SELF ORGANIZING HIERARCHICAL MULTICAST TREES AND THEIR OPTIMIZATION

F. Baccelli †, D. Kofman ‡, J. L. Rougier †
†Ecole Normale Supérieure
45 rue d’Ulm, 75005, Paris, France.
‡Ecole Nationale Supérieure des Télécommunications
46 rue Barrault, 75013 Paris, France.

Abstract—Multicast routing protocols suitable for wide-area networks are being developed for the Internet. Protocols based on hierarchical trees appear to be well suited for their superior scalability and flexibility.

In this paper we show how to construct a class of hierarchical multicast trees and we analyze their performances. This study gives insight into how the chosen hierarchical structure impacts the tree performance with respect to network resource consumption. Optimal structures which minimize resource consumption are deduced, which allows for simple dimensioning rules, such as how many hierarchical levels should be used.

A stochastic geometric approach sorted out to be well adapted for this study. This approach leads to explicit expressions for the average tree cost, as a function of the hierarchical clustering, from which the optimal tree structure can then be easily deduced.

Keywords—Multicast Trees, Hierarchical Center-Based Trees, Stochastic Geometry, Poisson-Voronoi, Optimization.

I. INTRODUCTION

Multicast routing protocols are being designed and deployed to reduce communication costs when dealing with applications involving communications between multiple users. In a multicast session, one of several sources transmits the same data to multiple destinations. Multicast routing protocols are in charge of building the dissemination trees. Existing multicast routing protocols (e.g. DVMRP [21], PIM-DM [7]) were designed for local area networks where receivers interested in multicast sessions are quite densely distributed. These protocols are neither efficient nor scalable for wide areas with sparsely distributed group members [7].

Routing protocols which build a bidirectional shared tree for the whole group (instead of many trees for each source and each group) turn out to be more scalable. However, shared trees are difficult to build and to maintain. For instance, the so-called Steiner tree construction is NP-hard and requires the location of all the members3. A more pragmatic solution [8], [22] is to elect a "center" (or "core"), all the branches of the multicast tree converging to this node. The core location affects dramatically the performance of such trees, particularly for inhomogeneous member distributions and large networks [9], [22].

Hierarchical multicast trees offer much more flexibility and scalability as local trees are built and maintained independently in each cluster [12], [17], [19]. Hierarchical trees are envisaged at the IETF for wide-area multicasting [13], [19], [4].

This paper concentrates more specifically on hierarchical center based trees (HCBT). In Figure 1, a HCBT is depicted with two hierarchical levels. The network is divided into five separate domains (e.g. Autonomous Systems, countries, continents) D1, D2, D3, D4 and D5. A core is placed in each domain. A router wishing to join a multicast session will be connected to the core of its domain. If not already on the tree, this core will in turn join a higher-level core (which is C4 in this case). Thanks to this simple algorithm, the cores which have no members in their domain will not be connected to the highest level core. Thus the cost of the tree adapts adequately to the distribution of the members. This self organization of the trees will be exemplified in section V. Furthermore, the nodes of a given domain only need to know the address of their local core. How core addresses are disseminated is outside the scope of this paper [13], [15].

Different algorithms adapted to build hierarchical trees have been presented [12], [17], [20]. These proposals have strongly influenced the development of wide area multicast routing protocols at the IETF [19], [4], [13]. The trees which are analyzed are those built by the algorithms presented in section V, when the multicast group is homogeneously distributed on the plane. Nevertheless, we expect that these models could also be used to approximate the hierarchical center-based trees obtained by different algorithms of the literature [12], [17], [20].

This paper focuses on several aspects of such hierarchical multicast trees: construction algorithms stressing how the trees can be built recursively and how they adapt to the topology of the multicast group and its evolution; performance evaluation issues, and in particular dimensioning rules for such trees, which indicate how to tune the parameters of the algorithms. This analysis allows one to gain insight into the impact of hierarchical clustering on the tree cost (i.e., network consumption) and shows how such hierarchical trees should be best deployed.

For instance, our analysis shows that the number of levels available in today’s Internet topology may be enough when cores

---

1 Distance Vector Multicast Routing protocol.
2 Protocol Independent Multicast — Dense Mode.
3 This information is hardly ever available in today’s internet.

---

Fig. 1. A two-level Hierarchical Center-Based Tree
are interconnected by tunnels but may be insufficient when intermediate routers are multicast-capable[5].

A stochastic geometric approach, inspired from [1], is used to model the spatial organization of the network. This approach, already used for different network optimization problems [1], [3], [16], happens to be well adapted to the study of multicast trees. The average tree cost can be explicitly expressed as a function of different relevant hierarchical structure parameters. Optimal structures can then be easily deduced.

This paper starts with the introduction of the stochastic geometric model in Section II. The performance analysis is presented in Sections III and IV. Tree construction algorithms are outlined in Section V. Finally, conclusions are drawn in Section VI. Note that for the sake of clarity, some technical calculation have been placed in the appendix.

