Optimization of Hierarchical Multicast Trees in ATM Networks

Sh. Barakat , J.L. Rougier
ENST, 46 rue barrault
75013 PARIS FRANCE
Email : {barakat , rougierj}@res.enst.fr

This paper concentrates on hierarchical multicast trees and their use in ATM networks. Multicast capabilities of ATM networks are still very limited. It poses many problems for an efficient support of new Internet applications. However, an intensive development and research effort is being made to enhance PNNI multicast capabilities. The introduction of point-to-multipoint connections and multipoint-to-point connections[1] allows to envisage to build sophisticated multicast services. It becomes possible to build multipoint-to-multipoint connections. In particular, hierarchical multicast trees have been shown to be easily implemented in ATM networks, using the inherent PNNI hierarchy [3]. Hierarchical trees have been introduced recently in the Internet community as they represent the most scalable multicast routing solution for use in large networks. Their utilization in ATM networks would allow to better optimize network resources.

This paper discusses the implementation of hierarchical multicast trees in ATM networks and analyses their performances. The total cost used by a hierarchical tree is evaluated by simulation as a function of the PNNI hierarchical structure. Simple dimensioning rules can be easily deduced.

Keywords

ATM networks, PNNI, Multicast, Routing optimization, Hierarchical shared trees, Network dimensioning, Looping.

1 Introduction

This paper analyses the new multicast routing capabilities that could be added to ATM networks. As PNNI, the ATM routing protocol of the ATM-Forum, is rapidly evolving to support multicast capabilities, it becomes possible to envisage to implement sophisticated multicast services. In this paper we show how hierarchical multicast routing protocols can be used in PNNI. Hierarchical multicast routing protocols have been introduced recently as they allows to scale to arbitrary large groups. The performance of such protocols is studied by means of simulation. Let's first start by a quick overview of the state-of-the-art in multicast routing protocols.

The support of multicast transmission becomes a necessity in nowadays networks especially with the new multimedia applications such as video and audio conferencing. With multicasting, a source doesn't need to send a particular information several times even if several members are willing to receive it. The destinations interested by this information join a multicast group and a single copy is sent to the group address. This copy passes on a tree joining the source to the group members and is duplicated when needed. The resources required are then reduced as a particular link is crossed at most once by a piece of information.

The most important factor affecting the performance of a multicast protocol is the efficiency of the tree built to join the group members together. This tree forms the path followed by the multicast traffic and then determines the quality of the reception at each destination in terms of QoS. Also, the resources required to build a multicast tree limit the scalability of the protocol. By scalability we mean the multicast protocol ability to give an efficient service even in case of wide networks. The tree consumes two types of resources: the bandwidth that may be reserved (or that is used) on tree branches and the volume of storage information needed to maintain the tree state.

[1] In PNNI Version 2

A good multicast protocol must optimize the resource utilization and adapt to the dynamic change in group membership. At the same time, it must provide an acceptable QoS for the recipients. These requirements become more stringent with the new high-bandwidth multimedia applications due to the hard QoS constraints they impose.

Many types of trees have been proposed to satisfy these requirements. Source specific trees (DVMRP, MOSPF, ATM point-to-multipoint VC) give the shortest delay between sources and receivers but they require an entry per source and per group at each node to maintain the tree state. Thus, these solutions doesn't scale for large multicast groups with several senders. Also, no control is done on the total bandwidth consumed.

At the opposite, a single bidirectional shared tree can be used to connect the members together. Each member will then send and receive traffic on the same tree. This way, the tree state is reduced to a single entry per group which eliminates the influence of the number of sources and lets the protocol scale to large groups — see figure 1.

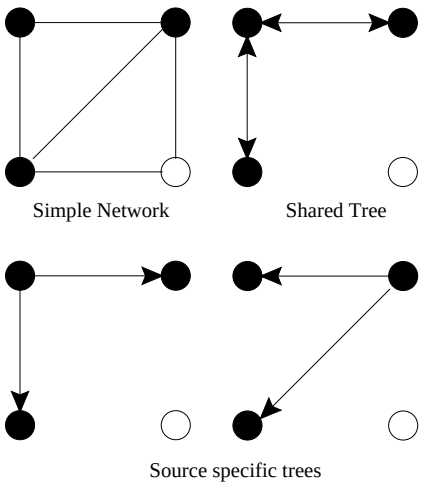

Figure 1: Some kinds of trees

In order to optimize network resources, shared tree can be designed to minimize the overall cost consumed by its branches. The cost can represent the number of links used, the capacity reserved etc. This optimization is known as the Steiner problem [5]. This problem has been proven to be NP-Complete [6]. Many heuristics have been proposed to build such trees in a polynomial time [7, 8, 9]. These heuristics propose solutions close to the optimal that can be computed in a distributed manner. Some of them consider, beside the minimization of the total cost, some QoS constraints such as delay and delay variation [10]. The implementation of Steiner trees in real networks remains difficult because of the global knowledge of the network and the heavy computation they require. Also, they cannot easily cope with the dynamic change in group membership, as they must be restructured periodically.

