COMPARATIVE ANALYSIS OF MULTICAST ROUTING ALGORITHMS FOR MULTIMEDIA COMMUNICATION OVER ATM NETWORKS

Frank Meylan(*), Luis Gustavo Gasparini Kiatake(*), Marcelo Zanoni Santos(*), Sergio Takeo Kofuji(*), Jean Pierre Courtiat(**)

(*) LSI-USP Laboratório de Sistemas Integráveis – Universidade de São Paulo
Av. Prof. Luciano Gualberto 158 tr.3 05508-900 São Paulo – SP – Brasil
Tel. +55 11 818-5589 Fax: +55 11 2114574
{meylan, kiatake, kofuji, zanoni} @ lsi.usp.br

(**) LAAS-CNRS 7, avenue du Colonel Roche 31077 Toulouse Cedex 04 - France
Tel : +33 5 61 33 62 44 Fax : +33 5 61 33 64 11
courtiat@laas.fr

Abstract

For a long time, the quality of the distributed multimedia applications was restricted by the limitations of the computer networking technologies. This scenery has changed a lot with the introduction of ATM (Asynchronous Transfer Mode). Characteristics like quality of service (QoS), support to different categories of traffic (voice, video and data), high scalability, among others, make ATM one of the most appropriate technologies for multiservice networks.

Another important characteristic is the support to multicast transmissions. In ATM, the use of multicast communication, as well as the resources' reservations are specified during the connection establishment phase. In this phase, connection establishment requests are routed among the switches in order to assure the necessary resources for the provision of the required communication services. In this case, the routing task involves aspects like quality of service (QoS), multicast communication, and resources reservation, becoming a complex task with specific characteristics.

This paper will analyze in details the multicast routing with QoS restriction, presenting a comparative study about the solutions that has been proposed to deal with this new problem. This study will cover theoretical analyses and simulations of four heuristics: KPP, CSPT, CCET and Hybrid. Parameters related to the order of complexity, cost of the multicast tree, delay, number of switches, execution time and maximum number of simultaneous multicast sessions will be evaluated, analyzed and compared.

Keywords: Multicast Routing, ATM Networks, QoS

1 Introduction

Recently, distributed multimedia applications like video on demand, distance learning, videoconference, cooperative work, among others have been developed and disseminated in an accentuated way. Such applications have some requirements that should be considered by the transmission network, in order to obtain a service with the expected quality.

The voice and video transmission in a videoconference, for example, request certain amount of available bandwidth witch is guaranteed during the existence of the session, besides delay controlled inside of established limits. These characteristics or demands imposed by the applications are called parameters of quality of service (QoS). According with those parameters and the available resources in the network, it is possible to evaluate if a service can be executed with the requested quality.

These applications involve, in many cases, several simultaneous participants in a same session, for example five researchers discussing through a videoconferencing. This demands an efficient mechanism of multicast communication from the network.

The ATM technology presents some characteristics that improve the support to the multimedia transmissions, as high transmission rates, differentiated categories of services and quality of service guarantees. This technology is connection oriented, and during the connection establishment the resources are allocated, from the origin, through all the intermediary equipments, until the addressee, making the continuity of the negotiated service guaranteed.

The routing in TCP/IP networks is a relatively simple task, once this technology doesn't allow the attribution of QoS and there aren't many applications that use the multicast concept. Therefore, routing algorithms are relatively simplified, but enough.

In ATM networks, routing is executed during the connection establishment, and the choice of the routes can take into account aspects related to quality of service, bandwidth requested and the maximum delay supported. Therefore, routing becomes a more complex and time expensive task, demanding efficient and elaborated algorithms.

Several proposals have been presented in order to solve the multicast routing problem when QoS parameters are involved. But, the works that compare these proposals according to a defined methodology are very scarce. Thus, this paper analyzes multicast routing with restriction of QoS, also known as restricted multicast routing, presenting a comparative study among the solutions that are being proposed to deal with this new problem. This study will be driven through theoretical analyses and simulations on the main heuristics described in the literature: KPP [Kompella93], CSPT [Sun95], CCET [Waters94] especially, a new heuristic of multicast routing proposed by Waters, Hybrid heuristic [Waters97], will be presented and compared with the others.

This heuristic has as innovative characteristic the fact of joining advantages of another heuristics optimized for specific cases of the multicast routing problem with QoS. However, this was not evaluated in the original work according to a methodology that facilitated its comparison with the heuristics existent. Therefore, this work presents as main contribution a detailed analysis of the Hybrid heuristic, as well as its comparison with the most important existent heuristics.

The paper is structured as follows: the section 2 presents the basic concepts about the representation of the multicast network model, section 3 describes the multicast routing problem with QoS restriction, section 4 describes the simulator and the analysis methodology, section 5 analyzes the results obtained through the simulations. In section 6 the conclusions are formulated and in the section 7, possible future works are proposed.

2 Basic Concepts

To make possible the study of the multicast communication in computer networks, it is necessary to define a representation language of the model in subject. This way, a computer network will be represented by a connected graph N=(V, E), where V represents a set of nodes and E represents a set of links among the nodes. The existence of a link e=(u,v) of the node u to the node v, also implies in the existence of a link e'=(v,u) for any u,v $\in$ V, that is to say, it is being assumed that the links are full-duplex.

Every link $e=(u,v) \in E$ has a cost function *C(e)* and a delay function *A(e)*. **The cost function C(e)** can be understood as monetary cost or cost in function of the existent traffic in a link. In this paper the cost function will be associated to the existent traffic in the link, once the objective is to study the efficient management of the network resources. This way, the more is the link utilization, the more is the link cost. In this work it is also assumed that the connections are asymmetric, that is to say, $C(e) \neq C(e')$, since this is a real fact in computer networks. The **delay function A(e)** shows the delay that the packages acquire when traversing a link *e*. This delay can be attributed at the time of transmission, due to the queuing, routing or commutation of the packages.

