A Survey of Proposals for an Alternative Group Communication Service

Ayman El-Sayed and Vincent Roca, INRIA Rhone-Alpes
Laurent Mathy, Lancaster University

Abstract

As expectations for the Internet to support multimedia applicationsgrow, new services need to be deployed. One of them is the group communication service for one-to-many or many-to-many data delivery. After more than a decade of important research and development efforts, the deployment of multicast routing in the Internet is far behind expectations. Therefore, a first motivation for an alternative group communication service is to bypass the lack of native IP multicast routing. Although less efficient and scalable than native multicast routing, such alternative services are generally suitable for the purpose. A second possible motivation is to go beyond the limitations of classic multicast routing for very specific working environments. In this article we identify, classify, and discuss some of these alternative approaches.

A group communication service refers to the ability to send information to several receivers at the same time, using either a one-to-many or many-to-many model. The any-source and source-specific multicast routing approaches provide such a service. Still, other solutions are possible, and this article proposes a survey of such alternatives. Although this survey aims to give a complete overview of alternative group communication service (AGCS) techniques, we do not claim to be exhaustive. Besides, we only consider the routing service (i.e., as a replacement for, or complement to, IP multicast) and ignore any upper-level service like reliability or congestion control. If some of the solutions we introduce largely impact these upper-level services, this is a by-product that will not be discussed in this article. Likewise, we do not cover differentiated services multicasting or, more generally, quality-of-service-based multicast routing.

An AGCS can be used as a way to *bypass the multicast routing deployment problems*. Indeed, group communication traditionally requires that each node at each site has access to a native multicast routing service. If intradomain multicast (within a LAN or site) is widely available, this is not the case for interdomain multicast. Today many Internet service providers (ISPs) are still reluctant to provide a wide-area multicast routing service [1]:

- There are *technical reasons*. Multicast is still a hot and complex research subject, many protocols are not yet finalized, and monitoring is not easy.
- There are *marketing reasons*. Multicast breaks the traditional pricing model where only the incoming flow is charged: should the source (who can serve a large number of receivers with a slow access line) or the receivers be charged, or should it be a free service?
- And finally, there is an "chicken and egg" problem. The use of multicast is still driven more by the academic community than by customer demand.

Several alternative solutions have recently been proposed specifically to bypass these limitations.

But other motivations exist. For instance, an AGCS can be used to go beyond the limitations of traditional multicast routing. An AGCS can offer a bridging service between several multicast-capable areas running different multicast routing protocols (e.g., between IPv4 and IPv6 multicast islands). An AGCS can also be used along with protocol-independent multicast sparse source mode (PIM-SSM). Since only the source S is allowed to send traffic to an (S, G) channel, G being the group addresses, no multicast back-channel is available for a receiver to provide feedback to the group. If the feedback rate is sufficiently low (e.g., with Real-Time Conferencing Protocol, RTCP), this feedback can be unicast to the source and echoed back onto the channel. If not, such an approach quickly results in source implosion, and this is where an AGCS can be of some help.

Finally, an AGCS can be used in working environments where traditional multicast routing is completely inappropriate. This is the case in ad hoc networks where there is no fixed infrastructure. Multicast routing, designed for a fixed hierarchical routing infrastructure with well identified multicast routers, is completely defeated. This is also the case when there are a very high number of small dynamic groups. The signaling load required by traditional multicast routing for each group prevents the whole system to scale in terms of the number of concurrent groups.

Several performance metrics have been defined to characterize AGCS performance and impacts on the network. Some of them focus on the data path:

- **Stress**: [2] defines the stress of a physical link as the number of identical packets it carries. The optimal value, achieved with native multicast routing, is of course 1.
- **Resource usage**: [2] defines this metric as the sum of the *delay* $*$ *stress* product over all the links that participate in data transmissions. This metric gives an idea of network resources used by the transmission process, assuming that links with high delays are more costly.
- **Stretch**: also called *relative delay penalty* in [2], the stretch metric between a source and a member is the ratio of the

