\{e\}\}\}\})$ which is a sub-tree pair with disjointed links and which does not interrupt flow.

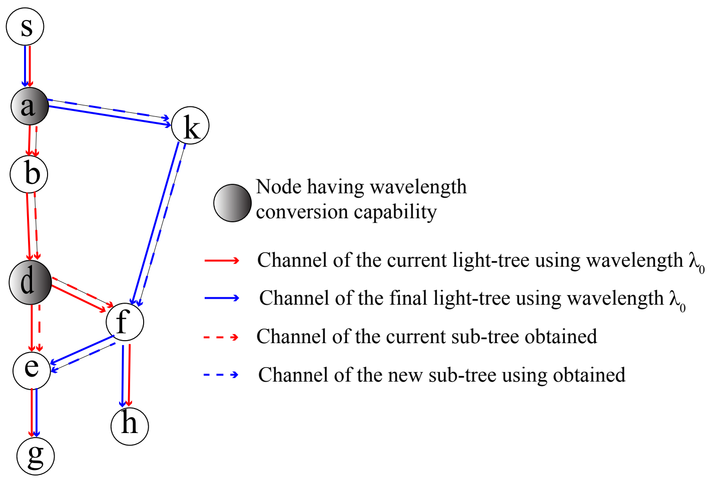

Figure 1: Problem instance containing a sub-tree pair with disjointed links

3 Sub-tree pair with shared links

3.1 Selection of a sub-tree pair with shared links

In order to select a Sub-tree pair with Shared Links (SSL), we propose the function called $Select\_SSL$ (see Algorithm 2). This function takes as input : the pair of light-trees (T_c, T_f) and a convergent node m. In fact, The existence of a sub-tree pair with shared links from a pair of light-trees implies the availability of at least one convergent node on the pair of light-trees. This function returns a sub-tree pair with shared links (ST_c, ST_f). The subset of destination nodes covered by ST_c and the subset of destination nodes covered by ST_f must be the same in order not to cause a flow interruption when reconfiguring this sub-tree pair with shared links.

The selection of a sub-tree pair with shared links (ST_c, ST_f) begins by the determination of the root node r of the sub-tree pair to be selected. r must belong to both sub-trees in order to allow flow transfer from ST_c to ST_f without flow interruption. By construction, r must be an ancestor of m on T_c and T_f . Also, (ST_c, ST_f) is a sub-tree pair with shared links, so the use of a spare wavelength must be required. Therefore, if r is different from the root node of the light-tree pair (T_c, T_f) then r belongs to set of nodes which has the wavelength conversion capability denoted by V_c (i.e. $r \in V_c$). Let $D(a, b, T)$ be represents the set of destination nodes of the tree T that have an ancestor on the segment from the node a to the node b. In short, the root node of a sub-tree pair with shared links (ST_c, ST_f) is equal to the node n (with $n \in V_c$) which is the youngest common ancestor of m on T_c and T_f such that $D(n, m, T_c) = D(n, m, T_f)$ (refer to line 1 of Algorithm 2). Note that if such a node n does not exist then the root node of the light-tree pair (T_c, T_f) is taken as the root node of the sub-trees pair to be selected.

Algorithm 2 Select_SSL

Input: T_c, T_f, m // T_c: The current light-tree;T_f : The new light-tree;m : a convergent node of (T_c, T_f)
Output: (ST_c, ST_f) // ST_c : The current sub-tree; ST_f : The new sub-tree
1: r = The node n of V_c which is the youngest ancestor on both T_c and T_f of m such that $D(n, m, T_c) = D(n, m, T_f)$ // V_c: The set of nodes which has the wavelength conversion capability ;
2: **if** $r == null$ **then**
3: $r = Root(T_c, T_f)$; // $Root(T_c, T_f)$: The root node of the light-tree pair (T_c, T_f)
4: **end if**
5: ST_c = The sub-tree of T_c rooted at r and having $D(r, m, T_c)$ as set of leaf nodes ;
6: ST_f = The sub-tree of T_f rooted at r and having $D(r, m, T_f)$ as set of leaf nodes ;
7: Return (ST_c, ST_f) ; // The sub-tree pair with shared links

After determining the root node and destination nodes of a sub-tree pair with shared links, the sub-trees can be selected. ST_c is the sub-tree of T_c rooted at the node r and having the elements of $D(r, m, T_c)$ as leaf nodes (refer to line 5). Similarly, ST_c is the sub-tree of T_f rooted at r and having as leaf nodes, the elements of $D(r, m, T_f)$ (refer to line 6).

