Cost-Optimal and Cost-Aware Tree-Based Explicit Multicast Routing

Miklós Molnár
University of Montpellier 2,IUT, Dep. of Computer Science, LIRMM
161 rue Ada, 34095 Montpellier Cedex 5 France
Email: Miklos.Molnar@lirmm.fr

***Abstract*— This paper aims to introduce the hard optimization problem of determining tree-based explicit multicast routes with minimum cost. Explicit multicast routing has been proposed as a technique to solve the problem of multicast scalability in IP-based networks. Tree-based explicit routing is a special routing technique, in which the multicast tree is computed at the source and encoded explicitly in the datagram headers. These enlarged headers may result in significant overhead traffic, so the cost minimization of this kind of routing is a relevant topic. In this particular multicast routing, the well known minimum cost spanning trees (Steiner trees) do not corresponds to the optimal solution: the overhead induced by the large header corresponding to a Steiner tree can be excessive. This paper proposes the optimization of the routing minimizing the communication cost per bit in tree-based explicit multicasting. If the multicast group is large and the header size is limited, several trees are needed to provide routing for the entire group. In this case, the optimization can be seen as a particular constrained partial spanning problem. It is demonstrated that the computation of the minimum cost tree and the set of trees with minimum cost are NP-difficult problems. The presented theoretical analysis is indispensable to find cost efficient routes for these kinds of multicast routing protocol. Some algorithmic issues of the tree set construction are also discussed in the paper: exact and heuristic algorithms are presented. In real routing protocols, expensive exact algorithms cannot be applied. So, the paper also aims with the presentation of some tree-based explicit multicast routing algorithms using polynomial execution time.**

***Keywords*-Communication theory; multicast routing; combinatorial optimization; minimum cost routing; Steiner problem; hierarchy; QoS-based routing;**

I. Introduction

Multicasting was proposed to minimize bandwidth and network resource usage (for instance in IP based networks) by Deering in [2]. This kind of communication allows messages to be sent to a set of destinations in a special way: at most one copy of each message is forwarded on each link of a multicast tree. There is a large variety of distributed applications including television, video on demand, games and video-conferences, which benefit from multicast communication. In IP based networks the deployment of multicasting has been delayed by the well known problem of scalability. Because IP multicast addresses do not contain any specific information (for example: localization of the destination), address based aggregation of multicast communications is not possible and thus multicasting does not scale with the number of multicast groups. Indeed, IP routers store an entry for each multicast group using the given router. The large number of multicast entries in the forwarding tables retard the forwarding process. Another problem for the deployment of multicasting is that currently not all routers in the Internet are multicast capable. To introduce multicast communication progressively, it is important to design protocols, which allow multicast via unicast forwarding in certain domains. For this reason, protocols such as REUNITE (cf. [3]) and HBH (cf. [4]) have been proposed. In these protocols forwarding is done in the traditional unicast way and the branching node routers store information on next destinations in special tables. Trivially, this kind of protocol does not resolve the scalability problem.

Explicit multicast routing protocols have been proposed that scale better with the number of multicast groups. When explicit routing is used the group forwarding information is stored in the header of the datagrams. The group information is generally collected by a particular router and this information should be available at the source to send the datagrams. So, this type of multicasting can be regarded as a source-based routing technique. Simple flat explicit multicast routing only encodes the set of destinations in the datagram headers. In the subsequently encountered routers, datagrams are forwarded using the header information by applying the locally available forwarding mechanism (often a unicast forwarding). Accordingly, there is no forwarding state information for the given groups in the forwarding tables.

The flat explicit routing protocols suffer from an important drawback: each intermediate router on the multicast route has to inspect the datagram header. The router should duplicate the datagram if there are several next hops to forward it toward the encoded destinations. This handling is obligatory even when the router is not a branching node of the multicast tree.

To avoid obligatory processing of the datagram headers in the intermediate routers, tree-based explicit multicast routing protocols have been proposed [5] [6]. In these protocols, the source (or an appropriate route computation element) computes the tree spanning the destinations and stores the tree structure in the datagram headers. Note that the tree can be encoded entirely by its significant nodes (destinations and branching nodes), and the data forwarding between two successive significant nodes can be performed using unicast routing. This allows tree-based explicit protocols to forward datagrams faster than flat explicit protocols.

An indisputable drawback of explicit multicast routing resides in the traffic overhead due to the enlarged header size. Moreover the header size may differ between one route and another, and this is particularly true for encoded trees. The more significant intermediate (and so encoded) nodes a spanning tree contains, the longer the datagram header becomes. The generated header related traffic must be taken into account, even for route computation and optimization. So, the optimization of the communication cost needs a new formulation of the explicit multicast routing problem, which is significantly different from the classic Steiner problem [7].

When IP protocols are used, explicit multicast routing must cope with datagram fragmentation. Because the amount of encoded routing information in the headers can be significant it is possible that these datagrams will be fragmented and data and header will be unfortunately separated. To avoid bad IP fragmentation, the segmentation of the destination set into several sub-sets has been proposed for flat explicit routing protocols [8]. Using this technique each sub-set of destinations can be encoded separately by a "small" datagram, which may be sent without fragmentation. However, the segmentation of multicast delivery trees for tree-based explicit protocols has not yet been investigated.

Because multicast datagrams can be fragmented and the multicast structure segmented, we analyze the optimality of routes with and without fragmentation. More precisely, we describe the optimal multicast structure, which generates the minimum communication cost per bit including the variable cost of the header transmission. Generally, by taking into account the header size limitation, this cost minimization corresponds to a constrained partial minimum spanning problem, which is NP-difficult even if the solution is a single tree.

Tree-based explicit multicast routing protocols can be solicited for different reasons and not only to tackle the scalability problem. Multicast communication may be constrained by a given policy of the source or of the application. The quality of service (QoS) requirement is one of the most frequently imposed constraints. Often the QoS is formulated on the basis of multiple criterion and the computation of feasible or optimal routes corresponds to a multi-constrained optimization. Finding the multicast graph respecting the defined QoS requirements and minimizing network resources is an NP-complete optimization task [9]. For example, Multicast Adaptive Multiple Constraints routing Algorithm (MAMCRA) [10] proposes the computation of routing structures constrained by multiple QoS criterion from the source to the destinations. In certain cases the result does not correspond to a tree but to a set of trees and paths rooted at the source and containing some cycles. Traditional IP multicast routing using a single IP address for the group cannot be used. Explicit routing is a good candidate to resolve the conflicts induced by the cycles. More generally, constrained multicast routing structures are tree-like structures called *hierarchies*. The use of this kind of structure for multicast routing in IP domain necessitates routing protocols that allow the crossing of branches (routes) in the same multicast route structure. The technique of tree-based explicit multicast routing also permits the encoding of hierarchical routing structures.

Another candidate for tree-based explicit multicasting is application level multicasting. Delivery trees can be computed at the application level and overlay links can be used among end systems handling the multicast packets. These solutions support naturally traffic engineering, can improve the reliability of multicast delivery, and facilitate secure group communications [11]. Generally, in traffic engineering solutions and QoS aware environments tree-based explicit multicasting may offer an interesting tunable multicast data delivery technique.