Other type of shared trees have been designed, in order to be more easily implemented. The most widely used is the *Center Based Tree* (CBT [11], PIM [14]). With CBT, a particular node of the network is selected as the *center* (designated "Core" in CBT or "Rendez-vous Point" in PIM). The members join the shared tree by connecting the center (along the shortest path) — see figure 2. This solution reduces the information volume needed to maintain this tree, but its performance can degrade significantly if the center is not suitably placed. The center placement problem is however NP-Complete.

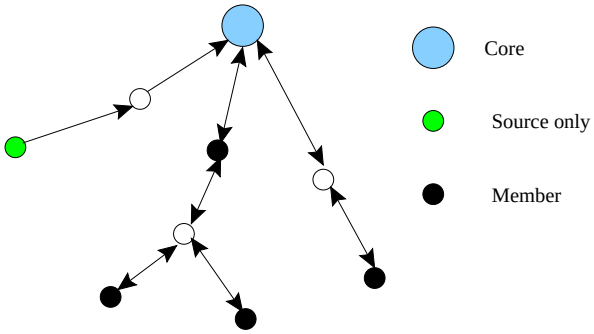

Figure 2: The Center Based Tree

The center location problem can be solved by using several cores interconnected hierarchically. These types of trees are presented in the next section.

This paper concentrates on hierarchical multicast trees in ATM networks. For the moment, only point-to-multipoint VCs are available in ATM networks. It means that a point-to-multipoint connection must be set-up for each source. The introduction of shared trees in ATM networks was not possible because of the AAL5 cell interleaving problem. Different cells would be mixed together at a merging point, and the recipient would not be able to rebuild the respective packets as there is no way to differentiate different flows that share the same

connection in AAL 5. This technical problem has been solved recently and multipoint-to-point VCs, with VC merging capabilities, are being defined in PNNI v2. It allows to envisage to build bidirectional multipoint-to-multipoint connections. For instance, Hierarchical shared trees can be easily introduced in ATM networks using the PNNI hierarchy as shown recently in [3]. This paper analyses the performances of this protocol. By simulations, we study the variation of the total cost of the tree as a function of the PNNI hierarchical structure. It allows to get simple dimensioning rules for multicast trees.

In the following section, the Hierarchical Multicast Trees are presented. In Section 3 the simulation model is explained. The results are interpreted in section 4. Finally, section 5 concludes this work.

2 Hierarchical Multicasting

The Center Based Tree, as all shared trees, reduces the volume of state at nodes and hence allows several large groups to exist simultaneously. However, the center location and the traffic concentration problems form the two drawbacks of this approach.

The computation of an optimal location of the center is an NP-complete problem. The solutions close to the optimal are function of the distribution of group members in the network. Because the group membership changes very dynamically, a particular placement may give a bad performance in some cases[2]. This problem becomes more complicated when the network size grows.

The concentration of traffic at the center causes the saturation of links around it especially if it is used by many groups. This saturation limits the number of groups and hence the scalability of the protocol.

The solution to this lack of scalability is the placement of multiple centers. A local group finds a center next to it which improves the performance of the multicast tree. The traffic will be concentrated at many points instead of one. Each center forms the root of the shared tree joining the members which are close to it. These trees are in turn connected by other centers placed at a higher level. The recursion continues until we get a single center at the highest level. Thus, a hierarchical shared tree is built having members at the lowest level and centers at different intermediate levels. If a core has no member in his area, then it is not active and does not try to join any higher-level node. This way, this hierarchical structure makes the multicast tree performance acceptable whatever the distribution of members is.

Many propositions have been presented in the Internet community to build a multicast tree based on a hierarchy of centers. OCBT[3] [12] assigns logical levels to multiple cores and describes a mechanism to join them together by a shared tree. It was also proved that the resulting tree is loop-free[4] and adapts very quickly in case of link failures. HPIM [15], the hierarchical extension of the Sparse Mode version of the Protocol Independent Multicast, has also been introduced. Some interesting advantages of this hierarchical protocol have also been outlined. However, the total bandwidth consumed by the tree built with these propositions hasn't been studied.

In ATM networks where QoS is guaranteed, the optimization of the bandwidth consumption is of paramount importance. For this reason, the multicasting scheme proposed in [3] has taken into account the total cost of the resulting tree. Because a part of our work is based on this proposition, a brief description is given in what follows.

