A New Path Probing Strategy for Inter-domain Multicast Routing

António Costa*, Maria João Nicolau†, Alexandre Santos* and Vasco Freitas*
*Departamento de Informática,
Universidade do Minho, Campus de Gualtar,
4710 Braga, Portugal
Email: {costa,alex,vf}@uminho.pt
†Departamento de Sistemas de Informação,
Universidade do Minho, Campus de Azurém,
4800 Guimarães, Portugal
Email: joao@uminho.pt

***Abstract*— Many already in use applications require the provision of QoS services from the underlying network infra-structure. This is particularly true for multicast, since it involves many participants at very sparse locations usually aiming to receive or send multimedia real-time streams. One way to provide QoS is through routing, since QoS aware multicast routing protocols can find feasible multicast trees.**

At large scale, scalability issues make the QoS multicast routing task a lot more difficult, since it is not possible to have a complete knowledge of the network topology and its path QoS metrics in a clear up-to-date way. The most promising proposals are therefore based on path probing strategies that evaluate a subset of the available connecting paths.

Assuming that each member can express its requirements as a combination of QoS metrics like available bandwidth, end-to-end loss probability and delay, it is up to QoS routing multicast protocols to build distribution trees connecting members through paths that can satisfy those requirements.

In this paper an inter-domain QoS multicast routing protocol is presented, specifically designed for the hierarchical inter-domain scenario, where requirements like intra-domain independency and policy awareness should be met. Emphasis is given to the path probing mechanism used to connect new members to the multicast tree, stressing how it differs from others.

Simulation results show that despite using a less aggressive and simplified probing mechanism - more suitable for inter-domain scenarios - the proposed routing strategy can build multicast trees with metrics similar to those build by more aggressive technics, with considerable less effort.

I. Introduction

To support group applications, such as video-conference, distance learning and cooperative work applications, multicast support is needed from the underlying network. Multicast routing protocols build a distribution tree (or a set of trees) to deliver data packets from sources to all receivers. When the network supports Quality of Service (QoS), applications specify their QoS requirements as a function of QoS metrics like delay, bandwidth or loss probability. In this case, multicast routing protocols should be modified in order to find feasible distribution trees. Such trees are distribution multicast trees which have enough resources to satisfy the QoS requirements specified by the applications.

The problem of finding feasible trees is very difficult to treat due to different reasons. First the dynamism of metrics like bandwidth, delay or loss probability is extremely high, which turns almost impossible to maintain the accuracy of the network state information. Second, the communication overhead of such metrics as well as the large amount of state information that is necessary to maintain turn the use of link state protocols too expensive.

Another approach is a method called ”path probing” that do not require the maintenance of global sate information in the nodes. Beyond the path probing strategy the path searching process is initiated by the new receiver which explores different alternative paths and evaluates them in terms of how well do they fulfill its requirements. We believe that this type of strategy is better suitable for large scale networks in particular to implement inter-domain QoS multicast routing.

As well as for unicast routing, the multicast routing can be treated at two different levels: intra-domain or inter-domain level. Intra-domain multicast routing protocols work within administrative domains (also called autonomous systems), and inter-domain routing protocols work between the domains. The nature of these two protocols types are different and a good intra-domain routing protocol would not necessarily be a good inter-domain routing protocol. There are a set of specific issues that must be taken into account when considering inter-domain routing, besides scalability.

An important issue related with inter-domain routing is the fact that the topology must be considered as an asymmetric one. This is because inter-domain routing policies might be such that two distant domains might not agree on the same transit domain. Most multicast routing protocols use RPF (Reverse Path Forwarding) concept in constructing multicast trees, including the multicast routing protocols based on the path probing model. This concept is based on the idea that an actual delivery path to a node is the reverse of the path from this node to the source. This concept fits well in symmetric environments, but in a routing environment where routing policies are applied, the guarantee that symmetrical path will exist between two network addresses is broken. Therefore reverse-path routing may not be used.