The present work focuses on the cost-optimal tree-based explicit multicast solutions taking into account the increased bandwidth usage due to the largest datagram headers. It is an extended version of [1], and it provides more detailed information about the problem formulation, the properties of the optimal solution and some algorithmic issues of the possible route computations.

The next section gives a rapid overview of the related work. The formulation of the tree-based multicast routing problem with minimum communication cost can be found in Section III. We demonstrate that the cost optimization of tree-based explicit routing is an NP-difficult computational problem. Some exact algorithms are presented in Section IV but these algorithm are very expensive. More practicable heuristic algorithms are also proposed for routing protocols. Our conclusions and perspectives close this initiative study.

II. Related Work

Initially, explicit multicast routing was proposed for small multicast groups (cf. Small Group Multicast in [12]) to decrease the number of multicast entries in routing tables.

Combined with the traditional IP multicast routing for large groups, scalability for all type of multicast can be achieved.

A simple approach to implementing explicit multicast routing is to simply store the set of destination addresses in each datagram. The basic protocol of this kind of flat explicit protocol is the Xcast protocol proposed in [13]. When an Xcast router receives an Xcast datagram, it performs a look-up for each valid destination in the header to determine the required next hop. Then it copies the incoming datagram to each required outgoing link. An improved version of this protocol is the protocol Xcast+, described in [14], which uses dedicated routers to reduce the header size. Simple explicit multicast routing eliminates tree construction and maintenance costs in the network and decreases the network control load. For these reasons it was also proposed for mobile ad hoc networks [15].

To resolve the main drawback of flat explicit routing protocols (which is the check of the destination list in each router) precomputed tree based explicit routing was proposed. The first tree-based explicit protocol was the ERM protocol proposed in [5]. In the ERM protocol the source encodes the IP addresses of the branching nodes and the destinations of the multicast tree in the datagram headers. Inside the routing domain, this header is analyzed and datagrams are routed using unicast forwarding mechanism. The protocol Linkcast, described in [6] improves ERM by proposing a new header encoding. Since the tree is encoded in the datagram header, a node can easily decide whether it is a branching node or not. Similarly, it is easy for a branching node to find its children. In [16] the trade-offs of the tree-based explicit routing protocol design are discussed and a performance analysis is presented. The analyzed metrics are the header size and the processing overheads. More detailed and appropriate tree information may reduce the processing overhead in return for larger header size and traffic overhead. The authors propose a modification to ERM called Bcast, which reduces the overhead of the protocol. In Bcast, a proactive bypassing mechanism helps to adjust the code size in response to inconvenient distribution of the receivers.

Using IP protocols, explicit multicast routing will inevitably experience datagram fragmentation. Because the amount of encoded routing information in the headers can be significant, it is possible that these datagrams will be fragmented and data and header will be unfortunately separated. The problem of IP fragmentation of multicast datagrams using flat explicit routing has been analyzed in [8]. The segmentation of the destination set into several sub-sets has been proposed to avoid cutting the headers in two. The optimal segmentation has also been analyzed and the authors have demonstrated that quasi-optimal communication cost can be obtained when header length is less than half the datagram size.

III. Communication Cost Optimization for Tree-Based Explicit Multicasting

In this section, we formulate the optimal tree based explicit multicast routing, which minimizes the communication cost (and not the cost of the used trees). We will show that communication cost minimization is a very hard problem when the traffic overhead due to explicit routing headers and segmentation must be taken into account. This problem corresponds to a special constrained Steiner problem with nonlinear cost function even if the maximal header size does not limit the tree. In the general case, when the limitation on the maximal header size should be taking into account, the problem becomes a special constrained partial spanning problem. In this case, the optimum corresponds to a special hierarchy: to a set of spanning trees.

Let $G = (V, E)$ be the undirected and connected graph corresponding to the network topology and $D \subset V$ the set of destinations of the multicast group originated at the source s. Let us suppose that the network topology is known at the source. Moreover, the size of the datagrams is limited by a value L_{max}.

We suppose that a homogeneous unicast routing mechanism exists in the routing domain and that this mechanism is known at the source. So, the source node can compare any spanning tree with the possible unicast routes in order to decide, which nodes of the tree should be encoded explicitly. Explicit multicast routing can then use the unicast routing mechanism between any two successive encoded nodes of the multicast tree. Evidently, the encoded tree must contain all nodes such that the set of unicast routes between them corresponds to the original tree. In the following, we call these nodes of the tree *significant nodes*. Figure 1 illustrates the encoding of the tree in a simple example. The source node s would send messages for the destination set $D = \{d_1, d_2, d_3, d_4, d_5\}$ using tree-based explicit multicast routing. Let us suppose that the unicast routing uses the shortest paths between any node pairs and the multicast tree is a partial spanning tree T as indicated in the figure. In this case the significant nodes are the nodes, which are indicated with a double line and correspond to the following parenthesized list:

$$a(c(d_1, d_2, g(d_3, d_4, d_5)))$$

The node a is significant, because the shortest path from s to c does not pass through a. So, to follow the route from s to c via a, a must be explicitly encoded.

To simplify, in the following the set of significant nodes S_T of a spanning tree T denotes the union of the branching

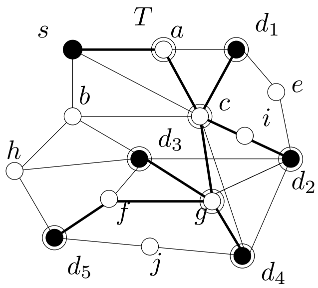

Figure 1. Significant nodes of a tree for unicast routing

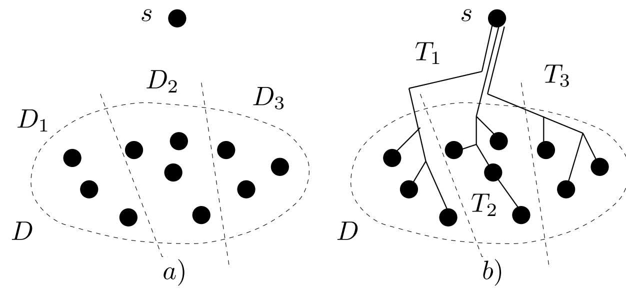

Figure 2. The optimization is a superposition of a) a partitioning and b) a minimum cost weighted spanning tree problem

nodes B_T of T and the destinations: $S_T = B_T \bigcup D$ (*i.e.*, the paths between these nodes are shortest paths).

We consider the minimization of the *total communication cost* as the objective of the tree-based explicit multicast optimization under the constraint on the maximal length of datagrams. We will show that this cost does not correspond to the sum of link costs as it is the case in simple multicast route computations. We distinguish two components of the communication cost: the cost of the transmitted payload and the cost of the overhead generated by the headers. The latter cost is proportional to the explicit routing header size. This header size depends on the number of encoded addresses and so on the structure proposed for the routing. We will show that the minimal cost routing structure is always a set of trees. Figure 2 illustrates the difficulty of the optimization. Generally, the optimal solution comprises several destination sub-sets (which should be spanned separately because of the constraints). Thus, the first question is related to the partitioning of the destination set (Figure 2/a). Then, for each sub-set of destinations a special minimum cost partial spanning tree should be built. This latter problem itself is NP-difficult (and can be seen as a special case of the Steiner problem, cf. in the next sub-section). The optimal routing problem is the superposition of these difficult optimization problems, since the cost of the partitioning and the spanning trees are inseparably related. If the trees resulting from the segmentation are large (in term of number of encoded nodes), then the payload in the datagrams is small and several datagrams should be sent to transmit the desired message. If the trees are small, then several trees are needed to cover the entire multicast group.