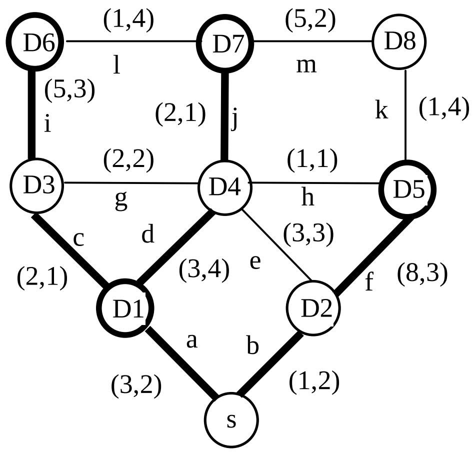

Figure 1 – Graph G, (cost,delay)

An alternating sequence of nodes and links is called **path**. So, a path $P(v_0, v_k) = v_0,e_1,v_1,e_2,\ldots,v_{k-1},e_k,v_k,$ such that every $e_i = (v_{i-1}, v_i) \in E$, $1 \leq \iota \leq k$. When a path contains repetition of nodes, it is said that it contains loops. If there is not such repetition, the path is loop free.

The **cost of a path** *P(v0, vk)* is defined as the sum of the costs of the links constituting the path:

$$Cost(P(v_0, v_k)) = \sum_{e \in P(v_0, v_k)} C(e)$$

According to this definition, the cost of P(s,D6), figure 1, can be calculated as C(P(s,V6)) = C(P(s,D1)) + C(P(D1,D3)) + C(P(D3,D6)) = 3 + 2 + 5 = 10.

Similarly, the **end-to-end delay of the path *$P(v_0, v_k)$*** is defined as the sum of the delays on the links constituting $P(v_0, v_k)$:

$$Delay(P(v_0, v_k)) = \sum_{e \in P(v_0, v_k)} A(e)$$

The delay of the path P(s,D6) can be calculated as: A(P(s,D6)) = A(P(s,D1)) + A(P(D1,D3)) + A(P(D3,D6)) = 2 + 1 + 3 = 6.

A ***multicast group*** $G = \{g_1, g_2, \ldots, g_n\} \subseteq V$, where $n = |G| \leq |V|$, is a set of nodes participating in the same network session. Every multicast group has a source node $s \in V$, witch may or may not participate in group G. Again in figure 1, the multicast group G is composed by G = {D1, D5, D6, D7}.

A ***multicast tree*** $T(s, G) \subseteq E$ is a set of links constituting loop free paths with a single source *s*, called root. The root is connected to all members of the multicast group *G*. In the example of figure 1, the tree witch connects all members of the group *G* to the source *s* is: T(s,G)={a,b,c,d,f,i,j}.

So, the **total cost of the multicast tree** can be calculated as the sum of the cost of all links in that tree:

$$Cost(T(s,G)) = \sum_{e \in T(s,G)} C(e)$$

In this way, the cost of the tree highlighted in figure 1 is: C(T(s,G))=C(a) + C(b) + C(c) + C(d) +C(f) + C(i) +C(j) = 3 + 1 +2 + 3 + 8 + 5 + 2 = 24.

And the **maximum end-to-end delay of a multicast tree** is the maximum delay from the source to any multicast group member:

$$Max\_Delay(T(s,G)) = \max_{g \in G} \left(\sum_{e \in P_T(s,g)} A(e) \right)$$

where $P_T(s, g)$ is the path from s to g along the tree $T(s, G)$. In other words, to know the maximum end-to-end delay of a multicast tree, it's necessary to calculate the delay of the paths that connects the source to every group member, selecting the maximum one. So Max_Delay(T(s,G)) = Max(A(P(s,D1)), A(P(s,D5)), A(P(s,D6)), A(P(s,D7))) = Max(2, 5, 6, 7) = 7.

3 Restricted Multicast Routing Heuristics

The first experience of multicast transmission in long distance networks happened in 1992, when IETF (Internet Engineering Task Force) transmitted an audiocast for twenty simultaneous participants in three continents [Deering92]. From this date, a lot of researches are being conducted in order to develop and/or improve multicast routing algorithms and protocols that take in consideration the following characteristics:

- Quality of Service parameters required by the real time multimedia applications;
- Efficient management of network resources;
- Scalability.

3.1 Constrained Multicast Routing Heuristics

Within the several heuristics existent, this work will restrict to the analysis of the restricted delay heuristics, more adapted for the real time multimedia applications. The restricted delay heuristics has as objective create a multicast tree connecting all the participants of the group to the source, minimizing the tree cost and respecting the maximum delay limit defined. Using a more formal language:

$$\begin{cases} Max\_Delay(T(s,G)) \leq \Delta \\ Cost(T(s,G)) = \text{minimum} \end{cases}$$

This is a NP-complete problem, so some heuristics were developed to solve it. Following, the heuristics KPP, CSPT, CCET and Hybrid will be presented.

3.1.1 KPP Heuristic

Kompella, Pasquale and Polyzos proposed in [Kompella93] a heuristic based in the Constrained Steiner Tree model, which will be called KPP in this paper.

The first step of the heuristic is to find for each pair of nodes belonging to the multicast group, the minimum cost path that respects the delay limit. This procedure is done by stepping through all the integer values of delay, starting from 1 until the maximum limit Δ. The reason for this is to increase the number of paths to be evaluated in function of its cost, consequently to increase the possibility to find the minimum cost path. In other words, it is found the closure of the graph formed by the participants of the multicast group.

Then, a constrained tree is found by the closure, where the root of this tree is the source. For that a greedy algorithm is used. In other words, the unnecessary links are successively removed from the graph based in their cost function. There are, also, alternative selection mechanisms, based on the cost function, on delay function, or on both. Then the links of the spanning tree are mapped on the original graph.

The computation of the constrained smallest path during the first phase of the heuristics is the one that consumes more time, tends a complexity of $O(\Delta n^3)$, where n is the number of nodes of the graph. The second pass of the heuristics, definition of the constrained tree, has complexity of $O(m^3)$, where m is the number of nodes of the multicast group. The last step, mapping of the solution on the original graph, has at the most a complexity of $O(n^2)$. Therefore the total complexity of the heuristics is given by $O(\Delta n^3)$.

The evolution of the KPP solution can be followed through the figure 2. The multicast communication has as origin the source F and the members of the multicast group are identified in the illustration with the labels B and D. The maximum delay established is 12.

The closure of the graph formed by the participants of the multicast group, first step of the heuristic, can be identified through the arrows stippled in the item B of figure 2. Through that closure, the algorithm will select the solution. In the example above, the solution will be formed by the links (F,B) and (F,D) with a cost of 22 and a maximum delay of 11, item C. This is a valid solution, since it satisfies the restriction of delay. The final step of KPP consists in mapping the solution of the closure in the original graph, resulting in the links (F,A), (A,B), and (F,C), (C,D), forming the constrained multicast tree presented in the item D of figure 2.