II. THE MODEL

The geometric approach presented in this section allows one to model the spatial organization of networks. Nodes (i.e., routers) are represented by random point patterns on \( IR^2 \), as it is usually done in multicast routing simulations [23]. Cores are also represented by other independent point patterns. The core location are chosen at random, as the cores are usually chosen administratively, independently of the member distribution. Note that the point patterns considered may not be faces or nodes also networks or multicast clouds. The trees could then be seen as the multicast interconnection structure.

The analysis is based on the random point-process theory and on the Palm calculus. An overview on these subjects can be found in the appendix of [1] (see "Point-Process Refresher") or more exhaustively in [10], [11], [6], [18].

The following notations will be used throughout this paper:

- For all finite \( B \subseteq IR^2 \), \(|B|\) denotes the surface of \( B \) (i.e., \( \int_B dx \)), for all \( x \in IR^2 \), \(|x|\) is the norm of \( x \). Finally, \( E_x^B [\xi] \) represents the mean value of \( \xi \) w.r.t. the Palm Probability \( P_x^B \), for all random variables \( \xi \) and \( k = 0, 1, \ldots, N \).

Let us define \( N \) independent homogeneous Poisson processes on \( IR^2 \): \( \pi_0, \pi_1, \ldots, \pi_{N-1} \). The point process \( \pi_0 \) represents the physical nodes (i.e., the routers) and \( \pi_k \) represents the set of cores of level \( k \) (for \( k = 1, \ldots, N-1 \)). Their respective intensities will be denoted \( \lambda_0, \ldots, \lambda_{N-1} \). The Palm probabilities with respect to these processes will be denoted \( P_0^\pi_0, \ldots, P_0^\pi_{N-1} \).

A node of \( \pi_k \) which wishes to join the group will be connected to the closest \( \pi_{k+1} \) core, for \( k = 0, \ldots, N-2 \). At the highest level, nodes of \( \pi_{N-1} \) are connected to the origin, which will be considered as a core of level \( N \). We will often look at the restrictions of these processes on a finite window, namely a finite set \( W_r \) of \( IR^2 \), which will be an open disk of radius \( r \), centered at the origin. This defines a model which is non-homogeneous (since all nodes of the multicast group and all cores will be located in \( W_r \)) but locally homogeneous (since these processes are the restrictions of fully homogeneous ones to \( W_r \)), with its core of level \( N \) located at the center of the window.

For all \( B \subseteq IR^2 \) and \( k \in \{0, 1, \ldots, N-1\} \), \( \pi_k (B) \) will denote the number of points of \( \pi_k \) which lie in \( B \), or equivalently the integral \( \int_B d\pi_k \). Let \( V_y (\pi_k) \) be the Voronoi Cell centered at \( x \in \pi_k \), i.e. the convex set defined by \( \{ y \in IR^2 : ||y-x|| < \forall z \in \pi_k, z \neq x \} \). With this definition, the following propositions are equivalent for all \( k = 0, \ldots, N-2 \):

(i) A node of \( \pi_k \) willing to join the group is connected to the closest \( \pi_{k+1} \) core.

(ii) A node of \( x \in \pi_k \) willing to join the group is connected to the core \( y \in \pi_{k+1} \) if and only if \( x \in V_y (\pi_{k+1}) \).

Thus, a Voronoi cell represents the "domain" for which a core is responsible. A graphical representation of a hierarchical tree, with \( N = 3 \), built according to these principles is shown in Figure 2.

Let \( \pi_0 \) be the point process obtained by thinning \( \pi_0 \). It will represent the routers which are members of a multicast group. \( \pi_0 \) can be easily sampled from \( \pi_0 \) by selecting independently the routers with a probability \( \pi_0 / \lambda_0 \). In this case, the members are homogeneously distributed on the plane (\( \pi_0 \) is also an homogeneous Poisson Process on \( IR^2 \)) and its intensity is \( \lambda_0^\pi = \pi_0 / \lambda_0 \).

More complex models could be considered. For the sake of simplicity, we will concentrate on homogeneous member distributions in the performance evaluation part. Non homogeneous distributions are discussed in §III-D, and in §V at the algorithmic level.

Let us also define \( \pi_1^*, \ldots, \pi_{N-1}^* \), the processes representing the sets of active nodes. These point processes are neither Poisson nor independent. They are nevertheless stationary. We will denote their respective intensities by \( \lambda_1^*, \ldots, \lambda_{N-1}^* \). Finally, let us denote \( \mu_k = E_x^\pi_k [\pi_{k-1} (V_0 (\pi_k))] \), \( k = 1, \ldots, N-1 \), the mean number of active or non-active nodes (or cores) of level \( k \) which lie in a typical Voronoi cell of \( \pi_k \), i.e. the maximal number of nodes for which a core of \( \pi_k \) is responsible. We have \( \mu_k = \lambda_{k-1}^* / \lambda_k \), for \( k < n \) and \( \mu_N = \lambda_{N-1} \pi_1^2 \) at the highest level (using our convention on \( W_r \)).

