GMAC: An Overlay Multicast Network for Mobile Agents

Pablo Gotthelf*, Marcos Mendoza, Alejandro Zunino*† and Cristian Mateos*†

*ISISTAN Research Institute, UNICEN University
Campus Universitario, Paraje Arroyo Seco,
Tandil (7000), Bs. As., Argentina. Tel/Fax: +54 (2293) 440362
†Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)
Email: azunino@exa.unicen.edu.ar

***Abstract*— Mobile agents, software entities able to migrate autonomously their execution, provide many advantages over traditional models for distributed systems such as client/server or code on demand. Despite these advantages, mobile agent applications are not in widespread use, because most platforms for mobile agents rely on group communication services, among other factors. IP multicast was supposed to provide group communication services in an efficient and scalable way. Nevertheless, IP multicast has failed to deliver its promises mainly due to its difficult deployment.**

This work proposes GMAC, an overlay multicast network for mobile agents. In particular, we limited the scope of our research to MoviLog, a platform for mobile agents for the WWW, since the platform itself relies on multicast for managing mobility. GMAC provides multicast services in a decentralized and scalable way, where end systems implement all multicast related functionality including membership management and packet forwarding.

Simulations comparing GMAC with other approaches in aspects such as throughput, protocol overhead, resource utilization and group bandwidth show that GMAC is a robust solution for providing multicast services to MoviLog or applications with similar requirements.

***Index Terms*— computer networks, multicast, overlay networks, mobile agents**

I. Introduction

MOBILE agents are software entities able to migrate autonomously their execution in order to achieve their users' goals. A mobile agent can move to the location where a required resource is located. As a consequence the interactions agent-resource are local instead of remote and network usage is reduced. Besides efficient network usage, mobile agents provide advantages [1] such as scalability, reliability and disconnected operations.

Several factors have hindered the achievement of true scalability of mobile agent systems in real settings. One of them is the lack of proper support for communicating large numbers of distributed agents. For example, some agent platforms such as MoviLog [2], require multicast services not only for providing communication among agents, but also for handling mobility. As a consequence, scalable and efficient multicast services are a crucial requirement for the success of mobile agents.

In order to support IP multicast in the Internet, the MBONE (Multicast Backbone) has been built. The MBONE is a virtual network extended across the Internet that allows IP multicast traffic between hosts [3].

Despite the need for multicast services, the usefulness of the MBONE is still limited due to two reasons. First, the MBONE does not reach all Internet users, since neither all routers, nor ISPs (Internet Service Providers) support it. Second, the MBONE uses UDP thus messages could get lost [3].

An alternative for multicast communications that do not require special routers are Overlay multicast networks [4], [3], [5]. Overlay multicast networks, instead of being supported at the network level, are supported by user-level applications. As a consequence, neither special routers nor extra ISP involvement is required.

In order to achieve scalability in mobile agent systems we have developed GMAC (Group Management Agent Cast), an overlay multicast network for mobile agents. In particular, we limited the scope of our research to MoviLog [2], [6], a platform for mobile agents for the WWW. MoviLog is an interesting domain for multicast since the platform itself relies on multicast for executing mobile agents and managing mobility.

The rest of this work is structured as follows. The next section describes the most relevant related work. GMAC is described in Section III. Experimental results and comparisons with other approaches are reported in Section IV. Finally, the paper concludes in Section V.

II. Related Work

Due to the limited adoption of the MBONE [7] and the increasingly number of applications requiring multicast services in the Internet, several alternatives have been proposed. Most of them provide multicast services by using overlay structures, thus there is no need to modify the network level of the operating system or use special network routers.

Some of the most relevant approaches for supporting multicast services are:

- ALMI [7]: creates a MST (Minimum Spanning Tree) as an overlay structure. It depends on a centralized component to generate and maintain this structure thus it is restricted to small groups.
- End System Multicast [4] improves ALMI by achieving decentralization, though it is still restricted to small groups because each group member is required to maintain a list of the other members.
- REUNITE [8], Overcast [9] and Scattercast [10] support diffusion of information by disseminating servers across the Internet.