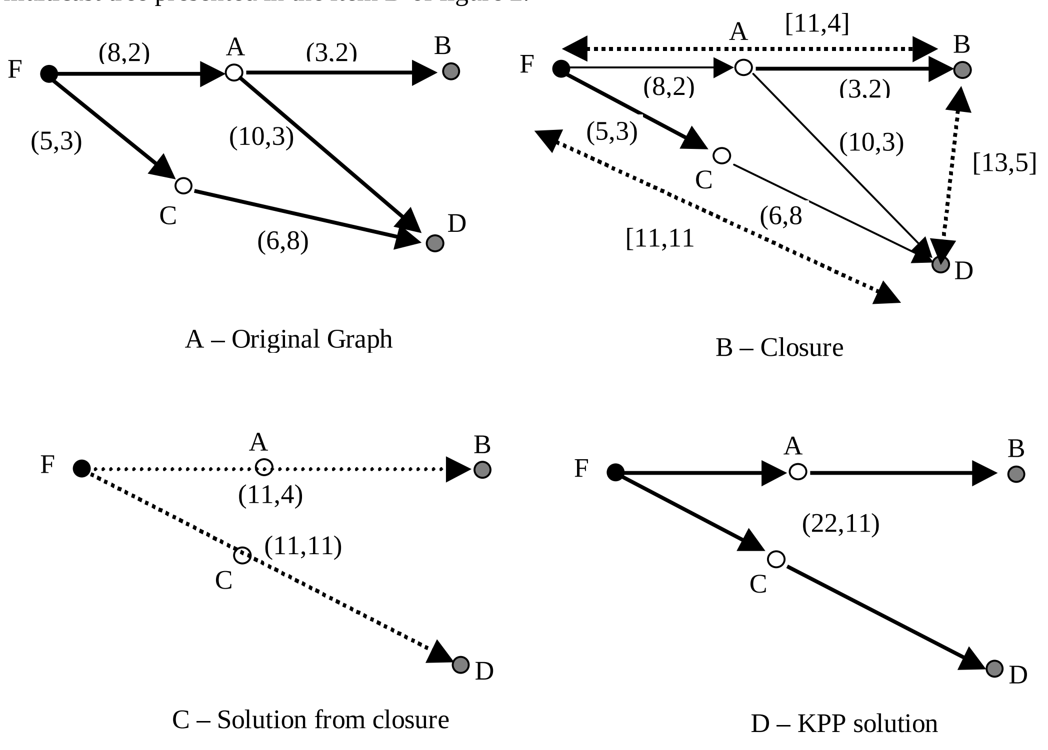

Figure 2 – KPP heuristic

3.1.2 CSPT heuristic

Quan Sun and Horst Langendörfer published in [Sum95] the CSPT (Constrained Shortest Path Trees) heuristic. It has three basic stages: First, a low cost scatter tree is formed trough Djikstra's algorithm. This tree is composed by the maximum number of components that Figure 2 – KPP heuristics belong to the multicast group, always respecting the maximum delay limit. The tree generated by this first execution usually does not cover all the multicast group members. So, a second execution of Djikstra's algorithm is performed, in order to produce a tree composed by lower delay paths, covering that elements that were not covered in previous step. The last stage consist in the union of these two trees, attempting to remove some loops and confirming that no path exceeds the maximum delay limit.

The first and second stages are essentially based in some adaptations of Djikstra's algorithm. So, the complexity of these stages is order $O(n^2)$. The last stage is $O(n)$. Thus, the this heuristic has a total complexity of order $O(n^2)$.

3.1.3 CCET heuristic

The CCET (Constrained Cheapest Edge Tree) heuristic was first described in [Waters94]. Since then some improvements were done. The original idea bind the delay restriction with the minimum broadcast delay or minimum multicast group delay. In this work, the delay restriction can have an arbitrary value.

First this heuristic calculates, trough Djikstra's algorithm, each node-source delay, generating a directed graph. During this stage, the edges that do not respect the maximum delay limit are removed from the tree (figure 3-2). Then, the nodes are arranged in the inverse order relating to their delays. Starting from the highest delay node, the heuristic searches for the lower cost link, generating a path from the underlying node to the source. As the networks are asymmetric, the link cost from node to source and from source to node can be different and the heuristic calculates the cost from node to source. After all node calculation, these paths are put in a broadcast tree (figure 3-3). In the end of this process, those nodes that do not belong to the multicast group are removed from the tree, resulting in the multicast tree (figure3-4).