⁴By "active", we mean a core which is involved in the tree, i.e., if its Voronoi cell contains at least one receiver or one lower level active core.
This general framework allows us to analyze the different costs of the hierarchical trees. Three main costs have been identified:

- **Tree construction and maintenance costs.** This represents the number of operations necessary to build the tree, to add a new member, to delete members etc.
- **Tree cost or network consumption.** The network resource used by the multicast tree are analyzed and minimized in this paper.
- **End-to-End Delays.** The mean delays or maximum delays between potential sources and members in the multicast tree are also very important performance criteria.

The network resource consumption is analyzed in the next section. A brief analysis of the end-to-end delays is also provided in section IV. The tree construction and maintenance costs, associated with various tree construction algorithms (e.g., in section V), will be studied exhaustively in a forthcoming companion paper.

### III. Analysis

#### A. Preliminaries

We will assume that the cost of a given link is represented by its length\(^7\), as it is traditionally done in the Internet community [8], [22]. The total cost of the tree will be calculated by summing all the costs of the different links.

We will also assume that nodes are directly linked to their core. This assumption is motivated by today's Internet architecture, where only a fraction of the routers are multicast capable. The multicast nodes (or clouds) are thus linked by tunnels\(^8\). It should be understood that in reality, it is impossible to connect directly any node to its core. The tunnels will probably pass through intermediate nodes of \(\pi_0\). The direct links are just used to represent the existence of a tunnel between two nodes — the cost of this tunnel being a function of the distance between these nodes.

Finally, we will look at the case where the multicast group is concentrated within a disc of radius \(r\). For this, we will look at the above processes in the window \(W_r\), and neglect certain border effects. The total number of routers in this network is thus on average \(M = \lambda_0\pi r^2\).

Under these assumptions, the tree cost can be evaluated, as shown in Appendix A:

\[
\mathcal{C} = \pi r^2 \left( \frac{2r}{3} \lambda_{N-1} p_{N-1}^* + \sum_{k=0}^{N-2} \frac{\lambda_k p_k^*}{2\sqrt{\lambda_{k+1}}} \right),
\]

where the \(\{p_k^*\}_{k=1,\ldots,N-1}\) can be approximated by the recurrence relation:

\[
p_k^* = 1 - \left( 1 + \frac{\lambda_k - 1}{\alpha_k} \right)^{-\alpha}, \quad k = 1, \ldots, N - 1,
\]

where \(\alpha \approx 3.57\). In the following sections, the cost calculated above is studied and interpreted in two special cases: the **sparse mode case** (where the group is densely distributed) is analyzed in Section III-C. The general case can be handled numerically. Possible extensions of this model are discussed in Section III-D.

#### B. Sparse mode case

The **Sparse Mode** assumption states that \(p_k^* \ll 1\) for all \(k = 0, \ldots, N - 1\). Then, the probability \(p_k^*\) can be approximated by \(p_k^* \approx \frac{\lambda_k}{\lambda_{N-1}} p_{N-1}^*\) (see Equation (2)). Similarly, since \(p_k^* \ll 1\), \(p_k^* \approx \frac{\lambda_k}{\lambda_{N-1}}\). By iteration, we get \(p_k^* \approx \frac{\lambda_k}{\lambda_{N-1}} \forall k = 1, \ldots, N - 1\).

The cost takes a degenerate form:

\[
\mathcal{C} = \pi r^2 \left( \frac{2r \lambda_{N-1}^*}{3} + \sum_{k=1}^{N-1} \frac{\lambda_k^*}{2\sqrt{\lambda_k}} \right).
\]

It is easy to see that the optimum is reached for a single hierarchical level \(N = 1\). This result can be explained by the geometric arguments given below.

Consider the tree represented in Figure 3. The dashed lines represent the links of the multicast tree with one hierarchical level. When adding a new hierarchical level (bold links), two phenomena can be observed:

- The path between each member and the central core becomes longer as it has to pass through intermediate cores, as shown in the gray cell of Figure 3.

- A link joining a core to the central core can be shared by several members. This concentration phenomenon may considerably decrease the cost of the tree.

There is usually a tradeoff between these two phenomena. Under the **Sparse Mode** assumption, the first one is predominant, as the probability that different members lie in the same cell is negligible.