2.1 Hierarchical Multicasting in ATM Networks

The idea is to use the logical hierarchy built by PNNI to place the cores. In PNNI [1] and at each level, the nodes which have the same address prefix form a Peer Group (PG). Each PG elects one of its node as a leader which represents it at the higher level. This leader aggregates the information on the PG and passes them up. It forwards down the information received from higher levels on the rest of the network. At the bottom of the hierarchy, we find the physical switches. At the other levels, the nodes and the links are logical. A logical node represents the set of PGs below it and a logical link aggregates the information on the physical links between the PGs represented by logical

[2] When the center is far from a local group.

[3] Ordered Core Based Tree.

[4] Even in case of unicast rooting loops.

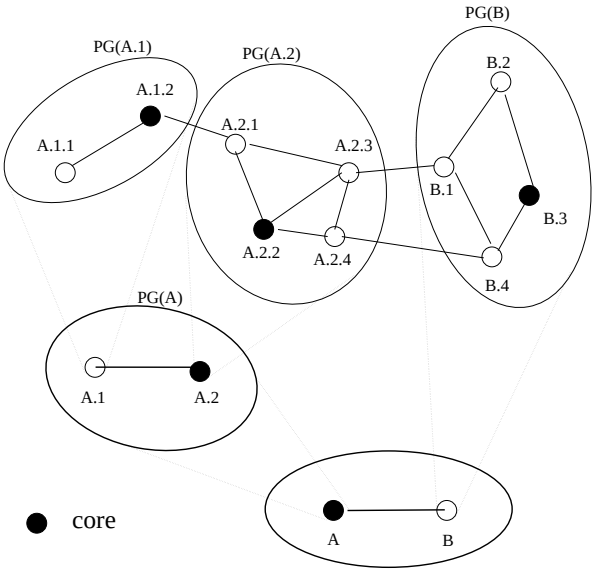

Figure 3: Placement of cores

nodes.

In each PG, a node has a complete vision of its PG and of those located between it and the highest level. In this multicast extension, a particular node of each PG is chosen as a core (Figure 3). The periodic flooding done in PNNI to update the routing tables is used to distribute the identity of the cores.

When a new member decides to join the group, it sends a message to the core of its PG. If this core finds that it is not on the multicast tree, it sends another message to the core of its parent PG. This will continue until a node that is already on the tree is reached. A shared multicast tree is then built at each level. A branch of the tree at a level X corresponds, at the lower level, to the path between the two cores of the PGs represented by its extremities (Figure 4). This mapping is repeated until the physical level is reached.

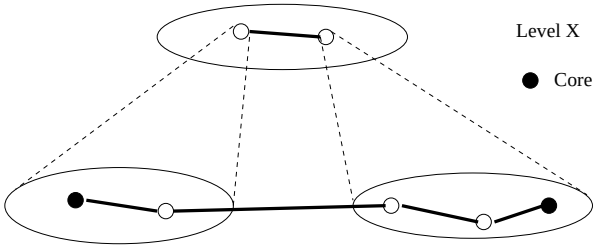

Figure 4: Mapping between trees of two levels

This multicasting scheme has the advantage of being scalable with respect to the network size because of its use of the PNNI hierarchy. The number of cores and their distribution follow the network topology. Also, it is scalable with respect to the group membership variation as only the cores which have a node in their PG are used. The tree is computed in a distributed manner and a member can join and leave the group without informing the others.

The structure and then the performance of the multicast tree is binded to the hierarchy built by PNNI. In this paper, we analyze the effects of the PNNI structure on the multicast tree, and determine the best hierarchical clustering in order to minimize the overall tree cost.

3 The simulation model

Using the scheme already described, the optimization of the resource utilization requires the computation of the best partition of the network in PGs and levels that minimizes the total cost of the multicast tree. We will assume that the same bandwidth is consumed on all the branches of the tree. If we consider that all the links of the network have the same capacity, a best representation of the cost to pay when adding a particular link to the tree will be its length. In practice, the installation of a long link is actually costlier than that of a shorter one having the same capacity. Also, because external links are generally long, the adoption of the length as a cost function will prevent the traffic from passing through other PGs when an internal route exists.

To model the hierarchy of an ATM network, we use the N-level hierarchical graph of GT-ITM[5] which is a package for generation graph models of internetworks. It uses the Stanford GraphBase for representation of graphs. This generator builds the graph in N steps which correspond to the N levels of the PNNI hierarchy. It begins at the top level N with a connected graph equivalent to the highest PG and at each step of the recursion, each node is substituted by another connected graph representing the PG located below a logical node (Figure 5). When replacing a node by its child PG, an external edge is connected to a randomly chosen point from the new graph.

The parameters of the model are the number of levels N, the graph size M and m_i, the av-

[5]Georgia Tech Internetwork Topology Models.