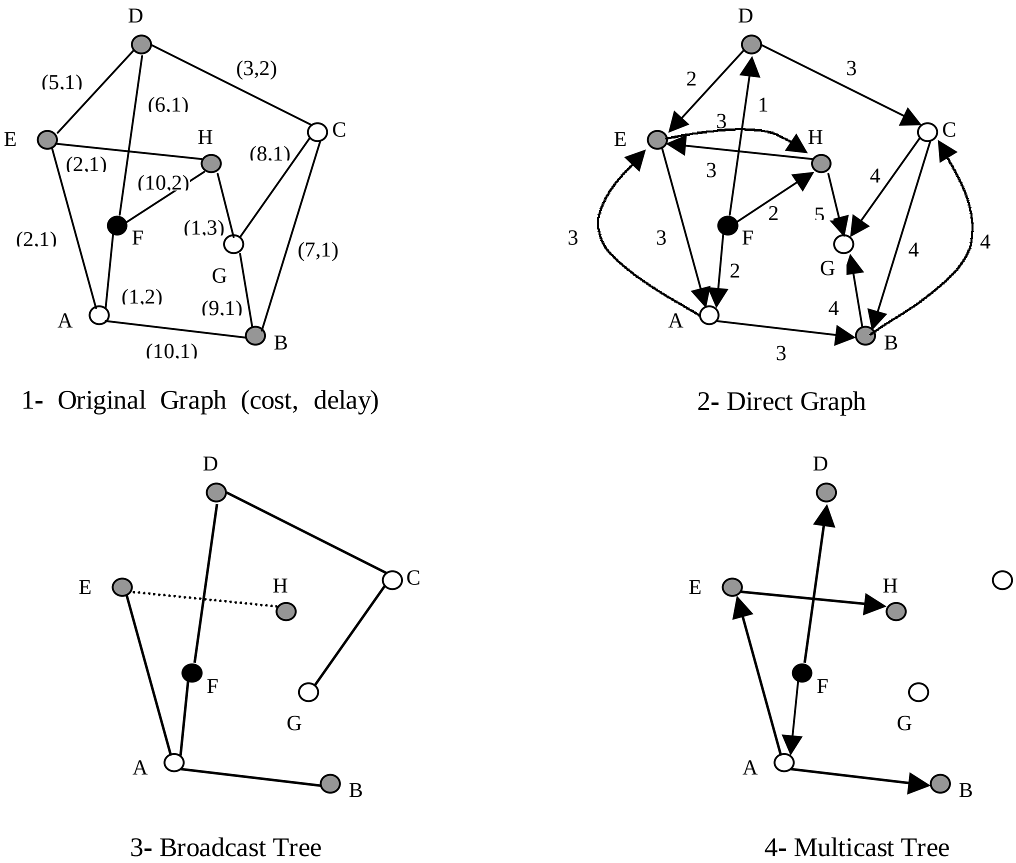

Figure 3 –CCET heuristic stages

About this heuristic complexity, one can see that the first stage uses the Djikstra's algorithm, and so it has an complexity order of $O(n^2)$. The second stage, that generates the broadcast tree, performs, for each node, an deep search in the tree until the source, resulting in an complexity order of $O(\max(N, |E|))$, where N is the node quantity and E is the edge set of the search tree, formed by all the possible paths from node to source [Gibbons89]. So, the total complexity order of this algorithm is $O(n^2)$.

3.1.4 Hybrid heuristic

John Crawford and Gill Waters in [Waters97] proposed the hybrid heuristic. This heuristic is formed basically by the integration of Dijkstra's algorithm and the CCET and CSPT heuristics. This integration was proposed because of the good results obtained by these heuristics for specific cases of the restricted multicast routing problem.

The calculation of the multicast tree is accomplished through the application of Dijkstra's algorithm creating the trees of smaller delay and smaller cost simultaneously. Then the CCET and CSPT selection methods are applied in order to generate the respective multicast trees. At this point, there are three possible trees to be considered:

i -) tree generated by Dijkstra's algorithm;

ii -) tree generated by CCET's selection mechanism;

iii -) tree generated by CSPT's selection mechanism.

Therefore the analysis comes in the following way:

- If some tree contains a path that exceed the maximum delay limit, it is automatically discarded of the analysis process;
- The total cost of the trees is calculated and that one which has the smallest cost will be selected.
- The complexity order of this heuristic is dominated by the CCET, so it results in $O(n^2)$.

4 Simulator and the Methodology of Analysis

Among several parameters involved in such simulation, some of them, as the complexity order, the amount of stored state information, etc., should be analyzed through analytic models. However, others of fundamental importance should be considered, especially in real network environments. Among these we can mention: time of execution, number of flaws, scalability and quality of the produced results. In order to reproduce the real environment of multimedia transmission flows in ATM networks, some procedures and definitions were created and implemented generating a multicast transmission simulator of ATM networks.

4.1 The simulator

The MCRSIM simulator was originally developed by Hussein Salama for his PhD Tesis [Salama96] presented at the North Caroline State University (USA). This simulator was properly modified and enhanced in order to support the Hybrid heuristic simulation beyond the others already implemented.

Based in the ATM network model, the simulator considers that all connections are full duplex with 155 Mbps bandwidth (OC-3). The network nodes are generated in a random way within a 4000x2400 km^2 area rectangle, that is a rough relation with the United States area.

The random node generator has an important play in simulations. This work uses a network links and nodes generator based in [Waxman88]. It was optimized in order to impose that all network nodes have at least order 2, i.e., each node must be bind to at least more two nodes. Some generator parameters were set in order to let the probability of the existence of a short link be higher than a long link, as occurs in real networks, and also to let the average nodes order equal to 4, as occurs in the Internet nowadays.

Each network node represents a non-blocking ATM switch. The link propagation speed was estimated in two thirds of the light speed. Cells buffering delays were not taken into account, and so the link delays are symmetrical, i.e. A (u, v) = A (v, u).

In order to simulate multimedia traffic, the multicast connections were set to use the VBR (variable bit rate) quality of service (QoS) category, assuming the transport of high quality MPEG-2 video flows. This QoS Category allows the reservation of certain amount of bandwidth for each video flow.

The cost function of a link is related with its utilization, i.e., the higher a link utilization the higher its cost. So this cost can vary as sessions that use that link are set or released. A link can accept new sessions until that the sum of the bandwidth reservation reaches 85 % of its capacity. In this condition, the link becomes saturated and new sessions are rejected. This admission control policy, in spite of simple, allows statistical multiplex and an efficient use of network resources, being enough for this work's studies.

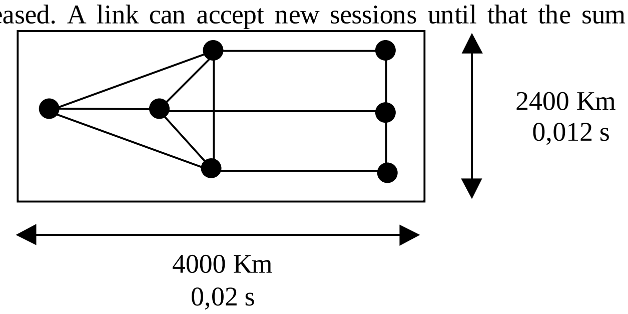

Figure 4 – Network dimensions and delays

The video and voice sessions have strict requirements about the maximum delay supported. According to the considerations done about the network dimension and the propagation delay, it is possible to estimate the east-west transmission delay as 0.02 seconds, and the north-south as 0.012 seconds (Fig. 4).

Thus, the maximum delay sender-receiver, called Δ, was set to 0.03 seconds, imposing a superior limit of the end-to-end network transmission. This relatively low value was used in order to let some time for the superior protocol process the information and do not affect the underlying service. Moreover this value is useful to identify more efficient algorithms.