In this section, we first present the objective function of the minimal cost tree construction even if segmentation is not needed (one tree can cover the entire multicast group and can be encoded in the header without segmentation and it corresponds to the minimal cost solution). Then, we show that this optimization problem is NP-difficult. Secondly, we present the explicit multicast routing structure optimization in the case where header segmentation is required. We will show that the optimal routing structure corresponds to a set of trees and the problem remains NP-difficult.

A. Minimum cost tree considering the header length

In order to determine the objective function of the explicit routing optimization progressively, we first focus on the simple case where only one encoded spanning tree is needed to cover the destinations.

More precisely, this case is produced when

- the source has only one sub-tree for spanning all of the destinations
- there is sufficient space in the packet header to store the encoded version of this unique spanning tree.

So, first we consider a unique spanning tree that covers the entire set of destinations where the tree is encoded and stored entirely in the corresponding datagram headers. If a spanning tree has several sub-trees at the source, then the datagrams sent on each sub-tree have distinct multicast tree encoding. Such a spanning tree can be considered as a set of its sub-trees rooted at the source. For instance, if a tree T can be decomposed at its source into two (disjoint) sub-trees T_a and T_b, then we say that, from the point of view of tree encoding, a set of trees $\{T_a, T_b\}$ covers the destinations. The trees of this set are encoded separately in different packet headers. Figure 3/a and Figure 3/b illustrate respectively the cases when the source has only one sub-tree, and where two disjoint sub-trees cover the destination set.

Lemma 1. *If the segmentation of the destination set is not needed and all of the destination are accessible via the same neighbor node of the source, the optimal structure is a partial spanning tree.*

Proof: Without segmentation, the optimal solution is a connected sub-graph. The datagrams are sent on each link in this structure. Since the edges are positively evaluated, an eventual cycle increases the cost. So, the solution is a partial spanning tree. ∎

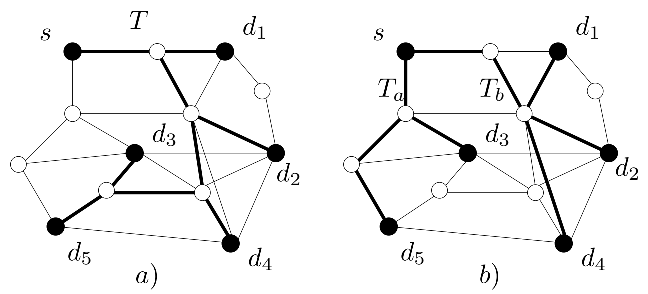

Figure 3. One or several sub-trees may be at the source

To cover the given set of destinations and the source with a single tree, different partial spanning trees can be found and enumerated (for example by a Steiner Tree Enumeration Algorithm, cf. [17]). Each tree contains a different set of significant nodes, corresponds to a specific header length and so involves a specific overhead and payload. The optimal solution is the tree, which minimizes the total communication cost. In the following, we talk about the partial minimum spanning tree for encoding (which is generally different from the Steiner tree of the given group).

To formulate the overhead generated by the explicit multicast headers, let us suppose that the significant nodes are encoded by their network addresses using l_a bytes, there are $k(T)$ significant nodes and the maximal size of messages is equal to L_{max} bytes. The encoding technique of the datagram header is out of scope of the paper. Only the impact of the encoded tree is analyzed, the rest of the header is considered to have a constant length. In this way, the size l_h of a header can be expressed by $l_h = k(T) \cdot l_a + c$, where c is the constant length of the rest of the header. Using datagrams with the maximum length, the maximum payload in a datagram corresponds to $l_p = L_{max} - k(T) \cdot l_a - c$ and to transmit a message of L bytes, $n_p = \left\lceil \frac{L}{L_{max} - k(T) \cdot l_a - c} \right\rceil$ datagrams must be used. So, the traffic generated by the transmission of the headers can be expressed by $L_h = n_p \cdot (k(T) \cdot l_a + c)$. The traffic corresponding to the transmission of the message of length L is

$$L_k = L + \left\lceil \frac{L}{L_{max} - k(T) \cdot l_a - c} \right\rceil \big(k(T) \cdot l_a + c\big) \quad (1)$$

Let us suppose that the communication uses a tree T of cost $d(T)$. Thus the total communication cost is

$$C_L(T) = L_k \cdot d(T) = \left(L + \left\lceil \frac{L}{L_{max} - k(T) \cdot l_a - c} \right\rceil \big(k(T) \cdot l_a + c\big) \right) \cdot d(T) \quad (2)$$

The optimization of the communication support should be independent from the message length L. The cost per bit better characterizes the cost of the communication and this cost should be minimized. The cost per bit can be obtained asymptotically as

$$C(T) = \lim_{L \to +\infty} \frac{C_L(T)}{L} = \lim_{L \to +\infty} \left(1 + \frac{\left\lceil \frac{L}{L_{max} - k(T) \cdot l_a - c} \right\rceil}{L} \big(k(T) \cdot l_a + c\big) \right) \cdot d(T) \quad (3)$$

Finally, the communication cost per bit using the tree T corresponds to

$$C(T) = \left(1 + \frac{k(T) \cdot l_a + c}{L_{max} - k(T) \cdot l_a - c}\right) d(T) = \frac{L_{max}}{L_{max} - k(T) \cdot l_a - c} d(T) \quad (4)$$

The optimal encoded partial spanning tree T_M^* is the tree, which minimizes this communication cost (Problem 1):

$$T_M^* : \arg \min_{T \in \mathcal{ST}} \frac{L_{max}}{L_{max} - k(T) \cdot l_a - c} d(T) \quad (5)$$

Theorem 1. *The optimization given by (5) is NP-difficult.*

Proof: Trivially, if a particular case of the problem given by (5) is NP-difficult, then the problem is NP-difficult. In the expression (5) the length $d(T)$ of the tree T is multiplied by a factor

$$f(T) = \frac{L_{max}}{L_{max} - k(T) \cdot l_a - c}, \quad (6)$$

that characterizes the tree (it depends on the number of significant nodes in the tree). Generally, this factor is different from one tree to another. Let l_a be chosen so that the factors $f(T)$ do not influence the choice of the optimal solution compared with the tree lengths. Concretely, for every pair (T_i, T_j) of possible spanning trees, such that $d(T_i) < d(T_j)$, let a value l_a^m be chosen, which guarantees that $C(T_i) < C(T_j)$. Taking into account the cost function, the condition for this can be expressed as

$$\frac{d(T_i)}{d(T_j)} < \frac{L_{max} - k(T_j) \cdot l_a^m - c}{L_{max} - k(T_i) \cdot l_a^m - c} \quad (7)$$

Since $\frac{d(T_i)}{d(T_j)} < 1$ such a value exists. With this value of l_a^m, the corresponding factors $f(T)$ do not influence the relation between the spanning trees: if $d(T_i) < d(T_j)$ then $f(T_i) \cdot d(T_i) < f(T_j) \cdot d(T_j)$. In this case, the shortest partial spanning tree from the set of all partial spanning trees is the solution of our problem. The selection of the minimum cost