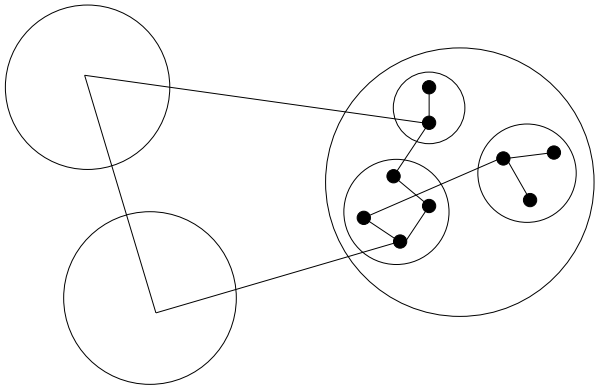

Figure 5: Generation of a 3-level graph

erage number of nodes per PG at level i[6]. The following equation is always verified:

$$M = m_N.m_{N-1}.....m_2.m_1 \quad (1)$$

An example of a graph generated for $N = 3$, $m_1 = m_2 = m_3 = 5$ is shown in figure 6.

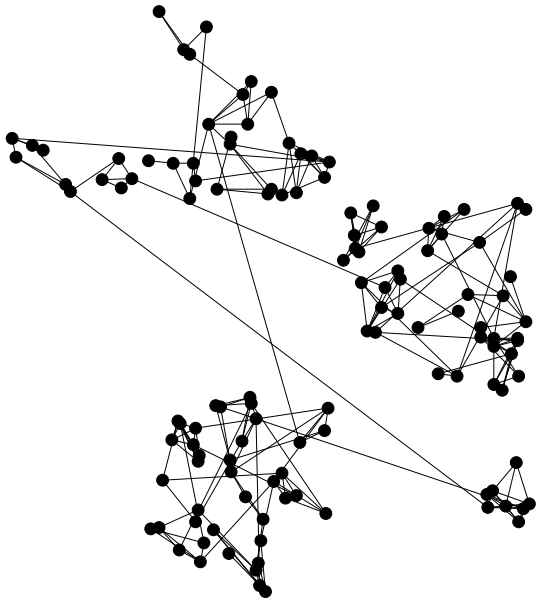

Figure 6: Example of a 3-level graph

Remark that with this generator, two PGs can be connected by at most one link which is not always met in practice. However, it has absolutely no influence in our case as only one external link will ever be used to exit a given Peer-Group. This is because only cores are able to pass traffic out of the PG and there is only one core per peer-group by construction. Even if there are many external links between two PGs, only one of them will ever be used at a given time.

After the generation of the graph, a level 1 core is randomly chosen in each PG[7]. Then, in a given level 2 Peer-Group, one of the cores of level 1 belonging to the same level 2 PG is in turn chosen randomly. This random selection continues until the m_N cores of level $N-1$ are determined, from which one is chosen as a root of the multicast tree.

Members are then randomly added. A new member joins the core of its PG and the shortest path between them is marked on the tree (i.e. this means that a connection is established to carry the multicast traffic). The Join message stops at this core or at the first tree node encountered. If the core does not yet belong to the group, it will in turn join its level 2 core, and so forth.

Whenever a core discovers that it is a leaf of the multicast tree with no attached members, it sends a Disconnect message and the branch which connect him is released.

3.1 Looping problem in Hierarchical Multicasting

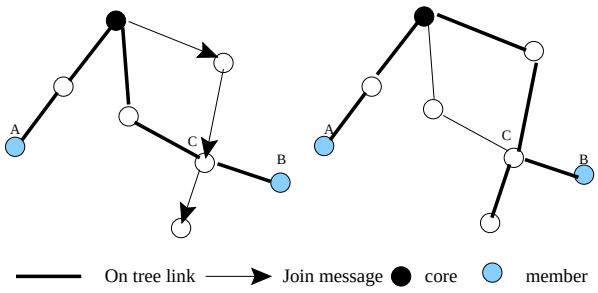

Figure 7: Solution to the looping problem

Loops can occur if no care is taken when marking the links between several cores. In figure 7, the core has several members in his Peer-Group. Thus, it must connect to its level 2 core. However, when trying to reach its core, the Join message reaches a node that is already on the tree. This would form a persistent loop. For instance, a packet sent by node B would loop between the core and node C. To let the message continue its way without creating a loop, the initial subtree must be changed. The branch between the encountered node and the core must be eliminated. The subtree rooted at

[6] The generator has been modified to accept real values for each m_i.

[7] Totally, we get $m_N.m_{N-1}.....m_2$ level 1 cores.

this node becomes attached to the new branch (Figure 7). This way, a priority is given to the establishment of a branch going from a core. In general, when a Join message sent by a core of level X finds in its way a node which is on a branch established by a core of level Y, it cuts this branch if $Y < X$ and stops at this point otherwise. The addition of this mechanism to our model guarantees the freedom of the multicast tree from loops as proved in [13].

