

# A Survey of Application-level Multicast Group Communication and a Proposal for Intelligent Reflectors

Janine Kniess<sup>1,2</sup>, Célio Vinicius N. Albuquerque<sup>1</sup>

1 Computer Science Institute, Federal Fluminense University (UFF)  
Passos da Pátria Street, 156, São Domingos, Niterói – RJ, Brasil - Zip  
Code 24.210- 240

{jkniess,celio}@ic.uff.br,

WWW home page: <http://www.ic.uff.br>

2 Computer Science Department, State University (UDESC)  
Campus Universitário Prof. Avelino Marcante, Bom Retiro, Joinville-SC,  
Brasil - Zip Code 89223-100

WWW home page: <http://www.joinville.udesc.br>

**Abstract.** The article proposes the state of the art in techniques and models designed to solve issues in application-layer IP multicast. Specifically, the issue of group communication for distributing video flows using reflectors to participants at a videoconference is considered. A proposed solution is introduced, involving an algorithm capable of dynamically discovering the appropriate reflector to meet a given client's needs. Simulations results show the efficiency of using reflectors in this type of application.

## 1 Introduction

As computer and network technologies have evolved, conditions have been created by which the transfer of multimedia data in real time can be supported. This means that the development of advanced applications, such as television and videoconferencing via the Internet, has become feasible.

Internet videoconferencing and TV belong to the category of group communication unlike others that are consisting of point-to-point conversations or file transfers. These new application often have a handful of sources and large number of receivers, and require large-scale video transmission. An efficient multicast service is required [1], since the use of multiple unicast channels is unfeasible in terms of bandwidth, and server processing, since the media, which are mainly audio and video, demand a variety of resources from the communications system.

In traditional group communication, each node must have access to the native multicast service. Data travels over a data link once, and is replicated by intermediate routers in order to serve those clients who wish to receive the content. This approach is not scalable as it frequently floods the network with new trees of content distribution. Additionally, network-layer IP multicast is a restricted technology that is not yet widely deployed. Internet Service Providers (ISPs) are reluctant to deploy it because it is difficult to manage and because of the security issues involved with such an approach.

Recently, application-layer IP multicast has been used over the Internet to support multimedia applications. In this case, the end systems deploy and implement all group communication functions. Application-layer multicast improves the efficiency of one-to-many and many-to-many multi-destination communication and makes it possible for a huge reduction to be achieved in bandwidth requirements, proving advantageous in transmitting continuous media, especially high quality video.

Nevertheless, the features made available by application-layer multicast services are yet insufficient for the scaleable distribution of high quality multimedia data to a large number of clients [2]. In order to cope with the limitations in distributing advanced multimedia video applications, especially videoconferencing, media reflectors have been used.

In traditional video distribution systems, reflectors serve as proxies for the virtual multicast (or overlay) network that interconnect the participants in a multimedia conference located in different local networks. Reflectors allow for a more efficient use of the available bandwidth since packets (video, audio and data) are simply sent along data tunnels that link up two reflectors. Additionally, the topology of the reflector network takes into account the geography and bandwidth available at each link of the network, optimizing their available paths.

In view of the benefits gained by using reflectors for distributing multimedia data to multicast groups on the Internet, this article introduces an algorithm capable of dynamically encountering an adjusted configuration for the more appropriate reflector for one or more participants in a multicast group that wish to receive video content and the deployment of mobile agents located in machines close to the clients, which would serve temporarily as reflectors.

Experiments with reflectors performed in simulations are presented with the goal of demonstrating the reduction in the average delay in the delivery of video data packets as the number of reflectors is increased, or as these reflectors become more "intelligent"; i.e. a mobile reflectors replicating the code of the reflector server to a specific point in the network close to the client or group of participating clients. The most recent achievements in the area of application-layer multicast group communication are also presented; i.e. techniques and models.