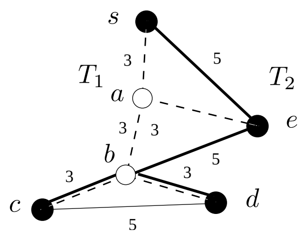

Figure 4. The impact of the header size on the optimal cost

partial spanning tree corresponds to the NP-difficult Steiner problem. ∎

The simple example of Figure 4 illustrates the impact of the header size on the optimal cost solution. Let us suppose that the maximal datagram length L_{max} is equal to 20 bytes, that there is no additional constant information in the header ($c = 0$) and the addresses are encoded on $l_a = 2$ bytes. The node s is the source of the communication and c, d, e are the destinations. The minimal cost Steiner tree T_1 covering the source and the destinations is marked with dotted line on the figure and has a length $d(T_1) = 15$. As there are two branching nodes on the tree, the number of significant nodes is $k(T_1) = 5$. So the factor corresponding to this tree is $f(T_1) = 2$. The total communication cost implicated by the tree $f(T_1) \cdot d(T_1)$ is 30. When taking the header size into account, we obtain the tree T_2 represented by a bold line on the figure. This tree is longer ($d(T_2) = 16$) but there are less branching nodes, $k(T_2) = 4$ and the factor $f(T_2) = 1,67$. The total communication cost of this tree $f(T_2) \cdot d(T_2) = 26,67$ is less then the cost per bit using the encoded Steiner tree.

B. Minimum cost solution with header segmentation

When the multicast group is large and the number of significant nodes in the multicast tree is high, a single encoded tree cannot ensure the coverage of the destination set. The group should be segmented in the optimal solution. Let us notice that in some cases some segmentation may be naturally given by the sub-trees at the source (cf. Figure 3/b)). These sub-trees are edge disjoint. In other cases the solution may contain non-disjoint trees. An example can be found in Figure 5, where the number of encoded significant nodes is supposed to be limited to 4. In the given graph, the five destinations cannot be spanned by a unique spanning tree from the source s. Segmentation is necessary. The figure illustrates a segmentation where two non-disjoint trees span the destination set. The nodes c and d_3 belong to both trees. T_1 should be encoded as $T_1 = (d_3(f(d_5, d_4)))$ and T_2 corresponds to $T_2 = (d_3(c(d_1, d_2)))$ for routing. Note that the node d_3, which belongs to both trees is a destination node. That does not mean that d_3 should consume any message twice. On the contrary, this node must receive the message for local consumption only once and the second message must be transmitted to the next node without local consumption. In other worlds: the node d_3 is a destination in only one tree and serves as relay node in the other. So, an *exclusively served destination node set* is associated with each spanning tree of a segmented solution. This exclusively served destination node set contains the real destinations in the tree (and not the relays even if they are destinations in the original problem).

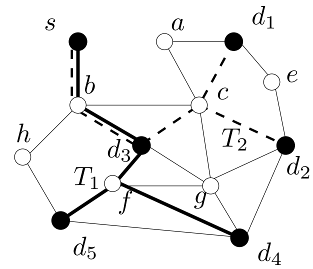

Figure 5. Two spanning trees with intersection

Lemma 2. *1) If the segmentation of the destination set is needed, the optimal explicit multicast routing structure is a set of partial spanning trees. 2) Each tree of this optimal set is rooted at the source and corresponds to a partial spanning tree minimizing the total communication cost for its exclusively served destinations.*

Proof: 1) The optimal solution Θ connects the destinations to the source. Since the maximum length of the datagrams limits the number of the significant nodes encoded in the headers, a single datagram header cannot be used for all destinations. A partitioning of the destination set is required. To connect a sub-set of the destinations to the source with a unique sub-graph in the optimal solution only a spanning tree can be used (cf. Lemma 1). So, the optimal solution Θ is a set of trees. 2) Each spanning tree in Θ should be a partial spanning tree minimizing the total communication cost relative to its exclusively served destinations. Let us suppose that a tree $T' \in \Theta$ does not minimize the communication cost relative to its exclusively served destination set $D_{T'}$. In this case, there is an other tree T'' minimizing the communication cost for the same destinations. So the overall solution containing T' cannot reach minimum cost. ∎

After segmentation, each header contains a tree spanning a sub-set of the multicast group. Let us suppose that the segmentation results a set of trees $F = \{T_i, i = 1, ..k(F)\}$ spanning $\{s\} \cup D$ with not necessary disjoint trees T_i.

Here $k(F)$ indicates the number of trees in the segmented solution.

Trivially, to reach the destinations, each tree of the solution should be rooted at the source node. So, the minimum cost multicast route forms a set of trees routed at the source. This kind of set of trees is often called a ”forest” in the literature. Recently, a new spanning structure was introduced in [18] to describe hierarchical structures which are, contrarily to trees, not obligatory exempt of redundancies. A *hierarchy* in a graph is a connected structure of consecutive nodes and edges that allows the some nodes and edges to be repeated such that for each node occurrence there is at most one predecessor node occurrence in the structure. In other worlds, a hierarchy is a tree-like structure permitting the repetition of the graph elements. A non-elementary path may contain a node several times. An elementary path is a path without repetition of the graph elements. The hierarchy is a more general concept than the tree concept and it may contain a node several times. Hierarchies without repetition are trees. Evidently, a set of trees routed at the same source node is a hierarchy since nodes and edges may be repeated in the set of the trees but each node occurrence has only one predecessor in the set. The source node can have several sub-hierarchies which are, in this particular case, spanning trees.

Generally, the different trees (sub-hierarchies) do not contain the same number of significant nodes. On the tree T_i, which has $k(T_i)$ significant nodes, the maximal payload per datagram is $p(T_i) = L_{max} - k(T_i) \cdot l_a - c$. It was shown in the last section that a tree T_i is optimal for the sub-set of destinations, if the total cost

$$C(T_i) = (1 + \frac{k(T_i) \cdot l_a}{L_{max} - k(T_i) \cdot l_a - c})d(T_i) \qquad (8)$$

is minimal. Using the previously mentioned set of trees (or hierarchy) F, the total transmission cost of a message of L bytes corresponds to

$$C_L(F) = \sum_{i=1}^{k(F)} \left(L + \left\lceil \frac{L}{L_{max} - k(T_i) \cdot l_a - c} \right\rceil \cdot \big(k(T_i) \cdot l_a + c\big) \right) d(T_i) \qquad (9)$$

The optimal solution (which results in the minimum cost per bit when L tends to infinite) is a hierarchy (set of trees) F_M^* spanning $s \cup D$ (Problem 2) such as:

$$F_M^* : \arg\min_{F \in \mathcal{SF}} \sum_{i=1}^{k(F)} \frac{L_{max}}{L_{max} - k(T_i) \cdot l_a - c} d(T_i) \qquad (10)$$

where $\mathcal{SF}$ denotes the set of hierarchies spanning $s \cup D$, each hierarchy is composed of partial spanning trees. The complexity of this new problem is discussed later, at the end of the next sub-section. Here we propose first a simplification of the data fragmentation.