3.2 Illustration of the selection of a sub-tree pair with shared links

In Figure 2, the set of solid red links forms the current light-tree T_c rooted at s. The set of solid blue links forms the final light-tree T_f rooted at s. a is a node having the wavelength conversion capability that is an ancestor of the convergent node g on both the current light-tree T_c and the final light-tree T_f. However, the set of destination nodes on the current light-tree T_c that have an ancestor on the segment from a to g is $D(a, g, T_c) = \{f, h, l\}$. The set of destination nodes on the final light-tree T_f which have an ancestor on the segment from a to g is $D(a, g, T_f) = \{f, h\}$. As $D(a, g, T_c) \neq D(a, g, T_f)$, we are looking for another ascendant of g. k is also a node having the wavelength conversion capability that is an ancestor of the convergent node g both on the current light-tree and on the final light-tree. In addition, $D(k, g, T_c) = D(k, g, T_f) = \{f, h, l\}$. Therefore, the root of the pair of sub-trees to select is node k. The current sub-tree ST_c is the sub-tree of T_c, rooted at k and covering the destinations l,

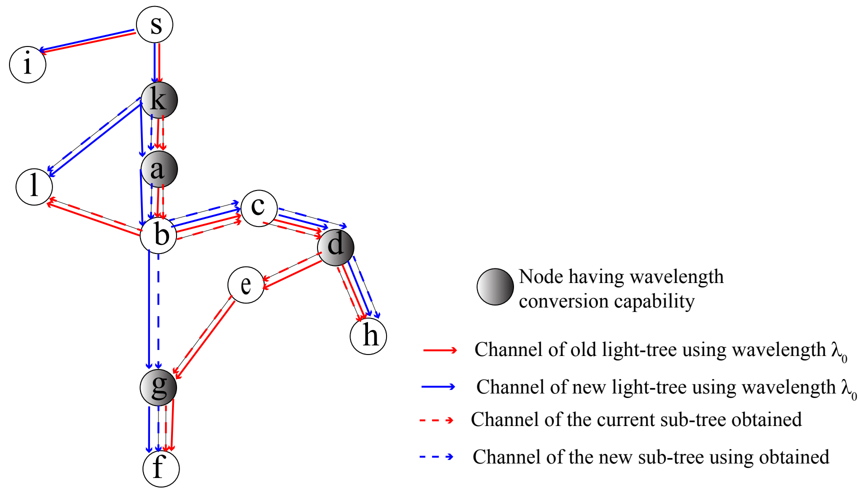

Figure 2: Problem instance containing a sub-tree pair with shared links

f and h. The new sub-tree ST_f is the sub-tree of T_f, rooted at k and covering the destinations l, f and h. The pair of sub-trees with shared links $(\{k\{a\{b\{l,c\{d\{h,e\{g\{f\}\}\}\}\}\}\}\},\{k\{l,a\{b\{c\{d\{h,e\{g\{f\}\}\}\}\}\}\}\})$ is then obtained as shown in Figure 2.a. The set of red dotted links forms the current sub-tree ST_c and the set of blue dotted links forms the new sub-tree ST_f.

4 Conclusion

Light-tree reconfiguration problem consists to migrate an optical flow from the initial light-tree to the final light-tree without interrupting the flow and use spare wavelength as few time as possible. For do this, light-tree reconfiguration process must be performed in several steps. At each step, a sub-tree pair belongs to a category of sub-tree pair with disjointed links or a category of sub-tree pair with shared links must be properly selected. Our work described in detail, how to select a sub-tree pair (with disjointed links or with shared links). Note that our work can be easily adapted to any study requiring the selection of a pair of sub-trees sharing (or not) links in a network that is not necessarily optical network. This work does not discuss how a sub-tree pair should be reconfigured or in what order the sub-tree pairs should be reconfigured ? To solve the reconfiguration problem, future works could be based on the current work and provide answers to these questions.

References

[1] Sreenath N, Murthy CSR, Mohan G. Multicast Routing in WDM Optical Networks. In: Du DZ, Raghavendra C, Ruan L, Du DZ., eds. *Optical Networks*. 6. Boston, MA: Springer US. 2001 (pp. 205–270)

[2] Cousin B, Adépo JC, Oumtanaga S, Babri M. Tree reconfiguration without lightpath interruption in WDM optical networks. *International Journal of Internet Protocol Technology* 2012; 7(2): 85–95.

[3] Adépo JC, Aka B, Babri M. Tree Reconfiguration with Network Resources Constraint. *International Journal of Computer Science and Telecommunications* 2016; 7(1): 4.

[4] Li H, Wu J. Survey of WDM network reconfiguration: topology migrations and their impact on service disruptions. *Telecommunication Systems* 2015; 60(3): 349–366. http://dx.doi.org/10.1007/s11235-015-0050-5 doi: 10.1007/s11235-015-0050-5