The paper is organized as follows. Section 2 sets out the techniques for providing group communication services currently available, designed to circumvent the distribution problems encountered in multicast routing. Section 3 presents the main studies related into application-layer IP multicast. In section 4, the deployment of a video broadcasting multimedia application using application-layer multicast is presented. The same section also contains the results obtained in the simulation. Section 5 presents conclusions and future studies.

## 2 Alternative Techniques for Multicast Group Communication Services

The efficiency of a control technique for group communication depends upon its satisfying factors such as robustness, security, scalability, performance and dynamic reconfiguration to discover new transmitters or receivers. The techniques described in this section are designed to provide solutions for the limitations encountered in group communication using IP multicast.

**Unicast/multicast reflectors and a individual data tunnels:** As stated by [3] and [4], in this approach, a reflector acts as a gateway between a network with multicast capacity (e.g. Mbone [5]) and a set of unicast hosts. Each multicast packet in a network is delivered to each unicast host and other unicast hosts, creating tunnels between the reflector and the end hosts. One shortcoming of this model is that it creates hot spots in the network in the proximity of the reflector.

**Permanent data tunnels:** Tunnels are deployed at the routing level and make use of IP encapsulation. [6]. They require access to privileged information to be deployed and are not generally updated by the end host. The tunnels are totally integrated into the multicast routing and provide connectivity between all the possible multicast groups. Mrouter DVMRP [7] is the most widely used solution for this approach.

**Gossiping-based solutions for peer-to-peer communication:** Within the context of overlay networks, the gossiping technique as set out in [8] can be used for data distribution. Each member of the group periodically sends a message containing a list of their neighbors. Each node builds up knowledge about the members of the group by hops. Thus, one node can have complete knowledge about the group. For very large groups, the periodic messages can cause considerable overheads. Additionally, the view that nodes in a large group have of the group is restricted to their neighbors.

**Communication routing services for a specific group:** The main feature here is that the routing services for group communication are established inside one of the routers. The drawback is that this service cannot be distributed on demand by end users. Two models, XCAST [9] and DCM [10], provide solutions for the limited scalability inherent to this technique. AMRoute [11], DMRP [12] and MAODV [13] set out protocols which permit the use of IP multicast in mobile ad hoc networks.

**Automatic Overlay Multicast:** Here, the multicast support for the central routers is shifted to the end systems. The end systems deploy all the functionalities of the group communication. Other models presented in [6, 14, 15, 16], among others, have been developed with this principle as their basis.

### 2.2 Discussions and Prospects

The techniques described above can be used in environments where traditional multicast routing is entirely inappropriate, such as ad hoc networks or in cases where there are a large number of small, dynamic groups. For this reason, many models have been developed based on these techniques. Given the advantages brought by

overlay multicast, which include its simple configuration, flexible implementation and the customization of some attributes, such as data transcoding, error recovery, flow control, scaling, management and security of different messages, this technique has become the subject of much research and is being used as a basis for the development of new models.

In the following section, we present the state-of-the-art in multicast solutions that use overlay multicast. Within the scope of this work, this choice can be justified by the fact that this approach has achieved satisfactory results in video on demand applications, such as videoconferencing.

### 3 Related Work

A considerable number of projects have explored implementing multicast at the application layer. They can be classified into two broad categories: mesh-first (NARADA [15], CANs [17] and SCRIBE [7]) and tree-first protocols (Yoid [18], ALMI [14] and Host Multicast Tree Protocol (HMTP) [19]). In the tree-first protocols, the participants construct a covering tree and organize themselves around the same one. At the same time they keep links of control for some neighboring members of the tree and, use these links of control to reorganize the covering tree. On the other hand, in the mesh-first, the participants use periodic measured who allow detecting the best ways to organize themselves in a mesh through the selection of the best links. A definite time this mesh, applies a guiding algorithm multicasting to determine one or more trees of covering.