Another inter-domain multicast routing key requirement is to keep, as far as possible, the independence between interior and exterior multicast protocols, in a way similar to unicast.

In this paper we propose a QoS aware multicast routing strategy which tries to address some of the problems faced by inter-domain multicast routing. Our strategy is based on the path probing model however it was improved in order to address important inter-domain requirements, like policy awareness, intra-domain independence and network asymmetries. In addition, heterogeneous receivers are supported, new group members may have different QoS requirements and may express them in terms of multiple parameters, like bandwidth, delay or loss probability.

II. Related Work

There are few proposals for inter-domain constrained multicast routing, most of them are based on the path probing strategy.

YAM[1] and QoSMIC[2] aim to discover multiple paths from an existing tree onto a joining node and afterward make a choice based on certain criteria.

YAM builds shared trees that have the capability to provide multiple routes to connect a new node onto an existing tree. It handles dynamic membership and does not require any global network state at routers, but it has excessive communication overhead because it relies on flooding to find a feasible tree branch to connect a new member.

QoSMIC alleviates the flooding behavior, but introduces a new complex element: the Manager Router. QoSMIC uses two different procedures to find a feasible tree: a local search and a multicast tree search. Local search is initiated by the new member router by flooding BID-REQ messages to its neighborhood, with scope controlled by TTL (Time To Live). Any in-tree router that receives a BID-REQ message becomes a candidate router and replies with a BID message, which is unicast to the new router. The BID message collects information about the path on its way that can be used for selection purposes. The multicast tree search occurs at the same time, initiated by a Manager after receiving a M-JOIN request from the new receiver. The Manager sends a BID-ORDER message to a set of in-tree routers, that become candidate routers and reply with BID messages exactly as described for local search procedure.

In QRMP (QoS-Aware Multicast Routing Protocol) [3] two search modes are defined: single path mode and multiple path mode. The routing process starts with the single path mode, attempting to search only the unicast routing path traveled by the join request through the multicast tree. The *join request message* carries the QoS requirements. As it travels, it checks the resource availability of every intermediate node and proceeds only when the node has the required resources. If an intermediate node does not have the required resources it triggers the multiple path mode by sending a *not acknowledge message* to the previous node. Upon receipt the *not acknowledge message* the previous node sends the *join request message* to all neighbor nodes except those from which the join request and not acknowledge messages were previously received. Once a feasible branch is detected an *acknowledge message* is sent back along the branch that triggers to multiple path mode. If more then one acknowledge message arrives at this node, the node will select the best branch and reject all the others. In QRMP, tree construction occurs from new receiver in direction to tree instead of from first in-tree node found backwards to receiver, therefore it does seem adequate to asymmetric topologies. In multiple path mode the join request messages are flooded. Besides, it does not support the establishment of multicast routing

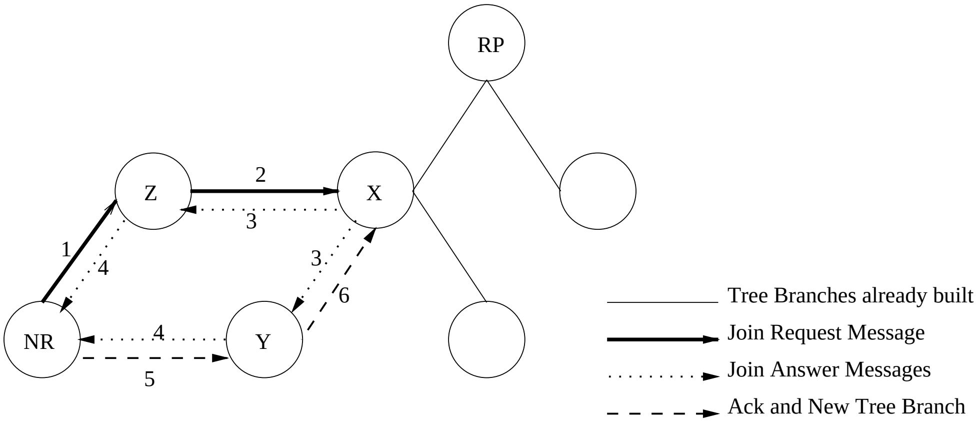


Fig. 1. Inter-domain tree construction. Actions are numbered in the order they occur

policies.