In order to reproduce real environments, the simulator generates random background traffic. It does not have a defined shape, but it establishes a minimum (Bmin) and maximum (Bmax) bandwidth. Thus, when a new connection is established the background traffic should also be considered by the traffic control mechanisms, besides those traffics already generated by existent connections.

4.2 Simulator adaptations and improvements

In order to make possible the comparison of the new heuristic presented in the previous section with those already implemented in the simulator, some adaptations and improvements in the original code were necessary.

The first step consisted in adapt the simulator code, written originally for Sun, DEC and RS6000 platforms, for Linux/Intel platform, especially because it is an open platform, presents great accessibility and is very spread in the academic world.

Then, it was necessary to write the code for the new heuristic, proposed by Waters, taking into account the preservation of the existing data structure already coded for the others heuristics. This concern attempts for the maintenance of data representation coherency, allowing a fair comparison among all heuristics.

Besides these modifications, some modules for specific simulation tests were developed. For example, it was developed the continuous algorithm simulator, which performs, for all random networks, simulations for all heuristics in the same configuration of background traffic, multicast group and transmission source. The results are stored in output files, according with the analyzed parameter.

4.3 Considerable parameters involved in multicast heuristic comparison

To perform the comparison of different heuristics, it is fundamental to define a comparison criterion. This definition requires the identification of the most considerable parameters and the adoption of a common metric to be applied over all heuristics [Tobagi94]. In this manner, the most significant topics to be considered in a heuristic analysis are:

- Efficient management of network resources;
- QoS parameters involved;
- Algorithm complexity;
- Amount of data necessary;
- Use for symmetric and asymmetric networks.

As the utilization of real networks for the development and improvement of new heuristics is not always possible, most of researchers prefer the use of simulations. As intention of these simulations is to represent the real conditions of computer networks, they need to execute the algorithm in several different topologies and network sizes, using real scenarios or some generated in a random way.

Congestion, delays, traffics and all other conditions should be as closer as possible to the reality. If a simulation does not follow these rules, its results may express results that would not occur in the real network.

The literature shows that simulation results of routing algorithms can presents very different values for different network sizes. So, it is common to find algorithm behavior analysis in two distinct cases: small networks, up to 20 nodes, and large networks, up to 200 nodes.

For each case the multicast group size influence over the underlying algorithm performance is studied. Two configurations were specified for the background traffic: the first vary the allocated background bandwidth between 45 Mbps (Bmin) and 85 Mbps (Bmax); the other allows a larger variation than the former, with the limits between 5 Mbps (Bmin) and 125 Mbps (Bmax). As the full-duplex link cost is directly proportional to its utilization, a larger variation in the background traffic can lead to a higher cost variation. This fact represents an higher link asymmetry, and in the simulations, high asymmetry networks means that their links have a background traffic with values between 5 Mbps and 125 Mbps.

Each simulator execution analyses the algorithm taking into account the quality of the resulting multicast tree (that means its total cost), average delay, average switches quantity in each path, execution time and failure occurrences quantity. The failure can be caused by the existence of a path that do not respect the maximum delay limit between source and receiver, or by the connection rejection establishment due to the lack of bandwidth for the video flow transmission.

Aiming to reach an expected, each configuration is executed 300 times, i.e. 300 random networks are generated, each one with a different background traffic and multicast group. Previous studies [Salama96] demonstrated that, in such kind of simulations, from 250 to 300 execution leads to a confidence value of 95% with a maximum variation of 5% from previous results.

5 Results and Analysis

The following sections present the results and analysis obtained from simulations of the restricted multicast routing heuristics. The analyzed parameters were: complexity study, total cost, delays, switch quantity involved, execution time and maximum quantity of established sessions.

5.1 Algorithm complexity study

The algorithm complexity study aims to evaluate the scalability and to esteem the execution time taking into account the network nodes quantity. So, it is possible to foresee if a certain algorithm is suitable for real implementation in protocols and equipment. As this work focus is the multicast routing for multimedia applications, which requires restrict and controlled end-to-end delays, the algorithm execution time is essential to an efficient performance of routes calculation. The following table summarizes the results for each algorithm.

Algorithms	Complexity Order
KPP	$O(n^3)$
CSPT	$O(n^2)$
CCET	$O(n^2)$
Hybrid	$O(n^2)$

Figure 5 – Algorithm complexity order

Only based in the theoretical study presented in Figure 5, and without any practical results, it is expected that the KPP algorithm will have a higher execution time than the others algorithms. Moreover, this time will arise faster than the others as the node quantity increases. These facts can affect the KPP's scalability and make difficult its utilization in networks that have many nodes.

5.2 Total cost of the generated tree

Total cost of the multicast tree built by the algorithm demonstrates its ability to allocate the switches' resources in an intelligent way and to transmit over low utilization connections.

Small networks (up to 20 nodes) Big Networks (up to 200 nodes)