3.2 Output of the simulation

Given a network size M, we vary the number of levels N and, for each value of N, we vary the set $\{m_i\}_{i=1,...,N}$ under the constraint given in equation 1. For each graph parameters, we take multiple values of group size G and then we build the hierarchical graph and the multicast tree joining the members. Next, the total cost of the tree is calculated. For each case (a given N, m_i and G), the computation of the tree cost is repeated many times in order to get mean values[8]. To evaluate the performance of the resulting shared tree, its cost is compared to a Steiner tree, generated by the heuristic for the Steiner problem described in [4]. At the end, the different results are plotted. The graphs shown correspond to a network of 100 nodes.

4 Simulation results

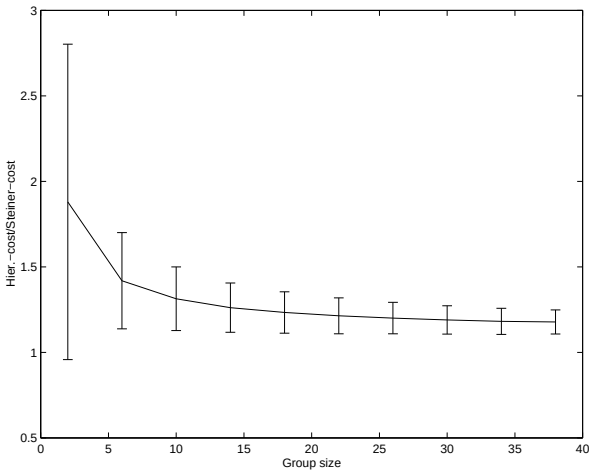

Figure 8: Tree performance vs. group size

[8]The consideration of many cases is necessary because of the diversity of network topologies and the variation of group distribution.

First, for a given group size G, we vary the network parameters N and m_i and we compute the ratio of the hierarchical tree cost to that of the Steiner tree. The different ratios obtained for a given G are then averaged and the variation of the tree performance is plotted (Figure 8). The improvement in the performance is evident and is the result of the core placement which becomes better with large groups. In spite of the bad results we got in some cases (We find a ratio of 10), the deviation is small and it decreases when the group size grows. For a given placement of cores, a small group sparsely distributed in the network leads to a bad performance tree. This is because the cores, which are placed randomly, may be badly placed — for instance the cores may be far from the members. These placement problems have less impact when the size of the group increases. Actually, the probability that a core is far from the members becomes small. Note that such hierarchical trees can be more efficient than the chosen Steiner heuristic if the members are close from each other.

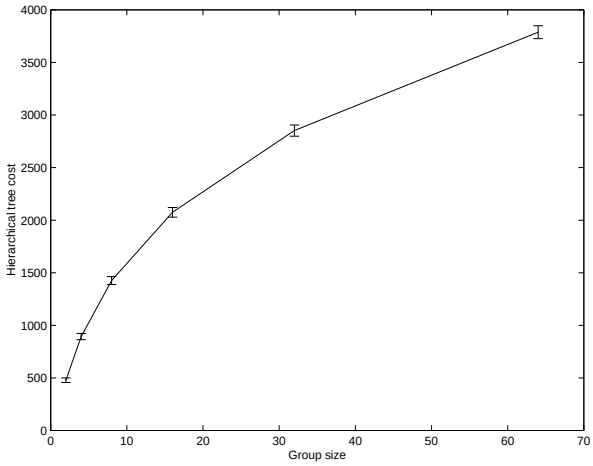

Figure 9: Total tree cost vs. group size

Next, the total cost of the tree is plotted vs the group size (Figure 9). The addition of a new member creates a new branch which increases the overall tree cost. Due to the multicast traffic merging in shared trees, the new branch stops at the first node of the tree. When the group size increases, the probability that the tree passes next to the new member increases and then the cost paid to establish the new branch decreases. This is illustrated by the decrease in the curve slope[9] in figure 9.

[9]The slope remains always positive.

