I R I S A
INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTÈMES ALÉATOIRES

PUBLICATION INTERNE
N° 1786

AN EFFICIENT MULTICAST PROTECTION SCHEME BASED ON A DUAL-FOREST

MOHAND YAZID SAIDI , BERNARD COUSIN AND MIKLOS MOLNAR

ISSN 1166-8687

IRISA
CAMPUS UNIVERSITAIRE DE BEAULIEU - 35042 RENNES CEDEX - FRANCE

INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTÈMES ALÉATOIRES
Campus de Beaulieu – 35042 Rennes Cedex – France
Tél. : (33) 02 99 84 71 00 – Fax : (33) 02 99 84 71 71
http://www.irisa.fr

An Efficient Multicast Protection Scheme based on a Dual-Forest

Mohand Yazid SAIDI * , Bernard COUSIN ** and Miklos Molnar ***

Systèmes communicants
Projet Armor

Publication interne n ° 1786 — Mars 2006 — 28 pages

Abstract: To provide fault-tolerance for multicast connections, different techniques of protection are developed. These techniques can be classed into reactive and pro-active approaches. Reactive approaches can have long recovery latency which is undesirable for many types of applications such as the real time ones.
In this paper, we focus on the multicast pro-active fault-tolerance schemes. One of the promising protection techniques is the dual-tree protection which is efficient to cope with single link failure but which cannot deal with node failures suitably. In this paper, we present an improved solution based on a dual-forest for multicast protection. Our proposition provides three improvements to traditional dual-tree protection. The first one concerns the capability to bypass both single link and node failures in a suitable and quick manner. The second increases the level of protection with the use of a forest as backup instead of a tree. The last permits the cost optimization of the dual-forest and of the delivery tree after recovery. Simulation experiments show that our protection scheme based on a dual-forest has better protection rate and causes less tree cost increase after recovery than the path-protection scheme.

Key-words: network, multicast, protection, dual-tree, dual-forest

(Résumé : tsvp)

* Mohand.saidi@irisa.fr
** Bernard.cousin@irisa.fr
*** Miklos.molnar@irisa.fr

Centre National de la Recherche Scientifique
(UMR 6074) Université de Rennes 1 – Insa de Rennes

Institut National de Recherche en Informatique
et en Automatique – unité de recherche de Rennes

An Efficient Multicast Protection Scheme based on a Dual-Forest

Résumé : Pour permettre une protection des communications multicast, différentes techniques de protection ont été développées. Ces dernières peuvent être classées en des schémas de protection réactive et en des schémas de protection proactive. Le premier type de schéma de protection induit des délais de récupération élevés et indésirables pour certains types d'applications notamment les applications multicast temps réel.

Dans ce rapport, nous nous focaliserons sur le second schéma de protection (proactive). Une des techniques prometteuses appartenant à cette classe de protection est l'arbre dual de protection. Cette technique contourne d'une manière efficace les pannes uniques des liens pour restaurer les communications multicast mais ne traite pas convenablement les pannes des nœuds. Ici, nous présenterons une meilleure solution basée sur une forêt duale. Notre proposition fournit trois améliorations au traditionnel arbre dual de protection. La première consiste à procurer une protection contre les pannes uniques de lien ou de nœud. La seconde augmente le niveau de protection avec l'utilisation d'une forêt comme secours au lieu d'un arbre. La dernière permet d'optimiser le coût de l'arbre multicast après récupération et le coût de la structure de routage utilisée pour le secours.
Les simulations montrent que notre technique de protection basée sur une forêt duale a un taux de protection plus élevé et un coût d'arbre multicast après récupération plus petit que ceux de la technique de protection par chemins disjoints.

Mots clés : Réseau, multicast, protection, arbre dual, forêt duale

1 Introduction

Due to the augmentation of delay sensitive network applications, fault-tolerance techniques become very important and necessary in the domain of routing. These techniques aim to ensure the non-interruption or minimal disruption of communications in a cost efficient manner (without traffic duplication). Thus, when a failure is detected, the protection techniques must determine quickly backup paths which will be used to bypass the failed component.

Existing protection schemes can be classed in two categories: reactive and pro-active. With reactive approaches, no computation of backup paths is needed before the failure. The restoration consists in finding and configuring new paths allowing the restoration of communications after the failure detection. However, in the pro-active approaches, the backup paths are pre-computed and possibly pre-configured beforehand. When a failure occurs, the protection mechanism switches from the primary affected paths to their backups.

In unicast IP protocols, failure detection leads to exchange of routing messages which will in turn cause the computation and the configuration of new paths getting round the failure. In some multicast IP protocols like MOSPF (Multicast Open Shortest Path First) [1], node or link failure will be interpreted as a change in network topology. As a result, the multicast forwarding information in the multicast routers will be updated in order to get new trees bypassing the failure. In some other multicast IP protocols like CBT (Core Based Tree) [2], only the affected part of the multicast tree (sub-tree downstream the failure point) is recomputed. Members belonging to the affected part leave the multicast tree and send a new *JOIN* request to the rendez-vous router in order to rejoin the tree via new paths.

All the above quoted protocols use reactive protection since new paths will be computed after the failure detection. Such type of protection works well enough for datagram communications and has the advantages of flexibility to cope with topology changes and decreasing the computational and maintenance costs. However, recovery latency measured using such type of protection is long and undesirable for many types of communications. In high-speed networks, long recovery from failure is very awkward and causes the loss of much data. Moreover and since efficient pre-reservation of bandwidth is not possible in the reactive protection techniques, the recovery can fail because the amount of available bandwidth on backup paths is not sufficient to receive the traffic of their affected primary paths.

To get around the previous problems, the tendency is actually to envisage the use of pro-active protection schemes. In such type of protection, the recovery is faster because the backup paths are pre-computed and generally pre-configured. Moreover, the pre-configuration of backup paths ensures the sufficiency of bandwidth and therefore the success of the recovery. Many pro-active techniques are developed for unicasting. In end-to-end protection, the whole path between the source and the destination is protected by only one vertex-disjoint backup path [3][4]. In one-to-one backup protection, the primary path is divided into a set of segments; each one has its own backup path and can be as small as a link [3][5].

For multicast communication, the routing structure to protect is a tree. Protection is in this case more challenging to achieve than in the unicast case since one network failure will affect all members downstream the failing component in the multicast tree.

The first trivial proposition for multicast pro-active protection scheme suggests to extract from the multicast tree all the paths from the root to each member and to protect each path with a unicast pro-active scheme [6][7]. The problem with this proposition is that it doesn't guarantee neither the non-duplication of the traffic over links, nor the non formation of loops after recovery.

In recent works, the tendency is to search a backup tree which is capable to protect all nodes and links of the primary tree. In [8], a dual-tree scheme for multicast fault-tolerant is proposed. In this protection technique, a new tree is built by the interconnection of the primary tree leafs without the use of any link or inner node of the primary tree. This tree provides paths which can be used to bypass failures. As we will see in the next section, the proposed dual-tree scheme can protect only against single link failure. In this paper, we propose a protection scheme based on a dual-forest. This forest, which provides the backup paths, interconnects the maximum number of primary tree leafs without the use of any link or inner node of the primary tree. Our solution makes several improvements to the dual-tree protection scheme. Indeed, it can cope with both link and node failures successfully, increases the level of protection and optimizes both the cost of delivery tree after recovery and the cost of backup paths (dual-forest).

The rest of this paper is organized as follows. In section 2, we review works related to multicast pro-active protection schemes and outline their insufficiencies. In section 3, we describe our proposition which improves the dual-tree scheme. Simulation results are presented and discussed in section 4. The last section is dedicated to the conclusions.