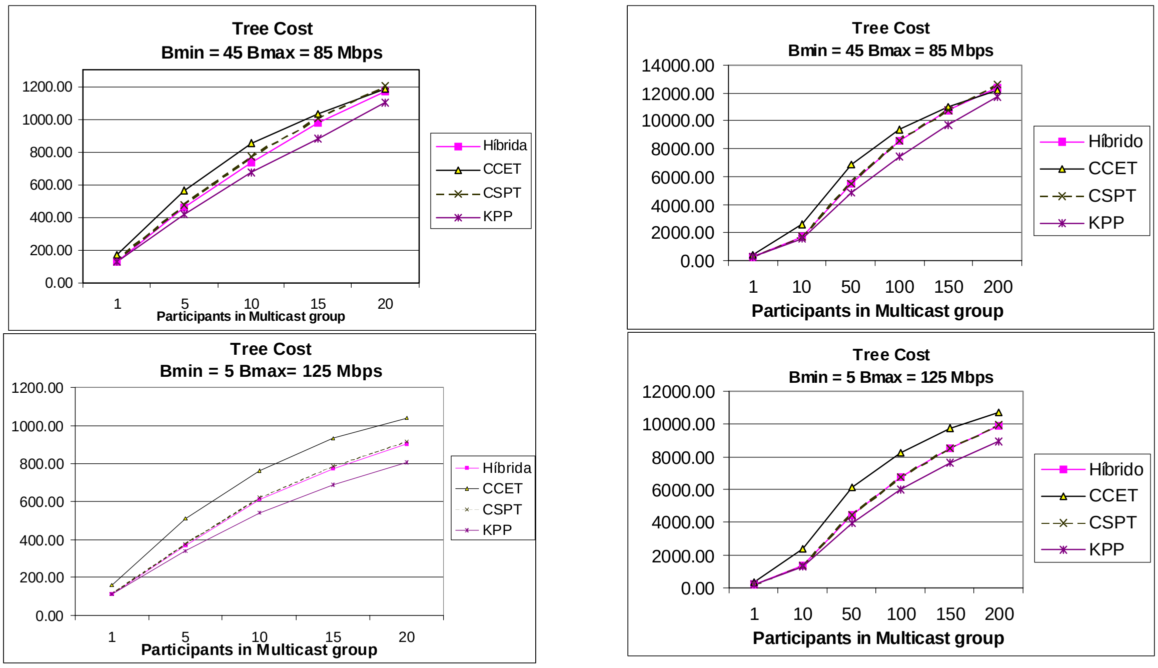

Figure 6– Cost of the trees generated by the algorithms

Figure 6 graphs shows that KPP heuristic generates the trees with smaller total cost for all tested cases, what was already expected, because minimize the resources utilization is the goal of the technique based on Steiner tree. Among the shortest path heuristics (Hybrid, CCET and CSPT), Hybrid presented the best results for small networks, up to 5% better than CSPT and 20% better than CCET. For big networks, Hybrid and CSPT showed improved performance, being up to 28% better than CCET.

An interesting behavior to be noted is the algorithms' performance difference, considering the largest asymmetry of the connections, that can be observed in the graphs with Bmin=5 Mbps and Bmax=125 Mbps. In this case, for small networks, the difference between Hybrid and CSPT decreases plenty, while the difference of these with CCET increases.

5.2.1 Multicast Tree Delay

The multicast tree delay is an important comparison parameter among the algorithms, because it permits to evaluate the algorithm capacity to use connections that impose the smallest delay in video flows transmission. In spite of all analyzed algorithms restrict the maximum delay, it is interesting that this parameter be the smallest possible value. This is due to the fact that for multimedia traffic, besides the temporal restriction, many processing capabilities are required for receiving the flows, packet processing, and presentation for the users.

Following, results of the trees' average delay are presented, when the participant number of the multicast group is varied.

Small networks (up to 20 nodes) Big Networks (up to 200 nodes)

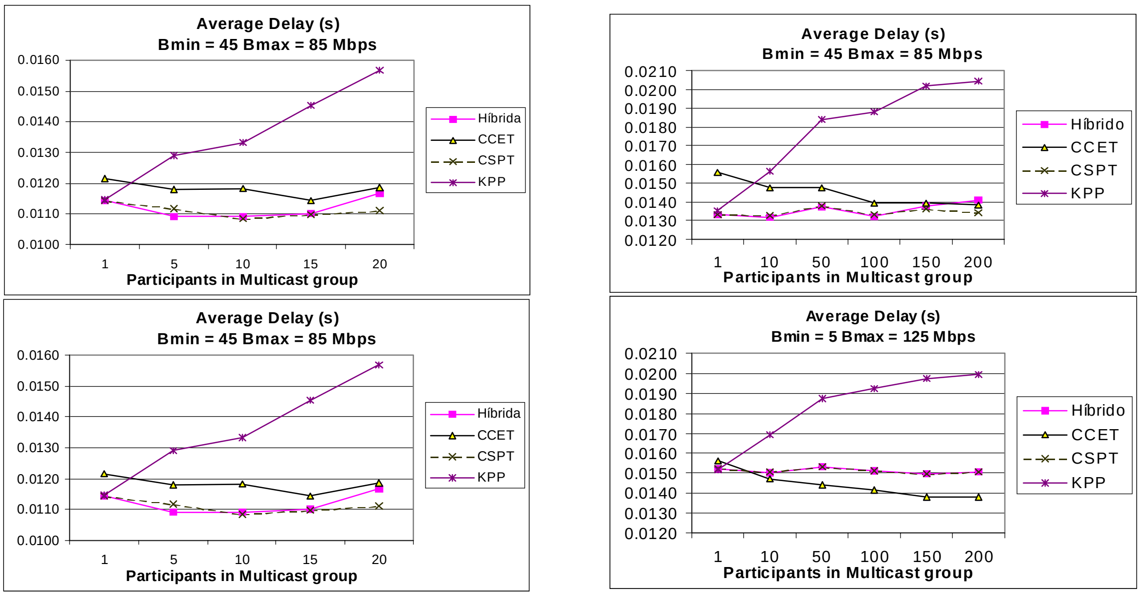

Figure 7 – Average delay of the generated trees

Observing the graphs in figure 7, one can note that the KPP heuristic demonstrated the most inefficient, considering the average delay of the generated trees for small and big networks. This is due to the fact that KPP heuristic aims to optimize just the cost of the tree, not considering other parameters.

Among the smaller path heuristics, it can be said that for small and big networks, Hybrid and CSPT heuristics presents much better results than CCET, considering low asymmetry. When high asymmetry is considered, i.e. when the difference between the cost of each connection direction is great, the behavior is practically the opposite of the first one. It would be necessary a deeper study, considering also implementation aspects to explain the heuristics behavior in relation to the larger or smaller symmetry of the networks connections.

5.3 Number of switches in each path

The average number of switches presents in the path that form the multicast tree is an important parameter to determine the network resources consumption for each algorithm, assuming that a switch is also a network resource. Although this paper don't evaluate the delay imposed by the commutation in each equipment, this work can be the starting point for a more detailed study, considering all delays involved in the voice and video transmissions.