In the optimal solution presented above it is possible that the header length and the payload are different from one tree to another. The differing fragmentation of the same message depending on the different trees may significantly complicate the data transmission procedure at the source. Organizing multicast communication around a set of trees that use the same data transmission procedure facilitates the explicit routing protocol.

C. Minimum cost solution with homogeneous fragmentation

Generally, the different trees in the segmented solution do not contain the same number of significant nodes. On the tree T_i, which has $k(T_i)$ significant nodes, the maximum payload per datagram is $p(T_i) = L_{max} - k(T_i) \cdot l_a - c$. The fragmentation of the message of L bytes is optimal in T_i, if this maximum payload is applied in the tree. To obtain the maximum payload a customized fragmentation is needed on each tree. In each tree, the data should be sent using different fragments, which results in a very complicated transmission procedure at the source.

Homogeneous fragmentation constraint. To simplify the fragmentation task at the source, let us suppose that the source implements a common fragmentation algorithm and always sends the same content (payload or fragment) on the trees covering the multicast group.

To satisfy the Homogeneous fragmentation constraint the maximum number of significant nodes per tree is trivially:

$$k_{max}(F) = \max_{T_i \in F} k(T_i) \qquad (11)$$

Consequently in a simple data transmission procedure, each header contains at most $k_{max}(F)$ encoded significant nodes and the payload is the same in simultaneously sent datagrams. To transmit a message of length L, the source should use

$$n_p = k(F) \left\lceil \frac{L}{L_{max} - k_{max}(F) \cdot l_a - c} \right\rceil \qquad (12)$$

datagrams. Using the aforementioned hierarchy corresponding to a set of trees F the total cost of the communication is equal to

$$C_L(F) = \sum_{i=1}^{k(F)} \left(L + \left\lceil \frac{L}{L_{max} - k_{max}(F) \cdot l_a - c} \right\rceil \cdot \big(k_{max}(F) \cdot l_a + c\big) \right) d(T_i) \qquad (13)$$

The optimal hierarchy (which induces the minimum cost per bit) corresponds to the set of trees spanning $s \cup D$ (Problem 3) such that:

$$F_M^* : \arg\min_{F \in \mathcal{SF}} \sum_{i=1}^{k(F)} \frac{L_{max}}{L_{max} - k_{max}(F) \cdot l_a - c} d(T_i) \quad (14)$$

where $\mathcal{SF}$ denotes the set of hierarchies spanning $s \cup D$ composed only from trees under the mentioned constraint.

The difference between the optimization problems 10 and 14 resides in the factor $f(T)$, which weights the different trees in the sums. These weights are typical of each tree in Problem 10 but they have the same value within a partition in Problem 14. So, optimization 14 with homogeneous weights is more simple but the complexity of both problems is high.

Theorem 2. *The optimization (10) and (14) are NP-difficult.*

Proof: In both problems, the optimal solution corresponds to a set F of trees. The destination sub-sets spanned exclusively by the different trees in F correspond to a partition $P = \{D_i, i = 1, \ldots, k(F)\}$ of D. Each sub-set of destinations D_i in this partition is covered by a partial spanning tree $T_i \in F$. Trivially, the tree T_i is of the minimum cost per bit regarding D_i, and the result corresponds to the partition minimizing the total cost (the sum of the sub-set costs). So, the solution corresponds to the selection of the minimum cost partition and this problem corresponds to the well known set cover problem, which is NP-difficult [19]. Trivially, each partial spanning tree T_i in the solution should be a partial spanning tree of $\{s\} \cup D_i$ inducing the minimum cost per bit while respecting the constraints (otherwise there is a solution with less cost when using the minimum cost spanning tree instead of T_i). For example, to find the optimal cost partial spanning tree of $\{s\} \cup D_i$ in a given partition, the Homogeneous fragmentation constraint should be respected. A tree with minimum cost per bit must be computed while respecting the maximum homogeneous header size and thus while respecting the maximum number of significant nodes. This latter computation itself is a NP-difficult problem (it can be considered as a particular case of Theorem 1). Combined with the optimal partitioning Problems (10) and (14) are NP-difficult. ■

IV. Algorithms to Find Cost-Aware Explicit Multicast Routes

Without completeness, some basic ideas to find minimum cost and cost aware solutions for the tree-based explicit multicast routing can be found in this section. Since the problem is NP-difficult, exact algorithms are expensive. Cost-aware but non-optimal solutions can be obtained by heuristics taking reasonable (polynomial) execution time.

A. Exact and heuristic solutions of Problem (5)

In Problem (5) we suppose that a single spanning tree is sufficient to solve the problem.

1) Exact algorithms: Modified Spanning Tree Enumeration Algorithms and Topology Enumeration Algorithms (cf. [17]) can be used to find the optimal tree. In the original algorithms, the possible partial spanning trees are enumerated and the tree with minimal cost is selected as the solution. The cost of each tree is computed as the sum of the costs of its edges. In our case, as Formula 5 indicates, this cost is weighted by the factor $f(T)$, which can be unique for each tree. In order to solve our problem, the enumeration algorithms can be applied but the tree with the minimal weighted cost should be selected. Let us notice that, in some cases, this factor $f(T)$ can also be used to eliminate excess trees in the enumeration algorithms. Since the function $f(T)$ is concave and increases rapidly depending on the number $k(T)$ of significant nodes, the optimal solution is probably among the spanning trees having few branching nodes. The complexity of the exact enumeration algorithms is always exponential and in $\mathcal{O}(n^2 2^{n-d-1})$ (where n denotes the number of nodes and d is the number of destinations) [20].

2) Heuristic algorithms: Contrarily, shortest path based heuristics originally proposed to find a 2-approximation for the Steiner problem cannot guarantee the same approximation ratio for the optimization problem (5). Indeed, the "penalty" factor $f(T)$ (which is a function of the number of significant nodes) cannot be included in the shortest path based heuristics.

The following simple example illustrates that a shortest path based Steiner heuristic finds an arbitrarily bad solution for Problem (5). Let there be a topology, a source s and a set of destinations D given as shown in Figure 6. Let us suppose that all the edges have a unit cost and $d = |D|$. In this particular topology, the optimal tree T^* (represented by a dotted line) uses a unique branching node. Shortest paths between the multicast group members do not traverse this node. A shortest path based heuristic (*e.g.*, the Takahashi-Matsuyama heuristic [21]) constructs the tree T_h (the continuous line in the figure).

The costs are $d(T^*) = 2(d+1)$ and $d(T_h) = 2d$ respectively. Since there are $2d$ significant nodes in T_h and $d+1$ in T^*, the approximation ratio in this case can be expressed as

$$A = \frac{C(T_h)}{C(T^*)} = \frac{L_{max} - (d+1) \cdot l_a - c}{L_{max} - 2 \cdot d \cdot l_a - c} \cdot \frac{d}{d+1} \quad (15)$$

Increasing the group size d causes this ratio to increase rapidly and an upper-bound cannot be given.

To find trees with low communication cost, we propose a modified version of the Takahashi-Matsuyama algorithm.

