Management Driven Multicast Protocol with End-to-End QoS

Radu Iorga
Telecommunication Dept.
University POLITEHNICA of Bucharest
Bucharest, Romania
radu.iorga@elcom.pub.ro

Eugen Borcoci
Telecommunication Dept.
University POLITEHNICA of Bucharest
Bucharest, Romania
eugen.borcoci@elcom.pub.ro

Radu Dinel Miruta
Telecommunication Dept.
University POLITEHNICA of Bucharest
Bucharest, Romania
radu.miruta@elcom.pub.ro

***Abstract*—Multicast technologies supporting various media services are increasingly seen in the current Internet and are expected to be used also in future Internet deployments. While traditional IP level multicast and overlay multicast are well known solutions, multi-domain multicast with quality of services (QoS) guarantees is still a research topic. This paper proposes a management driven hybrid multicast system and protocol, QoS enabled and spanning multiple IP domains. Starting from a previously defined architecture, the management system is developed and then, the protocol design, implementation and some performance evaluation are presented.**

Keywords-hybrid multicast; overlay, quality of services; virtualization; multiple domains; service level specification.

I. INTRODUCTION

Increasing demand for multimedia content distribution, while satisfying different levels of quality of services (and quality of experience for users), reinforced the interest for multicast technologies,. While IP level multicast is highly efficient it has not been largely deployed in multi-domain environments. On the other hand, the overlay (application layer) multicast is easier to be deployed, but is less efficient and does not exploit the IP native multicast capabilities where they exist [1]. For multi-domain multicast , one has also to consider that each domain is managed by an independent Network Provider (NP), or operator. Therefore, a *management driven multicast hybrid solution* seems to be attractive. This is true especially if QoS and flow distribution process supervision are wanted to be performed by the management, in order to fulfill requirements of *Service Level Agreements (SLA)* negotiated and agreed between the NPs and the multicast services users/clients.

The work [2] proposed an architecture of such a hybrid multicast framework which is QoS capable, where IP level intra-domain multicast is combined with inter-domain overlay multicast. This paper has started from the architecture in [2] and here the multicast management system is further developed. Then, the main contribution of this work is the *Management Driven Multicast Protocol (MDMP),* based on a powerful algorithm performing jointly a constrained QoS routing, resource reservation, and multicast tree mapping onto multi-domain network topologies, under control of a distributed management system. The specification, design and implementation of the protocol and also some performance evaluation have been accomplished and presented in the paper. The protocol offers a solution for real-time multimedia applications like IPTV, VoD in a multi-domain multi-provider scenario.

The multicast system discussed in this paper is currently under development in the European FP7 ICT research project, "Media Ecosystem Deployment Through Ubiquitous Content-Aware Network Environments", ALICANTE,[2][3].

The paper is organized as follows. Section II presents samples of related work. Section III introduces the MDMP, showing a high-level view and its main rationale. Section IV is focused on the most important design and implementation issues, including the protocol itself but also the algorithm used for multicast tree construction. Section V contains conclusions and outline of future work.

II. RELATED WORK

Several multicast solutions and protocols are already specified in IETF RFC documents and part of them are implemented and produced by equipment manufacturers. A comprehensive overview of multicast solutions is presented in [4].

However, in this study we are focus on adapting the chosen solution to the general multi-domain architecture defined in [2], [3]. There, *virtual content aware networks* with *guaranteed QoS* and *unicast/multicast enabled* - have to be constructed on top of multi-domain IP infrastructure, under management of several virtual network managers and network managers - the latter being aware of actual network topology and resources. Basically three solutions can be analyzed: IP level, overlay level and application level multicast (ALM). The latter is excluded from our scope given that it is performed in the end-host machines, while we need the network support. Additionally it is shown in [4] that IP multicast and overlay multicast (based on special nodes in the network) have better trade-offs between multicast tree cost and end-to-end delay than ALM.

The PAM protocol ("Adaptive Hybrid Multicast with Partial Network Support") defined in [5] combines the advantages of IP multicast with the ones of *ALM* and reiterates the idea of connecting native IP multicast islands with the use of IP-in-IP tunnels. Just as in the case of AMT

 ISBN: 978-1-61208-263-9

[6] this approach considers the IP multicast tree already constructed and does not offer any QoS support on the tunnels used between *m-routers*. The HOME (Host group based Overly Multicast Environment) approach defined in [7] takes the idea of ALM to move the multicasting towards the end nodes but relocates the terminations of the tree into the Designated Routers (DR) leaving the communication between DR and end-nodes as native IP multicast. The ALM is than created with the DR playing the roles of end nodes.

In [8], several QoS multicast solutions are analyzed. However, QoS constrained routing is not sufficient in our case, given that QoS guaranteed multicast connectivity services are needed. Therefore, some resource reservation is necessary. A QoS extension for OSPF (*QOSPF*) has been proposed in [9] and completed in [10] with multicast extensions. The solution is not appropriate, given the necessity of resource reservation.