Small networks (up to 20 nodes) Big networks (up to 200 nodes)

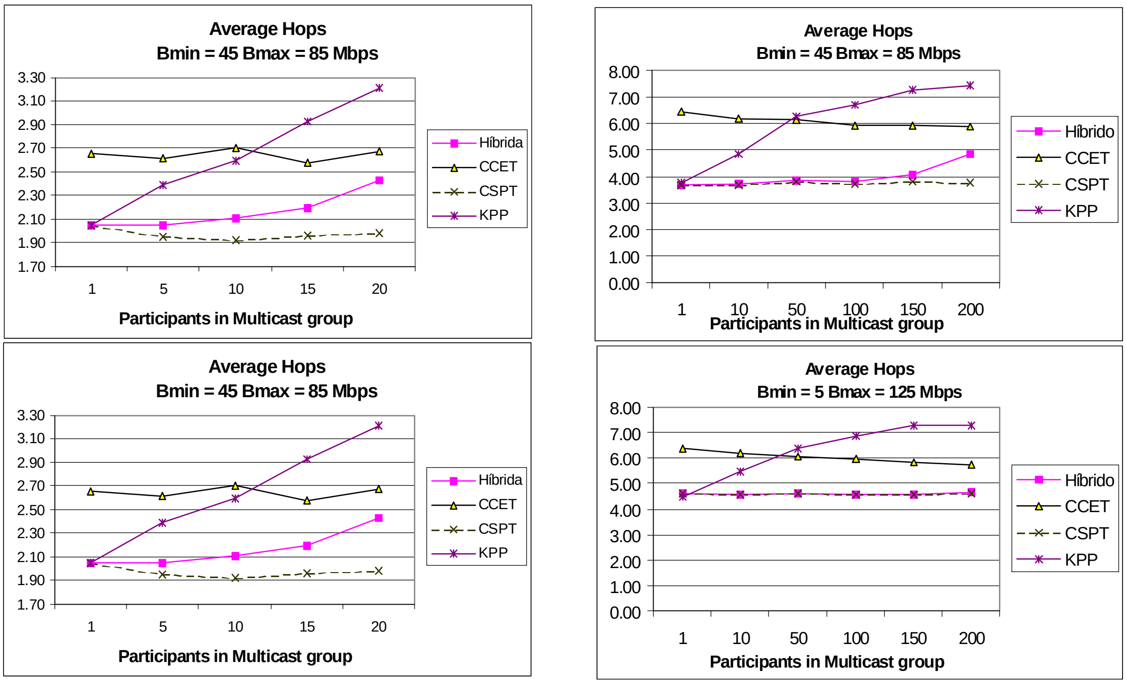

Figure 8 – Average number of switches on multicast tree path

When the parameter compared is the average number of switches allocated for a multicast tree path, figure 8 shows that CSPT heuristic presented the best performance, i. e. it allocated the smallest number of switches. Hybrid was up to 19% worse than CSPT in small networks and practically equal in big networks with great asymmetry. CCET obtained the worst result among the one of shortest path, about 42% larger in comparison to CSPT in big networks with little asymmetry. KPP, that is the only tested heuristic that implements the Steiner tree technique, showed growing results in function of the participants number the multicast group, and in general larger than the one present in the shortest path algorithms.

5.4 Algorithms' Execution Time

The analysis of algorithms' execution time is essential to compare their results to the cost (processing time) necessary for its calculation. Furthermore, this study is the first step toward a viability study for the implementation of the algorithms in protocols or specific equipment, as in LANS or WANs. Note that neither Hybrid heuristic nor the others already presented in the simulator were implemented taking into account the execution time optimization.

The results from figure 9 were achieved by executing the simulator in a Pentium II running Linux. Even so, the most interesting aspect is the difference between the values and not the values themselves.

We can see, in all analyzed cases, that KPP heuristic shows an execution time much greater than the shortest path algorithms. This fact, already described in the literature, confirm the thesis that algorithms that implement minimum Steiner tree technique are not suitable for big networks, even that they get better results.

Among the smaller path algorithms, hybrid heuristic presented the greatest execution time, almost the sum of the execution time of the other two heuristics. This fact can be explained by its operation mechanism that after creating a broadcast tree, uses two selection mechanisms, one used in CCET and the other used in CSPT, to create two multicast trees. Then, the costs of both trees are compared and the one that has the lowest cost will be chosen. Any way, the Hybrid cost is still much smaller than KPP, making possible its utilization for small and global networks.

Small networks (up to 20 nodes) Big networks (up to 200 nodes)

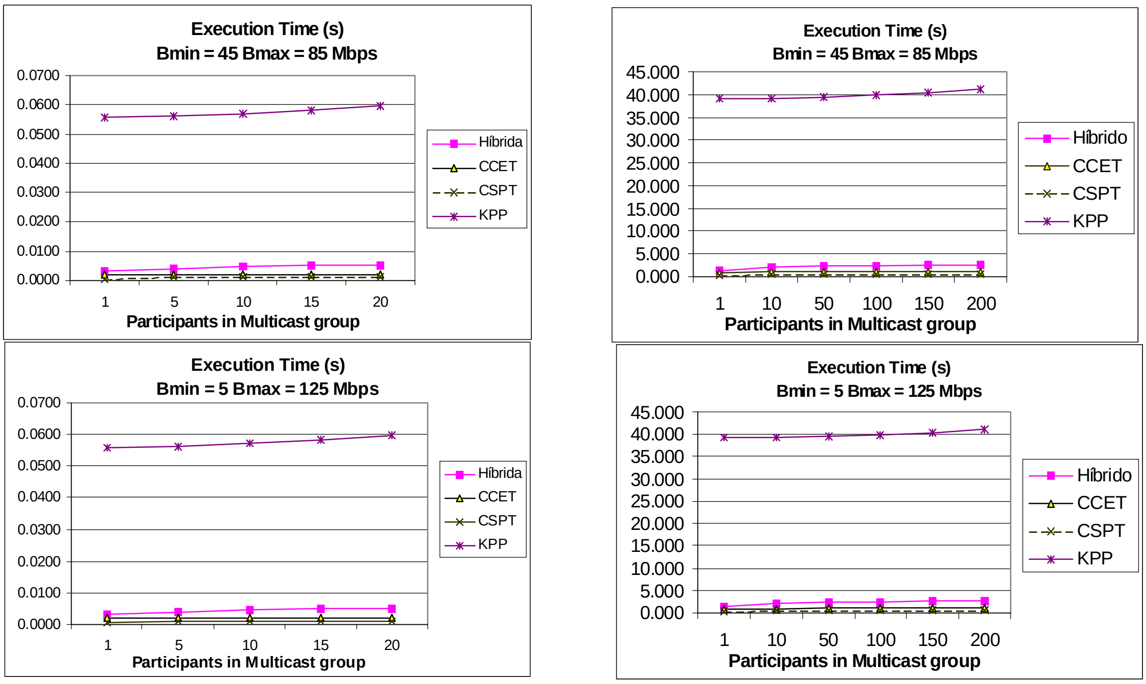

Figure 9 – Algorithms execution time

5.5 Number of sessions established with success

The tests of maximum number of simultaneous video sessions were performed by aiming to evaluate the algorithm efficiency to allocate switches resources and its ability to deal with connections saturation.