#### C. Dense mode case

Under the **Dense Mode** assumption, all the nodes of the network participate in the multicast group. Thus, \(p_k^* \approx 1\) and \(\lambda_k^* \approx \lambda_k\) for all \(k = 0, 1, \ldots, N - 1\) and the tree cost is

\[
\mathcal{C} = \pi r^2 \left( \frac{2r \lambda_{N-1}^*}{3} + \sum_{k=0}^{N-2} \frac{\lambda_k}{2\sqrt{\lambda_{k+1}}} \right).
\]

A tree structure with a minimal cost can then be determined by differentiating this expression.

\[
m_{N-1} = \frac{8}{3\sqrt{\pi}} \sqrt{m_N}
\]

\[
m_{k-1} = 2\sqrt{m_k} \quad \text{for} \quad k = 1, \ldots, N - 1.
\]
The previous equations means that, at the optimum, the concentration must be high at higher level cores (and small at lower level cores). Such a tree is represented in Figure 4.

The total cost of the tree at the optimum can be expressed as

$$C = r \frac{a^N - 1}{2^{N-1}} \left(3 \frac{2^N - 2}{2^N - 1}\right)^{\frac{1}{N-1}} (r \sqrt{\lambda_0})^\frac{1}{N-1}.$$  

It can be deduced that the optimum tree cost is reached for

$$N_{opt} = \log_2 \left( \frac{4}{3 \sqrt{\pi}} \sqrt{M} + 1 \right).$$  \hspace{1cm} (3)

The evolution of the tree cost with regards to the number of levels is represented in Figure 5 for various network sizes. It can be seen that, even for very large networks, the tree cost does not evolve significantly for $N > 3$. The Dense Mode case is in fact a "worst case" assumption so that a fairly small number of hierarchical levels is necessary whatever the size of the multicast group.

D. Remarks

The tree cost has been calculated using the length of the different links. However, the approach allows one to consider arbitrary cost functions. It is also of great interest to consider arbitrary member distributions. In this paper, we have chosen to concentrate on homogeneous Poisson processes for the sake of simplicity. However, the use of more complex point patterns can be considered, in order to incorporate such effects as clustering. The case of non-stationary point process remains difficult, as it would not be possible to use the Palm framework directly. The effects of clustering can however be studied using doubly stochastic Poisson processes (or Cox processes) for instance, as discussed in [11]. The calculations of appendix A can be generalized easily to doubly stochastic Poisson processes.

Finally, it was assumed that a node is directly connected to its closest core. This assumption is only relevant when multicast facilities are only available in very few nodes, namely the different cores. When tunneling is not used, concentration (as defined in section III-B) may also appear in intermediate nodes on the way from the node to its core. A new model, which we propose to call the Delaunay concentration tree then arises.

Given two point processes $\pi$ and $\pi^* \subset \pi$ and a core $x \in \pi$, one can associate such a Delaunay concentration tree which is a spanning tree with root $x$ and with leaves the points $y^*$ of $\pi^*$. This tree is obtained by building successively all shortest paths from $y_i$ to $x$ on the Delaunay graph (see [18][2]) of $\pi$, and by adopting the rule that when a path meets an already built path, it stops at the meeting point — hence the "concentration" effect.

The effect of concentration appears clearly on Figure 6(b). The cost of a Delaunay concentration tree obtained by simulation is plotted as a function of the group density $p_g$. As it can be seen, this curve is not linear — whereas the expression of the cost $C$ evolves in proportion with the group density $p_g$ at
all levels \((k = 1, \ldots, N)\) when direct links (i.e., tunnels) are
assumed. In this case, early simulations have shown that more
hierarchical levels seem to be necessary [5].

IV. END TO END DELAYS

In this section, a brief analysis of transit delays in a hierarchi-
cal tree is provided. A more extensive study of this perfor-
ance criteria will be available in a forthcoming companion paper.

Firstly, it should be remarked that, for the above optimal trees,
transit delays always include \(N^{opt}\) hops from the main core to
any destination. Several other issues pertaining to transit delays
can be addressed within our framework.

Secondly, another optimization problem could be posed
which consists in minimizing a cost function which is the sum of
the tree cost given in (1) and of a linear or polynomial func-
tion of the total number of levels \(N\), which coincides with the
total number of transit hops.

Thirdly, one can also investigate how much is spared in terms
of network resource consumption due to the multicast tree. Con-
sider two routers \(a\) and \(b\) which are members of the multicast
group and located at distance \(\rho\) one from the other. The mul-
ticast tree builds one route from \(a\) to the central core and another
route from \(b\) to the central core, which may have a certain num-
ber of links in common. This number measures how many re-
sources are saved by the multicast mechanism for serving these
two users. Results derived in [3], allow us to determine the dis-
tribution of the numbers of links which are common. It is well
known that the probability \(Q(\rho)\) that two points of the plane dis-
tant of \(\rho\) belong to the same Voronoi cell of a Poisson process
with intensity 1 is

\[
Q(\rho) = 2\rho^2 \int_0^{\infty} \frac{d\phi}{\phi^2} \int_0^\infty \alpha \exp(-\rho^2 A(\alpha, \phi)) d\alpha,
\]

with

\[
A(\alpha, \phi) = \alpha \sin \phi + \alpha^2 (\pi - \phi) + (\alpha^2 + 1 - 2\alpha \cos \phi) \times \left(\pi - \arccos \frac{1 - \alpha \cos \phi}{\sqrt{\alpha^2 + 1 - 2\alpha \cos \phi}}\right),
\]

(see e.g., [10]). A simple scaling argument gives that the same
probability for a Poisson process with intensity \(\lambda\) is \(Q(\sqrt{\lambda} \rho)\).