In the unidirectional core-based trees the existence of a central node, which might not be on a QoS path between source and receiver, raises even more challenges than in *Shortest Path Tree (SPT)* cases. The best thing that can be done in this case is to try to assure local QoS and not end-to-end [8]. The major core-based protocol, *PIM-SM* [11] has no QoS extensions.. However, the fact that it depends on the underlying unicast routing protocol, makes it good candidate to support QoS constraints. PIM-SM is the main multicast protocol in [12] which proposes a hybrid multicast architecture where IP multicast is used only in the last domain where the receivers are. More, the solution offers QoS guarantees only if the unicast routing protocol used by PIM-SM can offer any. A hybrid multicast system is also proposed in [13] to create an E-Cast tree composed of unicast QoS pipes for inter-domain and use of PIM-SM inside the domains as opposed to MDMP which creates two combined multicast trees with QoS.

The work [14] proposes a QoS framework for a multi-domain multicast service. The QoS control is performed by choosing the best available inter-domain multicast route in order to respect end-to-end connection requirements. A Multicast Inter-Domain Entity is introduced in each network domain to search and test multiple paths from domains already in the multicast group (tree-domains) towards a new joining-domain. The difference of our solution is that we take benefit from existence of Virtual Network Manager and also Network Manager in each domain and the multicast tree computation is performed there so no other entity is needed.

The paper [15] presents a system called Multi-servIce Resource Allocation (MIRA), where a multicast-aware resource reservation protocol for class-based networks that consider routing asymmetries is proposed. MIRA agents are distributed in the network. The difference of our solution is that we do not need MIRA agents inside the network and the trees are not per session based but the multicast tree life is that of a virtual network to which it is associated.

III. MANAGEMENT DRIVEN MULTICAST PROTOCOL

This section will present additional motivations to define the *MDMP* framework. Then the principal architectural features of the MDMP are described. Its name comes from the fact that there is a central manager that computes the tree, based on input data, and programs each router along the tree with the computed information.

An important MDMP usage is in *virtualization based on overlays,* which is seen today as a major method to make the Internet (and also Future Internet) more flexible, [16],[17], [18] by slicing it in parallel isolated planes. Towards this aim, *Network Infrastructure Providers (NIP)* can offer their sliced resources to some *Virtual Network Provider (VNP)* which constructs *Virtual Networks (VNet)* by merging several network infrastructure resources. Each NIP, while cooperating to the VNet, still manages independently its resources and maintains knowledge on their availability. A multicast capable VNet needs a multi-domain tree, where each domain has to construct its own part of the tree.

General *virtual networks mapping* [16],[17],[18] is needed, *abstracting the subset of network resources* (links, nodes). An overlay multicast VNet seems to be the first approach. However it is better to benefit in some domains by native IP multicast capabilities, therefore a *hybrid solution* is more appropriate and so is adopted in this work.

The VNets are *created on demand,* by some *generalised "VNet Users" or Requestors" using a VNP* management framework. The VNP management will perform the multicast VNet planning, advertising, offering, negotiation, provisioning, and commands their operation (installation, modification, manipulation, monitoring, and termination), while cooperating with NIPs. A VNP might be seen as containing one or several VNet Managers (VNetMgr) which at their turn can be associated in one to one relationship with each Net Manager (NetMgr) of the NIP. This approach satisfies the requirement of having distributed management and control by avoiding a single central manager. Creation of *multicast VNets* fulfilling QoS requirements needs vertical and horizontal *negotiations* and SLA concluded between entities while preserving each domain's resource management independence.

VNets optimized mapping [19], is necessary onto multi-domain substrate. This is supported by the MDMP which combines several functions: constrained routing, admission control for resource reservation, and final tree mapping combined with inter-domain QoS-enabled routing and resource reservation [19]. An appropriate metric should be defined for tree construction to be used in the selection of tree paths and then a reservation action is performed at the level of VNetMgr and respectively NetMgr. An additive metric incorporating QoS needs has been defined. Scalability of the solutions is also necessary.

The result of the above considerations is the architectural solution shown in Fig. 1, where MDMP constructs the multicast VNets. The actors involved are: *Multicast VNet requester – (*e.g., a high level Service

Provider); *Multicast VNet Manager (MVNetMgr)* –the block actually computing and controlling the tree installation. For each tree there is a MVNet Manager, which is the initiator of the process and this is seen as a central management node with enough information (topology, capacities, etc.) and political rights to take all the construction of the tree in its responsibility; *Network Manger* that manages a single domain resources and in particular has the responsibility to map its part of the multicast tree on its real topology.. Also at the request of MVNet Manager, it commands the routers to install the tree. Assuming that VMnetMgr knows the topology of the network, a modified Dijkstra algorithm is used to compute parts of the tree. The existence of a source-based tree means that *Source Specific Multicast (SSM),* [20] should be used as addressing scheme. This solves any issues with group address management.

The major steps of *MDM* algorithm are: *1.Get request and topology; 2. Compute SPT on the graph after removing the link that do not meet the QoS constraint; 3. Enforce the decision of the algorithm (config. multicast routing tables and do the actual reservation of resources in the routers).*

IV. DESIGN AND IMPLEMENTATION

The MDMP system has been implemented as part of Alicante work and installed on a pilot testbeds (Fig. 2). It consists in three fully meshed domains, with all nodes as Linux routers with IP multicast and QoS support enabled. In this example, the managers are located in the same physical machine as the routers (see C12 in domain1) but the communication between different entities is made over TCP connections allowing physical separation.