III. Policy Aware QoS Inter-Domain Multicast Routing Protocol

In this section our multicast routing strategy, called PAQoSIDMR (Policy Aware QoS Inter-Domain Multicast Routing Protocol) is described. It is based on shared unidirectional multicast distribution trees, one per multicast group. We called the tree root Rendez-Vous Point (RP), as in PIM-SM proposal. The tree construction is based on the path probing model (as YAM, QoSMIC and QRMP), but, as mentioned before, some improvements are proposed in order to address important inter-domain multicast routing requirements.

Network asymmetries are more likely to happen at inter-domain level mainly due to distinct domain routing policies. So it is not possible to construct the inter-domain multicast distribution tree by ignoring them and assuming symmetric inter-domain links.

The right way to deal with asymmetry is to start a tree construction from its root towards the new leaf member. This solution has been proposed in [4] and has two known drawbacks: a greater join latency caused by excessive control messages and an excessive load on root routers.

In order to avoid root routers overload, in PAQoSIDMR proposal the join requests are handled by the first in-tree router that receives them, thus relieving the RP of that task. This enhancement gains extra importance at inter-domain level, if we consider that it can reduce the number of domains involved in tree branch setup, but also reduces the possibilities of finding a feasible path.

If the first in-tree router fails to connect a new member, a controlled number of retries may be conducted by other in-tree routers, including, as a last retry, the RP.

Figure 1 illustrates a successful tree branch construction. The new receiver host (NR) should creates a *Join-Request* message, including the required QoS parameters in it, and addresses it to the Rendez-Vous Point of the group.

The *Join-Request* message is forwarded, hop by hop, until it reaches a router already connected to the multicast tree. (in-tree router). This router will then initiate the new branch construction by sending *Join-Answer* probing messages back to the New Receiver. The in-tree router must send one *Join-Answer* for each available, policy consistent, return path registered in its multicast routing information base. The current multicast frameworks [5] [6], allows routers to maintain multicast specific routing information by means of M-RIB. Like the standard unicast RIB, M-RIB also contains unicast prefixes and their attributes, but is only intended for multicast usage. This M-RIB allows a domain to announce the same unicast prefix with different attributes specifically for multicast purposes. Therefore *Join-Answer* messages could be addressed to the unicast address of the *Join-Request* originator taking into account the multicast specific policies established.

Before forwarding *Join-Answer* messages, all routers must collect dynamic QoS metrics and append them to the message. If the accumulated path QoS metric does not meet the QoS requirements included in the request, the *Join-Answer* message must be discarded,

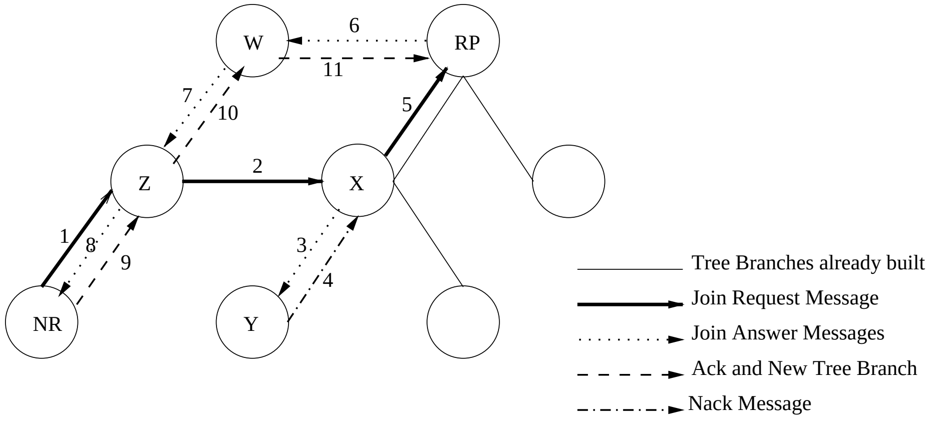


Fig. 2. Inter-domain tree construction after one branch setup fail. Actions are numbered in the order they occur

and a *NAck* message is sent back to the *Join-Answer* originator. Those *NAck* messages can be forwarded using unicast routes.