Therefore, the probability that two routers of the multicast
group, a distance of \(\rho\) apart, have only one transit hop which
is not in common is \(p_1(\rho) = Q(\sqrt{\lambda_1} \rho)\), where \(\lambda_1\) is the optimal
value given above. Using the independence assumption between
the point processes, it is easy to show that the probability that the
two routes have \(k < N = N^{opt}\) hops which are not in common
is

\[
P_k(\rho) = Q(\rho\sqrt{\lambda_k}) \prod_{h=1}^{k-1} \left(1 - Q(\rho\sqrt{\lambda_h})\right).
\]

The probability of two routes with no hop in common is

\[
P_N(\rho) = \prod_{h=1}^{N-1} \left(1 - Q(\rho\sqrt{\lambda_h})\right).
\]

The mean number of hops spared is then

\[
\mathbb{E}[S(\rho)] = N - \sum_{h=1}^{N} h p_h(\rho).
\]

A fourth issue is that of the use of the multicast tree in a many
to many multicast application. In this case, the tree is often used
backwards from one of its leaves as follows: the path from this
leaf to any other leaf in the tree is simply the shortest path in this
tree, namely the route going up to and down from the common
node of smallest level which is an ancestor of both leaves. It is
then interesting to know the average number of hops in this path,
as a function of the Euclidean distance between the two leaves.
This issue can be addressed via the above approach. The mean
number of hops in this route is then

\[
\mathbb{E}[R(\rho)] = \sum_{h=1}^{N} 2h p_h(\rho).
\]

V. HIERARCHICAL TREE CONSTRUCTION

In this section, we describe informally the two types of algo-
rithms allowing one to build hierarchical center based trees in
a distributed way, and more precisely how to select the cores
of various levels in a way which is adapted to the topology of
the multicast group. These algorithms admit as input the point
process \(\pi_0\) representing the routers, and the sub-process \(\pi_0^*\)
of \(\pi_0\) which describes the receivers\(^9\) of the multicast group. There are
two main steps:

1. Selection of the \(N\) core point processes \(\pi_1, \ldots, \pi_N\), (all are
sub-processes of \(\pi_0\));
2. Construction of the multicast tree: for each core, say \(x\) of
level \(1 \leq k \leq N\), a spanning tree is established between \(x\) and
the set of cores of level \(k - 1\) which belong to \(V_k(\pi_k)\).

We shall not specify here how these spanning trees are built.
Typical examples are Delaunay concentration trees as defined in
§III-D, at least when the network has multicast capabilities, or
direct connection trees as in Figure 4 in case of tunneling.
The points of \(V_k\) will be denoted \(\{x^*_k\}\), and those of \(\pi_k\), \(\{y^*_k\}\).

A. Principles of the algorithms

The main parameters of the algorithms are the real numbers
\(\rho_1, \ldots, \rho_{N-1}\) with \(1 > \rho_1 \geq \rho_2 \geq \cdots \geq \rho_{N-1} > 0\).

A.1 Static Algorithm

Here are the basic principles of the static algorithm, which is
most appropriate for the case of a homogeneous multicast group
over the whole plane, i.e., the case where \(\pi_0^*\) is a stationary sub-
process of \(\pi_0\), for instance obtained by a so-called Bernoulli
thinning of \(\pi_0\):

1. A single central core of level \(N\) is placed at the origin.
2. For all \(k\) decreasing from \(N - 1\) to 1, a Poisson point process
\(\pi_k\) is sampled by thinning the points of \(\pi_0\) with parameter \(p_k\).
3. For all \(k\) decreasing from \(N\) to 1, a spanning tree is estab-
lished from each core \(x^*_k\) to the cores of level \(k - 1\) which belong
to \(V_k^+(\pi_k)\), at least when there exist such cores.

\(^9\)More precisely the routers which have receivers directly connected to them.
4. Cores that have no spanning trees attached to them are discarded.

A variant of this consists in establishing a concentration spanning tree from each node \( x \) of \( M_k = \sum_{i=k}^{N} \pi_i \) to the cores of level \( k-1 \) which belong to \( V_{2k}(M_k) \).

The reason for having the central core chosen at the origin stems from the assumption that \( \pi_0 \) is stationary. This is also the reason for sampling the core point processes everywhere on the plane at random. Note that this only makes sense when concentration trees are used. In the case of tunneling, we would indeed have a central core with an infinite number of direct connections to the the infinite number of active cores of level \( N-1 \).