MDMP can deal with any topology provided that there is some entity able to find and disclose it (discovery mechanism is out of this paper's scope). For the ease of portability, an xml format has been defined to represent the topology:

<ip>141.85.43.124</ip>
<intf>eth1</intf>
<nip>141.85.43.130</nip>
<bwd>3</bwd>

The implementation is made in C under Linux, hence the name of the interfaces have the Linux format. The topology should be read like this: Node .124 has a neighbor node .130 reachable through interface *eth1* with available bandwidth of *3 Mb/s*.

Only bandwidth is used in this implementation as being the easiest case, but in a form of any additive metric [19] (cost of a link is 1/bandwidth). This representation of topology can be used for both inter-domain and intra-domain, even though in inter-domain case the *<ip>* represents the MVNetMgr IP while in intra-domain the *<ip>* represents the router IP.

An open-source library called *libxml2* [23] is used to read and parse the xml file describing the topology. Each node is given a unique ID. As this is a multithreading environment *mutex-es* are used to ensure the uniqueness of the IDs. The extracted data is kept in simple linked lists: one that keeps , all the nodes and each node has a linked list with its neighbors (reduced structures are presented):

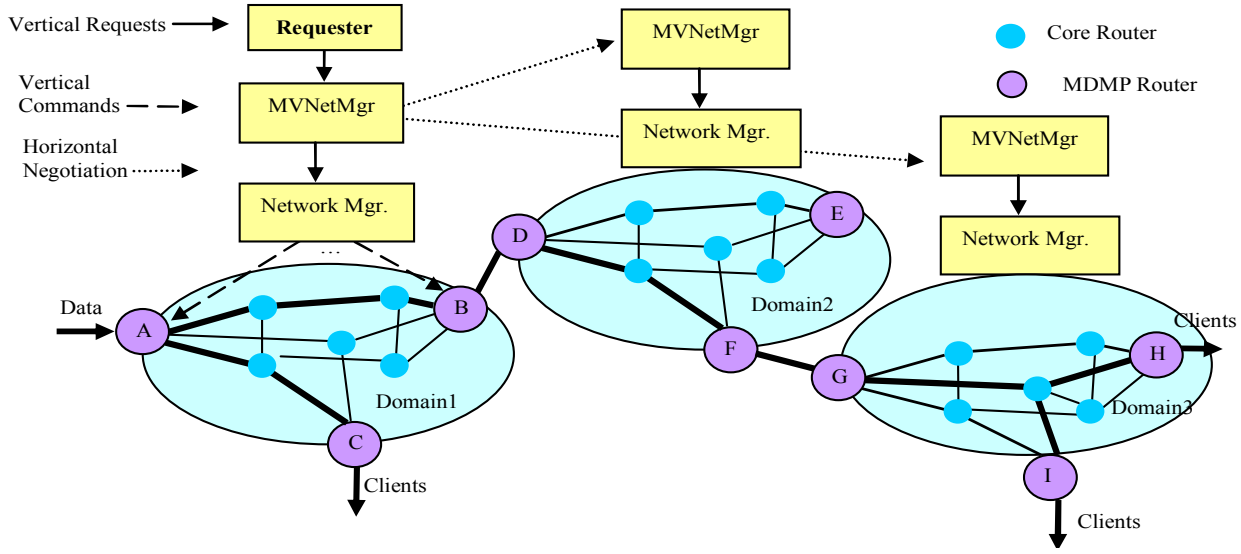

Figure 1. A MDMP typical architecture

```
/*Linked list holding ALL nodes in topology*/
struct mcast_node {
        int   node_id; //The ID of the node.
        char  node_ip[IP_MAX_SIZE]; // Node's IP
        struct neighbor   *neigh_head; // Head of neighbors
                                       linked list
           ... //some other internal variables
           struct mcast_node *next; //next node.};
  /* intf struct: A linked list holding ALL neighbors*/
struct neighbor {
```

```
char intf_name[INTF_NAME_SIZE]; //
char    intf_id; // intf id : eth0->0
int     bwd; // Bandwidth to neighbor
char    neigh_ip[IP_MAX_SIZE];
 ... //some other internal variables
struct neighbor *next; // next neigh};
```

This is just a proof of concept on a small size testbed, hence the performance is not very stringent. However, in a real life implementation, advanced data structures can be used to speed up things. For instance, now if we need to lookup a node, a *O(n)* complexity will be achieved, but if, in an advanced implementation, a hash table is used and the hashing function is correctly chosen, a complexity of *O(1)* can be obtained.

A. Inter-domain MDMP

Message Sequence Chart (MSC) is presented in Fig. 3 showing the MDMP signaling to build the multicast tree spanning multiple domains. The trigger to MDMP is a request for a QoS assured multicast tree and it must contain the source IP of the tree (Src_IP), the desired QoS (only bandwidth in our implementation) and the receivers (routers M13, M22, M23 and C32 in Fig. 2). We consider a request for 3 MB/s in this example. Several requests can be handled because we have a separate thread for each new tree requested and semaphores to protect the data.