Each one of these alternatives provide multicast support for different types of requirements. REUNITE, Overcast and Scattercast are best suited for communications with a single sender and multiple receivers, whereas mobile agents require communications with multiple senders. Moreover, these approaches rely on special routers spread across the Internet thus they are hard to deploy. On the other hand, ALMI and End System Multicast allow all group members to send data. Both approaches support diffusion groups by adding functionality to clients, but are restricted to small groups.

Many mobile agent platforms rely on group communications for supporting interactions among agents. MoviLog [2], [6] goes a step further by requiring multicast for managing mobility. The idea is that each host capable of executing mobile agents has to announce its available resources (code, data or services). In this way, MoviLog is capable of automatically migrating agents based on their resource needs. Therefore it is desirable to support multicast groups as large as possible for achieving scalability.

All in all, since none of the previous approaches provide enough scalability, reliability, support for multiple senders and easy deployability, we designed GMAC to cope with the requirements imposed by MoviLog.

III. GMAC

GMAC is an application level multicast infrastructure. GMAC uses a binary tree as an overlay structure where each node of the tree corresponds to a host or application belonging to a group and the links between them are unicast connections.

GMAC was developed to provide multicast services based on the following characteristics and requirements of the MoviLog platform:

- *Burst transmission:* Group members transmit for short periods of time (no data streaming).
- *Little volatility:* Members are supposed to use the service for long periods of time.
- *Solidarity among group members*: each group member is interested in delivering all messages it gets to all the other members, even when these mesages are not originated by itself.
- *Homogeneous groups:* members have similar characteristics and behavior, so that connection resources and communication demands do not differ too much from one another.
- *Unknown Topology:* In contrast with other kinds of networks, it is very difficult to determine or take advantage of the Internet topology.
- *Connection restricted clients:* Multicast services to hosts behind Network Address Translators (NAT) or firewalls should not be denied, but they may receive a degraded service.

In order to achieve multicast functionality, each host in a group transmits data only to their neighbors. These, in turn, retransmit received data in the same manner. In this way, GMAC relieves the transmitting host. Furthermore, members with better connection capabilities are not overloaded or misused, since every member retransmits data to other two group members at most.

The whole functionality of GMAC is implemented by each node in a decentralized manner, as the responsibilities for message delivery, tree building and recovery are distributed among group members.

Some of the properties of GMAC can be appreciated by comparing it with sequential unicast, where each group member has to send a copy of the message to all the other group members. Assuming that each unicast link takes one second to transmit a message, the total time for sending a message is $n-1$ seconds, where n is the number of group members. On the other hand, the time required by GMAC is at most $(\lceil \log_2 n \rceil - 1) * 3 - 1$ seconds.

A. GMAC Components

GMAC is composed of two classes of components: group members and a GMAC registry. A group member is identified by an IP address and a port number, which are used by the rest of the participants to initiate their unicast connections. The registry is used by hosts willing to join a group to obtain the address of the group root.

As shown in Figure 1, nodes are connected by two unicast links: one for control messages and one for data. Neverthless a single link schema is also possible.

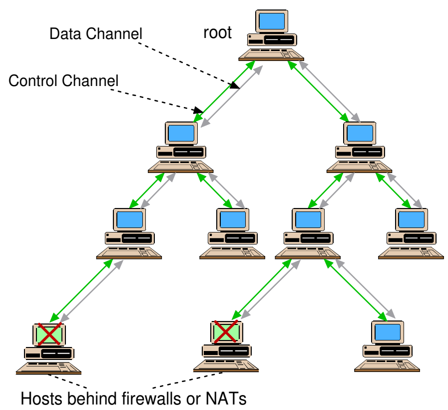


Fig. 1. GMAC tree

GMAC also supports hosts with connectivity restrictions. Two restricted hosts are not able to communicate to each other, since none of them accept incoming connections. GMAC only supports this kind of hosts as leaves of the tree, restricting its number to at most $n/2+1$ where n is the total number of group members.

