A NEW ARCHITECTURE FOR SPARSE-MODE INTER-DOMAIN MULTICASTING

Jean-Jacques Pansiot, Abdelghani Alloui, Thomas Noel, Dominique Grad
Université Louis Pasteur – LSIIT, Computer Science Department
Boulevard S. Brant, F-67400 Illkirch, France,
Email :{pansiot, alloui, noel, grad} @dpt-info.u-strasbg.fr

ABSTRACT

Multicast routing, especially in the inter-domain case, is a complex problem involving many aspects such as scalability, routing policies, access control, and robustness. We propose a new multicasting architecture based on several features. A new level of addressing called logical addressing. It allows on the one hand providing an efficient multicast service for *multihomed* hosts and mobiles, and on the other hand to construct reduced bi-directional trees consisting of logical edges or tunnels connecting nodes playing an active role in the tree. Also a group manager controls tree membership and multicast trees are constructed from root to leaves contrary to most current solutions. After reviewing requirements for inter-domain multicast routing, we show how this architecture may help to solve some problems of inter-domain multicast routing, in terms of policy, scalability, and robustness.

1. INTRODUCTION

S. Deering's work [4] has lead to the definition of a multicasting model for the Internet, called the Host Group Model. This model extends the classical IP unicast model to provide multipoint-to-multipoint communications. Later, several multicast routing protocols have been defined and implemented [17,11,15] and many multicast applications have been developed such as audio and video tools, shared whiteboard… Some of these protocols and applications have been tested either on a limited domain, or on the Mbone, a virtual multicast network built on top of the Internet.

An attractive feature of the Host Group Model, is anonymity of receivers: senders don't need to know where receivers are located, nor who they are. This property is useful in the routing process, and is an implicit consequence of the identification of a set of destinations by a unique identifier. But without some access control this is likely to lead to situations where a multicast traffic is forwarded to networks where there is no authorised receiver, resulting in a wasting of network resources. We think that multicast routing protocols must implement some access control on group receivers. Likewise, the traditional model "senders just send" should be coupled with some (implicit) access control on routers, to prevent any host to send multicast packets to any group.

In the current Internet, each host is identified by one or more IP address used for routing, meaning that they must contain enough information for a router to forward packets. The routing process itself is executed independently of any active data communication, by exchanging routing information.

The association between an object and an address is becoming less and less fixed. (i) Mobile hosts may be connected through different networks at different times with different network addresses. Mobile networks are also considered. (ii) Hosts or networks may be multi-homed, that is they may be reachable by several paths, some of which may fail. (iii) Addresses may change because they reflect the inter-network topology. This is the case for IPv4 addresses with the use of CIDR. This will become even more common with IPv6, where renumbering is a feature.

Now, supporting these evolutions in the current IP multicast service is not provided efficiently.

Many people think that multicast applications could rapidly use a big share of Internet traffic if a general inter-domain multicast infrastructure were available. What is currently missing is an inter-domain multicast routing protocol that will scale to the whole Internet. Some propositions are already underway, such as BGMP [9], and newer proposals have been made such as QoSMIC [7], Simple Multicast [13] or EXPRESS [8].

In this paper we propose a new multicasting architecture called LAR (Logical Addressing and Routing) with the following goals:

- Some of the burden of multicasting should be supported by hosts whenever possible, since most multicast communications are triggered by applications on hosts. Note that this is different from unicast routing, which is rather statically set up, independently of user applications. In particular, group membership should be controlled by a manager, together with some group policy: is the multicast tree bi-directional or not, are non-member sources authorised?
- As few routers as possible should be involved in a multicast tree, and as few trees as possible should be constructed for a group. This leads to consider reduced trees, and filters to implement sub-trees.
- Multicast trees should be as independent as possible of the underlying network layer, when routing changes, hosts move...This leads to define a separate logical addressing level above the network level.