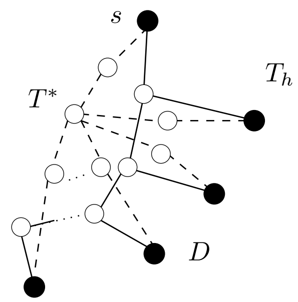

Figure 6. Shortest path based heuristics give an arbitrary solution

A simple objective function can be formulated if the costs (the cost of the usage of the edges and the overhead due to the headers) are expressed by additive metrics. Edge costs are basically additive. Moreover, the use of new branching nodes in the multicast tree can be penalized by additional cost factors. Let B_T be the set of branching nodes of the tree T and let us suppose that the inclusion of a new branching node $v \in B_T$, which is not a destination, corresponds to an additional cost $b(v)$. So, a partial spanning tree resulting in a low communication cost can be obtained by minimizing the sum of edge and node costs :

$$T_D^* : \arg\min_{T \in \mathcal{T}} (d(T) + \sum_{v \in B_T \setminus D} b(v)) \quad (16)$$

This expression can be considered as an approximation of the communication cost. Trivially, similarly to the Steiner tree problem, this problem is also NP-difficult. The advantage of the formulation (16) is that simple and efficient Steiner heuristics can be adapted to resolve it. Starting from this modified problem formulation we propose a heuristic to compute advantageous partial spanning trees for explicit tree based multicast routing.

Avoidance of Branching node Creation (ABC) algorithm

Following the objective function given by (16), a simple algorithm can be designed by modifying the well-known Steiner heuristic proposed by Takahashi and Matsuyama [21]. In each step of the original algorithm, the nearest destination node is added to the tree using the shortest path from node to tree.

In the modified ABC algorithm, the creation of a new branching node in the tree is penalized. For this reason, the ”distance” $\bar{d}(n,T)$ between the tree T and the node n is defined as

$$\bar{d}(n,T) = d(n,m) + \begin{cases} 0 & \text{if } m \in D \cup B_T \\ c & \text{otherwise} \end{cases} \quad (17)$$

where $m \in T$ is the node connecting n to T, $d(n,m)$ is the distance from m to n and c is the penalty associated with creating a new branching node in the tree. This modification does not affect the favorable complexity of the algorithm. Figure 7 illustrates one step of the algorithm. Let us suppose that each edge has unit cost. The cost of the nodes in the tree T are indicated in the figure. To connect the node n to the tree, the algorithm does not use the shortest path (n,b) but an alternative (the path (n,m)), which connects n to the leaf node m. This connection results in a lower communication cost because new branching nodes are not created.

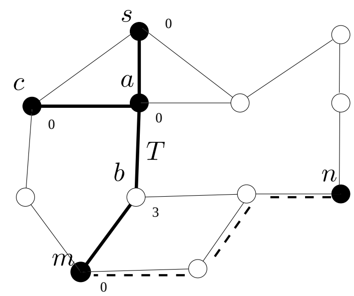

Figure 7. Add a new destination to the tree using the ABC algorithm.

To illustrate the performance of the ABC algorithm, simulation has been performed in the Eurorings topology, which has 43 nodes and 55 edges (cf. an example in [22]). In this topology, the Shortest Path Tree (SPT) algorithm, the Takahashi-Matsuyama (TM) heuristic and the ABC algorithm have been executed for different multicast requests the group size of which varied between 10 and 35. For each group size, 100 groups were generated randomly. Figure 8 shows the number of significant nodes in the computed multicast trees. Supposing a maximal packet size equal to $l_{max} = 1600$, addresses encoded in 128 bits and a constant part in the headers occupying 200 bytes, the communication cost corresponding to the three different trees is illustrated in Figure 9. In this network, the ABC algorithm reduces the communication cost by 10 - 20 % compared to the shortest path tree and the approximated Steiner tree using explicit routing.

B. Exact and heuristic solutions of Problems (10) and (14)

To the best of our knowledge, exact algorithms are not known that solve the recently formulated size-constrained minimum-cost partial spanning problem. Since a single tree is not always sufficient, Steiner Tree Enumeration Algorithms do not work. A trivial exact solution can be proposed as follows.

1) Exact algorithm: As demonstrated in Section III and illustrated with Figure 2, the optimal solution corresponds to an optimal partition of the destination set. So, exact algorithms solving the Set Cover Problem (cf. [19]) can be applied with the following adaptation: the cost associated to

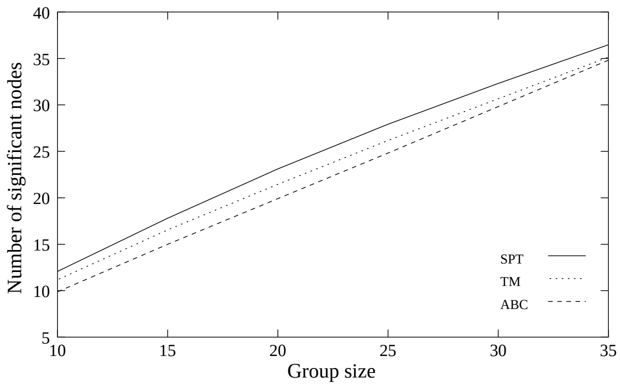

Figure 8. The number of significant nodes in the multicast tree.

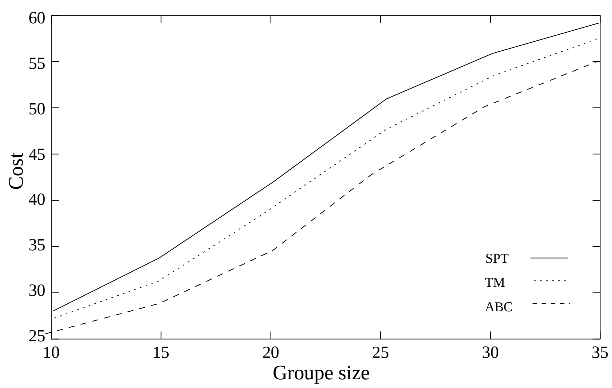

Figure 9. The communication cost associated with the multicast tree.

a sub-set of destinations in the partition is the communication cost of the partial spanning tree minimizing this cost.

Most of the exact algorithms to solve the Set Cover Problem are brute force and dynamic programming based algorithms [23]. In both cases, the associated optimal communication cost per bit must be computed for each sub-set of destination nodes in the partition. For this, a simple modified Steiner Tree Enumeration Algorithm (cf. [17]) can be used as indicated in the previous sub-section.

Let k_{max} be the maximal number of significant nodes in our size-limited spanning problem. The maximal number k of destinations in a spanning tree respecting the size constraint is given by $k(k-1) = k_{max}$. If P_D^k denotes the number of k-limited partitions of D, then the exact algorithm complexity is bounded with $\mathcal{O}(P_D^k 2^{k+1} n^2)$.

2) Some heuristic algorithms: Since the exact computation is very expensive, only heuristic algorithms can compete for potential use in networks. Heuristic solutions can be obtained in two different manners.

- The heuristics in the first group aim to *directly* build a set of trees with respect to the size constraint (moreover, the algorithms can eventually balance the size of the trees).
- The second type of algorithms works in three phases to compute the final solution:
  1) at first a low cost partial spanning tree is computed (regarding the overhead generated by the headers)
  2) then this unique spanning tree is segmented into several trees when the size constraint is exceeded
  3) the size of the trees may also be balanced.