0890-8044/03/$17.00 © 2003 IEEE

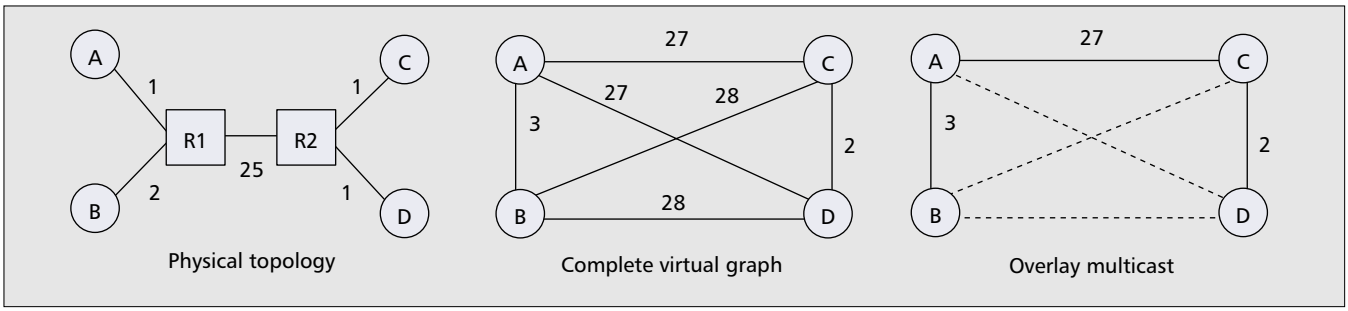

Figure 1. *The physical and overlay topologies.*

delay between them along the overlay distribution topology, to the delay of the direct unicast path.

Another set of metrics focuses on end host performance:

- **Losses after failures**: This metric counts the average number of packet losses after an ungraceful failure of a single node [3, 4]. It highlights robustness to the occurrence of unpredicted events.
- **Time to first packet**: [3] defines the time required for a new member to start receiving a data flow when joining an ongoing session.

Finally, some metrics focus on the control part:

- **Control overhead**: Maintaining the AGCS topology has a cost, in terms of control information exchanged (number of messages processed and bandwidth) [5].

The broad diversity of performance metrics shows there is no single answer to the question "what is the best solution?" Some proposals can deliberately favor some of these metrics at the expense of others (e.g., the multi-unicast approach used by reflectors, below, offers high robustness to member failures, other than the reflector itself, at the cost of high link stress near the reflector).

The remainder of the article is organized as follows. We provide a taxonomy of AGCS approaches, and the pros and cons for each class of solutions; we discuss some key points. Finally, we conclude this survey.

A Taxonomy of AGCS Proposals

Unicast/Multicast Reflector and Punctual Tunneling Proposals

Principles — In this category we find solutions whereby a host having access only to unicast routing contacts a reflector, which is a user-level gateway between a multicast-enabled network (e.g., the Mbone) and the set of unicast hosts. Each multicast packet coming from the multicast network (from a unicast host) is forwarded to each unicast host (to the multicast group and the other unicast hosts). These solutions are often called *tunneling* approaches too since they create tunnels between the reflector and the end hosts, but they are completely different from *permanent* tunneling approaches.

The first key aspect is its *application-level feature*. The communication between a host and the reflector can be more or less elaborate: multicast packets can be captured by a BPF tool and encapsulated in unicast datagrams. A simpler solution consists of opening a UDP socket and forwarding only the payload, without the initial packet headers. In that case the source address and port are lost, but upper protocols (e.g., RTCP) may recover the source identity.

Second, this service is usually *set up for a limited time and for a limited number of groups* (usually there is one reflector per group).

The UMTP [6] and Mtunnel [7] proposals fall in this category.

Discussion — This approach is clearly not the most efficient since it creates hot spots in the network near the reflector. However, it is easily set up and the reflector has full control of the service, its duration, the multicast groups forwarded, and the set of unicast authorized hosts. Therefore, its global impact on the network over a longer period is limited.

A straightforward extension that largely improves scalability consists of having a topology of reflectors, each controlling a multicast-capable area. All the receivers of a domain are thus hidden behind their local reflector.

Permanent Tunneling Proposals

Principles — Permanent tunneling proposals differ from the reflector proposals from several points of view. First of all, tunneling is performed at the routing level and uses IP encapsulation. Its creation requires privileges and is usually not set up by an end host.

Second, if a reflector answers a punctual need within a well identified group of people, tunneling solutions offer permanent connectivity for a whole site.

Third, tunnels are fully integrated in the multicast routing protocols and offer connectivity to all possible multicast groups.

The Mrouted DVMRP implementation is undoubtedly the most popular tunneling solution and has long been used in the Mbone. AMT [8] is midway between the reflector and permanent tunneling categories. It manages the multicast traffic exchange for any groups between isolated multicast-enabled sites, but does not include a routing protocol, unlike DVMRP/Mrouted.

Discussion — This class has long been the only way to connect isolated multicast islands. Because of performance problems tunnels create (potential for a high physical link stress and loops), it is now banned from modern multicast routing protocols. However, multicast tunneling is still used in some situations, for instance, when crossing IPSec virtual private network (VPN) tunnels that do not support multicast packets [9].