In section 2 we present a survey of the current IP multicast service, dealing with the Host Group Model, and multicast routing protocols. We also present some limitations of the current service.
In section 3 we present the main features of our architecture: logical addressing, group operations, tree construction and maintenance, *multihoming* and mobility. In section 4 we address more specifically the inter-domain aspects: policy routing, scalability in terms of size and number of groups, and finally in section 5 we summarise what has already been done and further work that should be done.

2. BACKGROUND AND MOTIVATIONS

2.1. The Host Group Model

The current IP multicast model, called the Host Group Model, was defined by Deering a decade ago [4]. Under this model, a set of destinations of an IP packet is called a group and is identified by a multicast address. Multicast addresses are taken in the same addressing space as unicast addresses. To send a packet to a set of destinations, a sender simply places a multicast address in the address destination field of the IP packet. Receivers must join a group to receive multicast packets sent to the group. In addition a multicast routing protocol must be run on routers, to ensure that packets sent by any sender are forwarded to all group members.

2.1.1. Multicast routing protocols

Several multicast routing protocols have been defined. Some are based on a limited flooding technique, such as DVMRP [17] or PIM-DM [3]. These dense mode protocols are suitable for use in a limited domain, with a high density of group members. Link-state multicast protocols such as MOSPF [11] also have a tree for each active source, but there is no flooding of the whole domain. The tree is also limited to a single domain, due to the cost of computing a whole tree in each node for each pair (source, group).
Then protocols for sparse groups such as CBT [1] or PIM-SM [5] have been proposed. They rely mainly on a shared tree, constructed by explicit join requests sent to a specific node (core or rendezvous point: RP). These protocols need less state since there is only one tree per group. Also there is no flooding. One difficulty with these protocols is the way the RP is defined and used. In the current proposals, each router must know the RP of each group in order to forward join requests. An election mechanism based on a BootStrap Router (BSR) has been defined but it does not scale well. Therefore, all these protocols are intra-domain routing protocols.
Newer protocols have been proposed for inter-domain routing. BGMP [9] is a routing protocol that constructs an inter-domain tree that interconnects intra-domain trees, constructed from intra-domain routing protocols. Each group (multicast address) has a root domain, and the inter-domain tree construction is quite similar to the construction of a sparse mode intra-domain tree. Ranges of addresses are allocated to domains, through the MASC [6] protocol, with the intent that groups created from inside a domain use multicast addresses from these ranges. This means that the root domain is likely to contain group members and is a logical location for the root. These ranges of addresses are then advertised through an inter-domain unicast routing protocol such as BGP4+ [2]. When a domain wishes to join a group, a lookup in the group routing information base (G RIB) generated by BGP4+ indicates which is the root domain for the group, and the next hop towards this domain.

2.2. Open issues

2.2.1. Access control management

Neither current multicast model nor current implemented multicast routing protocols provide any mechanism to limit the access to a multicast group for sender and receiver.
A malicious or careless sender could send a high rate of data to a group degrading reception quality of the group members and wasting network resources.
Also, a malicious or careless user can join many groups implying high data traffic in the LAN and access network. Finally, a host could randomly join arbitrary multicast addresses, even if no corresponding group exists, creating useless state information in routers and useless signalling.

2.2.2. *Multicast tree construction*

Sparse-mode protocols (PIM-SM and CBT) rely on an explicit join technique. Upon reception of a group membership report, a local multicast router, generates a protocol-specific join request. This request is forwarded hop-by-hop towards a particular router (called Rendezvous Point or Core router), installing a routing state on intermediate routers. This way, multicast trees constructed are **reverse shortest path trees**. In the case of networks with asymmetrical links, data flowing from the root towards the members will not take the optimal path. In addition, any router between the new member and the first on-tree node of the tree will be added to the existing multicast tree, and has to maintain a multicast forwarding state for this group, even if it forwards data to only one interface. The number of these routers depends, obviously of several parameters, mainly network topology, Core/RP placement, and density of group members. However, in the case of sparse groups, one can expect this number to be large particularly on the backbone [12].