Yoid and HMTP defines a distributed tree building protocol between the end-hosts, while ALMI uses a centralized algorithm to create a minimum spanning tree rooted at a designated single source of multicast data distribution. One shortcoming of the model is that the centralized algorithm is entirely responsible for controlling the data path. If the algorithm fails in some way, all the operations related to the group may be put in jeopardy. The Overcast protocol [16] organizes a set of proxies (called Overcast nodes) into a distribution tree rooted at a central source for a single source multicast. In order to obtain specific content, a client is referred to a URL containing the address of the parent of the group, plus the name of the resource. The parent selects the most appropriate content server in the tree and redirects it to the client in a way that is transparent to the user.

The aim of the algorithm in an Overcast tree is to optimize the parent node's bandwidth for all the other nodes. In order to do so, the algorithm includes a new node as far as possible from the parent node, without affecting its bandwidth.

A distributed tree-building protocol is used to create this source specific tree, in a manner similar to Yoid.

NARADA, ALMI and CANs are designed for medium-sized groups. NARADA keeps a complete list of all the group members. NICE [20], HMTP, Overcast, and SCRIBE are designed for large-sized groups. The CANs protocol can be expanded to support large multicast groups. In CANs, heuristics are used to minimize the number of duplicate messages forwarded. This is achieved by the storage in cache of the

message identifications received. Cache which obtained by storing the identifiers of the message received in a cache. Thanks to this, no duplicate message is forwarded. Some projects (CANs) and Pastry [21]) have also addressed the scalability issue in creating application layer overlays. CANs defines a set of end hosts implement a hash table on an Internet-wide scale for storing two member peers ( $i,j$ ). The nodes are distributed into coordinate space, in which each member is the owner of their space. Patry is a self-organizing overlay network of nodes, where logical peer relationships on the overlay are based on matching prefixes of the node identifiers. Scribe is a large-scale event notification infrastructure that leverages the Pastry system to create groups and build efficient application layer multicast paths to the group members for dissemination of data on groups based on publish-subscribe. NICE also chooses overlay peers based on network locality which leads to low stretch end-to-end paths.

We summarize the above as follows: For both NICE and CANs, members maintain constant state for other members, and consequently exchange a constant amount of periodic refreshes messages. This overlay paths for NICE and SCRIBE, have a logarithmic number of application level hops, and path lengths in CANs asymptotically have a larger number of application level hops.

### 3.2 Analysis and Comparison of the Overlay Multicast Protocols Described

The models set out above have the main advantage of not requiring any support from network routers. Thus, overlay networks are particularly attractive for managing multicast group communication, as they improve scalability in terms of the number of concurrent groups, and allow for the creation of a robust communication system in a variety of different types of application (e.g. peer-to-peer and ad hoc networks). Other models which are no less relevant to application-layer multicast, but which are not included in this paper, can be found at, SHDC [22], ZIGZAG [23] and Bayeux [24].

In the simulation model presented below, we apply the ideas introduced in this article to construct a distributed multimedia application in C++ and show that the use of reflectors can reduce the mean end-to-end delay between nodes in the network that are receiving multimedia content. In this specific example, the content being transmitted is video.

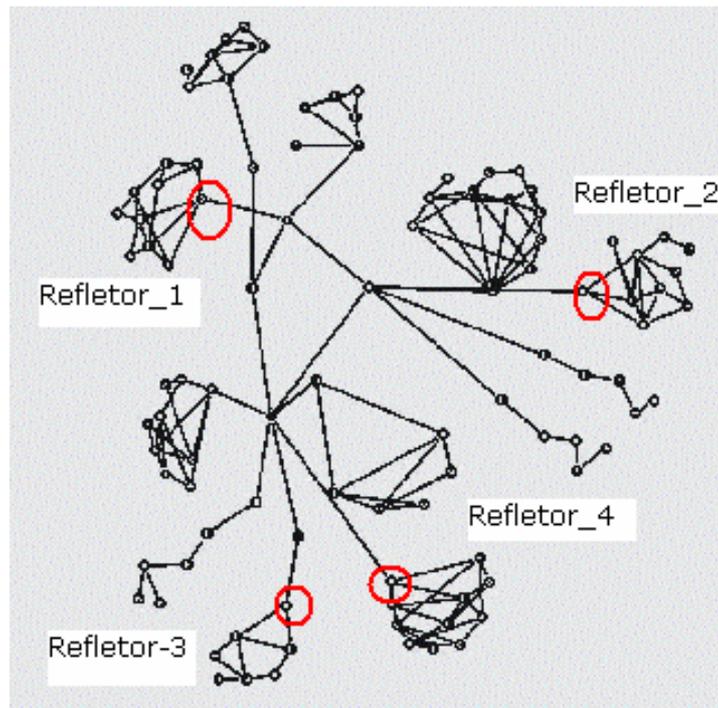
## 4 Intelligent Reflectors: Simulation Model and Results