A simple algorithm in the first category can be obtained by modifying the ABC algorithm proposed in the last sub-section.

ABC algorithm with respect to the size constraint

The modification of the ABC algorithm presented in the previous section consists of the insertion of the size constraint. Let k_{max} be the maximum number of significant nodes. In the modified version, the destination associated with the lowest additive cost (in term of edge cost and new significant node creation cost) is added to the tree if and only if the number of significant nodes in the tree under construction is less than k_{max}. Otherwise, a new tree is created by connecting the nearest unspanned destination to the source node.

The second class of heuristics can be designed as follows.

Spanning tree segmentation

- At first, a partial spanning tree computation algorithm is used to compute a tree spanning the destination set (for example the original algorithm of Takahashi and Matsuyama or the original ABC algorithm can be used for this purpose). This unique tree does not necessarily respect the size constraint.
- In the second phase (which is the segmentation of the unique spanning tree), this low cost tree is segmented by distributing the destinations between several subtrees taking the size constraint into account.
- If the tree set contains unbalanced numbers of significant nodes in the different trees, then a final balancing algorithm can be applied to obtain a balanced tree set.

In the following, we present our proposals for tree segmentation and charge balancing. In the segmentation problem, a tree spanning the entire destination set is given but the number of significant nodes exceeds the size upper bound k_{max}. The result of the segmentation is a set of trees; each tree in the set corresponds to a sub-tree of the delivery tree and the number of significant nodes in each tree is less than the size constraint. The segmentation can also be considered as a particular case of the Set Cover Problem. A heuristic segmentation approach has two potential objectives:

- minimize the number of k_{max}-limited trees

- minimize the overall cost of the set of k_{max}-limited trees covering the original tree.

The solutions obtained by the two different objectives can be different as illustrated in Figure 10, where the first figure shows the original delivery tree. Let us suppose that $k_{max} = 5$. Figure 10/b) presents the result when the number of trees is minimized. There are two trees to span the 8 destination nodes and the total length of this solution is equal to 18. Figure 10/c) illustrates the minimal cost solution under the constraint k_{max}. In this case, there are three trees and the cost is equal to 15.

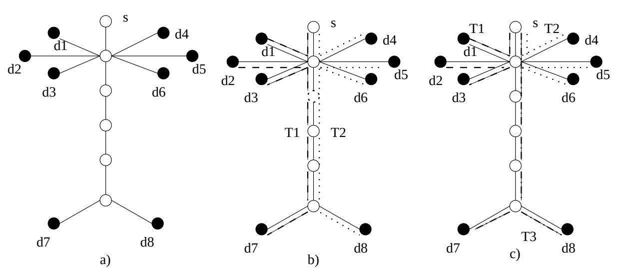

Figure 10. *Segmentation with two different objectives*

In the following, we propose a heuristic solution for this particular Set Cover Problem.

Maximal Common Path First algorithm

The Maximal Common Path First (MCPF) algorithm proposes a new alignment of the destinations in the spanning tree T. To achieve this it uses a new metric $\kappa(d_i, d_j)$ between two destinations d_i and d_j corresponding to the number of common edges of the paths from s to d_i and from s to d_j in T.

$$\kappa(d_i, d_j) = |path(s, d_i) \cap path(s, d_j)|.$$

Using this metric, a complete graph (a special metrical closure) can be computed for the destinations. Figure 11/b) illustrates the metrical closure of the tree presented in Figure 11/a).

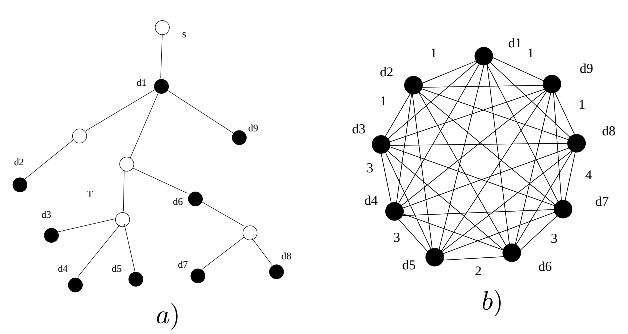

Figure 11. *The metrical closure of the destinations with the new metric*

The MCPF algorithm computes a k_{max}-limited spanning forest in the metrical closure. It is based on the well-known Prim's algorithm and consists of extending a tree T_i started at the source until no new destinations can be added. At each step, the destination with the maximal number of common edges is added to T_i, If a tree is saturated according to the size constraint, a new tree is initiated from the source to the next destination. The pseudo-code of MCPF is given by Algorithm 1.

Algorithm 1 MCPF algorithm using the Prim approach

Require: a tree T spanning the multicast group (s, D), the maximum number k_{max} of significant nodes
Ensure: a set $F = \{T_i, i = 1, \ldots, p\}$ such that each tree T_i has no more than k_{max} significant nodes
Initialization
Build the metrical closure $\bar{G}$ of the set of members D, using the "distance" κ
$F \leftarrow \emptyset$
$i \leftarrow 1$
$T_i \leftarrow$ a new tree initialized with the source s
repeat
  (d, m) is an edge of $\bar{G}$ of maximum value, such as d is in D and m is in T_i
  if $T_i \cup path(d, m)$ has no more than k_{max} branching nodes **then**
    connect d to T_i
    $D \leftarrow M \backslash \{d\}$
    recompute the cost of the edges in $\bar{G}$
  else
    $F \leftarrow F \cup T_i$
    $i \leftarrow i + 1$
    $T_i \leftarrow$a new tree initialized with the source s
  end if
until $D = \emptyset$
$F \leftarrow F \cup T_i$

Let d denote the number of destinations and t the number of trees after segmentation. Let us suppose that the destinations are distributed uniformly in the trees and there are $\lceil d/t \rceil$ destinations per tree. In the worst case, there are $\lceil d/t \rceil - 1$ branching nodes per tree to cover $\lceil d/t \rceil$ destinations. So

$$2 \left\lceil \frac{d}{t} \right\rceil - 1 \leq k_{max} \tag{18}$$

This relation gives the following approximated upper-bound of the number of trees after segmentation :

$$\frac{2d}{k_{max} + 1} \leq t \tag{19}$$

To examine the real number of trees after segmentation with the MCPF algorithm, simulations in the Eurorings topology have been executed. For each group size 100 groups have been generated randomly and a multicast tree has been computed using the Takahashi-Matsuyama algorithm. The size limit k_{max} on the headers has been set to 20. Figure 12 shows the observed number of trees per group after the segmentation.

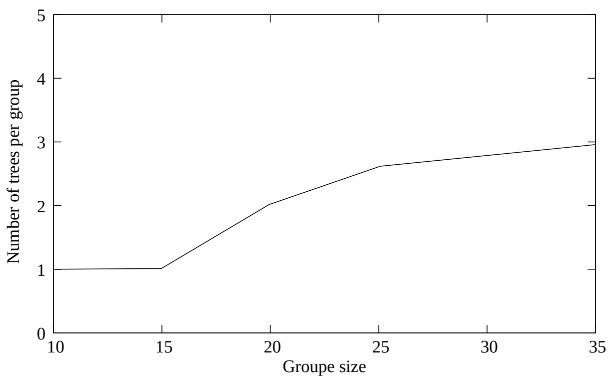

Figure 12. The number of trees after the segmentation by the MCPF algorithm.