2.2.3. *Multihomed hosts and mobiles*

The current multicast service may not behave correctly in the presence of multihomed hosts. Indeed, if a *multihomed* host sends to a group, it must choose one of its interfaces for initial transmission of data. Routers have in charge to forward multicast data to other networks. Particularly, to the other networks attached to the host. This may produce a non-optimal routing. Moreover, if the *multihomed* host switches transmission of data to another interface (for example because of failure of the initial interface), applications using network addresses to identify senders, will likely have an abnormal behaviour.

A similar problem may arise with mobiles sending to a multicast group. If a mobile using Mobile IP [18] wishes to continue to take part into a group communication, after a move, it has two possibilities. (i) The mobile uses its permanent address and packets are tunnelled between its permanent network and its current network, implying a waste of resources (ii) the mobile uses its current address and it must rejoin the group each time it moves. A problem similar to the *multihomed* host problem may arise.

3. LAR: A LOGICAL ADDRESSING AND ROUTING ARCHITECTURE

3.1. Overview

LAR is intended to provide efficient multipoint communications. It uses two types of addresses: network (or routing) addresses, such as current IP unicast addresses are used in the routing process, and logical addresses (called LAR addresses) are used to uniquely identify logical objects. Logical objects considered are (i) **LAR node**: a host or router implementing LAR. In addition to one or more network addresses, each node has a unique LAR address. The LAR address remains fixed through renumbering or mobility. (ii) **LAR group**: a set of LAR nodes having a common interest, and implied in one or several communications. The LAR address allocated to a group is derived from the host (creator) LAR address by adding a suffix. (iii) **LAR tree**: a distributed structure used to forward data to (a subset of) the members of a group. A LAR group may use several trees. LAR tree addresses are derived from LAR group addresses by adding a suffix.

Each LAR group has a manager, which deals with the control of the group. By default the creator of a group is also its manager. However it can delegate this task to another host. The group's manager publishes the address of the group and the address of its manager using the Domain Name Service (DNS), or other means.

To join a group the new member sends a join request towards the group manager, which, after some access control, accepts or rejects the join request. If the access control succeeds, a join acknowledgement is sent through the tree root and downwards along the existing LAR tree towards the new member, using the underlying unicast routing table. A new LAR edge connects the existing tree to the new member. Therefore a tree constructed by LAR is a shortest path tree from the root to members, even in topologies with asymmetrical links. Advantages to construct trees from source to receivers have been discussed in [7]. Note also that LAR trees are reduced trees. This means that a logical edge of a LAR tree (a tunnel) may be a multi-hop unicast route at the network level. The number of LAR nodes involved in a LAR tree will be smaller than the number of routers involved in a classical multicast tree connecting the same members, especially for sparse groups. By default, LAR trees are shared bi-directional trees.

3.2. Addressing and naming

Hosts have **logical addresses**. A simple way to allocate logical addresses is for example to derive them from the host's Fully Qualified Domain Name[1]. These addresses are independent of routing, and do not change when network addresses change because of host mobility or network renumbering.

[1] This has two main advantages: the hierarchy for LAR addresses is exactly the same as the hierarchy of names, and it is possible to have inverse queries (from LAR address to name, and from network address to LAR address).

Groups have logical addresses derived from the creator's logical address so there is no need for a protocol to allocate group addresses. Some groups may require several distribution trees (for example, a shared tree and some source-rooted trees). These trees have logical addresses derived from the address of the corresponding group. The default bi-directional shared tree is used for intra-group signalling and as default for sending data.
Since a group and a multicast tree are identified by a logical address, this identifier is not tied to a given root or root domain. Therefore, the root may change during the lifetime of the group.
The two levels of addresses correspond to two levels of header in packets. The first (lower) level is the usual network level, using addresses with routing semantics. Source and destination addresses identify the end points of a logical edge (a tunnel). The second level is a new logical routing (LAR) level. It contains the logical addresses of the sender and of the group of receivers.