The MVNetMgr receiving this requests will be called the *Initiator* and it will execute some preliminary checks to determine if the tree can be accepted or some negotiations are needed (*Neg_req()* and *Neg_rsp()*). The next step is to build the Overlay Tree which must start with determination of the domains with receivers: in our implementation, where the domain is represented by the MVNetMgr IP, this kind of mapping is stored in a database. It is outside the MDMP scope how this mapping is obtained.

In Fig. 2 the mapping will mean that for M13 the IP of MVNetMgr1 will be found, for M22 and M23 the IP of MVNetMgr2 will be found and so on (action 1 at MVNetMgr1). The second action is to pick a Group IP address. Note that we have also the source of traffic so the pair (Src_IP, Grp_IP) is unique and will be referred to as (S,G). At this point we have a graph with domains as nodes, we have the receiving domains and we have the 3Mb/s constraint. First of all, the non-compliant links are

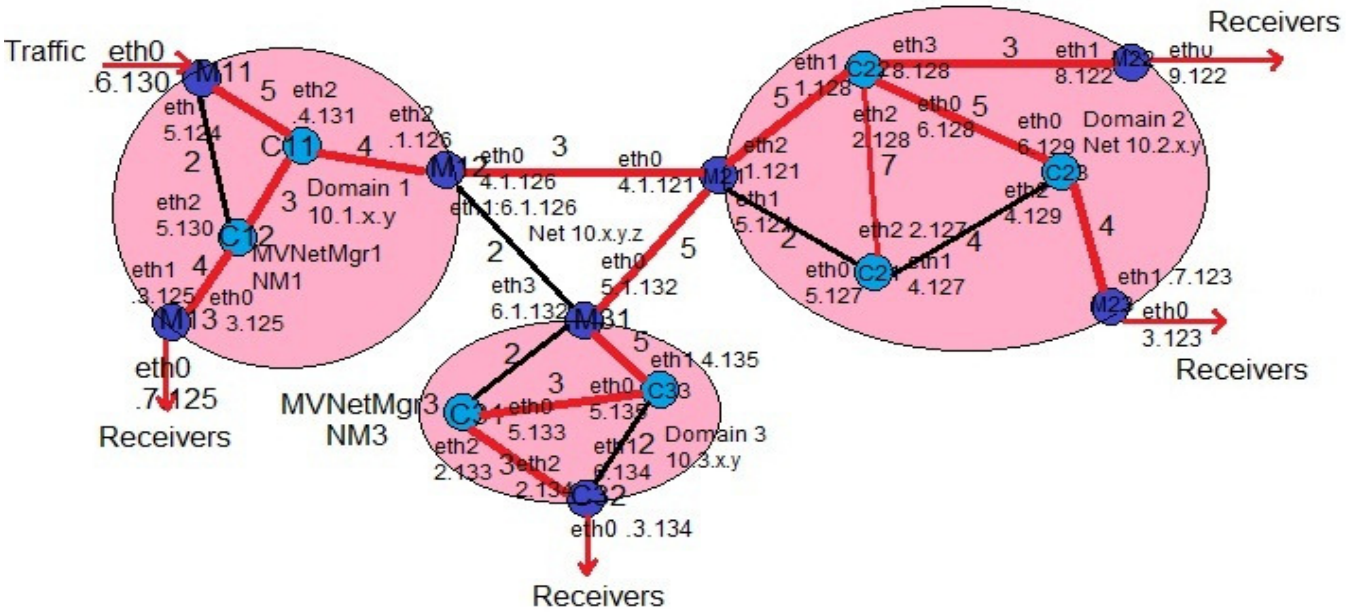

Figure 2. The testbed topology

eliminated (i.e., link between M12 and M32) and based on the new graph and using Dijkstra algorithm the Shortest Path Tree (SPT) is computed (action 3). Details of the algorithm itself are presented in section C. But SPT is covering all nodes in a graph so we have a special API that traverses the graph and prunes the nodes that have no receivers or are not on the path to any receiver. At this point we have the Overlay Tree, so the Initiator will now try to contact each domain, including own domain, and request the IP multicast tree. As there might be many domains we developed a multithreading environment using Linux pthreads. For each domain belonging to the Overlay Tree we create a pthread to handle the negotiation. After all the signaling is done all threads are terminated to save CPU resources.As shown in Fig. 3, a *Req_tree()* is sent to MVNetMgr2 and MVNetMgr3. The request sent to other MVNetMgr are similar to the one received from the SP. However the receivers are different: the request sent from MVNetMgr1 to MVNetMgr2 has as source the IP of MVNetMgr1 and one of the receivers is now MVNetMgr3.

 ISBN: 978-1-61208-263-9

Using the mapping database, MVNetMgr1 will determine that the source of the IP multicast tree is M21 and that M21 is also a receiver as it is the route towards MVNetMgr3 which is a receiver. But this is normal as different interfaces of M21 are used as source and receiver. Apart from the Initiator all the other MVNetMgr have fewer actions to perform: get the mappings from database and just relay the request to own NM. The *Req_tree()* message from MVNetMgr2 to NM2 is very similar to the one from MVNetMgr 1 to MVNetMgr2. The main difference is that now the MVNetMgr IP is replaced with the needed router IP. The order of actions is presented sequential in the MSC to make it easier to understand, but in reality it is hard to know the exact order due to the multithreading behavior of our solution. All the above communication is realized over a TCP socket in order to make it reliable. The socket is created using the IPs involved and predefined TCP ports:

#define ROUTER_PORT 39393 // to enforce on the routers
#define INTRANRM_PORT 19191 // to request from own domain
#define CANMGR_PORT 29292 // to negotiate with other domains

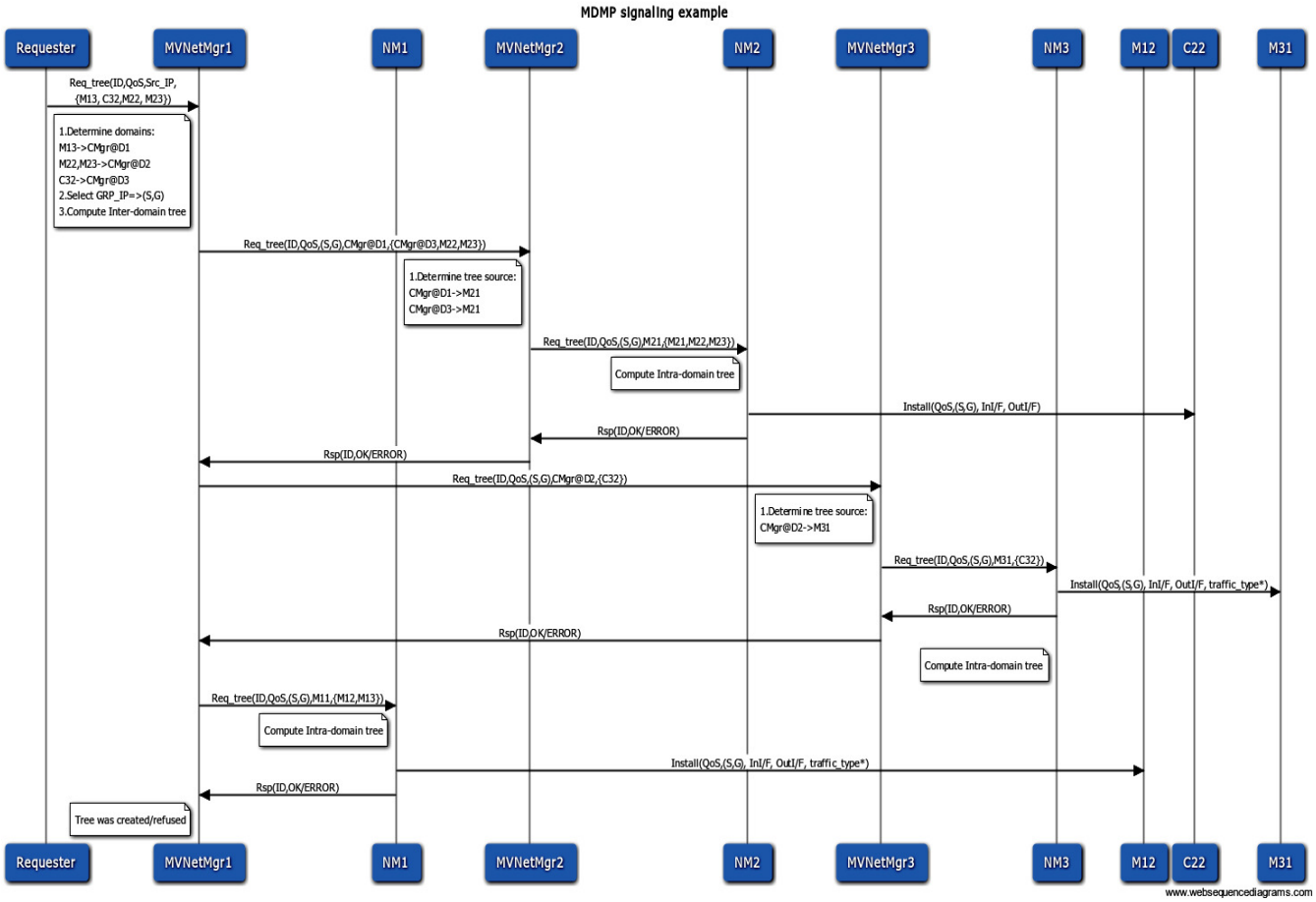

Figure 3. Control messages for inter and intra domain

B. *Intra-domain MDMP*

Switching focus to NMx will duplicate most of the actions presented in inter-domain scenario at intra-domain level. From the protocol point of view, a request is received and the same algorithm from section C is used. We reiterate the discussion about the difference between SPT and the multicast tree in order to emphasize in Fig. 2 a case where a node (C21) is part of the SPT (thin red line) but, as it has no receivers nor is on the path to any receiver, is pruned from the actual multicast tree (thick red lines). The NM will have to enforce the result on the routers. We present the three most relevant moments to install the tree: right after the tree is computed; at a later time triggered by an Invocation message from the requester; at a later time specified from the beginning by the requester. We've implemented the first case as it offers the most testing possibilities.

For each router that needs instructions a separate thread is opened. In an intra-domain topology this might became a problem as there might be many routers and the CPU might get over-loaded. However, the processing needed in each thread is not very big, so the CPU overloading issue might not be a top priority issue. Details on router configuration are presented in SectionIV-D.