2 RELATED WORKS

Pro-active protection techniques for multicast can also be grouped in two categories: global level and local level [3].

In the global level protection techniques, the switching from primary to backup paths when a failure is detected is done by the source node. As a result, the failure notification message must reach the source node in order to start recovery procedure. In the case of link or node failure far from the source node, the delay may be significant and undesirable. However, in local level protection techniques, the recovery is faster because the activation of backup paths is triggered by the node detecting the failure (or by one of its upstream nodes) which is nearer to the failed component.

In all the following sections, the network topology is represented by a directed graph *G*. The set of destination nodes *Mi* corresponds to the set of multicast group members which is noted *M*. The source node and the primary tree are noted S and T_p respectively.

In the rest of this section, we give a survey of the existing pro-active protection techniques for multicast (firstly at a global level protection and secondly at a local level) with the mention of their advantages and drawbacks.

2.1 Global level protection

2.1.1 Path protection

This protection technique is inspired from unicast. For each path of the primary tree from the source to a multicast group member, a vertex-disjoint path connecting the source to the destination is pre-computed and possibly set up as backup [3][4]. In Fig. 1(a), the primary path (S, C, M1) (resp. (S, C, D, M2)) connecting the source node to the multicast group member M1 (resp. M2) is protected by the vertex-disjoint backup path (S, A, B, M2, D, M1) (resp. (S, A, B, M2)).

When a failure is detected, the source node which is informed determines all the affected multicast group members (which correspond to the multicast group members belonging to the sub-tree newly disconnected from the source node) and activates their backup paths by sending multicast traffic on the (residual) primary tree and on the activated backup paths.

In Fig. 1(b), after the failure detection of link (C, M1), the source node S concludes that only the member node M1 is affected and activates its backup path (S, A, B, M2, D, M1). However and in order to avoid possible loops resulting from the activation of new backup paths (see link (D, M2) in Fig. 1(b)), the node source should send multicast traffic on each activated path and on the (residual) primary tree with different connection identifiers.

This protection technique is easy to implement but presents several drawbacks:

1. it doesn't guarantee the non-duplication of packets over links. This point is very important since multicast traffic duplication is not always possible because of lack of bandwidth;

2. a recovery procedure can imply the reconfiguration of several paths. For instance, a failure of link (S, C) in Fig. 1(a) involves the reconfiguration of two backup paths: (S, A, B, M2) and (S, A, B, M2, D, M1);
3. the cost of backup paths is not optimal (in Fig. 1(b), the path (S, C, D, M1) is better than the backup path chosen by the path protection technique);
4. the existence of backup paths depends on the chosen primary tree.

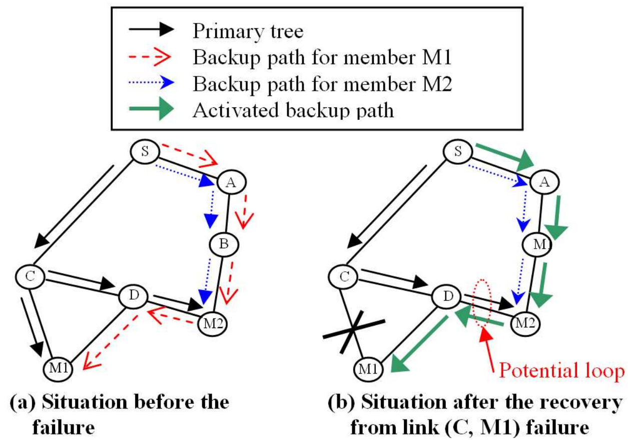


Fig. 1: Path disjoint protection

2.1.2 Redundant tree Protection

The (vertex-)redundant tree consists in a directed tree covering all nodes of ($\{S\}$ U M) and not containing any link or any another node of the primary tree [9] (Fig. 2). When a failure is detected, the source switches from the primary to the redundant tree.

This technique is simple to implement and facilitates the recovery procedure because it is common to all node or link failures. However, the cost of the redundant tree is very significant and awkward. Moreover, the existence of a covering redundant tree requires very restricting conditions (a high graph redundancy ratio) on the network graph. This limits the applicability of this technique.

2.2 Local level protection

2.2.1 One-to-one backup protection

This protection technique also is inspired from unicast. Each node along the primary path (from the source to a given multicast group member) has a backup path getting around its

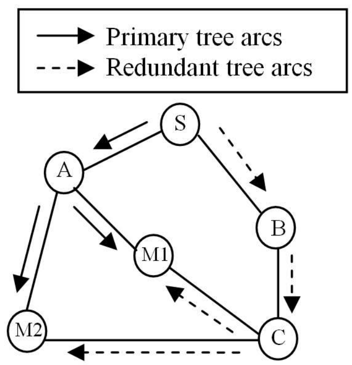


Fig. 2: Redundant tree protection

downstream node (the downstream node corresponds to the next hop on the primary path) [3][5]. In Fig. 3, node B is protected against failure of node D by the backup path (B, A, M1).

In some cases, a node must have several backup paths to cope with the failure of its any downstream node on the primary tree. In Fig. 3, S has two backup paths: (S, A, D) and (S, B, M2) to cope with possible failure of node B so that both members M1 and M2 continue to receive multicast packets.

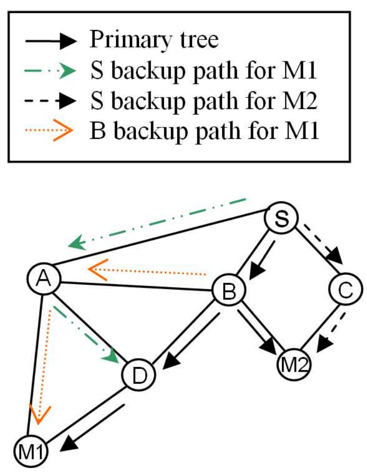


Fig. 3: One-to-one backup protection

In Fig. 3, if node D fails, node B will take locally the decision to activate the backup path (B, A, M1) in order to reach the multicast group member M1.

This solution has a very important advantage: it requires only a bi-connected graph to be sure (to protect all links and nodes). A graph is said to be bi-connected if there exists two vertex-disjoint paths between any two vertices in the graph.

However and like the path protection technique, one-to-one backup protection doesn't guarantee the non-duplication of packets over links. Moreover, this technique generates much control packets to build and maintain backup paths. Furthermore, the information necessary to the recovery procedure is very significant (one or more backup paths per primary tree node) and can pose problems to its storage and its searching (scalability).

2.2.2 Dual-tree protection

The dual-tree [8] is built by the interconnection of all primary tree leafs without the use of any link or inner node of the primary tree (so the dual tree is vertex-disjoint with the primary tree except primary tree leafs). The source node is regarded as a leaf if it has only one child in the primary tree.