Any *Join-Answer* that reaches the New Receiver contains information about a valid feasible tree branch. A selection procedure must then be executed to select one of them according to some criteria, and, finally an *Ack* message establishes the branch, including the necessary state information in each router. The *Ack* message must travel in the exact opposite direction followed by the *Join-Answer* message selected.

A. Dealing with tree branch setup failures

The tree branch construction may fail if none of the possible alternative paths can meet the QoS requirements. This situation is illustrated in figure 2.

The in-tree that handles the *Join-Request* and initiates the tree branch construction, can also detect branch setup failures. It will always receive an *Ack* message when a feasible path is found, and a *NAck* message for all eliminated alternatives. In the example shown in figure 2, there was only one possible path available in the first in-tree router found, and a *Join-Answer* was sent on that path. However, when the router Y handle the *Join-Answer*, it calculates the cumulated QoS metrics and concludes that the path can not meet the requested QoS, and immediately sends a *NAck* message back to router X. As router X does not have other possible path to the new receiver, it forwards the *Join-Request* message to its upstream neighbor in the tree, which is, in this particular situation, the Rendez-Vous Point.

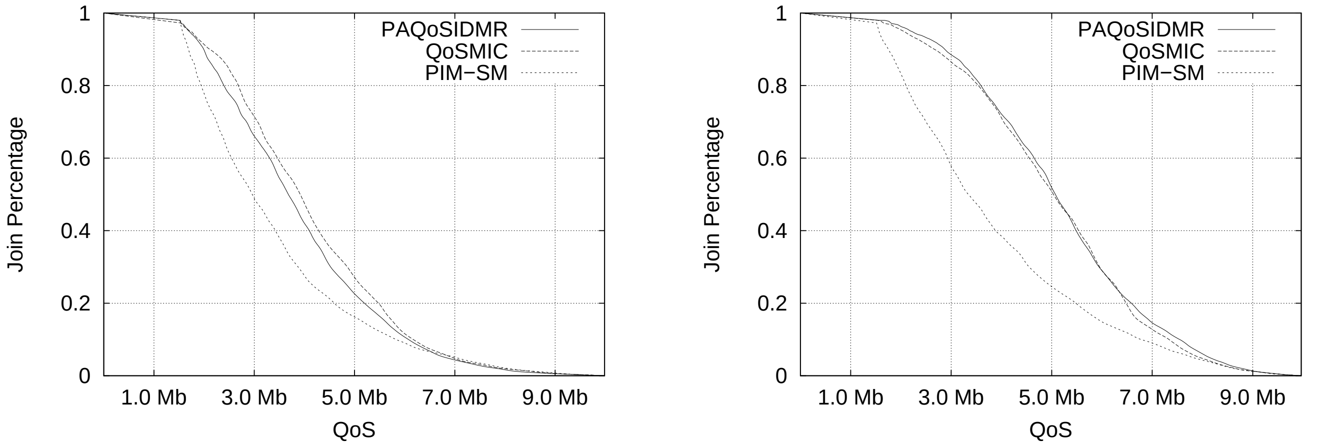
In extreme situations, with consecutive retry failures, many in-tree routers may be involved in the join procedure without success. This results in a very large join latency. Note that even in the worst case, when the *Join-Request* reaches the RP, the procedure terminates. If there is no possible path a *NAck* message is sent back to the New Receiver router.

In order to control better the join procedure, a retry counter is also included in the *Join-Request* to reduce the number of retries to an acceptable limit. The counter may be initialized with a value of 2 or greater.