The GMAC registry (Figure 2) has two responsibilities: attending join requests by sending the root address and port when the correct group name and password are given, and providing a mechanism for replacing a group root in case it fails or leaves the group. This component can be implemented as an independent application. The only requirement is that it has to be globally accessible. The registry is implemented as a cluster of redundant servers thus reliability is not compromised.

B. Joining a Group

The general idea to achieve decentralization is that when a node receives a join request, it incorporates the requesting node as a child. If the node already has two children, it will delegate the join request to its least weighted child, this is, with the smallest subtree. Therefore, a host willing to join a group will descend into the tree until it is inserted as a leaf. As the overlay structure is a binary tree, a host joining a group will have to traverse at most $\log_2 n$ nodes, being n the total number of nodes in the tree.

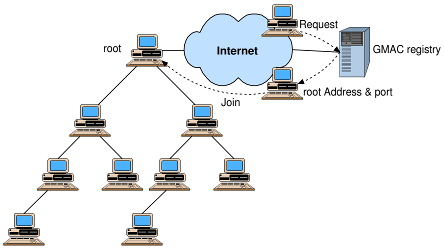


Fig. 2. GMAC registry

To reduce reconnections a more complex heuristic was used in GMAC. The idea is that each host in the tree knows where the next connection point downwards in the tree is. In addition, GMAC allows hosts with restricted connections as leaves by using tree rotations. Therefore, restricted hosts are moved downward the tree when non restricted hosts join the group.

C. Failure recovery

When a node fails or leaves a group, the tree must be restructured in order to continue providing multicast support to the rest of the group. This reorganization is done in a decentralized way by the other members as follows:

- a parent node, which had the failing node as its child, just closes the connections to it, updates its state information and sends it to its own parent.
- children nodes, which had the failing node as parent, must reconnect to the tree by sending a connection request to the root. In case the root has failed, one of them will succeed when claiming the GMAC registry to become the root.

For the non adjacent nodes downward the failing one, reorganization is transparent, as they will be reconnected together with the children nodes.

In this way, a node failure is handled by reconnecting its children, which is accomplished in a decentralized manner. In addition, tree reorganization involves at most two reconnections, thus its computational cost is logarithmic over the number of nodes.

IV. Experimental Results

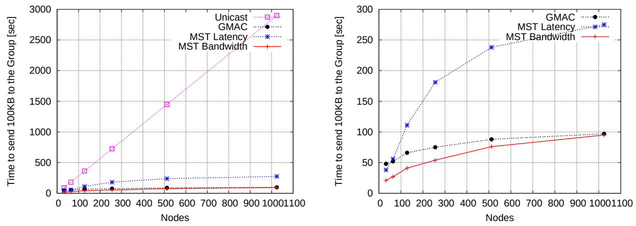
The alternatives described in Section II provide multicast support to applications with particular multicast requirements. Solutions involving broadcasting with a single sender try to maximize throughput, while conferencing oriented applications require low latency as well, though they may allow graceful degradation.

GMAC requirement is that every member gets messages as soon as possible and without data loss. Moreover, no minimum group bandwidth is required. The main GMAC features are decentralization and scalability, while transmission requirements are not considered critical.

The following metrics where used for evaluating GMAC:

- *Latency:* is the maximum delay, from the application point of view, that exists between a sender and the receivers. In a distribution tree latency is the longest path between any pair of nodes.
- *Bandwidth:* is the maximum throughput, from a receiving application point of view, that a group can provide. In a distribution tree, nodes with less bandwidth may act as bottlenecks, limiting the overall group bandwidth.
- *Resource utilization:* is the amount of network resources consumed by the process of data delivery to all receivers. At application level it is generally assumed that more latency implies more cost.
- *Protocol overhead:* this metric takes into account the network traffic that do not contain data, including control messages required to build and maintain the overlay structure.