3.3. LAR data structures

LAR nodes contain two data structures: (I) **The logical routing table** (LAR table) contains an entry for each active tree in this node. Each entry is of the form $<TA, \{L_1, L_2, L_3...\}>$ where TA is the LAR address of the tree, and each L_i is the logical address of a neighbour in the tree. (II) **The LAR cache table** contains an entry for each LAR host address in use by this node. Each entry is of the form $<LA, \{N_1, N_2, N_3...\}>$ where *LA* is the LAR address of a node, and N_1, N_2... are usable network addresses of this node (possibly sorted by preference).

3.4. **Group operations**

3.4.1. Group creation

A host (the **creator**) creates a group. This host chooses a **manager** for this group. For most simple cases, the manager will just be the creator. The manager then constructs a logical address and a name for the group, and advertises them, for example by a dynamic name server update. The DNS will contain the association (group name, group address, and manager address). The manager then chooses the initial **root** of the tree. For most simple cases, the root is just the manager, but it could be a conveniently located router. In general the root address is not advertised. To increase scalability and reliability, several managers may be set up.

3.4.2. Joining a group or a tree

When a host wishes to join a group (whose name or logical address has been learnt by any mean: mail, web, session directory), it first queries the DNS to learn the group manager. The group manager may also be advertised by the same means. The host may also learn any particular control method for membership.
The host then sends a join request to the manager. After some checking, the manager decides whether the host may join or not. In a positive case, a join acknowledgement is sent to the tree. This acknowledgement will travel along the tree and trigger the creation of a new branch in the tree, connecting the new member. The new member receives the acknowledgement from its neighbour in the tree, together with the logical address of this neighbour. In the same way, a host may join a particular tree of the group.

3.4.3. Leaving a group or a tree

A host wishing to leave a group or tree just sends a *leave* message to its neighbour in the tree. A host may also leave implicitly, for example after a failure. The tree maintenance mechanism will automatically prune this member.

3.5. Tree construction and maintenance

3.5.1. Tree construction

The first node of a tree is the root. When the membership of a new host *M* has been accepted, the manager sends a *join acknowledgement* to the root. This message travels hop-by-hop down the tree, until the point where the unicast route to the new member leaves the tree. Note that there are several cases:
1. At some tree node *A*, the next hop toward *M* is different from the next hop toward any tree node. Then a new edge *(A, M)* is created.
2. At some tree node *A*, the next hop towards *M* is the same as the next hop towards some tree node *B*, but *B* is not on the unicast route towards *M*. This means that the route from *A* to *B* and the route from *A* to *M* are the same up to a router *C*, *C* not yet a LAR node in the tree. In this case, the edge *(A, B)* is split into two edges *(A, C)* and *(C, B)* by adding a new tree node *C*. Then an edge *(C, M)* is created.
In both cases, *M* knows it has been inserted in the tree when it receives the edge creation message. This message contains the logical and network addresses of its neighbour in the tree.

Fig 1 shows an instance of LAR tree, compared to a tree that would be built by a sparse-mode protocol like PIM-SM or CBT.

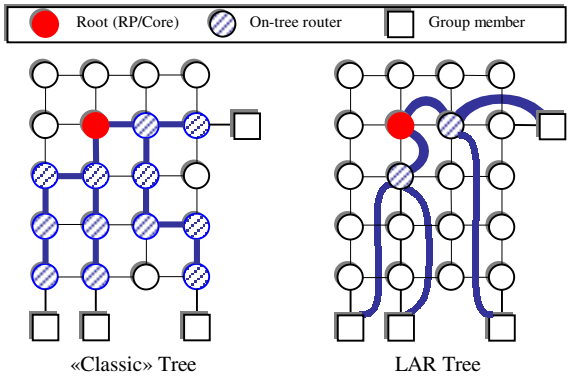

Figure 1. Example of a LAR tree.