IV. Simulation Analysis

In order to evaluate the performance of the proposed protocol, a large set of simulations were conducted on Network Simulator (NS)[7]. Besides PAQoSIDMR, two other reference protocols were used: PIM-SM and QoSMIC. PIM-SM is the current commonly used best effort protocol and it is used here as a bottom line reference. QoSMIC is a superset of YAM that can be configured to find paths using large portions of the already connected in-tree members, under the supervision of a tree manager. In order to fairly compare it with PAQoSIDMR, we used it to build unidirectional core rooted trees instead of the originally proposed bidirectional shared ones. For simplicity and improved efficiency the core node is always the tree manager.

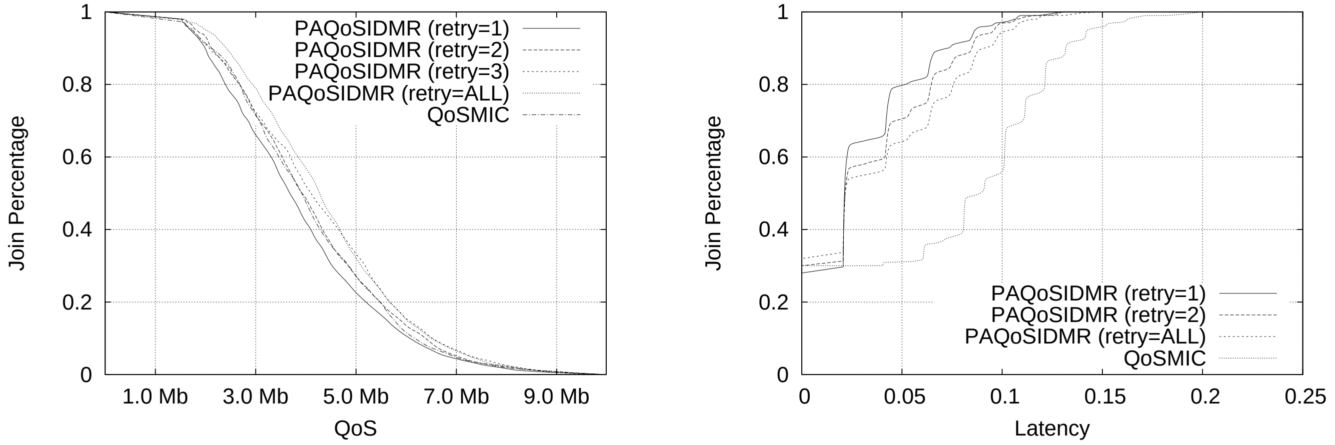
GT-ITM [8] and BRITE [9] were used to generate 100 nodes inter-domain topologies. 100 simulations were run for each topology. In each simulation a node was randomly chosen as group core. Then, one by one, 60% of the nodes join the group. Each node is randomly chosen among all unconnected ones. Since the focus