In order to simulate scenarios similar to the worldwide web, the network simulator ns-2 [25] was used together with the topology generator Georgia Tech Internetworking Topology Models (GT-ITM) [26]. This generates network topologies similar to the Internet. In this study, GT-ITM was used to generate a 100-node (1-100) transit-stub topology with one transit domain. The Transit-Stubs model does not currently support representation of the host systems. Thus, all nodes are of

the same type and produces connected sub-graphs by repeatedly generating a graph according to the edge count, and checking the graph for connectivity.

We choose 4 groups of clients from this topology to make up the overlay network, which would in turn provide the multipoint communication through point-to-point channels. Any node in the network, with the exception of the reflectors (that they are selected with a circle), could act as a video distribution source. For the purpose of simulation, one chose the node of number 54 as source and nodes (14, 58, 63 and 93) as reflectors.

Figure 1 illustrates the topology generated using GT-ITM.



**Fig. 1.** Topology generated using GT-ITM

In the first simulation, just one reflector was used to forward a video stream using the IP multicast to the groups of clients. The same experiment was repeated, though each time new reflectors were added; in total, experiments were performed using 1, 2, 3 and 4 reflectors. In the experiments, the reflectors were the node closest to the backbone of the network, as shown in Figure 1. The simulator was configured to transmit a video stream at 2Mbps for a 15-minute period (900 s). The capacity of link is of 5 Mbps, the transmission delay is of 1.6 ms and the delay between links varies of 20 up to 490 (ms).

The graph in Figure 2 shows the results obtained for the average delay in the communication between the participant nodes for the different number of reflectors.