3.5.2. Tree maintenance

As in other multicast routing protocols, the root periodically sends *hello* messages down the tree. This allows tree nodes to verify that they are still connected to their upstream neighbour. In the absence of *hello* message for some time, a node sends a *rejoin* message to the manager. The address of the manager may be stored when the tree is constructed, or it may be recovered from the group address by a DNS query. The *rejoin* message will travel downstream from the root in order to graft back the node. While waiting to rejoin the tree, a node continues to send *hello* messages downstream for some time, in order to maintain its own sub-tree.

Symmetrically, nodes send *hello reply* messages to their upstream neighbour. In the absence of *hello reply* message for a specified time, the upstream node may delete the corresponding downstream neighbour.

3.6. Mobility.

In the LAR architecture, a mechanism similar to mobile IP may be used for multicasting: a mobile just updates the LAR cache of its neighbours in multicast trees it belongs to. Each time the set of usable network addresses of a node changes, an update is sent to all LAR neighbours. The duration of this update is about the same as in the unicast case.

3.7. Handling of data packets

3.7.1. Sending data

When a host sends data to a group/tree, two cases are possible. (i) If the host is a member of the tree, then data is forwarded along the tree towards all other members if the tree is bi-directional or the host is the root of the tree. (ii) If the host is not a tree member, then it learns the manager address, and sends data to the manager. Note that since we use two levels of addressing the LAR destination address is the address of the group, and the network address is the address of the manager. The manager may then choose several actions:

- Forward data to the tree (no new encapsulation is needed, just change the network header)
- Discard data (if it is not allowed to send data from outside the group)

When a non-member sender learns the address of the root, it sends data towards the root, using a hop-by-hop option: when data packets reach a tree node for the first time, they are forwarded along the tree, provided it is an open bi-directional tree.

Note that non-member sources are harder to control, and should be avoided when possible. In our architecture, each member may specify a filter (see section 3.7.2). In particular, it is easy to construct send-only branches for sources not wishing to receive any data packet.

3.7.2. Forwarding Data

Consider a LAR packet containing N_S and N_D as source and destination network addresses, and L_S, L_D as source and destination logical addresses. When a node A receives this packet, two cases are possible:

- N_D is not a network address of A. Then A is not the destination of the packet, and it is forwarded towards N_D, without any action at the LAR layer: this is just usual routing at the network layer.
- N_D is one of A's network addresses. This means that A is the end point of the LAR edge. The LAR layer then processes the packet: An entry for L_D is searched in the logical routing table. If one is found, the list L_1, L_2, L_3... of neighbours in the tree is retrieved. N_S is used to determine if the sending node is a neighbour in the tree. If the sender is a neighbour, the packet is forwarded to all other neighbours. The new network source address is a network address of A, and the new network destination address is a network address of the neighbour, as found in the LAR cache table. If the sending node is not a neighbour, and the group is not open the packet is discarded. Otherwise it is propagated to all neighbours in the tree as previously. In all cases, the LAR header is not modified.

We propose to associate some flags to each tree identifier, specifying how to handle data. This allows implementing per tree policies. For example we consider (i) ***uni-directional*** flag: indicates that only data coming from the root may be forwarded. If the tree is bi-directional, data coming from any neighbour in the tree is forwarded to all other neighbours, according to possible filtering. (ii) ***Open tree*** flag: indicates that data coming from outside the tree may be accepted, and be forwarded to all neighbour nodes.

In addition we propose to associate a mask to each (outgoing) edge. The *hello* and *hello reply* messages

sent by a node to its neighbours may contain a *mask* field. A node aggregates *mask*s (by ORing them) it has received and propagates them to other neighbours, using the same messages. Similarly, data packets have a *subtree* field. A data packet is sent to a neighbour only if the AND of the outgoing edge *mask* and its *subtree* field is not all zero. The semantics of the *subtree* and *mask* fields is left to the application level: the sender chooses the *subtree* value, and the receiver chooses the *mask* value. Signalling messages are not subject to filtering. Possible applications of this mechanism are: (i) different sources may use different bits in the *subtree* field allowing source filtering in a shared tree. (ii) A source may send different layers of a hierarchically encoded video in different flows allowing heterogeneous receivers with only one tree. (iii) A host wishing to send data but not to receive data may become a member and set a null mask. This allows bi-directional trees while avoiding having non-member sources. In contrast most current proposals need several trees and non-member sources.