A.2 Adaptive Algorithm

The above static algorithm only makes sense in the case of multicast groups distributed uniformly over the whole plane. In case this is not true, the sampling of the core processes over the whole plane is a clear waste of computational effort.

Here are the basic principles of a somewhat more complex adaptive algorithm, which should be used for non-homogeneous and sparse multicast groups, as it allows one to restrict the sampling of the point processes to a subset \( W \) of the plane centered on the multicast group (e.g., the convex hull of the set of points \( \{y_i\} \) or a disc containing these points as in §III).

1. A single core \( x_N \) of level \( N \) is sampled. The location of this core depends on the multicast list.
2. For \( k = 1, \ldots, N-1 \), a point process \( \pi_k \) is sampled by a thinning of \( \pi_0 \) restricted to \( W \).
3. Each node of \( \{y_i\} \) selects the core of \( \pi_1 \) which is the closest, which defines a collection of cores \( \{y'_1\} \); a path is established between each core \( y'_1 \) and each of the nodes of \( \pi_0^k \) which belong to \( V_{y'_1}(\pi_1) \).
4. For all \( k = 1, \ldots, N-1 \), each point of the set \( \{y'_k\} \) selects the core of \( \pi_{k+1} \) which is closest, which defines a set \( \{y'_k\} \) of active cores of level \( k+1 \), which are connected to the cores of level \( k \) via paths as above.
5. A path is established between each node of \( \{y'_N\} \) and \( x_N \).

A variant of this consists in replacing \( \pi_k \) by \( M_k = \sum_{i=k}^{N} \pi_i \) in the last three steps.

B. Distributed Versions of the Algorithms

Our aim in this subsection is to show that decentralized procedures allowing one to construct such trees are feasible. By decentralized, we mean here that the computations and the data can be distributed. We will limit ourselves to one example, which is that of the variant of the adaptive algorithm based on the \( M_k \) processes (this is in fact the most difficult case). This algorithm requires that the central core have a knowledge of the whole multicast group, but it allows one to distribute the computations in a nice adaptive and recursive way.

In contrast, the static algorithm, which is not detailed here, does not require that the central core have a knowledge of the multicast group, but it is costly in terms of computations as explained above. Actually, the core locations must be disseminated in the network by some special protocols with the static algorithm.

Of course, the knowledge of the whole multicast group by the source is not scalable for large multicast groups. Intermediate solutions can be envisaged, which would require only some limited information at the source, while still allowing to distribute computations. Such intermediate solutions are not considered in this paper. For the sake of clarity, we concentrate on the following simple algorithm.

The procedure is expressed in terms of communicating agents (the routers) exchanging messages, which can be buffered. For the sake of simplicity, it is assumed that the time between the emission of a message and its reception is a constant \( \text{contime} \), and that computation times of agents are 0. These assumptions allow us to decompose the algorithms into phases naturally associated with its recursive nature. The identity of the emitter of any received message is supposed to be known and it is referred to as \( \text{sender} \) in what follows.

B.1 Behavior of the central core \( x_N \)

The behavior of the central core \( x_N \) is summarized in the following pseudo-code, which uses two lists \( \text{son-list} \) and \( \text{link-list} \) describing respectively the set of (IP addresses of) potential cores of level \( N-1 \) (not all will be used) and the list of links to active cores of level \( N-1 \) (which is the basis for establishing the spanning tree from \( x_N \) to these cores), respectively.

The procedure is initialized when the central core receives an \( \text{init-mail} \) with argument \( \{y'_N\} \), the set of receivers. After this, it samples \( \pi_{N-1} \) and asks the receivers to elect some cores within the set of points of \( \pi_{N-1} \) via request messages (\( \text{req-mail} \)). Finally it waits for the reception of link messages (\( \text{link-mail} \)) from elected cores of \( \pi_{N-1} \), \( \pi_{N-2} \) etc.

\begin{verbatim}
(begin)

(central core \( x_N \))
wait for init-mail (rlist) from sender;
\( \text{domain} := W(rlist); \)
\( \text{sample a } \pi_{N-1} \text{-thinning } \{x_{N-1}\} \text{ of } \pi_0 \text{ on domain}; \)
\( \text{son-list} := \{x_{N-1}\}; \text{U } \{x_N\}; \)
broadcast req-mail (son-list, N-1, x_N, domain) to rlist;
until \( \text{3*contime} \) do
begin
\( \text{wait for link-mail (rlevel) from sender; \}
\text{link-list} := \text{link-list U (rlevel}(x_N, \text{sender}); \)
end
\text{behave as a core;}
\end{verbatim}

B.2 Behavior of a receiver \( y'_i \) of the multicast group.