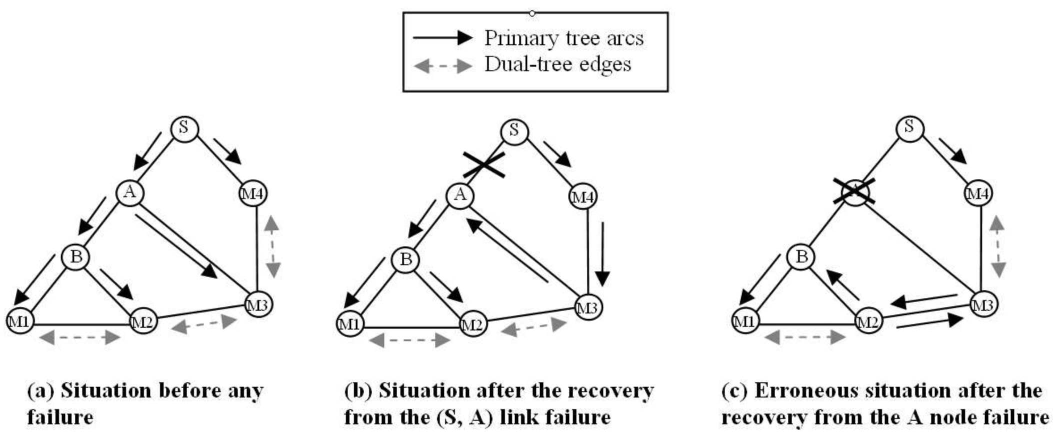


Fig. 4: Dual-tree protection

In Fig. 4(a), a primary tree is built to convey multicast traffic to the different members (M1, M2, M3, M4) and a dual-tree is pre-computed to deal with failures. The dual-tree is obtained by the interconnection of the four primary tree leafs M1, M2, M3 and M4 without the use of any link or inner node of the primary tree.

When a node detects a failure on its upstream interface, it runs the restoration algorithm of the dual-tree to repair the multicast communication. In Fig. 4(b), node A detects a failure on its upstream interface and starts recovery procedure.

Firstly, it divides the set of primary tree leafs into two sets: affected nodes (AF_A) and unaffected nodes (NA_A). Affected nodes consist in primary leaf nodes which belong to the primary sub-tree rooted at A and unaffected nodes correspond to the rest of primary tree leaf nodes. Here: AF_A = {M1, M2, M3}, NA_A = {M4}.

Secondly, node A determines one path interconnecting directly (or via intermediary nodes which are not a primary leafs) in the dual tree one node of AF_A to one node of NA_A. The path consisting in link (M3, M4) is returned.

Finally, node A sends a *Reconfig message* to node M3 which will be its new parent. When node M3 (which is a descendant of A in the old primary tree) receives the Reconfig message, it forwards the massage to M4 and changes its old parent (which is A) to be a child. At a reception of Reconfig message by node M4 (which is an unaffected node), it creates a new multicast state information to serve member node M3.

More formally, the specification of the restoration algorithm of dual-tree can be summarized in the following steps:

1. When a node x detects a failure on its upstream interface, it computes the two sets AF_x and NA_x. Remember that AF_x is composed of primary leafs which belong to sub-tree rooted at x and NA_x corresponds to the rest of primary tree leafs.

2. Node x selects one node y in AF_x which is connected to a node z in NA_x using a path p. The inner nodes of p must not belong to (AF_x U NA_x).

3. Node x sends a Reconfig message to its child leading to y, updates its parent to be that child. Reconfig message will be forwarded in the primary tree until it reaches y, every node receiving this message changes its old parent to be a child, and its old child (on the path to y) becomes its new parent. Reconfig message is then forwarded to z along path p and corresponding multicast information states are installed in the intermediate node(s).

Contrarily to the previous protection techniques and since the source node has no particular role in the dual-tree protection, this technique can be used in both source specific multicast (SSM) model and in any source multicast (ASM) model. For simplicity, we presented here only the directed SSM tree model of this protection which could be easily generalized to the undirected ASM tree model.

Note that, in dual-tree protection, nodes must know (or deduce) the structures of primary and secondary trees. One way to achieve this is to distribute the topology and group membership information to all the multicast routers. Moreover, all links of the different dual-trees must be bi-directional since we assumed that dual-tree is undirected.

The dual-tree protection is sure in the link failure cases but protection is not guaranteed in the node failure cases. Fig. 4(c) shows a case of node failure where recovery procedure fails to repair multicast communication. In the figure, node A fails. As a result, node M3 and B detect a failure on their upstream interfaces and start recovery procedure. Let suppose M3 is the first node discovering the failure. M3 computes its AF_{M3} and NA_{M3} sets (AF_{M3} = {M3}, NA_{M3} = {M1, M2, M4}), selects a path (M3, M2) to get around the failure and sends Reconfig message to node M2. Then in same manner, B computes its AF_B and NA_B sets (AF_B = {M1, M2}, NA_B = {M3, M4}), chooses the only path (M2, M3) allowing the interconnection of one node in AF_B to one node in NA_B and sends the Reconfig message to M3. The recovery procedure ends with the configurations illustrated in Fig. 4(c) in which members M1, M2 and M3 are disconnected from the source multicast.

Note that we supposed in the example of Fig. 4 that nodes are not able to differentiate node failures from link failures. This is generally the case in actual networks. However and even if nodes are capable to determine that it is a node failure, the dual-tree protection cannot be sure. Indeed, nodes M3 and B of Fig. 4(a) compute their affected set and unaffected set which are the same ($AF_{M3,B}$ = {M1, M2, M3}, $NA_{M3,B}$ = {M4}) and use the same path (M3, M4) to bypass the failure. As a result, some members (M1 and M2 in Fig. 4(a)) will not receive the multicast traffic because they are not connected to the source node.

3 DUAL-FOREST PROTECTION

An efficient protection technique will cope with both link and node failures without creation of loops and does not require very restricting conditions on the topology to be applicable. Hence, all protection techniques presented in the previous section are not practical since none of them verifies all the above mentioned criteria. In this section, we propose an efficient multicast protection scheme based on a dual-forest (or the dual-forest protection scheme) which not only satisfies the previous criteria but which has also very significant advantages. It has a good protection rate and it does not require many control messages to achieve the restoration (section 4).

Like the dual-tree protection technique, the dual-forest protection scheme uses a reduced topology to build the backup paths. This reduced topology is obtained by the elimination in the global topology of all links and inner nodes of the primary tree. Hence and since the reduced topology can be unconnected, we use a forest instead of a tree to best protect the multicast communication.

In sub-section 3.1, we present the restoration algorithm of our dual-forest protection. Thanks to the modifications made to the algorithm presented in section 2.2.2, this algorithm can cope successfully with both node and link failures. In sub-section 3.2, we ascertain and prove that the *KMB forest* covering all primary leafs is a good backup structure (the union of backup paths) which optimizes both the cost of backup paths and the multicast tree cost increase after recovery. We define a KMB forest as a forest of *KMB trees* on each connected component and we point out that a KMB tree [10] is an approached Steiner tree built by the interconnection of the closest terminal nodes (primary tree leafs) until forming a tree.

3.1 Restoration algorithm of our dual-forest protection

To be able to deal with node failures, we propose some improvements to the algorithm described in section 2.2.2. For simplicity, we present the directed tree model (SSM) of this protection which can be generalized to the undirected tree model (ASM). We give below some definitions necessary to its understanding.

When a node x detects a failure on its upstream interface in the primary tree, it divides the set of primary leafs into three sets: surely affected nodes SA_x, possibly affected nodes PA_x and unaffected nodes NA_x.

Surely affected nodes correspond to leafs of the primary sub-tree rooted at x. Possibly affected nodes consist in leafs of the primary sub-tree rooted at the parent of x and not belonging to the set of surely affected nodes. Unaffected nodes correspond to the rest of the primary tree leafs.