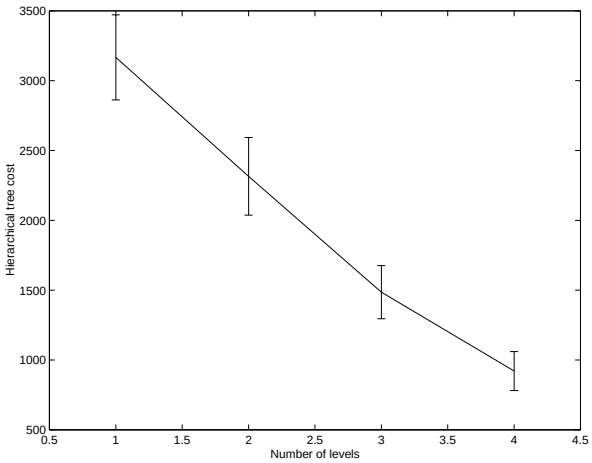

Figure 10: Tree cost vs number of levels

In Figure 10, the impact of the number of hierarchical levels is analyzed. For each number of levels, the tree cost is averaged over a set of m_i and group sizes. This curve shows an improvement in tree performance for a great number of levels. This result can be explained by the concentration that the hierarchy creates. Actually the addition of new levels allows to concentrate the multicast traffic around the different cores. It allows to reduce the number of "external" links that are used to interconnect the peer-groups. For instance, let's consider the PG depicted in figure 11. Two close members may be connected to their core by different links. When we insert a new hierarchical level, this large PG will be partitioned into some smaller PGs. Putting these close members in the same PG will force them to join the same core before joining the initial one. The different paths that exist between these members and the rest of the PG due to the existence of many external links will be substituted by the shortest path between the new core and that of his parent PG. The multicast traffic is then concentrated, hence the reduction of the cost.

Now, given a certain number of levels, we study the effect of the size of the PGs (i.e. the distribution of cores at different levels) on the cost of the tree. Let's consider first a network with two levels. Here, because a single PG exists at level 2, m_2 represents its size and the number of PGs at the physical level. It represents also the number of cores at level 1[10]. For a group size G, we study the variation of the tree

[10]One of these cores is chosen as a root of the tree.

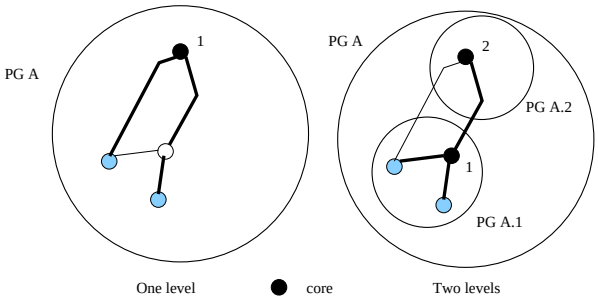

Figure 11: The concentration of multicast traffic

cost as a function of m_2. We find that increasing m_2 always makes the tree costlier. However, this phenomenon doesn't have the same importance in case of large and small groups. For small ones (Figure 12), the cost increases slowly. On the other hand, the variation is more important in case of large groups (Figure 13).

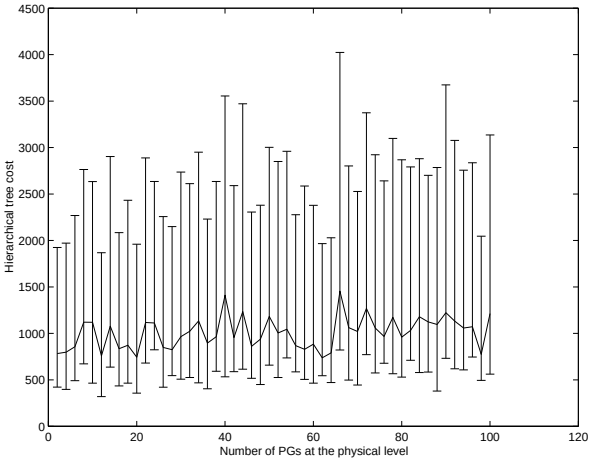

Figure 12: Cost variation for small group(Average, Min & Max)

The increase in the cost is caused by the dispersion of multicast traffic which is the opposite of the concentration introduced by the insertion of a new level. Here, a large PG is substituted by smaller ones which belong to the same level. This partition gives each core of a new PG the possibility to choose its own route to the higher core. Different paths may be chosen especially if many external links exist. This disperses the traffic in the backbone connecting the cores which increases the total cost — although the tree inside a PG becomes smaller (Figure 14). In case of small groups, the members are sparsely located and the traffic is naturally dispersed. Thus, the addition of PGs

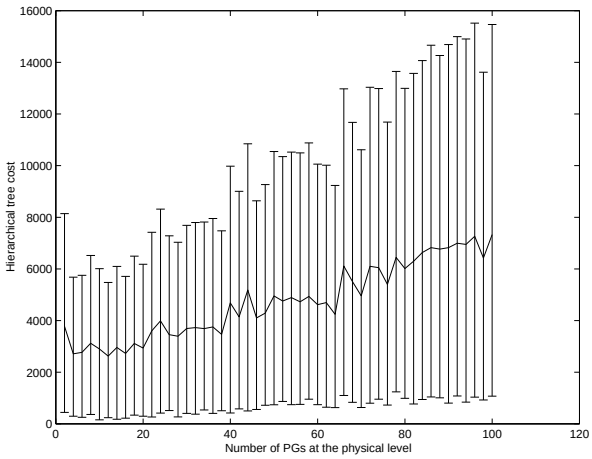

Figure 13: Cost variation for large group(Average, Min & Max)

doesn't affect the tree performance.