Several simulations were made in order to evaluate GMAC with sequential unicast and two Minimum Spanning Trees (MSTs), one maximizing bandwidth and the other minimizing latency.



(a) Including unicast alternative

(b) Without unicast alternative

Fig. 3. Time in seconds to send a 100KB message

Figure 3 shows the time in seconds needed by a random node to send 100KB to all the other group members. It can be seen (Figure 3(a)) that the alternative of using only unicast is not viable when groups are large. Figure 3(b) shows the results for GMAC and the two MST alternatives. The latency MST approach has poor performance because it does not take into account bandwidth metrics. On the other hand GMAC scales well.

Figure 4 shows the results for resource utilization. The Latency MST is considered as the one which better resource utilization achieves (it is the approach used by the MBONE). ALMI relies on this metric to build its overlay tree. End System Multicast, in contrast, suggests that this may not be the best choice, because the shortest path is not always the fastest one. Furthermore, the assumption that latency represents distance is not necessarily true. As a consequence, it is very difficult to achieve good network resource utilization with overlay approaches.

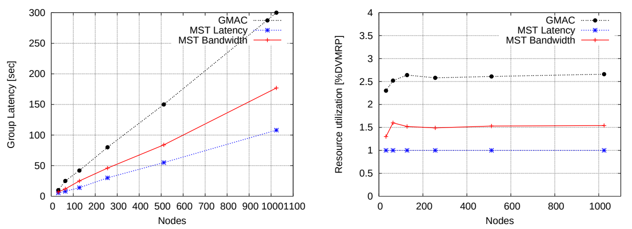


Fig. 4. Resource utilization

Figure 5 shows the group bandwidth that can be achieved with the different approaches. This metric is important for applications requiring audio/video streaming, which is not the case. In distribution trees, less capable nodes may act as bottlenecks, limiting the group to their bandwidth. In GMAC, bottleneck nodes may retransmit messages, thus the overall group bandwidth would be restricted to half of the less capable host bandwidth. It is worth noting that GMAC was developed for homogeneous groups, thus bottleneck nodes are less likely to appear.

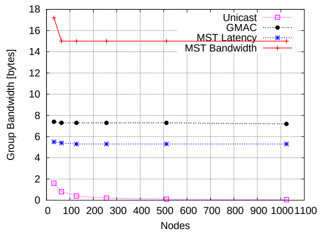


Fig. 5. Maximum group Bandwidth

The protocol overhead metric considers network traffic that does not represent useful data. In GMAC the information required to build and maintain the binary tree is minimum, because this information is transmitted only when a node fails or joins a group. In contrast, alternatives which use some kind of optimization introduce considerable overhead. This is because many measurements are required for building and maintaining these optimized structures.

Another important issue is the possibility of failure of some node. When this occurs in an overlay structure, members who depend on the failing host, like their subtrees, can get disconnected. As explained in Subsection C, GMAC quickly recovers from a node failure by reincorporating immediately the orphan nodes and their subtrees. However, optimized overlay structures recovery such as ALMI is much harder, because a host failure may trigger the reoptimization of the whole structure. In addition, it may require complex heuristics in order to keep the diffusion group working until a new optimization is performed. This is one of the issues that renders approaches based on optimized overlay structures not scalable.

V. Conclusions

This paper presented GMAC, an overlay multicast support for MoviLog. Using a binary tree as overlay structure, GMAC allows group communication between hosts spread across the Internet. The entire functionality of GMAC is implemented in each host at application level in a decentralized way, thus the responsibilities for message delivery, tree building and recovery are distributed among group members. Therefore, special routers are not required, achieving great robustness and easy deployment. Another virtue of GMAC is that it supports large groups, in opposition to most related approaches.