The idea of the restoration algorithm of the dual-forest protection scheme is to give priority to backup paths[1] interconnecting nodes of SA_x to nodes of NA_x. Indeed, these paths are sure and can deal with both node and link failures. We recall that during normal operation, traffic is delivered through the primary tree. The dual-forest which provides

[1] The backup paths form the dual-forest

Algorithm 1. Dual-forest algorithm executed by node x detecting a failure

1. **deduce_sets**$(x, T_p, SA_x, PA_x, NA_x)$; {
The node x which detects the failure deduces the three sets SA_x, PA_x and NA_x relating to the primary tree T_p }

2. **deduce_backup_path**$(F_d, SA_x, PA_x, NA_x, bp_x)$; {
deduces a shortest path bp_x which belongs to the dual forest F_d and which interconnects one node in SA_x to one node in NA_x
if such path does not exist, a shortest path interconnecting one node in SA_x to one node in PA_x is assigned to bp_x
if no path is determined, an infinite path is assigned to bp_x }

3. **if** $bp_x =$ *infinite path* **then goto** end ; **endif** {
recovery with the dual-forest protection fails }

4. **split_backup_path**$(PA_x,\ bp_x)$; {
if bp_x includes a node of PA_x then a shortest sub-path interconnecting the first extremity node of bp_x (which belongs to SA_x) to the closest node of PA_x is assigned to bp_x
do nothing if bp_x does not include any node of PA_x }

5. **extremities**(bp_x, e_1, e_2) ; {
the first extremity of the path bp_x belonging to SA_x is returned in the parameter e_1
the second extremity of the path bp_x is returned in the parameter e_2 }

6. **create_Reconfig_message**(x, T_p, bp_x, r_msg) ; {
all nodes on the path in T_p from x to extremity e_1 are listed in the Reconfig message r_msg, in this order
These nodes will be followed by nodes of bp_x (from e_1 to e_2), in order too }

7. **send_Reconfig_message**(x, r_msg) ; {
r_msg will be sent to the **succ**(x, r_msg) node which is the successor of x in the Reconfig message list }

8. **parent**$(x, T_p) \leftarrow$ **succ**(x, r_msg) ; { the successor of x in r_msg list becomes its new parent in the delivery tree T_p }

9. **del_child**$(x, T_p,$**succ**$(x, r\_msg))$;
{ the **succ**(x, r_msg) node is deleted from the child list of x in T_p }

Algorithm 2. Dual-forest algorithm executed by node y receiving a Reconfig message

1. **if** y ! $= e_2$ **then** {
   e_2 is the last node in the Reconfig message list }
   send_Reconfig_message(y, r_msg) ; {
   r_msg will be sent to the **succ**(y, r_msg) node }
   parent$(y, T_p) \leftarrow$ **succ**(y, r_msg) ; {
   the parent of y in T_p will be the **succ**(y, r_msg) node }
   del_child$(y, T_p,$ **succ**$(y, r\_msg))$; {
   the **succ**(y, r_msg) node is removed from the child list of y in T_p }
   endif
2. **add_child**$(y, T_p,$ **pred(y, r_msg)**$)$; {
   the predecessor of y in r_msg list becomes a new child of y in T_p }

the backup paths interconnects primary leafs in the reduced topology. The number of trees forming this dual-forest must be equal to the number of connected components in the reduced topology.

The restoration algorithm of the dual-forest works as follows:

1. When node x of the primary tree detects a failure on its upstream interface, it runs the routine depicted in Algorithm 1.
2. When node y receives the Reconfig message, it runs the routine depicted in Algorithm 2.

In addition to the enhancement allowing the recovery from both single link and node failures, the dual-forest protection scheme produces other improvements. Firstly, the choice to use a forest instead of a tree as backup increases the protection (section 4.3.1). Indeed, if it is not possible to determine a tree which interconnects all primary leafs (the reduced topology is not connected), the use of a forest will allow a protection of some parts of the primary tree. These protected parts consist in the sub-trees whose leafs are completely interconnected by one of the (backup) trees belonging to the dual-forest. Secondly, the restoration delay can be decreased since steps 1 up to 6 of Algorithm 1 can be performed before the detection of a failure. Indeed, each node of the primary tree can suppose the failure of its upstream interface and carry out the computations necessary to quickly restore the multicast communication.

Let us illustrate the operation of the restoration algorithm of the dual-forest protection by applying it to the network of Fig. 4(a). Suppose node A fails then, nodes M3 and B will detect the failure on their upstream interfaces. Let suppose that M3 is the first to start its computations (recovery procedure).

Recovery procedure for M3

As it is described above, when a node detects a failure on its upstream interface, it runs the routine depicted in Algorithm 1. Thus:

M3 execution trace of Algorithm 1

1. $SA_{M3} \leftarrow$ {M3}, $PA_{M3} \leftarrow$ {M1, M2}, $NA_{M3} \leftarrow$ {M4}
2. $bp_{M3} \leftarrow$ (M3, M4) { (M3, M4) is the only path interconnecting one node in SA_{M3} to one node in NA_{M3} }
3. do nothing { because bp_{M3} != $infinitepath$ }
4. do nothing { because bp_{M3} does not include any node of PA_{M3} }
5. $e_1 \leftarrow$ M3, $e_2 \leftarrow$ M4
6. Reconfig message is created. The list of nodes included in the message is: [M3, M4]
7. Reconfig message is sent to node M4 which is the successor of M3 in the Reconfig message list.
8. parent(M3, Tp) ← M4 { node M3 sets M4 as its parent in Tp }
9. del_child(M3, Tp, M4) { do nothing because M4 is not a child of M3 in Tp }

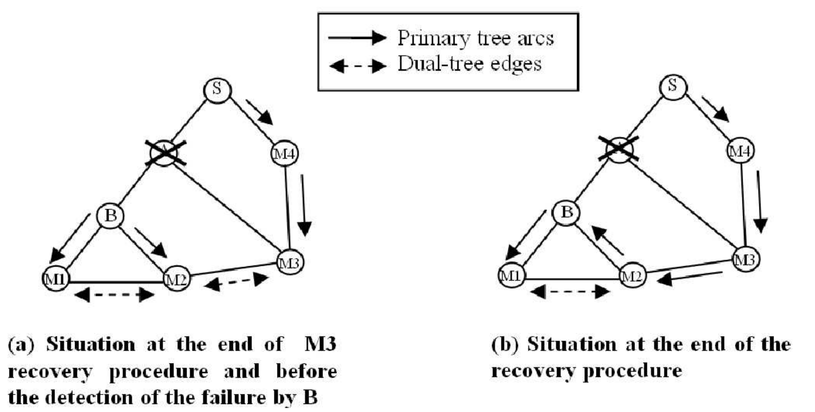


Fig. 5: Dual-forest protection scheme

Since M3 sends a Reconfig message to M4 (in step 7), this latter will receive the message and will run the routine depicted in Algorithm 2 accordingly to the restoration algorithm of the dual-forest protection.

M4 execution trace of Algorithm 2

1. do nothing { because M4 = e_2 }
2. add_child(M4, T_p, M3) { M3 becomes a child of M4 in T_p }

At this point of execution, the M3 recovery procedure ends with the tree structures shown in Fig. 5(a). Independently from the time when M3 starts and ends its recovery procedure, B will detect the failure on its upstream interface and starts the computations relating to the recovery procedure in order to cope with the failure.