The simulation consists of generating a random network, without background traffic. Over this network will be created, in a random way, multicast groups with a video VBR flow source, generating a multicast session. Note that the multicast session is only established if there are available resources in all the intermediary switches belonging to the multicast tree. The sessions are created continually until the algorithms fail in finding routes with enough bandwidth for video traffic or routes that don't respect the maximum specified delay.

For each multicast group size, three hundred different networks are generated and for each of them one hundred of multicast session establishment trials are performed. The results described in figure 10 represent are the average of the values obtained in the simulations.

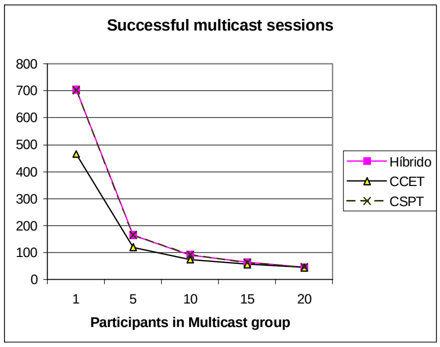

Figure 10 – Number of multicast sessions established with success

The KPP simulations could not be performed due to its high execution time. The results using KPP and CSPT heuristics were closer and the main fail reason was due to the connection saturation. In CCET tests, it was noticed a number of simultaneous session establishments up to 39% smaller. This is due to the great number of fails caused by the maximum delay violation (0,03 seconds), besides its smaller ability in selecting less congested connections. Observing figure 10, one can see that when the participants number of multicast group grows, the number of sessions that the algorithms establish before the network saturation decreases. This can be explained due to the fact of a larger multicast group demand larger number of connections to form the multicast tree. Therefore, the network resources are consumed quickly in big multicast groups.

6 Conclusions

This article analyzed the main multicast routing algorithms, accomplishing analytic studies and simulations, aiming to explore the main restrictions of the multimedia communication, such as maximum delay of the multicast tree, number of switches, maximum number of multimedia. Other parameters were also evaluated, such as complexity, total cost of the tree and execution time.

Observing the results one can see that KPP heuristic presented the best results of total cost of multicast tree, but it also presented a high execution time, what makes very difficult its utilization in networks with many nodes. Hybrid heuristic didn't present results as good as KPP, even so its relation cost/execution time make it has more conditions of generating protocols or equipment of multicast routing with QoS.

The average delay of the multicast tree was the only parameter where the CCET heuristic obtained some advantage compared to the other heuristics. Hybrid and CSPT obtained very close results, while KPP introduced the largest delay. CSPT heuristic got the smallest number of intermediary switches in the multicast tree paths, proceeded closely by Hybrid, standing out the ability of these heuristic to deal with the network resource consumption problem. These same results were verified in the maximum number of simultaneous multicast sessions' tests. This test could not be accomplished for KPP heuristic due to its high execution time.

Therefore, we can conclude that Hybrid heuristic demonstrated better or the same results of the other heuristics in the majority of appraised parameters, but maintaining a execution time that doesn't make unfeasible its utilization, being the best solution for the multicast routing for multimedia communication in ATM networks.

7 Future works

Since ATM Forum doesn't have any standardization for multicast routing with QoS restriction, there is so much research to be performed in that area. Therefore, the study of new algorithms or the integration of already existent approach, like we did in this paper, will be very useful, in a close future, in the protocol specification, or switches with multicast routing capabilities with QoS restrictions.

The algorithms analyzed in this article didn't take into account the inclusion or exclusion of elements in the multicast group, what may generate routes modifications. This point may be a possible extension of this work.

It was also assumed during the simulations that the video transmissions used the service category VBR. A comparative analysis of bandwidth allocation with CBR or ABR could originate results about the behavior of the algorithms when different resources reservation policies are used.

Finally, it was seen that the algorithms that implement the minimum Steiner tree technique has as the great disadvantage, its execution time. Therefore, these algorithms could be modified in order to execute in a distributed or parallel environment.

8 Bibliography

[Crawford94] J. S. Crawford, “Multicast Routing: Evaluation of a New Heuristic”, master thesis dissertation, University of Kent, England, 1994

[Deering92] S. Casner and S. Deering, “First IETF Internet AudioCast”, in Proceedings of ACM SIGCOMM, Computer Communications Review, vol. 22, no.3, July of 1992

[Gibbons89] Alan Gibbons, “Algorithmic Graph Theory”, Cambridge University Press, 1989

[Kompella93] V. Kompella, J. Pasquale and G. Polyzos, “Multicast Routing for Multimedia Communications”, IEEE/ACM Transactions on Networking, 1(3):286-292, 1993

[Salama94] H. F. Salama, D. Reeves, Y. Viniotis and T. L. Sheu, “Comparison of Multicast Routing Algoritms for High Speed Networks”, Tech. Rep. TR 29.1930, IBM, September of 1994

[Salama96] H. F. Salama, “Multicast Routing for Real-Time Communication on High-Speed Networks ", Ph.D. dissertation, Department of Electrical and Computer Engineering, N. C. State University, November of 1996

[Sun95] Q. Sun and H. Langendoerfer, “Eficient Multicast Routing goes Delay Sensitive Applications”, in Proceedings of the Second Workshop on Protocols goes Multimedia Systems (PROMS ’95), pp. 452-458, October of 1995

[Tobagi94] C. Noronha and F. Tobagi, “Evaluation of Multicast Routing Algorithms goes Multimedia Streams”, in Proceedings of the IEEE International Telecommunications Symposium, August of 1994

[Waters94] G. Waters, “A New Heuristic for ATM Multicast Routing”, in 2nd IFIP on performance Modeling and Evaluation of ATM Networks, pages 8/1-8/9, July 1994

[Waters97] G. Waters and J. Crawford, “Hybrid Approach to Quality of Service Multicast Routing”, Fifth IFIP Workshop on Performance Modeling and Evaluation of ATM Networks, 1997

[Waxman88] B. Waxman, “Routing of Multipoint Connections”, IEEE Journal on Selected Areas in Communications, vol 6, no. 9, pp. 1617-1622, December of 1988