4. LAR FOR INTER-DOMAIN MULTICAST ROUTING

The key problems for an inter-domain protocol are mainly scaling and policy. Some work has been initiated on this subject [10] [15], with an emphasis on the issues that block the deployment of current multicast routing protocols on a large scale.

An inter-domain multicast routing protocol must work properly on a large scale. Thus, it is crucial that all its structures and mechanisms be scalable. A good parameter illustrating this important property is the size of routing tables. Indeed, in an inter-domain context (many groups, with a huge number of sources) the protocol must minimise the size of the states maintained by routers. Particularly, it is not conceivable to maintain a state for each source. Another parameter that must scale is the amount of control traffic exchanged between routers.

Traffic concentration is also an important parameter. The lack of a load sharing mechanism may lead to situations where some inter-domain links are saturated with multicast traffic, whereas other links are underused. Thus, an inter-domain multicast routing protocol must enable some form of load sharing, such as load balancing between equal cost paths.

Multicast routing between autonomous systems must be subject to control in the same way as unicast routing is. An autonomous system needs an authorisation (agreement) of a nearby autonomous system before using its resources for relaying traffic. An inter-domain multicast protocol must take into account policy constraints and hence offer a policy model.

A first problem is the difference between unicast and multicast topologies. This can be the result of a partial deployment of multicast, or because of different policies for forwarding unicast and multicast traffic. A related issue is what is known as “The third party dependency problem”. Whenever possible, a communication between two domains (AS) should not depend on the resources of a third domain. This problem may arise with protocols which use shared trees and impose that the multicast data must initially travel through the root before reaching group members.

4.1. Interaction with unicast inter-domain routing

Unicast inter-domain routing is based on domains (or Autonomous Systems: AS), connected through Border Routers (BR). These routers exchange routing information by an inter-domain routing protocol, mainly BGP4. This protocol advertises AS-paths to networks that may be used by the recipient of the advertisement together with other attributes. The routing policy of a domain is mainly implemented by advertising only a subset of known routes. This policy concerns primarily destinations. For example if domain A advertises destination D towards domain B, it means that packets coming through B with destination D are allowed to transit through A. The notion of routing policy for multicast is more complex. We will consider the following type of policy for a domain A, concerning a destination D, and a neighbour domain B: **Policy 1**: A allows groups originating (whose creator is on the B side) on the B side to have members in destination D. **Policy 2**: A forbids groups originating on the B side to have such members.

BGP4+ is an extension of BGP allowing advertising other types of routes in addition to unicast IPv4 routes. This allows implementing different policies for unicast and multicast. In order for LAR to support policies 1 and 2, we assume that a boolean attribute MM (multicast member) is added to all destinations. Paths to D with MM set are advertised by A towards B if and only if multicast trees originating on the B side are allowed to transit through domain A towards members in D. The management of this attribute should not be a big extension to BGP. Note that we may have two routes to destination D advertised in the same direction: one with MM set, and one with MM reset. Routes received with MM set will be stored in the M-RIB (Multicast Routing Information Base) of the BR.

To comply with multicast routing policy as described above, the tree construction algorithm of section 3.5.1 is modified as follows. While the join acknowledgement message travels from the root to the new member M hop by hop:

- If the current node is not a border router or, if the current node is a border router, and the unicast route and multicast route towards M are the same, apply algorithm of section 3.5.1. That is, the unicast routing table is used.
- If the current node is a border router and the unicast route differs from the multicast route, send the join message along the multicast route to the ingress border router of the next AS. This router will become a LAR node of the tree in order to insure that the logical edge (which is a unicast route) complies with the multicast policy.