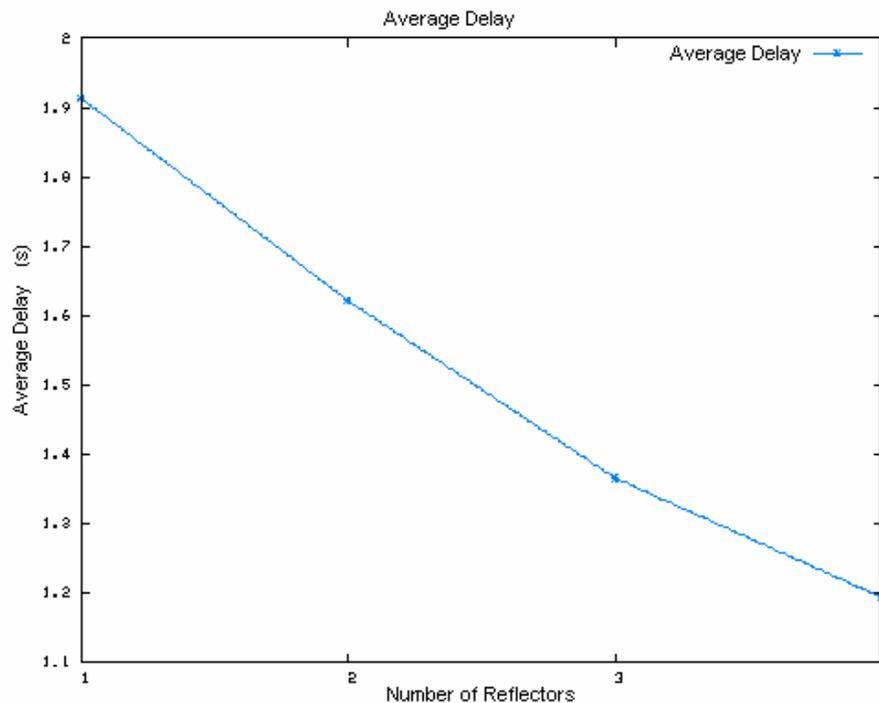


Fig. 2. Relation between average delay and the number of reflectors

This graph shows that as more reflectors are deployed in the network, the average delay is reduced. With reflectors, the video distribution source is not overloaded in terms of bandwidth, while the presence of reflectors makes the system more scalable and tolerant of failures.

In this study, it was observed that a support protocol for the dynamic discovery of reflectors was lacking. For example, in a videoconferencing system, users may be added and removed, and there are other features typical of group communication. This article explores the feasibility of using an algorithm which discovers the most appropriate reflector the moment a request is received by a client to receive video content. Figure 3 presents this algorithm.

```
//client requests access to the videoconference (join)
//search for the most suitable reflector in the reflector
//list ...
list<unsigned> ::interator i = listReflectors.begin();
while (i != listReflectors.end()){
//search for the reflector with the lowest load and
//ascertain whether the number of
//videoconference users has exceeded the maximum limit
```

```
if (value_Load <=(*ptr)->Load && maxUsers<=(*ptr)->MAX ){
...
//Add one channel to the reflector
init_reflect_sockets(channels);
...
//ascertain that the host is not already a participant
if!(inChannel(channel, verif)){
//add host to the communication channel
add_host_to_channel (channel,verif);
...}
...
}...
}
```

**Fig. 3.** Algorithm – Discovery most Appropriate Reflector

The reflector discovery algorithm dynamically ascertains which is the appropriate reflector to meet the client's needs, based on parameters such as capacity and maximum number of users present at the videoconference. Other parameters could be added, such as estimated delay and bandwidth. Currently, we are exploring the possibility of creating mobile reflectors, i.e. replicating the code of the reflector server to a specific point in the network close to the client or group of participating clients. In this case, new machines could install the mobile agent and act as reflectors to transmit the video. In this model, the discovery protocol first seeks the most suitable reflector, should the user be geographically very distant from the reflectors already established, or should they be unable to serve the client at the time that they make the request.

## 5 Conclusions and Future Studies

In this article, the latest achievements, techniques and models, in the area of application-layer multicast group communication have been presented. Furthermore, in section 3, this article proposes and shows the benefits of using reflectors to forward video streams for the practical deployment of application-layer video broadcasting. The results showing a reduction of the mean video delivery delay in the network are promising such as a reduced mean delay between the nodes in the network, are promising.

As a future improvement of this work to include the results of simulations using the reflector discovery protocol, how the clients subscribe the suitable reflector and the deployment of mobile agents located in machines close to the clients, which would serve temporarily as reflectors on an on-demand basis.