Each receiver waits for a succession of \( \text{req-mails} \) asking him to elect some closest core of level \( N-1, N-2, \ldots, 1 \) within a list. Once such a message is received, an election message (\( \text{elec-mail} \)) is sent by the receiver to the elected core:

\begin{verbatim}
(receiver node \( y'_i \))
until a req-mail with rlevel=0 is received do
begin
\text{wait for req-mail (rlist, rlevel, mode, rdomain) from sender; \}
select the closest point \( y \) of rlist;
send an elec-mail (rlist, rlevel, mode, rdomain) to \( y; \)
end
select the closest point \( y \) of rlist;
\text{send link-mail (0) to } y; \end{verbatim}

B.3 Behavior of any core

The code of a core \( x \) uses the same two lists as for the central core, with the same interpretation (but for the level). The
cast-list contains the subset of \( \{x^i\} \), which will eventually be served through this core. Below, the function Voronoi
(list, point), computes from list, the Voronoi cell w.r.t. this list with nucleus point.

A core knows that it becomes an active core of level \( k < N \) when it receives an elec-mail with level parameter \( k \). It then waits to collect the full list of receivers which have selected him. After that, it establishes a link to its father and then behaves exactly as a central core, but on a smaller domain:

\[
\text{(core } z) \text{ \hspace{1cm} wait for elec-mail (list, rlevel, mode, rdomain) from sender;}
\]
\[
\text{cast-list} = \text{cast-list} \cup \text{sender;}
\]
\[
\text{until comtime do}
\]
\[
\text{wait for elec-mail (list, rlevel, mode, rdomain) from sender;}
\]
\[
\text{cast-list} = \text{cast-list} \cup \text{sender;}
\]
\[
\text{end}
\]
\[
\text{send link-mail (rlevel) to mode;}
\]
\[
k := \text{rlevel}-1;
\]
\[
\text{if } k > 0 \text{ then}
\]
\[
\text{domain} := \text{Voronoi (list, } x \text{) \cap \text{rdomain;}}
\]
\[
\text{sample a } p_k \text{-thinning of } x \text{ on domain, with points } \{x^i\};
\]
\[
\text{son-list} := \{x^i\} \cup \{x\};
\]
\[
\text{broadcast req-mail (son-list, } k, x, \text{ domain) to cast-list;}
\]
\[
\text{else}
\]
\[
\text{broadcast req-mail } \{x\}, 0, x, 0 \text{ to cast-list;}
\]
\[
\text{end}
\]
\[
\text{behave as a core;}
\]

The algorithm establishes the tree in \( N - 1 \) phases, all of length \( 3^* \text{comtime} \). The first phase leads to the construction of the links between \( x_N \) and its core sons of level \( N - 1 \); the second one between the elected cores of level \( N - 1 \) (which may include \( x_N \)) and the cores of level \( N - 2 \) etc. Once the algorithm terminates, one retrieves the spanning tree structure recursively from \( x_N \) via the link-list variables. An execution of this code is illustrated in Figure 7. The final tree is depicted in Figure 8.

B.4 Addition to or deletion from the multicast group.

When a node \( z \) or a set of nodes \( S \) is added to the multicast group, the principle consists in locating, for each added node, the already active core to which it is the closest, and if this node is of level \( k \), in constructing the spanning tree (here with one terminal node) from this point to \( z \) by using the same procedure as above.

This is illustrated in Figures 9. The dual procedure allowing one to remove members is immediate. An interesting feature of these two complements is that the trees obtained by an addition (or a deletion) are statistically equivalent to those which would have been obtained when applying the initial algorithm to the group with the additional member (or to the group without the deleted member). In other words, such trees have the capability of evolving with the topology of the multicast group, and of nevertheless preserving a form of statistical invariance.

B.5 Robustness w.r.t. degradation.

Assume a given link or set of links in a subpart of the tree enters into some degraded mode, e.g. measured by round trip delays. Then the algorithm can be used as follows: detect the core say \( y \) of the smallest level such that all the links which are in a degraded mode are contained in the Voronoi cell of \( y \). Then re-sample the subtree rooted in \( y \) w.r.t. the sublist of receivers which are served through this core. Such a re-sampling, which again preserves the statistical properties of the global tree, is illustrated in Figure 10.
VI. CONCLUSION

In this paper we have analyzed the construction and the performances of hierarchical multicast trees. A stochastic geometric approach based on [1] was used to compute the average tree cost (network consumption) as a function of the hierarchical structure. The goal was to get some hints on how hierarchical multicast trees should be best deployed to minimize network utilization.

In the case where tunneling is used to interconnect the routers involved in multicast routing, it was possible to calculate the tree cost explicitly (for arbitrary cost functions), at least in the case of a multicast group which is locally Poisson in a disc, and then deduce optimum structures for which network consumption is minimal. The optimum was outlined to be reached for a particular structure with high concentration at the higher levels, and a small number of hierarchical levels. However, these results are not accurate when tunneling is not used, where more hierarchical levels seem to be necessary.

Possible tree construction algorithms were also discussed which highlighted several interesting properties of hierarchical trees, namely their flexibility and robustness to deletion and addition of nodes, and their adaptability to local degradations.