Note that in the LAR architecture, it is not necessary to advertise group addresses as in BGMP, since trees are constructed from root to members. In addition, if we consider that most members will usually be receivers only, policy is applied in a consistent way with unicast.

4.2. Scalability issues

Inter-domain multicast brings several scalability issues related to group size and to the number of groups.

4.2.1. Address allocation, root advertisement

Our architecture does not need a specific address allocation protocol, or protocols to advertise root (Core, RP) for trees. This saves both states in routers and bandwidth. The root of a tree is chosen by the group's manager and may be replaced by the same manager. The manager may be dynamically retrieved from the group address via the DNS. Therefore managers, instead of routers, handle several problems with multicasting. The group manager supports a significant part of the load due to a group, which is more scalable when the number of groups becomes large.

4.2.2. Tree state in routers

The amount of state to be maintained in a router involved in a LAR tree is similar to that of other sparse mode protocols (CBT, PIM-SM): it is proportional to the number of neighbours in the tree. However, fewer routers will be involved in a given tree, especially for very sparse groups.

If we consider the example of Figure 1, there are 3 routers involved in the LAR tree, whereas sparse-mode protocols would likely imply 13 routers.

Obviously, if multicast becomes widely used in the Internet, routers may still have a huge number of trees to manage, and tree aggregation is a hard problem [15]. The notion of filtering and subgroups is a first step to avoid the creation of many similar trees (see section 3.7.2).

4.2.3. Very large groups

Our architecture was more specifically designed for sparse groups. One difference between our architecture and other proposals (except for QoSMIC) is that join requests are sent to a manager. This brings more membership control, but could be a single point of failure in case of manager failure and a bottleneck if join requests are sent at a high rate.

We propose the following solution: The initial manager may decide to set up additional managers, either because the group needs high reliability or because the group is expected to become very large. These managers may be geographically distributed. New managers may be added dynamically while the group grows or other managers die. All managers are advertised in the same way. If possible, new members choose the closest manager, for example by anycasting. A join request accepted by a manager triggers a join acknowledgement forwarded to the root. Usually a join acknowledgement will not need much processing power in the root, since it will be forwarded to a child node in the tree.

Another problem with very large groups is the reconnection of whole sub-trees when a node fails. Our notion of logical edge should simplify the grafting of a sub-tree back to a tree. With LAR it should not be necessary to flush a whole sub-tree in case of failure, including root.

5. CONCLUSION AND FURTHER WORK

In this paper we have presented a new multicasting architecture. The main features and their consequences may be summarised as follows

a) Joining a group is done through one or more managers: A first level of access control may be achieved, without any router involvement, and allowing application specific membership policies. Routers don't need any a priori knowledge of groups (address of RP, address of group)

b) Tree construction is done from root to member (contrary to most protocols except for QoSMIC). Routes are evaluated in the direction in which they will be mostly used. In case of asymmetric links, this will give better results than routes based on the reverse path. Policy routing may be implemented very similarly for unicast and multicast. There is no need to advertise multicast addresses, except if there is also a policy on group creators. Changing the root of a tree is possible without rebuilding most of the tree. As a possible drawback, the root could be a bottleneck in the case of a huge number of members joining at the same time.

c) A **logical addressing** layer is added on top of the usual network header. Reduced multicast trees may be constructed, reducing the number of routers involved in a given tree, especially for sparse groups. Multicast

trees are more stable. In many cases, a change in unicast topology will not lead to a change of the logical tree. The traffic disruption will be kept to a minimum. If the unicast level implements some form of load sharing, LAR will implicitly make use of it, since a LAR edge is a unicast route. This will be particularly true of very sparse groups. Efficient multicasting for mobile nodes is easy to implement, in a way consistent with the unicast case. The price is the overhead due to the additional header.

d) Flags and filters are associated to a tree. Defining sub-trees reduces the number of trees, implying globally less state and less signalling. Sender only members may be part of the tree, mostly avoiding the problem of non-member sources. Tree specific policies may be defined, for example trees may be uni or bi-directional and they may be open or not.