Recovery procedure for B

Like node M3, node B will run the routine depicted in Algorithm 1:

B execution trace of Algorithm 1

1. $SA_B \leftarrow$ {M1, M2}, $PA_B \leftarrow$ {M3}, $NA_B \leftarrow$ {M4}
2. $bp_B \leftarrow$ (M2, M3, M4) { (M2,M3, M4) is the shortest path which interconnects one node in SA_B to one node in NA_B }
3. do nothing { because bp_B != $infinitepath$ }
4. $bp_B \leftarrow$ (M2, M3) { because bp_B includes node M3 which belongs to PA_B }
5. $e_1 \leftarrow$ M2, $e_2 \leftarrow$ M3
6. Reconfig message is created. The list of nodes included in the message is: [B, M2, M3]
7. Reconfig message is sent to node M2 which is the successor of B in the Reconfig message list.
8. parent(B, T_p) $\leftarrow$ M2 { node B sets M2 as its parent in T_p }
9. del_child(B, T_p, M2) { M2 is removed from the child list of B in T_p }

When node M2 receives the Reconfig message sent by node B, it runs the routine depicted in Algorithm 2 accordingly to the restoration algorithm of the dual-forest protection.

M2 execution trace of Algorithm 2

1. Reconfig message is sent to node M3 which is the successor of M2 in the Reconfig message list
   parent(M2, T_p) $\leftarrow$ M3 { node M2 sets M3 as its parent in T_p }
   del_child(M2, T_p, M3) { do nothing because M3 is not a child of M2 in T_p }

2. add_child(M2, T_p, B) { B becomes a child of M3 in T_p }

As in step 1 of Algorithm 2, node M2 forwards the Reconfig message to node M3, this latter node runs also Algorithm 2:

M3 execution trace of Algorithm 2

1. do nothing { because M3 = e_2 }
2. add_child(M3, T_p, M2) { M2 becomes a child of M3 in T_p }

At this point of execution, the recovery procedure ends with success. Indeed, the configuration results illustrated in Fig. 5(b) show that the source node can deliver multicast packets to all multicast group members through the tree composed of links: (S, M4), (M4, M3), (M3, M2), (M2, B), and (B, M1).

In step 2 of Algorithm 1, notice that the purpose of searching a shortest path interconnecting one node in SA_x to one node in PA_x is to allow link protection where node protection is impossible.

3.2 An efficient heuristic for dual-forest computation

3.2.1 Introduction

The restoration algorithm of the dual-forest protection uses paths interconnecting primary leafs for recovery. These paths which form the backup structure are built with only the use of nodes and links of the reduced topology. Thus and in order to maximize the level of protection with the dual-forest protection scheme, the number of connected components of the backup structure must be equal to the number of connected components of the reduced topology.

In this section, we endeavor to determine an efficient heuristic to pre-compute the backup structure which does not decrease the protection level (the backup structure must provide the same protection level as that of the reduced topology). The heuristic has the objective of optimizing the two following criteria under the constraint of maximizing protection: the cost of the backup structure (criterion 1) and the cost increase of the multicast tree after recovery (criterion 2).

To optimize the first criterion without decreasing the level of protection, the backup structure must correspond to a Steiner forest. We recall that a Steiner forest is a forest of Steiner trees on each connected component (one tree per connected component) [11]. As the determination of a Steiner tree is an NP-complete problem, we approximate the solution with the use of a Steiner tree heuristics.

To optimize the second criterion without decreasing the level of protection, the backup structure must include for each possible failure affecting the primary tree, at least one of backup path sets which minimizes the cost increase of the multicast tree after recovery. In the dual-forest protection scheme case, the multicast tree cost after recovery is equal to the residual primary tree cost plus the cost of path(s) bp_xi deduced by the restoration algorithm of the dual-forest protection (Algorithm 1 of section 3.1). Since the residual primary tree cost is given and depends only on the failure, the optimization of the multicast tree cost increase involves the cost minimization of the path(s) bp_{xi} chosen by Algorithm 1 of section 3.1.

There exist different backup structures belonging to the reduced topology which include a set of backup paths minimizing the multicast tree cost increase after recovery. For instance, the backup structure corresponding to the reduced topology graph is a trivial solution which is not interesting since its cost is very high. Here, we search for one backup structures (BS_{opt}) which optimizes the above two criteria without decreasing the level of protection. Thus, the heuristic may determine an approached Steiner forest (criterion1) which includes backup paths minimizing the cost increase of the multicast tree after recovery (criterion 2). To determine (BS_{opt}), we distinguish the link failure case from the node failure case. Whereas in the link failure case it must be made sure that a path of minimal cost allowing the recovery belongs to (BS_{opt}), it is necessary to be sure in the case of node failure that a Steiner tree (a set of paths) allowing the recovery belongs to (BS_{opt}).

3.2.2 Link failure case

In case of link failure detected by node x, the backup path minimizing the multicast tree cost increase and allowing the recovery is the shortest path belonging to the reduced topology which interconnects one node in SA_x to one node in (NA_x U PA_x); therefore, such path (or an equivalent path in cost) must belong to BS_{opt}. As x can be any node (different from the source) of the primary tree, we conclude that BS_{opt} must include for each value of x, one shortest path interconnecting one node in SA_x to one node in (NA_x U PA_x). The property below ensures that BS_{opt} in the case of link failure corresponds to the minimal KMB forest which covers all primary leafs in the reduced topology.

Property Let G an undirected graph and let T be a tree built on G. Let $\{E_1, E_2\}$ be a partition of the set of T leafs. The cost of the shortest path interconnecting one node in E_1 to one node in E_2 in the reduced graph (obtained by the elimination of all links and inner nodes of T) is equal to the cost of the shortest path interconnecting one node in E_1 to one node in E_2 in the KMB forest which covers all T leafs.

Proof By construction:

Suppose that the cost of the shortest path interconnecting one node of E_1 to one node of E_2 in the reduced graph is c and suppose that all paths of cost lower than c (which do not create loops) are already added to the building KMB forest. At this moment, we know that no node of E_1 is connected to node of E_2. In the next steps of the KMB forest building, a path of cost c will be chosen to interconnect one node of E_1 to one node of E_2 since the nodes of the 2 sets are completely disjoint (no risk of loop formation).

It is trivial that if there is no path interconnecting the two sets in the reduced graph, there will be no path interconnecting them in the KMB forest. ∎

Since $\{SA_x, NA_x \text{ U } PA_x\}$ forms a partition of T_p leafs independently of values of x, we deduce from the property above that the cost of the shortest backup path allowing recovery in the reduced topology is equivalent to the cost of the shortest backup path allowing recovery in the KMB forest. Hence, if the backup structure used for protection corresponds to the KMB forest which is an approached Steiner forest (criterion 1), one backup path minimizing the multicast cost increase will be a candidate for recovery (criterion 2). This path is often selected but there are cases where we prefer another path in order to ensure recovery.

3.2.3 Node failure case

In case of failure of the parent of a node x in the primary tree, the structure which minimizes the multicast tree cost increase and which repairs the multicast communication consists in branches (forest) to add to the residual primary tree so that the resulting tree covers again all nodes of set $(SA_x \text{ U } PA_x)$ and the multicast tree cost increase is minimal.