 ISBN: 978-1-61208-263-9

C. MDMP algorithm

Because the different approaches between unicast and multicast situations, for this last case we propose an enhanced version of the algorithm, more detailed presented in [19], but focused for this time on multicast. The new improvements here assume, apart from specific adaptations of multicast the concept of prioritizing requests..In order to alleviate the impact of the unsolved requests, a better situation will be if these requests will be the least important ones. All requests form the received set are grouped based on the source node and *group priority* is defined (lower value means higher priority). In the case of several groups with the same priority, the algorithm will permute the processing order obtaining the best cost. Based on our simulations results just after a few permutations the total cost is not decreasing significantly, so it is not need to run the n! permutations, where n is the number of different group with the same priority. To offer a maximum flexibility solution w.r.t. Requester interests, one should admit that the Requester can specify a priority for each individual request. The group priority has precedence over the individual request priority.

The used algorithm for unicast is a little bit more extensive that the current one for multicast in terms of requested bandwidth. For the multicast case, we should have the same bandwidth values for different requests even if they are part of the same group. Even if the requested bandwidth is the same for each request, the order of solving requests inside each group becomes important: honoring one request before the other can exhaust the available bandwidth for a segment. Other specific adaptation for the multicast case, compared with the unicast situation presented in [19] is the non using of the blind search (when constraints are applied on the STP found by the modified Djikstra algorithm, the path can take other way, through other nodes, different from the ones indicated by Djikstra). Leaving the STP found by the modificd Djikstra algorithm, takes us away from the multicast context. In order to obtain more accurate results we simulated a two-level hierarchical network with 31 ASes, each AS containing a maximum of 8 routers, totaling a number of 186 nodes using a dedicated tool for topology generation from Scilab [21].

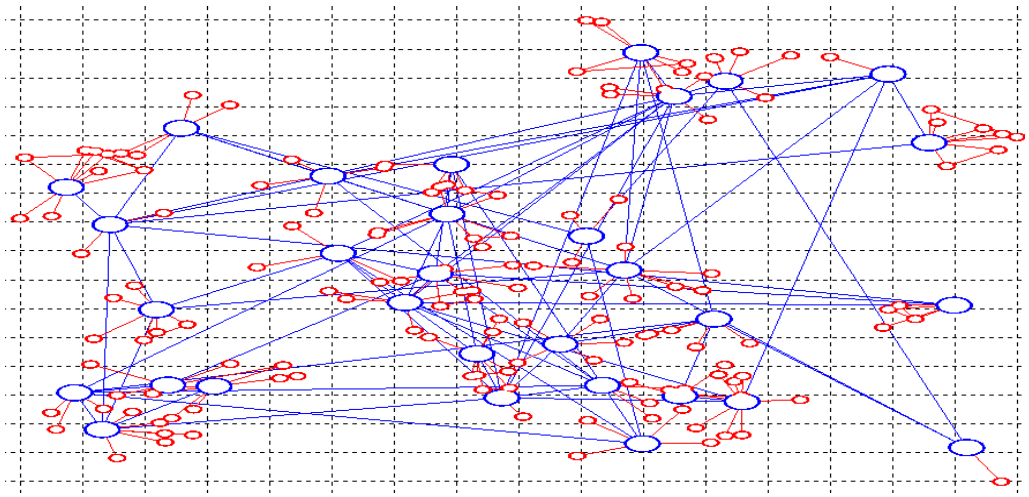

Figure 4. Two levels hierarchical network topology

The obtained topology is the one from Fig. 4 where the circles with larger diameter and their associated segments represents the inter-domain topology and the smaller diameter circles with their associated links are the intra-domains. Each segment for both inter and intra-domain areas has an associated bandwidth generated in respect to a Gaussian distribution centered in 100.

Because the NARVAL_T_HierWaxmanConD function from NARVAL [22] was created to generate a network graph with much more layers, we modified it for only two.

Each segment for both inter and intra-domain areas has an associated bandwidth generated in respect to a Gaussian distribution centered in 100. Because the NARVAL_T_HierWaxmanConD function from NARVAL [22] was created to generate a network graph with much more layers, we modified it for only two. The network backbone of size n (31 in our case) and thereafter the second layer added were created based on the Waxman model. The topology was generated using the following parameters values (more details about the significance of each parameter can be found at the NARVAL help):

a=0.4;//first parameter of the Waxman model
b=0.5;//second parameter of the Waxman model
n=31;//network backbone size
l=1000;//network squared area side
nl=8;//maximal quantity of nodes per subnetwork
db=25;//original diameter of nodes

 ISBN: 978-1-61208-263-9

dd=15;//diameter difference between successive network layers
cv=[2 5];//color of each network layer
[g,d,v]=NARVAL_T_2layers(a,b,n,l,nl,db,dd,cv);//application of NARVAL_T_HierWaxmanCon
And centered using the following script :
Lxmin=100; Lxmax=900; Lymin=100; Lymax=900;
nodex=g.node_x; nodey=g.node_y;
xmin=min(nodex); xmax=max(nodex);
ymin=min(nodey); ymax=max(nodey);
nodex=(Lxmax-Lxmin)/(xmax-xmin)(nodex-xmin)+Lxmin;*
nodey=(Lymax-Lymin)/(ymax-ymin)(nodey-ymin)+Lymin;*
g.node_x=nodex; g.node_y=nodey;
ind=1;//window index
f=NARVAL_G_ShowGraph(g,ind);//graph visualization
After this step we successfully extracted the adjacency matrix containing at this moment zeros (no link) and ones (presence of a link) and we created other scripts in order to insert in this matrix the generated bandwidth values attached at each link.