The set of trees should be balanced if necessary. For example, using the MCPF algorithm, the resulting trees contain a number of significant nodes near to the given limit k_{max} except the last tree, which may contain only a few members. The equilibrium of the number of significant nodes in the different trees decreases the maximal length of the explicit routing header information and so increases the payload. So, the balancing operation can decrease the total cost of multicast communication.

Member Switching Algorithm

To balance the trees, the Member Switching algorithm considers the tree having the largest encoding. It removes a destination from it, then adds this destination to the tree having the smallest encoding. This process is repeated until the encoding of the largest tree is close to the encoding of the smallest tree.

The worst-case time complexity of the algorithm is $\mathcal{O}(|D| \cdot |N|)$, where $|D|$ is the number of destinations and $|N|$ is the number of nodes of the graph.

V. CONCLUSIONS AND FUTURE WORK

Explicit multicast routing is an alternative solution to resolve the scalability of multicast routing in IP. Flat explicit routing protocols generate significant overhead in routers due to the intensive processing of the datagram headers. Tree-based explicit routing could simplify the task of the routers by encoding the multicast tree in the datagrams and by using conventional unicast data forwarding between the significant nodes of the tree. The computation of the multicast route corresponding to the minimum communication cost per bit is a hard optimization problem. We formulated and illustrated this optimization in several cases: when the multicast group can be spanned with only one tree but also when several trees are needed for the group due to limitations on header size. In this latter case, we introduced the important homogeneous message fragmentation constraint to avoid complicated data transmission procedures at the source. The optimization problems are NP-difficult in these aforementioned cases and well known Steiner heuristics cannot guarantee limited cost solutions. To illustrate the introduced problems, some exact algorithms were presented but they are very expensive. For explicit multicast routing, we also proposed heuristics providing low cost, explicitly encoded multicast routes. These algorithms find approximate solutions in polynomial execution time. In particular, the ABC algorithm permits the construction of multicast trees with low communication cost when the tree should be encoded in the packet headers. If the number of significant nodes is high, tree segmentation and balancing can be performed with good performance using the presented MCPF and Member Switching algorithms.

Algorithm 2 Member Switching Algorithm

Require: a set of trees $F = \{T_i, i = 1, \dots, p\}$
Ensure: the balanced set F
repeat
  $T_s \leftarrow$the tree of F of smallest encoding
  $T_l \leftarrow$the tree of F of largest encoding
  if $encoding(T_l) > encoding(T_s) + 2$ **then**
    remove the destination d from T_l such as the significant father of d in T_l has the lowest degree
    add the member d to T_s
  end if
until $encoding(T_l) \leq encoding(T_s) + 2$

REFERENCES

[1] M. Molnár, “On the Optimal Tree-Based Explicit Multicast Routing,” in *Second International Conference on Communication Theory, Reliability, and Quality of Service CTRQ, IARIA*, Colmar, France, July 2009.

[2] S. E. Deering, “Multicast Routing in a Datagram Internetwork,” Ph.D. dissertation, Stanford University, December 1991.

[3] I. Stoica and al., “REUNITE: A Recursive Unicast Approach for Multicast,” in *INFOCOM 2000*, March 2000.

[4] L. Costa, S. Fdida, and O. Duarte, “Hop By Hop Multicast Routing Protocol,” in *ACM SIGCOMM’01)*, August 2001.

[5] J. Bion, D. Farinacci, M. Shand, and A. Tweedly, “Explicit Route Multicast (ERM),” IETF, Draft, June 2000, draft-shand-erm-00.txt.

[6] M. Bag-Mohammadi, S. Samadian-Barzoki, and N. Yazdani, “Linkcast: Fast and Scalable Multicast Routing Protocol,” in *IFIP Networking*, 2004, pp. 1282–1287.

[7] F. K. Hwang, D. S. Richards, and P. Winter, “The Steiner Tree Problem,” *Annals of Discrete Mathematics*, vol. 53, 1992.

[8] A. Boudani, A. Guitton, and B. Cousin, “GXcast: Generalized Explicit Multicast Routing Protocol,” in *IEEE Symposium on Computers and Communications (ISCC)*, June 2004.

[9] F. Kuipers and P. V. Mieghem, “Conditions that Impact the Complexity of QoS Routing,” *IEEE/ACM Transactions on Networking*, vol. 13/4, 2005.

[10] ——, “MAMCRA: A Constrained-Based Multicast Routing Algorithm,” *Computer Communications*, vol. 25/8, pp. 801–810, 2002.

[11] T. Braun, V. Arya, and T. Turletti, “Explicit routing in multicast overlay networks,” *Computer Communications*, vol. 29, no. 12, pp. 2201–2216, August 2006.

[12] R. Boivie, N. Feldman, and C. Metz, “Small Group Multicast: A New Solution for Multicasting on the Internet,” *IEEE Internet Computing*, vol. 4, no. 3, pp. 75–79, May 2000.

[13] R. Boivie, N. Feldman, Y. Imai, W. Livens, D. Ooms, and O. Paridaens, “Explicit multicast (Xcast) basic specification,” IETF, Draft, June 2004, draft-ooms-xcast-basic-spec-06.txt.

[14] M.-K. Shin, Y.-J. Kim, K.-S. Park, and S.-H. Kim, “Explicit Multicast Extension (Xcast+) for Efficient Multicast Packet Delivery,” *Electronics and Telecommunication Research Institute (ETRI) Journal*, vol. 23, no. 4, December 2001.

[15] L. Ji and M. S. Corson, “Explicit Multicasting for Mobile Ad Hoc Networks,” *Mobile Networks and Applications*, vol. 8, pp. 535–549, 2003.

[16] M. Bag-Mohammadi and S. Samadian-Barzoki, “On the efficiency of explicit multicast routing protocols,” in *ISCC ’05: Proceedings of the 10th IEEE Symposium on Computers and Communications*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 679–685.

[17] S. L. Hakimi, “Steiner’s Problem in Graphs and its implications,” *Networks*, vol. 1, pp. 113–133, 1971.

[18] M. Molnár, “Hierarchies for Constrained Partial Spanning Problems in Graphs,” IRISA, Rennes, France, Tech. Rep. 1900, 2008.

[19] R. M. Karp, “Reducibility among combinatorial problems,” in *Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher, Eds. Plenum Press, 1972, pp. 85–103.

[20] P. Winter, “Steiner problem in networks: A survey,” *Networks*, vol. 17, pp. 129–167, 1987.

[21] H. Takahashi and A. Matsuyama, “An approximate solution for the Steiner problem in graphs,” *Mathematica Japonica*, vol. 24, no. 6, pp. 573–577, 1980.

[22] J. Moulierac, A. Guitton, and M. Molnár, “Multicast Tree Aggregation in Large Domains,” in *IFIP Networking, Springer-Verlag, LNCS*, no. 3976, January 2006, pp. 691–702.

[23] F. V. Fomin, F. Grandoni, and D. Kratsch, “Measure and Conquer: Domination - A Case Study,” in *32nd International Colloquium on Automata, Languages and Programming (ICALP 2005), Springer LNCS vol. 3580*, 2005, pp. 191–203.