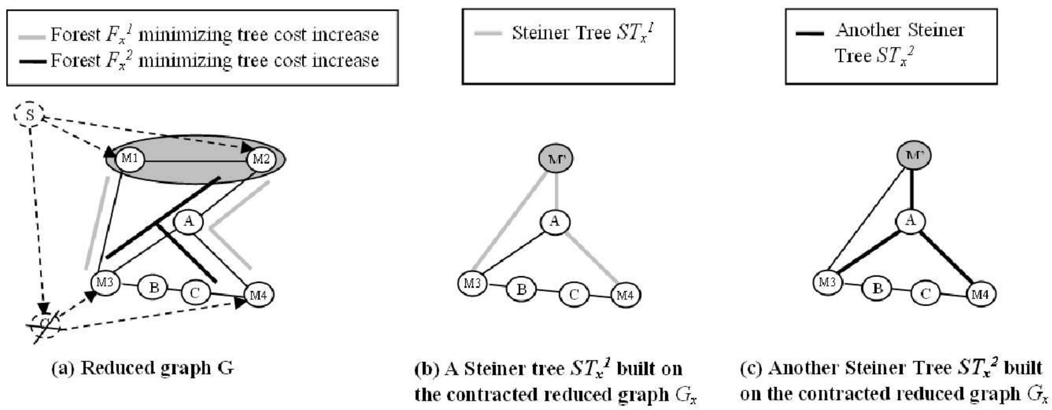


Fig. 6: Reduced graph transformation

Fig. 6(a) shows a reduced graph G (continuous lines) resulting from the elimination of all links and inner nodes of a primary tree (dashed arrows). Nodes M1 and M2 constitute the $NA_{x(x=M3 \text{ or } x=M4)}$ set, and nodes M3 and M4 constitute the $(SA_x \text{ U } PA_x)$ set. The forests in G connecting all nodes of $(SA_x \text{ U } PA_x)$ set to nodes of NA_x set and which have minimal cost are the best solutions allowing the recovery from the failure. We show below that determining these forests in G corresponds to finding a Steiner trees built on another graph G_x (Fig. 6(b) and Fig. 6(c)). This graph G_x is deduced from G by the contraction of the NA_x nodes into one contracted leaf node M'. Thus, link (M1, M3) (resp. link (M2, A)) in G corresponds to link (M', M3) (resp. link (M', A)) in G_x and link (M1, M2) in G is removed in G_x. We can verify easily that the forests $F_{xi(i>0)}$ which connect all nodes of

(SA_x U PA_x) set to nodes of NA_x set and which have minimal cost in G correspond to a Steiner trees $ST_{xi(i>0)}$ in G_x which covers all nodes of (SA_x U PA_x U {M'}).

For simplicity, we consider that all links in Fig. 6 have the same cost. In Fig. 6(b), the Steiner tree ST_{x1} represented by bold gray edges which covers M3, M4, M' in G_x is constituted of links (M', A), (A, M4), and (M', M3). The corresponding forest F_{x1} in G is composed of links (M2, A), (A, M4), and (M1, M3). Another Steiner tree ST_{x2} represented by bold black edges in Fig. 6(c) which corresponds to another forest F_{x2} can be determined. ST_{x2} consists in links (M', A), (A, M3) and (A, M4) and the corresponding F_{x2} is composed of links (M2, A), (A, M3) and (A, M4). Hence, links of one of the two forests F_{x1} or F_{x2} must belong to the global optimal backup structure BS_{opt} in order to optimize the multicast tree cost increase after recovery.

Suppose now that the failed node is the parent of another node y (the father of x is unscathed) then, other Steiner trees $ST_{yi(i>0)}$ will be determined in G_y (and correspondingly other $F_{yi(i>0)}$ in G) so that one of the forests $ST_{yi(i>0)}$ belongs to the optimal backup structure BS_{opt}.

We conclude that the optimal BS_{opt} must correspond to a Steiner forest (criterion 1) and must include for each node x of the primary tree (different from the source node) at least the links of one forest $F_{xi(i>0)}$ (criterion 2). Recall that determining a forest $F_{xi(i>0)}$ is an NP-complete problem since it corresponds to a Steiner tree computation on graph G_x.

In general, there is not an exact solution BS_{opt} which minimizes the two criteria above together. Thus and due to the NP-completeness of the problem to find the Pareto's zone[2] (Steiner tree problem), we search for a heuristic optimizing the two criteria together.

To approximate a solution optimizing the first criterion, the heuristic may determine an approached Steiner forest whose Steiner nodes are the primary tree leafs. To approximate a solution optimizing the two criteria and since each tree of the different forests $F_{xi(i>0)}$ corresponds to a Steiner tree[3] in the reduced topology, any sub-tree of the approached Steiner forest whose leafs are a primary leafs[4] must be an approached Steiner tree. The Steiner nodes of the different sub-trees correspond to their nodes which are a primary tree leafs. Among Steiner tree heuristics there is one (KMB heuristic [10]) which can be used to optimize the two criteria together. Indeed, any sub-tree of a KMB forest whose leafs form a sub-set of the set of global Steiner nodes is a KMB tree. The Steiner nodes of each sub-tree consist in its nodes which are a Steiner nodes of the KMB forest.

In conclusion, we deduce that the use of the KMB heuristic to compute the dual-forest provides a 2-approximate solution optimizing the backup structure cost. Moreover, the KMB dual-forest includes backup paths which optimize (minimize in the link failure case) the multicast tree cost increase after recovery (with the dual-forest protection scheme).

[2] A solution x belongs to the Pareto's zone if there is not any other solution y better than x according to all the criteria

[3] One Steiner node of each tree must belong to NA_x. The rest are elements of the set (SA_x U PA_x)

[4] The forests used for restoration interconnect nodes of (SA_x U PA_x) to nodes of NA_x. Thus, all leafs of the trees of each forest must belong to (SA_x U PA_x U NA_x)

4 SIMULATION

In order to evaluate the quality of the dual-forest protection (IDFP), we choose to compare it to the dual-tree protection (DTP) and to the path protection (PP). The purpose of selecting the DTP is to show the difference between the level of protection that it provides and that provided by the IDFP. We also opt for the PP technique because it is the only applicable technique among those presented in section 2. Indeed, the problem of loop formation in the local one-to-one backup protection cannot be solved easily as in PP. As a result, a distributed method which determines a new connection identifiers is required (then new control messages are necessary). Concerning the redundant tree protection, it requires a very restricting conditions on topology (which are not satisfied in IP networks in general) to be applicable.

For the comparison, the metrics described in the sub-section 4.1 are used.

4.1 Comparison criteria

The following metrics have been selected in order to evaluate the quality criteria of the IDFP, DTP and PP schemes: protection rate (PR), average tree cost increase after recovery (ATCI), average cost of the longest reconfiguration path (ACLRP) and average number of control messages required to complete the restoration (ANCM).

PR measures the survivability of the network, the higher is this rate the better is the protection. It is defined as the ratio between the number of cases where the protection technique successes to repair the multicast tree and the total number of (failure) cases.

The tree cost increase is determined as the ratio between the cost of routing structure used after the restoration and the cost of tree used before the failure detection. A significant ATCI indicates a wasting of bandwidth.

Among the reconfiguration paths, the longest one is the path which has the highest cost. We point out that a reconfiguration path is a path defined by the succession of nodes that the Reconfig message travels. ACLRP can be used to approximate the average restoration delay. Indeed, if we assume that the time necessary for a Reconfig message to travel a given path is higher when the path is longer then, the restoration delay will depend directly on the longest reconfiguration path. In this case, the recovery technique is better when ACLRP is small.

The number of control messages required to complete the restoration corresponds to the total number of Reconfig messages sent on each link of the network in order to repair the affected communication. More significant is the average number of these messages (ANCM), more bandwidth is wasted and more the network is likely to be congested.

4.2 Simulation model

In the simulation presented here, all links are bi-directional and of cost equal to 1 (without lack of generality, our model can be easily extended to any link cost). 400 connected graphs (sufficient number in experiments which reduces significantly the correlation between the chosen graphs and the values of the different metrics) are randomly generated with the Waxman approach [12] in which a link between two nodes x and y is chosen to be in the graph according to probability $p(x,y) = \alpha e^{(-d(x,y)/\beta L)}$, where *d(x, y)* is the Euclidean distance between x and y, L is the maximum distance between any two nodes, α and β are parameters verifying $0 < \alpha, \beta < 1$. α is chosen equal to 0.25 to have connectivity characteristics of Internet networks and we varied β and the size of graphs to have different average node degrees.