Simulation results
We created a set of 10 requests divided into 5 different groups with different priorities as in Fig. 5, this representing the Traffic Distribution Matrix.

1	1 75 31 1 2	6	7 100 31 1 2
2	1 115 31 1 1	7	7 142 31 1 1
3	1 19 31 1 3	8	14 112 25 3 1
4	5 78 27 2 1	9	14 18 25 3 2
5	5 121 27 2 2	10	21 175 55 4 1

Figure 5. Set of requests

Each line of the matrix specifies an individual request as *{source node, destination node, requested capacity, group priority, individual request priority}*.
Because of the multicast context each request from a group must have the same requested bandwidth value.
Using the adjacency matrix of the above topology and this set of requests as the input data of our algorithm it produces the following results (it was introduced only 2 permutations in case of groups with the same priorities):

==========
Input file multicast.in:
Request 1->115, carry 31: 1 14 15 4 22 16 115
Request 1->75, carry 31: 1 14 18 8 75
Request 1->19, carry 31 unsatisfied. Node unreachable
Request 7->142, carry 31: 7 17 15 24 23 142
Request 7->100, carry 31: 7 17 14 13 100
…..
Cost: 19.360294. Satisfied requests: 9 / 10

Request 7->142, carry 31: 7 17 15 4 23 142
Request 7->100, carry 31: 7 17 14 13 100
Request 1->115, carry 31: 1 14 6 23 4 22 16 115
Request 1->75, carry 31: 1 14 18 8 75
Request 1->19, carry 31 unsatisfied on 0->19. Node unreachable
……
Cost: 20.120386 . Satisfied requests: 9 / 10

Best cost: 19.360294
Satisfied Requests: 9 / 10
==========
Total time: 0.004000

The total processing time displayed here includes the time for printing functions which is quite significantly and it was obtained using a personal PC equipped with Intel Core 2 CPU, T5600@ 1.83 GHz, 2,00 GB RAM and a 32-bit OS. *Removing these printing functions, for the current input data, the processing time is too small for this six zeros granularity (0.000000)*
As it can be observed for both permutations only 9/10 requests were solved with a better cost associated for the first order. The request remained unsolved in this case is due to the situation of an unreachable node. Even if the network graph is constructed as a connex one, because of some optimization techniques used during the implementation (remove from graph all segments which do not respect the condition: avail_bwdb >= min_req_bwd value from the group) some nodes could become unreachable.
All requests were solved in respect with their group priorities and in respect with their individual priorities inside the group also.
It can be observed that we have 2 groups with the same priority (group 1 and 7) and because of this the algorithm performs 2 permutations, thus obtaining the best cost.

D. MDMP routers

There are two types of routers involved in MDMP: the core routers (any router with IP multicast and QoS capabilities), and the special edge routers, [2]. An ingress router receives unicast traffic and will send it as multicast; egress router receives either unicast or multicast, and will send unicast. An ingress router should determine the (S,G) tree the given packet belongs to. The incoming packet has as IP destination the IP of the egress router. So other type of information needs to be used, several of which have been proposed in Alicante [2]: the (SIP, DIP, proto, SrcPort ,DstPort) tuple configured by control plane; some special information inserted in the packet such as Content Aware Transport Information (CATI); NAL unit inside SVC header; deep packet inspection. The egress router has to change the IP destination of the packet to the IP of the directly connected edge router of the neighboring domain. This can be done by UDP tunneling the packet or by rewriting the destination address.
In Fig. 3 an *Install* message containing information about (S,G) and QoS is sent. For the edge routers the sequence of action is more complicated and has been described

 ISBN: 978-1-61208-263-9

above. But for the core routers, a simple system call to *smcroute* is made to install the routes into the kernel:

```
sprintf(cmd,"smcroute -a eth%d %s %s %s",
        r_cfg.in_intf, r_cfg.src_ip, r_cfg.grp_ip, oif);
system(cmd);
```

In our implementation we have used Linux and the traffic control tool *(TC)* for QoS, based on Hierarchical Token Bucket (HTB),.To apply the QoS requirements using *TC*'s HTB there are two steps: create the class of traffic based on the requested QoS (with the option of borrowing bandwidth if unused by other flows) and create the filter (based on (S,G) or the tuple above or any field in the packet) for the traffic to be classified inside the class. All traffic matching the filter should be guaranteed the traffic class. Using the same programming method as for adding the multicast routes, we create the strings needed for classes and filters based on received *Install* command parameters, and we make a system call to instruct the Linux kernel of our needs. Our implementation just tries to demonstrate that MDMP works. Of course that if the routers support more sophisticated QoS rules they can be applied with ease, as the protocol poses no restrictions.

V. CONCLUSION AND FUTURE WORK

The paper addresses the problem of building multicast QoS enabled trees spanning multiple domains. Some implementation details have been shown to prove that the concept. Scalability and performance aspects have been discussed. The MDMP trees are built under management actions that are supposed to be rare, so the response time of the protocol is not a constraint. These challenges towards a more scalable and performing software are in the front list of our future work plans related to MDMP. Further research is needed to solve in a timely manner any possible updates (prunes or grafts) to already installed trees.