Finally, it should be remarked that the models presented in this paper allow one to consider other important performance problems. In particular, the general model where tunneling is not used is being investigated. Furthermore, other cost functions such as the tree construction cost are being considered.

APPENDIX

I. TREE COST EVALUATION

We will denote $\mathcal{C}$ the average cost of the multicast tree in the window $W_r$, and $\mathcal{C}_k$ the mean total cost of the links joining the cores of $\pi_{k-1}$ to their cores of level $k$. By construction, one can write $\mathcal{C} = \sum_{k=1}^{N} \mathcal{C}_k$ with

$$\mathcal{C}_N = \mathbb{E} \left[ \int_{W_r} ||x|| \pi_{N-1}(dx) \right]$$

$$\mathcal{C}_k = \mathbb{E} \left[ \int_{W_r} \int_{V_\gamma(\pi_k)} ||z - y|| \pi_k(dx) \pi_k(dy) \right].$$

From Campbell's formula ([1, appendix]), $\mathcal{C}_N = \lambda_{N-1} \int_{W_r} ||x|| \, dx$ so that finally $\mathcal{C}_N = \lambda_{N-1} \frac{2n}{3} r^3$. Similarly, we can write

$$\mathcal{C}_k = \pi r^2 \lambda_k \mathbb{E} \left[ \int_{V_\gamma(\pi_k)} ||z|| \pi_k^2(\pi_{k-1}(z)) \right].$$
Denote $z_k^\bullet$ the point of $\pi_k$ which is closest to the origin. From Neveu’s exchange formula ([1, appendix]),

$$ C_k = \pi^2 \lambda_k^{-1} E_0^{T_k-1} \left[ | | z_k^\bullet | | \right] = \pi^2 \lambda_k^{-1} E_0^{T_k-1} \left[ | | z_k^\bullet | | | \pi_{k-2} (V_0 (\pi_{k-1}) > 0) \right], $$

where the last relation follows from the conditional distribution interpretation of Palm probabilities. $| | z_k^\bullet | |$ is a functional of the process $\pi_k$, whereas the event $\{ \pi_{k-2} (V_0 (\pi_{k-1}) > 0) \}$ is a functional of the processes $\pi_{k-1}, \pi_{k-2}, \ldots, \pi_1$ and $\pi_0$, which are independent of $\pi_k$. Therefore,

$$ C_k = \pi^2 \lambda_k^{-1} E_0^{T_k-1} \left[ | | z_k^\bullet | | \right]. \quad (6) $$

As proved in [1, page 6],

$$ E_0^{T_k-1} \left[ | | z_k^\bullet | | \right] = \frac{\Gamma \left( \frac{2}{3} + \frac{1}{5} \right)}{\pi \lambda_k^{\frac{2}{3}}}, \quad (7) $$

where $\Gamma$ denotes the Euler gamma function. Thus, $C_k = \pi^2 \frac{\lambda_k^{-1}}{3 \sqrt{2 \lambda_k}}$. Finally, the expression of the cost takes the simple form:

$$ C = \pi^2 \left( \frac{2 \pi N_1 - p^*_N}{3} + \sum_{k=1}^{N-1} \frac{\lambda_k p_k^*}{2 \sqrt{\lambda_k + 1}} \right). $$

II. DETERMINATION OF $\{p_k^*\}_{k=1, \ldots, N-1}$

The probability $p_k^*$ is defined as

$$ p_k^* = P_0^{a_k} \left\{ \pi_{k-1} (V_0 (\pi_k) > 0) \right\}. $$

If the processes $\{\pi_k\}_{k=1, \ldots, N-1}$ were Poisson and independent of $\pi_k$, we could write:

$$ p_k^* = 1 - P_0^{a_k} \left\{ \pi_{k-1} (V_0 (\pi_k) = 0) \right\} = 1 - E_0^{a_k} \left\{ \exp \left( -\lambda_{k-1} V_0 (\pi_{k-1}) \right) \right\}. $$

There are no explicit formulas for the distribution of the size of a typical Voronoi Cell. Nevertheless, fine approximations are available, most of which are enumerated in [14]. It has been shown that $|C_0 (x)|$ admits a density close to

$$ f(x) = \frac{b^\alpha}{\Gamma (\alpha)} x^{\alpha-1} \exp (-bx), $$

with $b \approx \alpha \lambda$, if $\lambda$ denotes the intensity of the underlying Poisson process, and $\alpha \approx 3.57$. Hence the expression:

$$ p_k^* = 1 - \left( \frac{\alpha \lambda_k}{\lambda_k + \alpha \lambda_k} \right)^\alpha, \quad (8) $$

Note that the processes $\{\pi_k\}_{k=1, \ldots, N-1}$ are not Poisson processes. However, simulations have shown that this approximation gives accurate results.

REFERENCES