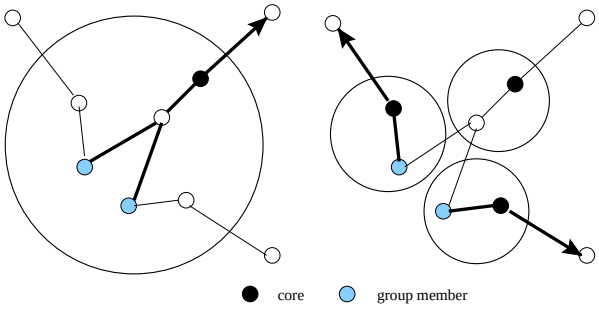

Figure 14: The dispersion of multicast traffic

Note that for other values of N, we find the same result. The lowest cost is always obtained for the smallest values of $m_N, m_{N-1}, ..., m_2$. Hence, the best partition that reduces the multicasting cost is the one that considers a great number of levels and small PGs of size 2 for levels $N, \ldots, 2$. The size of PGs at level 1 (m_1) is then easily deduced from equation 1,

$$m_1 = \frac{M}{2^{N-1}}$$

As a result of this partition, the best multicast tree for a given number of levels N involves one core at level N and 2^i cores at level i. This guarantees the best concentration of multicast traffic.

It should be understood that it doesn't mean that all graphs must be divided in Peer-Groups of size 2. Actually, the graphs on which the tree cost is calculated are different at each iteration. The total number of points remains the same, but their respective location and the links which interconnect them vary depending on the chosen hierarchical parameters.

Given an arbitrary flat network, it is generally not possible to divide it into several Peer-groups. A peer-group must actually be a connex sub-graph (for routing information dissemination). It may not be possible to find any connex sub-graph in the original network. The results presented above give very general dimensioning rules but are not suitable for optimization of arbitrary networks. However, the simulations allow to conclude that, if a hierarchical structure *already* exists (i.e. the original graph can be divided iteratively into several connex sub-graphs), then the best multicast trees are obtained by using as many hierarchical levels as possible and by defining a core in each peer-group.

Furthermore, it sorts out that the Peer-Groups should not be partitioned into several multicast routing domains if they can't be further divided into Peer-Groups. Actually, simulations on general flat networks (which have a priori no inherent hierarchical structure) have also been made. Such a flat graph is depicted in figure 15. In this model, the cores are placed randomly in the graph.

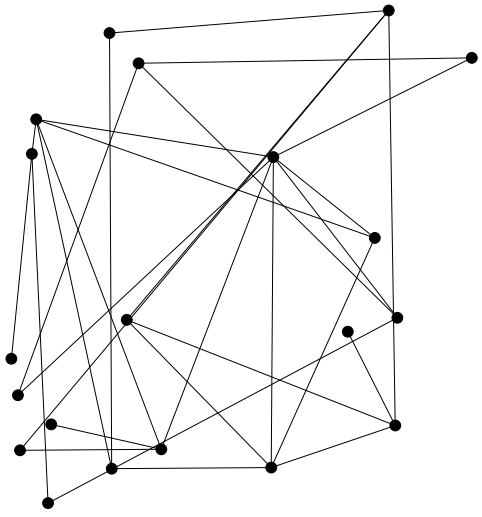

Figure 15: Example of a flat graph

It sorts out that, in this case, the addition of hierarchical levels increases the cost of the multicast tree (see figure 16) — exactly the opposite as compared to the previous model. This increase is due to the core location problem. As there is no "canonic" choice for the core placement, they are located randomly in the graph which may be not suitable and may disperse the

traffic. In the previous model, on the opposite, the cores where placed accurately (one per each peer-group) so that it concentrates the traffic and decreases the cost of the resulting tree.

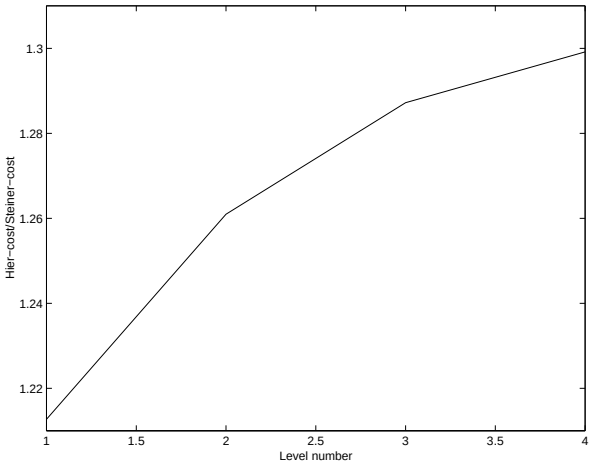

Figure 16: Tree cost vs number of levels for flat networks

To put it in a nutshell, cores should not be added if one does not know where to locate them ! In average, it would just disperse the traffic and the performances would decrease. The efficiency of a hierarchical multicast tree depends strongly on the hierarchical structure of the underlying network. The best multicast tree is obtained by dividing the network using as many levels as possible, and then by placing a core (and only one) in each peer-group.