After the graph generation, we vary the multicast group size between 2 (sparse mode) and 30 (dense mode), and we build randomly for each graph and each size 100 multicast groups (100 is sufficient value in experiments to reduce significantly the correlation between the chosen multicast groups and the values of the different metrics). One node of the multicast group is randomly chosen as the source. For each multicast group (and each graph), the primary tree is computed using the Dijkstra shortest path tree algorithm.

Two types of failures are considered: link failures and node failures. The failing links are selected randomly among the primary tree links and the failing nodes are chosen randomly among the inner nodes of the primary tree (nodes different from the source and leaf nodes). For each failure type, the different protection schemes are used to repair the multicast communication and the values of the metrics described in sub-section 4.1 are computed.

As the conclusions concerning the best protection technique are the same in all our simulation cases (graph sizes equal to 50 and 100, and average node degrees equal to 2.75, 3, 3.5, 4, 4.4 and 8), we present and analyze here only the results depicted in Fig. 7 and Fig. 8 obtained for β equal to 0.08 and graph size equal to 100 (the average node degree is equal to 4.4).

4.3 Comparison and analysis

4.3.1 Dual-tree protection Vs Dual-forest protection

To illustrate the importance of the dual-forest protection scheme, we start initially by using the protection rate metric to compare it with the dual-tree protection scheme.

The results depicted in Fig. 7 show a great difference between the PR of IDFP and that of DTP in the two types of failures (link and node failures). Indeed and except in the unicast case (group size equal to 2) where the two schemes are same, the performances of IDFP are widely better than those of DTP. We see that the difference between the PR of the two schemes in the link and node failure types increases monotonically with the augmentation of the group size. This is due to the fast decrease of the PR values of DTP whereas those of IDFP seem to be stabilized (see section 4.3.2 for the explanation) with very light variations.

Note that the usage of DTP is drastically restricted because of the fast decrease of its PR. In Fig. 7 for instance, the group size must be lower than 5 to have an average protection rate upper than 0.75. The reason of the fast decrease of the PR in DTP is due essentially to the augmentation of the primary tree size with the increase of the multicast group size. Indeed, the probability to obtain a connected reduced graph (and thus a dual-tree) after the

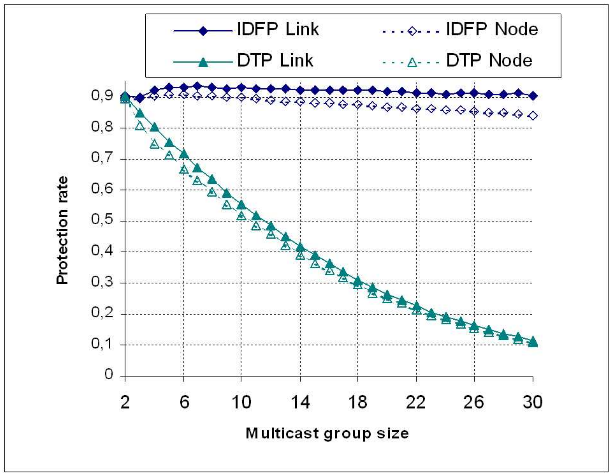


Fig. 7: Protection Rate (PR)

elimination of all links and inner nodes of a primary tree is smaller when the primary tree is larger.

Another important observation refers to the difference between the PR of DTP link failure type and that of DTP node failure type. We recall that the PR in the link failure type corresponds also to the rate of cases where the dual-tree exists. Thus, the PR of DTP in the node failure type is always smaller than that of DTP in the link failure type because of the cases where the restoration leads to the formation of loops (section 2.2.2).

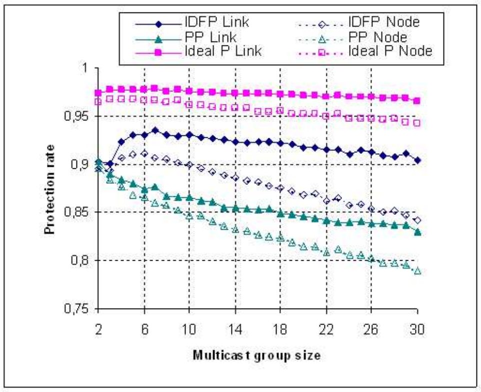
Since the PR of the DTP is small and undesirable, we focus in the rest of simulation only on the comparison between the PP and IDFP schemes.

4.3.2 Path protection Vs Dual-forest protection

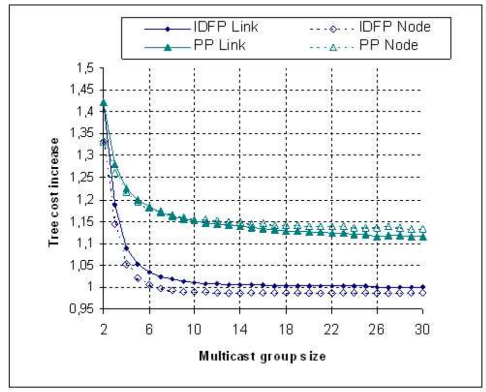
Concerning the protection rate metric (Fig. 8(a)), the IDFP is always better than the PP.

The values of PR are identical for a multicast group size equal to 2 (one source and only one destination) in both link and node failure types because the IDFP and the PP act in the same manner to restore unicast communication. Indeed, the two techniques use a same backup path (or two backup paths having a same cost) which is disjoint to the primary path and which connects the source to the destination.

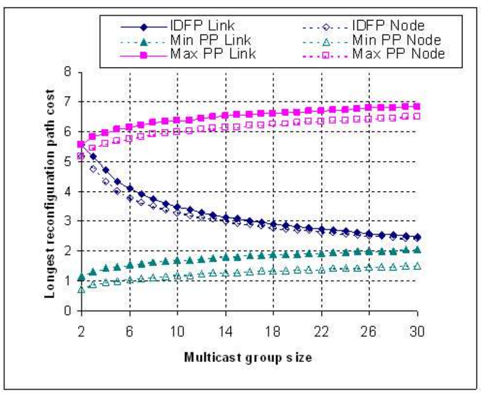
When the size of multicast groups is higher than 2, the curves of the IDFP go up rapidly until the group size reaches value 5. This can be explained by the augmentation of the number of primary leafs as the group size increases. Adding a new leaf node can then make possible for a non protected primary leaf node to have a backup path by its interconnection to this new primary leaf node.



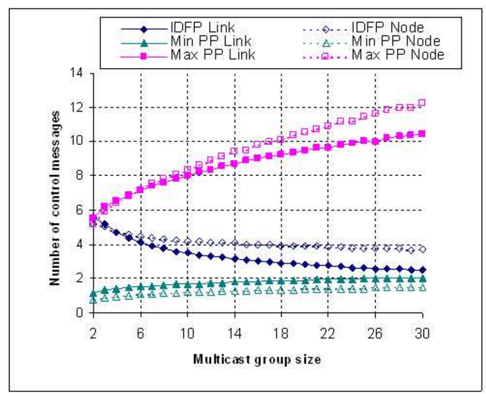
(a) Protection Rate (PR)



(b) Average Tree Cost Increase (ATCI)



(c) Average Cost of the Longest Path (ACLRP)



(d) Average Number of Control Messages (ANCM)

Fig. 8: Performances of the dual-forest protection and the path protection schemes

When the group size is higher than 5, the protection rate of the IDFP seems to be stabilized with very light diminution especially in the case of node failure. That is due to the primary tree size which increases and which becomes significant, restricting the reduced topology on which the dual-forest is built. The PR in the case of failure node decreases more rapidly than in the case of link failure because a node failure involves in general the reconfiguration of more than one backup path whereas only one backup path is used to bypass a link failure.