(a) *QoS Obtained (edge probability 0.033, average node degree 3.5)*

(b) *QoS Obtained (edge probability 0.066, average node degree 6.4)*

Fig. 3. Simulation Results - QoS Metrics



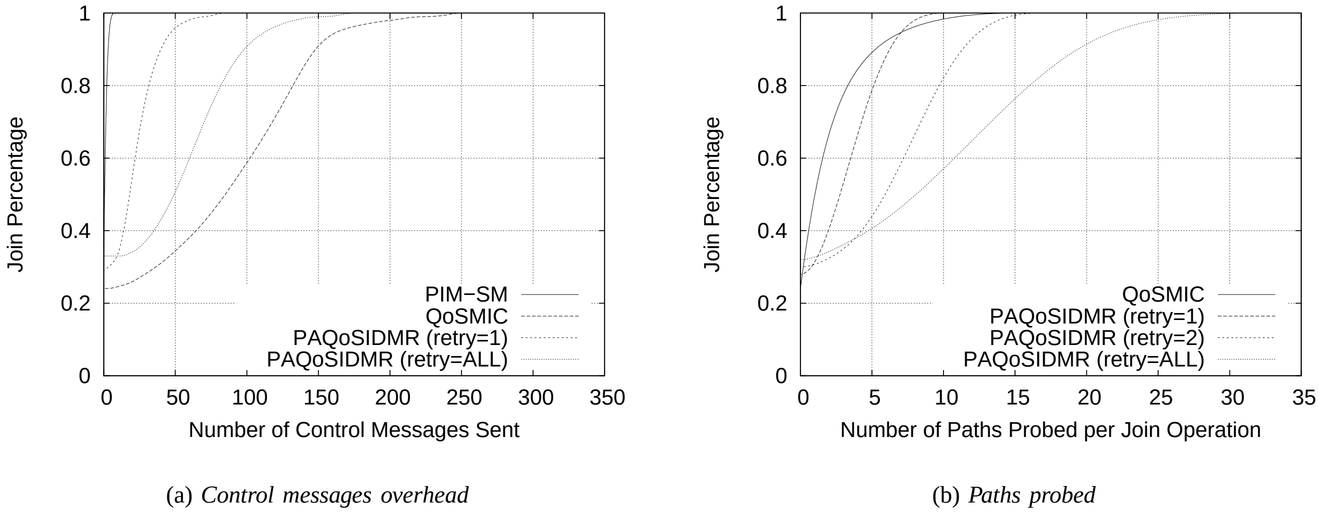
(a) *Retry mechanism (inside tree)*

(b) *Probing Latency*

Fig. 4. Simulation Results - Retry Mechanism

here is on the probing efficiency, we used a join only scenario. We also varied some topology parameters, like the number of nodes, the link bandwidth distribution (heavy tailed, exponential, uniform) and the node degree. From extensive conducted experiments with different scenarios, only the node degree function seems to affect the results. Other parameters do not impact protocol comparison. Unless otherwise stated, results refer to topologies generated with uniform link bandwidth distribution (1.5Mb-10Mb). Multicast receivers try to connect on paths with higher available bandwidth. One source generating CBR traffic is attached to the core node.

Four performance metrics were used: *ObtainedQoS*, *ControlMsgsOverhead*, *PathsProbed* and *ProbeLatency*. Figures 3(a) and 3(b) show the results for the *ObtainedQos* metric. Instead of probing for each receiver QoS requirements, the protocol modules were configured for connecting over the best possible path found. This means that each receiver will have to wait a fixed amount of time for all answers. The BW QoS obtained in each join operation is then registered. For each value we compute the percentage of join operations that registered a greater or equal BW value. Graphs 3(a) and 3(b) show a smooth line plot of every value measured. Result analysis



(a) *Control messages overhead*

(b) *Paths probed*

Fig. 5. Simulation Results - Overhead

show that PAQoSIDMR can perform 15% better than PIM-SM for an edge probability of 0.033 (fig 3(a)) and performance increases up to 30% when we double the edge probability (fig 3(b)). In the first case PAQoSIDMR performs 5%-7% worst than QoSMIC. For higher edge probabilities PAQoSIDMR performs as good as QoSMIC. In both cases PAQoSIDMR retry mechanism was not used, meaning that only the first in-tree node found was used in the probing process. Indeed, as already stated, PAQoSIDMR includes a retry mechanism based on further forwarding the request towards the core, issuing probes from inner in-tree nodes. A retry counter limits these efforts to a reduced number of nodes. As shown in figure 4(a), PAQoSIDMR performs better than QoSMIC for retry values greater than 2, and at least as good as QoSMIC for a retry value of 2.