To validate the main ideas of the LAR architecture, we have implemented part of it above IPv6, on FreeBSD, including mobile hosts in multicast groups (but not the policy and filtering part). The LAR header is implemented as an IPv6 destination option. LAR addresses are taken as a subset of IPv6 addresses defined by a specific prefix. Applications may use LAR without much change since LAR addresses are compatible with network addresses. Obviously many things remain to be done. Among them: study how to deal efficiently with broadcast mediums, and give a precise specification of the LAR routing protocol to construct a LAR tree, especially the interaction with policy routing. Note that LAR could be run in parallel with other multicast protocol since it uses a different encapsulation.

REFERENCES

[1] A. Ballardie, *Core Based Trees (CBT version 2) Multicast Routing*, http://www.ietf.org/rfc/rfc2189.txt.

[2] T. Bates, R. Chandra, D. Katz, Y. Rekhter, *Multiprotocol Extensions for BGP-4*, http://www.ietf.org/rfc/rfc2283.txt.

[3] S. Deering et al, *Protocol Independent Multicast Version 2 Dense Mode Specification,* http://search.ietf.org/internet-drafts/draft-ietf-pim-sm-v2-new-00.txt

[4] S. Deering, *Multicast Routing in Datagram Internetwork*, PhD thesis, Stanford University, 1991.

[5] D. Estrin et al, *Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification*, http://www.ietf.org/rfc/rfc2362.txt.

[6] D. Estrin et al, *The Multicast Address-Set Claim (MASC) Protocol*, http://search.ietf.org/internet-drafts/draft-ietf-malloc-masc-06.txt

[7] M. Faloutsos, A. Banerjea, R. Pankaj, *QoSMIC: Quality of Service sensitive Multicast Internet protoCol*, ACM SIGCOMM'98, September 1998.

[8] H. Holbrook and D. Cheriton, *IP Multicast Channels: EXPRESS Support for Large-scale Single-source Applications*, ACM SIGCOMM'99, September 1999

[9] S. Kumar et al, *The MASC/BGMP architecture for inter-domain multicast routing*, ACM SIGCOMM'98, September 1998.

[10] D. Meyer, *Some Issues for an inter-domain Multicast Routing Protocol*, work in progress, internet-draft, November 1997.

[11] J. Moy, *Multicast Extensions to OSPF*, RFC 1584, March 1994.

[12] J.-J. Pansiot, and D. Grad, *On Routes and Multicast Trees in the Internet*, ACM Computer Communication Review, Vol. 28, n° 1, 1998.

[13] R. Perlman et al, *Simple Multicast: A Design for Simple, Low-Overhead Multicast*, work in progress, draft-perlman-simple-multicast-02.txt, February 1999.

[14] C. Shields, and J.J. Garcia-Luna-Aceves, *KHIP - A Scalable Protocol for Secure Multicast Routing*, SIGCOMM 99, September 1999.

[15] M. Sola, M. Ohta and T. Maeno, *Scalability of Internet Multicast Protocols*, INET'98, Geneva, Switzerland, July 1998.

[16] D. Thaler, D. Estrin and D. Meyer, *Border Gateway Multicast Protocol (BGMP): Protocol Specification*, http://search.ietf.org/internet-drafts/draft-ietf-bgmp-spec-01.txt.

[17] D. Waitzman, S. Deering and C. Partridge, *Distance-Vector Multicast Routing Protocol*, http://www.ietf.org/rfc/rfc1075.txt.

[18] D. Johnson and C. Perkins, *Mobility Support in IPv6,* http://search.ietf.org/internet-drafts/draft-ietf-mobileip-ipv6-12.txt.