ACKNOWLEDGMENT

This work was supported partially by the EC in the context of the ALICANTE project (FP7-ICT-248652) and partially by the project POSDRU/88/1.5/S/61178

REFERENCES

[1] H. Asaeda et.al., "Architecture for IP Multicast Deployment: Challenges and Practice", IEICE Transactions on Communications, Vol. E89-B, No. 4, 2006, pp. 1044-1051.

[2] E. Borcoci, G. Carneiro and R. Iorga**,** "Hybrid Multicast Management in a Content Aware Multidomain Network**",** AFIN 2011, The Third International Conference on Advances in Future Internet, pp. 90-95 http://www.thinkmind.org/index.php?view=article&articleid=afin_2011_5_20_70107

[3] ALICANTE, Deliverable D2.1, ALICANTE Overall System and Components Definition and Specifications, http://www.ict-alicante.eu/, Sept. 2011.

[4] Li Lao, J.-H. Cui, M Gerla, and D Maggiorini, "A Comparative Study of Multicast Protocols: Top, Bottom, or In the Middle?", Technical Report TR040054, 2005,Computer Science Department, UCLA,http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.59.4904 (last accessed March 4, 2013)

[5] H. Luo, K. Harfoush; "Adaptive Hybrid Multicast with Partial Network Support" - http://www4.ncsu.edu/~kaharfou/Papers/hybrid_multicast.pdf

[6] D. Thaler, M. Talwar, A. Aggarwal, L. Vicisano and T. Pusateri; "Automatic IP Multicast Without Explicit Tunnels (AMT)" – internet draft 2010 - http://tools.ietf.org/html/draft-ietf-mboned-auto-multicast-10

[7] D. Kim, Ki-Sung and Yu, A Scalable Hybrid Overlay Multicast Adopting Host Group Model for Subnet-Dense Receivers International Journal of Computer Science and Network Security(IJCSNS 2007).

[8] B Wang, J.C. Hou, "Multicast Routing and Its QoS Extension :Problems,Algorithms and Protocols", IEEE Network January/February 2000, pp. 22-36

[9] R. Guerin et al., "QoS routing mechanism and OSPF extensions" Internet draft – 1998 - http://tools.ietf.org/html/rfc2676.html

[10] C. Tseng, C. Chen, "Multicast Extensions to QOSPF", Internet and Multimedia Systems and Applications, EuroIMSA 2005, Grindelwald, Switzerland, pp. 370-376, February 21-23, 2005 IASTED/ACTA Press, 2005.

[11] B. Fenner, M. Handley, H. Holbrook and I. Kouvelas, RFC 4601 - Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification (Revised)

[12] R. Iorga, E. Boroci, and S. Obreja, "QoS Enabled IP multicast in a Multi-domain Multimedia Distribution System", Telecomunicatii 2008

[13] E. Borcoci, A. Pinto, A Mehaoua, L. Fang and N. Wang "Resource Management and Signalling Architecture of a Hybrid Multicast Service for Multimedia Distribution", Springer, Lecture Notes in Computer Science Volume 5274, pp. 39-51, 2008

[14] E. Guainella, C. Sansone, B. Angelini and N. Angelini, "The DAIDALOS approach to IP multicast, Inter-domain QoS control", http://www.eurasip.org/Proceedings/Ext/IST05/papers/524.pdf (last accessed March 4, 2013)

[15] A. Neto, E. Cerqueira, A. Rissato and E. Monteiro, "A Resource Reservation Protocol Supporting QoS-aware Multicast Trees for Next Generation Networks, Proc. of Computers and Communications, ISCC 2007. 12th IEEE Symposium on, pp. 707 – 714, 2007

[16] N.M. Mosharaf Kabir Chowdhury and R. Boutaba, A survey of network virtualization, Computer Networks, Volume 54, Issue 5, 8, pp. 862-876, April 2010, ISSN 1389-1286, 10.1016/j.comnet.2009.10.017.

[17] T. Anderson et al, "Overcoming the Internet Impasse through Virtualization", Computer, vol. 38, no. 4, pp. 34–41, Apr. 2005.

[18] N. Wang; et al; , "A two-dimensional architecture for end-to-end resource management in virtual network environments," Network, IEEE , vol.26, no.5, pp. 8-14, September 2012.

[19] R. Miruta, E. Borcoci and E. Pallis - Planning and Provisioning of Virtual Content Aware Networks over IP Infrastructures – 2012 – TEMU, IEEE Conference Publications pp. 118-123 http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6294701&contentType=Conference+Publications&sortType%3Dasc_p_Sequence%26filter%3DAND%28p_IS_Number%3A6294693%29%26rowsPerPage%3D50

[20] H. Holbrook, B. Cain (August 2006). RFC 4607: Source-Specific Multicast for IP - http://tools.ietf.org/html/rfc4607

[21] http://www.scilab.org/

[22] http://atoms.scilab.org/toolboxes/NARVAL/1.0

[23] http://www.xmlsoft.org/

 ISBN: 978-1-61208-263-9