5 Conclusion

This paper concentrated on hierarchical multicast tree dimensioning in ATM networks.

The introduction of point-to-multipoint and multipoint-to-point connections in PNNI makes it possible to implement multipoint-to-multipoint connections. In the presented approach, a single multipoint-to-multipoint VC is established to inter-connect the group members. As all shared trees, this approach was shown to reduce the network resources consumption as there is only one connection for the group, regardless of the number of sources. The proposed multipoint-to-multipoint connections are built using the hierarchical structure of PNNI, as proposed in [**?**]. It uses many cores placed at different levels to build the multicast tree. It was shown that this core hierarchy allows to solve the center location problem that arises in center-based trees. The hierarchical trees concentrates the traffic at multiple points instead of a single one, which reduces the effect of a bad core placement on the performance of the multicast tree.

This paper determined how the hierarchical structure of PNNI impacts on the total cost of the multicast tree. Simulations were used to compute the total cost of the multicast trees built in various PNNI networks, created by a random graph generator. Our goal was to find how the network should be dimensioned in order for the multicast trees to use as little network resources as possible.

The results showed that, due to the concentration of multicast traffic at each level of the hierarchy, the higher the number of hierarchical levels is used, the less network resources are consumed by the multicast tree. Also, it sorted out that PGs of small size must be chosen.

Note that this partition of the network is convenient to reduce the cost of multicast communications. Its effect on the unicast routing must also be taken into consideration. The volume of routing tables and the efficiency of the connection set-up procedure depend on the PNNI hierarchy. In case of a conflict, a compromise between multicast and unicast performances must be considered.

The implementation of such trees in ATM networks raises many other interesting open issues. For instance, traffic management policies appropriate for such trees still remain to be defined.

References

[1] The ATM Forum Technical Committee. Private Network-Network Interface Specification Version 1.0. March 1996.

[2] The ATM Forum Technical Committee. Private Network-Network Interface Specification Version 2.0. September 1997.

[3] R. Venkateswaran, C. S. Raghavendra, X. Chen, and V. Kumar. Hierarchical Multicast Routing in ATM Networks. *In IEEE Intl. Conf. on Communications*, volume 3, pages 1690-1694, June 1996.

[4] L.T. KOU, K. MAKKI. An even faster approximation algorithm for the steiner tree problem in graphs. *Congressus Numerantium*, 59 (1987), pp. 147-154.

[5] E. N. Gilbert and H. O. Pollak. Steiner minimal trees. *SIAM Journal on Applied Mathematics*, 16(1):1-29, January 1968.

[6] R. M. Karp. Reducibility among combinatorial problems. *Plennum Press*, New York, 1972.

[7] B. W. Waxman. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, 6(9): 1617-1622, December 1988.

[8] L. Kou, G. Markowsky, and L. Berman. A fast algorithm for Steiner trees. *Acta Informatica*, 15:141-145, 1981.

[9] M. Doar and I. Leslie. How bad is naive multicast routing. *IEEE INFOCOM'93*, pp.82-89, 1993.

[10] G. N. Rouskas and I. Baldine. Multicast routing with end-to-end delay and delay variation constraints. *IEEE Journal on Selected Areas in Communications*, 15(3):346-356, April 1997.

[11] T. Ballardie, B. Cain, and Z. Zhang. Core Based Trees (CBT version 3) Multicast Routing Protocol Specification.*Internet draft*, March 1998, Work in progress.

[12] C. Shields and J. J. Garcia-Luna-Aceves. The Ordered Core Based Tree Protocol. *Proceedings of the IEEE INFOCOM*, Kobe, Japan, April 1997.

[13] C. Shields. Oredered core based trees. *Master's thesis*. University of California, Santa Cruz, California, June 1996.

[14] D. Estrin, D. Farinacci ,A. Helmy,D. Thaler,S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, and L. Wei. Protocol Independent Multicast-Sparse Mode (PIM-SM) Protocol Specification.*Internet draft*, September 1997, Work in progress.

[15] M. Handley, J. Crowcroft, and I. Wakeman. Hierarchical protocol independent multicast (HPIM). *University College London*, November 1995.