Figures 5(a) and 5(b) show the results for the *ControlMsgsOverhead* and *PathsProbed* metrics. For each connection request the total amount of control messages sent by all nodes is registered. Besides the connection requests themselves, this includes the path probing messages, the tree branch setup messages and all QoS state refresh messages. PAQoSIDMR, for instance, needs to keep an up-to-date multi-value metric in all in-tree nodes. This feature implies a periodic refresh down the tree. After having waited a fixed amount of time, each new receiver enters a selection procedure to choose the higher QoS path probed. The number of paths probed is also registered. Values are plotted against the percentage of requests that used a less or equal number of probes. Results show that tree construction overhead is significantly smaller in PAQoSIDMR. However, the number of paths probed per request is bigger, meaning that QoSMIC is more efficient in finding paths with less common parts. That’s a consequence of its complex candidate selection mechanism.

Last discussed metric is the *ProbeLatency*. Because in this scenario all receivers wait a fixed amount of time, the total join latency time is bounded by this value. However, by registering the time of each probe arrival we can plot the ”best path” probe latency (Figure 4(b)). Values are plotted against the fraction of the receivers that experienced less or equal latency. Values observed largely depend on topological delays and network load. By using a fixed link delay topology with traffic loads far behind congestion, we can stress up the protocol differences, which are not significant. Probing latency is greater in QoSMIC because it uses a tree manager to conduct the in-tree candidate selection. After a scope limited local search procedure, requests are sent to the tree manager and then multicasted to the in-tree probing candidates, thus increasing probing latency.

V. Conclusion

In this article we have presented and evaluated a proposal for inter-domain multicast routing that builds QoS aware inter-domain unidirectional multicast distribution trees. Evaluation was focused on the performance of the path probing strategy.

New members are connected through tree branches that meet some QoS requirements specified by them. Tree construction is requested by new members but initiated from in-tree members, in order to better deal with asymmetries. Alternative paths are probed with messages that collect QoS path metrics, and one of them is selected and established by the new receiver. No assumptions are made about intra-domain choices.

Our simulations show that the proposed strategy efficiently finds feasible QoS Paths to connect new members with very low message control overhead, which is a desirable feature for the inter-domain scenario. Simulations also show that the retry mechanism can largely improve path selection results. By further forwarding the join requests inside the constructed tree, chances of finding feasible paths do improve.

References

[1] K. Carlberg and J. Crowcroft, "Building shared trees using a one-to-many joining mechanism," *ACM Computer Communication Review*, pp. 5–11, 1997.

[2] Michalis Faloutsos, Anindo Banerjea, and Rajesh Pankaj, "Qosmic: quality of service sensitive multicast internet protocol," in *Proceedings of the ACM SIGCOMM '98 conference on Applications, technologies, architectures, and protocols for computer communication*. 1998, pp. 144–153, ACM Press.

[3] Shigang Chen, Klara Nahrstedt, and Yuval Shavitt, "A qos-aware multicast routing protocol," in *INFOCOM (3)*, 2000, pp. 1594–1603.

[4] Horst Hodel, "Policy tree multicast routing: An extension to sparse mode source tree delivery," *ACM SIGCOMM Computer Communications Review*, vol. 28, no. 2, April 1998.

[5] Kevin C. Almeroth, "The Evolution of Multicast: From the MBONE to Interdomain Multicast to Internet2 Deployement," *Network, IEEE*, vol. 14, no. 1, pp. 10–20, January/February 2000.

[6] S. Kumar, P. Radoslavov, D. Thaler, C. Alaettinoglu, D. Estrin, and M. Handley, "The MASC/BGMP architecture for inter-domain multicast routing," in *Proceedings of ACM SIGCOM*, Setembro 1998, pp. 93–104.

[7] K. Fall and K. Varadhan, "*The NS Manual*," Jan 2001, URL=http://www.isi.edu/nsnam/ns/ns-documentation.html.

[8] K. Calvert and E.W. Zegura, "*GT-ITM: Georgia Tech internetwork topology models* (software)," 1996, URL=http://www.cc.gatech.edu/fac/Ellen.Zegura/gt-itm/gt-itm.tar.gz.

[9] A. Medina, A. Lakhina, I. Matta, and J. Byers, "BRITE: An Approach to Universal Topology Generation," in *Proceedings of the International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunications Systems- MASCOTS '01*, August 2001.