Contrarily to IDFP curves, the PP ones go down continuously. Indeed, as each member node is protected by a distinct path, the probability of success of the restoration procedure (probability to determine a backup path for each affected member) decreases with the increase of the size of affected members which depends on the multicast group size.

The last important point is related to the difference between IDFP curves and ideal ones which does not exceed 0.06 in the link failure case and 0.1 in the node failure case. We recall that an ideal protection technique will ensure the restoration of the multicast communication after a failure if all members belong to the same connected component in the topology graph obtained after the failure.

With regard to the multicast tree cost increase (Fig. 8(b)), the curves of the two techniques (IDFP and PP) have a similar form with a significant and almost constant difference (equal to 0.1).

The IDFP introduces small increases in the ATCI for groups of small size. This is due to the distance between the primary leafs which is more important for groups of small size. This increase in tree cost is not awkward because the cost of primary tree is small (ATCI is a ratio). Thus, the difference in cost between the routing structure after recovery and the used tree before the detection of the failure is not important. For groups of medium and large sizes, the ATCI is negligible and in most cases of node failure lower than 1. This is due to the close distance between primary leafs in one side, and to the use of a KMB forest as backup on the other side. Indeed, the replacement of an affected part of the multicast tree which is computed according to Dijkstra algorithm by another part determined with the use of a KMB forest decreases in most cases the cost of the multicast tree.

However, in PP the backup paths have a higher cost than the primary ones since they are computed using a restricted topology (while the global topology is used for the computing of primary paths). As a result, a significant tree cost increase is observed in PP. Thus, more than 11% of the total quantity of bandwidth allocated on the multicast tree is needed for a recovery from only one failure.

In Fig. 8(c) and Fig. 8(d), two cases of restoration in PP are taken into account: min PP and max PP. When a node detects a failure on an interface leading to a child node in the primary tree, it sends a Reconfig message to the source node. In min PP, the backup paths are already configured and thus, the source switches directly after the reception of Reconfig message from the affected primary paths to their backups without sending any additional message. The longest reconfiguration path cost and the number of control messages are the same and correspond to the cost of the primary path from the upstream node of the failed component to the source node. In Fig. 1(b), the corresponding longest reconfiguration path consists in link (C, S). However, in max PP, the backup paths have not been configured. Hence, the source forwards the Reconfig message on all backup paths participating to the restoration (technique used in [6][8]). In Fig. 1(b), the corresponding longest reconfiguration path is (C, S, A, B, M2, D, M1) and the number of control messages is equal to 6.

The results illustrated in Fig. 8(c) show that the ACLRP of min PP is smaller than that of IDFP which is in turn smaller than that of max PP (for the two types of failures). Note that while the different curves of PP go up continuously, those of IDFP go down with the augmentation of the group size. In max PP, it is obvious that with the augmentation of the group size more backup paths are necessary. As a result, its ACLRP increases. At the opposite, the ACLRP of IDFP decreases because of the diminution of the distance between primary leafs in one side, and because of the localization of failures which are more close to the primary leafs in other side (in a tree, there are more nodes nearer to the leafs than to the source). The other important remark which makes the IDFP interesting concerns the distance between its curves and those of min PP which is very small for large groups (small than 0.5 for group size equal to 30 in the link failure case).

In Fig. 8(d), the ANCM of min PP, max PP and IDFP are depicted. Note that the curves of min PP in both link and node failure types and the curve of IDFP in link failure type are identical to those of the longest reconfiguration path cost. Also and for the same reasons as in Fig. 8(c), the curves of max PP go up continuously. However and unlike the curve of IDFP in the node failure type in Fig. 8(c), that of Fig. 8(d) seems to be stable (very small diminution) for group size between 6 and 30. This is due to the increase of the average number of affected primary leafs (which involves to send more Reconfig messages) with the increase of the group size. Thus, the effects of the distance diminution between primary leafs and of the localization of failures which are more close to the primary leafs are decreased.

5 Conclusion

In this paper, we gave an overview of the existing pro-active protection schemes for multicast. We outlined the problems and insufficiencies of each scheme and we proposed an efficient multicast protection scheme based on a dual-forest. Our proposition makes several improvements to the traditional dual-tree. Indeed, the proposed multicast protection scheme based on a dual-forest can cope rapidly with both node and link failures without loop formation. The level of protection is also increased with the use of a forest instead of a tree as a backup structure. Moreover, we have elaborated that the use of KMB heuristic to compute the dual-forest is a good technique which decreases both the cost of the backup structure and the cost of the multicast tree after recovery.

We evaluated the performances of our protection scheme by comparing it to the dual-tree and to the path protection schemes. The results show that the dual-forest protection is a promising protection scheme since it has a better protection rate than those of the dual-tree and path protection schemes. Its tree cost increase after recovery is also better than that of path protection, its average cost of the longest configuration path is small and the average number of control messages that it involves is not high.

An improvement can be made to the dual-forest protection in order to enhance its protection rate. Typically, the contraction of a (primary) sub-trees already protected into a contracted leaf nodes can provide opportunities for some (non-protected) components to be protected by their interconnection to the contracted leaf nodes.

References

[1] J. Moy. Multicast Extensions to OSPF. RFC 1584, Mar 1994.

[2] A. Ballardie. Core Based Trees (CBT) Multicast Routing Architecture. RFC 2201, September 1997.

[3] P. Meyer, S. Van Den Bosch, and N. Degrande. High Availability in MPLS-Based Networks. Alcatel Telecommunication Review, 4th Quarter 2004.

[4] K. Murakami, and H. S. Kim. Optimal capacity and flow assignment for self-healing ATM networks based on line and end-to-end restoration. In IEEE/ACM Trans. on Networking, vol.6(2), pp.207-221, April 1998.

[5] P. Pan, G. Swallow, and A. Atlas. Fast Reroute Extensions to RSVP-TE for LSP Tunnels. RFC 4090, May 2005.

[6] C. Wu, W. Lee, Y. Hou, and W. Chu. A New Preplanned Self-healing Scheme for Multicast ATM Network. In Proceedings of IEEE ICCT'96, vol.2, pp.888-891, May 1996.

[7] C. Wu, W. Lee, and Y. Hou. Backup VP Preplanning Strategies for Survivable Multicast ATM Networks. In Proceedings of IEEE ICC'97, vol.1, pp.267-271, June 1997.

[8] A. Fei, J. Cui, M. Gerla, and D. Cavendish. A "Dual-Tree" Scheme for Fault-Tolerant Multicast. In Proceedings of IEEE ICC 2001, June 2001.

[9] M. Medard, S. Finn, R. Barry, and R. Gallager. Redundant trees for preplanned recovery in arbitrary vertex-redundant or edge-redundant graphs. IEEE/ACM Transactions on Networking, vol.7(5), pages 641-652, October 1999.

[10] L. Kou, G. Markowsky, and L. Berman. A fast algorithm for Steiner trees. Acta Informatica, vol.15, pages 141-145, 1981.

[11] D. Pe'er, and A. Wigderson. On Minimum Spanning Trees. December 1998. http://www.math.ias.edu/ avi/STUDENTS/dpthesis.pdf

[12] B M. Waxman, Routing of multipoint connections, IEEE JSAC, vol.6(9), Pages 1617-1622, December 1988.

Contents