GMAC goal is to provide multicast services to non critical burst transmitting applications, where the key idea is that overloaded hosts are relief by delegating their transmission responsibilities to their neighbors. In addition, GMAC can be widely deployed, because it does not rely on special routers and its application program interface is compatible with the well known Java datagram multicast service, so that GMAC can be easily used by existing Java applications. Furthermore, GMAC only assumes unicast as subjacent service, allowing any host in the Internet to use it, even those with connectivity restrictions, such as hosts behind NATs or firewalls.

GMAC is an application level solution. As a consequence, network resource utilization is rather bad. This is one of the main drawbacks of GMAC, however this is also present in most overlay based approaches.

Although GMAC was not developed to attend critic transmission requirements, experimental results show that it is an effective choice for groups of considerable size. In contrast, with approaches based on optimized structures, problems such as increased complexity of generating these structures, management overhead and probability of failure arise.

In GMAC, these problems are not present, as the complexity for generating its structure is not affected by the group size. In addition, network overhead is minimal and failure recovery is fast and effective. Furthermore, GMAC obtained very good experimental results. Some of them can be explained because GMAC nodes use all its resources to transmit to their parents first, as they know that this is the longest path a message will have to traverse, whereas other alternatives do not use this kind of priority scheme.

All in all, GMAC proposes a new way of providing multicast services at application level opening new research directions, such as data recovery, optimization, group publication policies and security.

References

[1] D. B. Lange and M. Oshima, "Seven good reasons for mobile agents," *Communications of the ACM*, vol. 42, no. 3, pp. 88–89, Mar. 1999.

[2] A. Zunino, C. Mateos, and M. Campo, "Reactive mobility by failure: When fail means move," *Information Systems Frontiers. Special Issue on Mobile Computing and Communications*, vol. 7, no. 2, pp. 141–154, 2005, ISSN 1387-3326.

[3] H. Eriksson, "MBONE: the multicast backbone," *Communications of the ACM*, vol. 37, no. 8, pp. 54–60, Aug. 1994.

[4] Y.-H. Chu, S. G. Rao, and H. Zhang, "A case for end system multicast," in *Measurement and Modeling of Computer Systems*, pp. 1–12, 2000. [Online]. Available: citeseer.ist.psu.edu/chu01case.html

[5] Y. Chawathe, S. McCanne, and E. Brewer, "RMX: Reliable multicast for heterogeneous networks," in *Proceedings of the 2000 IEEE Computer and Communications Societies Conference on Computer Communications (INFOCOM-00)*, pp. 795–804. Los Alamitos: IEEE, Mar. 26–30 2000.

[6] A. Zunino, M. Campo, and C. Mateos, "MoviLog: A platform for prolog-based strong mobile agents on the WWW," *Inteligencia Artificial, Revista Iberoamericana de I.A.*, vol. 4, no. 21, pp. 83–92, 2003, ISSN 1337-3601.

[7] D. Pendarakis, S. Shi, D. Verma, and M. Waldvogel, "ALMI: An application level multicast infrastructure," in *Proceedings of the 3rd USENIX Symposium on Internet Technologies and Systems (USITS '01), March 26–28, 2001, San Francisco, California, USA*, USENIX, Ed. Berkeley, CA, USA: USENIX, 2001.

[8] I. Stoica, T. S. E. Ng, and H. Zhang, "REUNITE: A recursive unicast approach to multicast," in *Proceedings of the 2000 IEEE Computer and Communications Societies Conference on Computer Communications (INFOCOM-00)*, pp. 1644–1653. Los Alamitos: IEEE, Mar. 26–30 2000.

[9] J. Jannotti, D. K. Gifford, K. L. Johnson, M. F. Kaashoek, and J. W. O'Toole, Jr., "Overcast: Reliable multicasting with an overlay network," in *Proceedings of the Fourth Symposium on Operating Systems Design and Implementation (OSDI 2000), October 23–25, 2000, San Diego, California, USA*, USENIX, Ed. Berkeley, CA, USA: USENIX, 2000.

[10] Y. D. Chawathe, "Scattercast: an architecture for internet broadcast distribution as an infrastructure service," Ph.D. dissertation, University of California, Berkeley, Spring, 2000.