## 6 References

1. L. Costa, S. Fdida and O. Duarte, Hop by Hop Multicast Routing Protocol, ACM SIGCOMM'01, San Diego, CA, Aug. 2001.
2. F. Kon , R. H. Campbell and K. Nahrstedt, Using dynamic configuration to manage a scalable multimedia distribution system, Computer Communications Journal (Special Issue on QoS-Sensitive Distributed Systems and Applications), vol. 24, pp. 105-123, Jan, 2001.
3. R. Finlayson, The UDP Multicast Tunneling Protocol, work in progress, draft-finlaysonumtp- 07.txt, Sept. 2002.
4. P. Parnes, K. Synnes and D. Schefstrom, Lightweight Application Level Multicast Tunneling Using Mtunnel, Computer Communication, vol. 21, no. 15, Apr 1998.
5. K. C. Almeroth and K. Sarac, Monitoring IP Multicast in the Internet: Recent Advances and Ongoing Challenges, IEEE Communications Magazine. October 2005.
6. A. El-Sayed, V. Roca and Laurent Mathy, A Survey of Proposals for an Alternative Group Communication Service, IEEE Network, Jan 2003.
7. D. Thaler et al. IPv4 Automatic Multicast without Explicit Tunnels (AMT), work in progress: draft-ietf-mboned-auto-multicast-01.txt, Apr. 2002.
8. M. Castro, P. Druschel, A-M. Kermarrec and A. Rowstron, SCRIBE: A large-scale and decentralized application-level multicast infrastructure, IEEE Journal on Selected Areas in Communications (JSAC) (Special issue on Network Support for Multicast Communications), 2002.
9. R. Boivie et al. Explicit Multicast (Xcast) Basic Specification, work in progress, drafttooms- xcast-basic-spec-03.txt, June 2002.
10. L. Blazevic and J.-Y. Le Boudec, Distributed Core Multicast (dcm): A Multicast Routing Protocol for Many Groups with Few Receivers, ACM SIG-COMM Computer Communications, vol. 29, no. 5, Oct. 1999.
11. M. Liu, R. Talpade, and A. McAuley, Amroute: Adhoc Multicast Routing Protocol, Technical Report. TR 99-8, CSHCN, (unpublished), 1999.
12. S.-J. Lee, W. Su and M. Gerla, On-Demand Multicast Routing Protocol (ODMRP) for Ad Hoc Networks, IETF Internet Draft, work in progress, 2002.
13. S.-J. Lee et al. A Performance Comparison Study of Ad Hoc Wireless Multicast Protocols, IEEE INFOCOM 2000, Mar. 2000.
14. D. Pendarakis, S. Shi, D. Verma and M. Waldvogel, ALMI: An application level multicast infrastructure. Proceedings of the 3rd, USENIX Symposium on Internet

Technologies and Systems (USITS '01), pp. 49-60, mar 2001.

15. C. Abad, W. Yurcik and R. H. Campbell, A Survey and Comparison of End-System Overlay Multicast Solutions Suitable for Network Centric Warfare, (December, 2005); <http://www.citeseer.ist.psu.edu/735587.html>.

16. J. Jannotti et al. Overcast: Reliable Multicasting with an Overlay Network, In Proc. Of OSDI, October 2000.

17. S. Ratnasamy, M. Handley, R. M. Karp and S. Shenker, Application-Level Multicast Using Content-Addressable Networks, Proceedings of the Third International COST264, isbn: 3-540-42824-0, pp. 14-29. 2001.

18. P. Francis, Yoid: Your own internet distribution, Technical report, ACIRI, (unpublished), (April, 2000); <http://www.aciri.org/yoid>.

19. B. Zhang and S. Jamin and L. Zhang, Host multicast: A framework for delivering multicast to end users, In Proceedings of IEEE Infocom, June 2002.

20. S. Banerjee, B. Bhattacharjee and C. Kommareddy, Scalable Application Layer Multicast, ACM SIGCOMM '02, Pittsburgh, PA, Aug. 2002.

21. A. Rowstron and P. Druschel, Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems, IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), pp. 329-350, November, 2001.

22. L. Mathy, R. Canonico and D. Hutchison, An Overlay Tree Building Control Protocol, 3rd Int'l. Wksp. Networked Group Communication, Nov. 2001.

23. D. Tran, K. Hua, and T. Do, ZIGZAG: An Efficient Peer-to-Peer Scheme for Media Streaming, IEEE INFOCOM, 2003.

24. S. Zhuang et al. Bayeux: An Architecture for Scalable and Fault-Tolerant Wide-Area Data Dissemination, Proceedings of NOSSDAV, June 2001.

25. S. McCanne and S. Floyd, The Network Simulator-ns-2. (January, 2006); <http://www.isi.edu/nsnam/ns>.

26. E.W. Zegura, K. Calvert and S. Bhattacharjee, How to Model an Internetwork, IEEE Infocom, 1996.