Automatic Overlay Multicast Proposals

Principles — This broad class of proposals shifts the multicast support from core routers to end systems. End systems now implement all group communication functionalities, including membership management, packet replication, and distribution. For instance, in Narada [2] group members communicate via an overlay structure built on top of unicast paths between various pairs of hosts (Fig. 1). The physical topology is abstracted as a complete virtual graph on which Narada constructs a spanning tree using a Reverse Path Forwarding algorithm. Since the group is dynamic, a mechanism is defined to add or drop links, repair partitioned topologies, and incrementally improve the virtual topology.

The overlay multicast (OM) approach differs in many respects from traditional multicast routing:

- A forwarding node in the overlay topology can be either *an end host* (i.e., running the application), *a dedicated server*

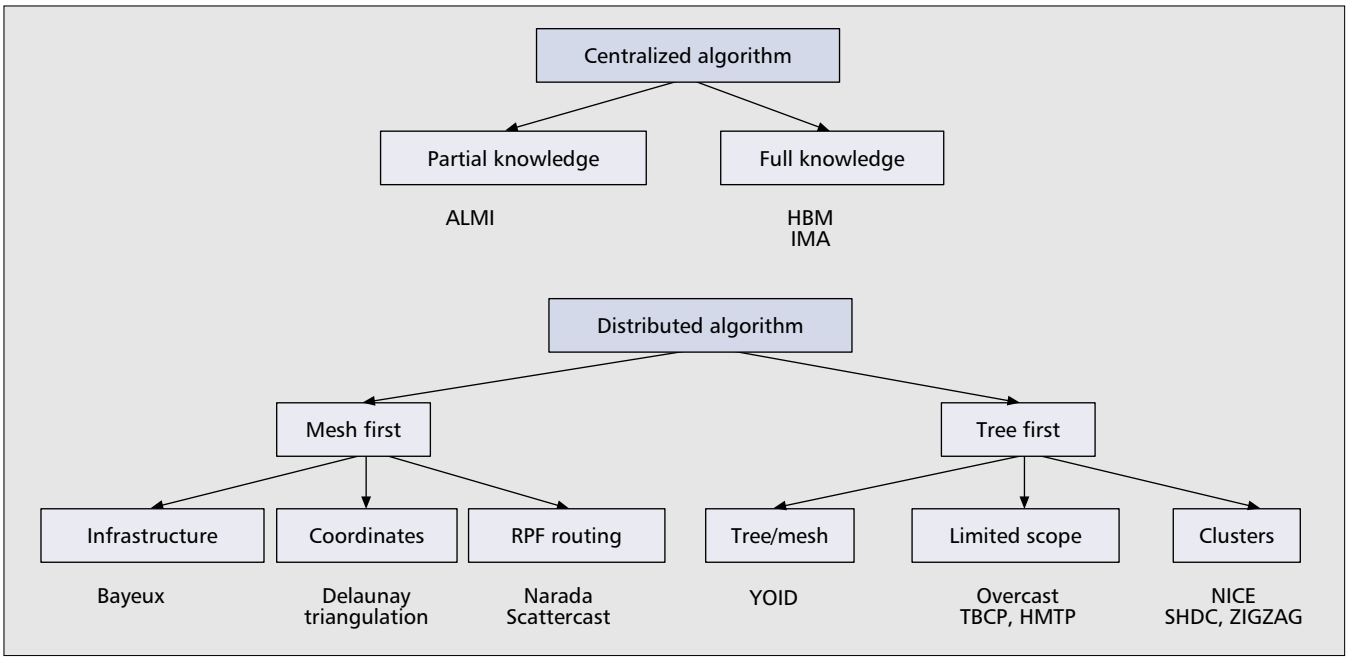

Figure 2. *A taxonomy of overlay multicast proposals.*

within the site, or *a border router*. On the contrary, traditional multicast trees only include core routers.

- With an overlay topology, *the underlying physical topology is completely hidden*. A directed virtual graph is created between all the nodes. Undirected graphs can also be used if the possibility of having asymmetric routes is overlooked. Such graphs are built and optimized according to some form of metric measurements taken between some or all nodes.
- In traditional multicast, the membership knowledge is distributed in the multicast routers. With an OM *group members are known* either by a rendezvous point (RP) [4], the source, or everybody, or is distributed among members [2, 10].
- The *overlay topology is potentially under complete control*. For instance, [4] takes advantage of the additional knowledge centralized at the RP (the node/link specificities and their stability) during the topology creation process.

Figure 2 classifies the proposals according to the centralized or distributed topology building algorithm. Centralized approaches are further classified according to full (HBM [4]) or partial (ALMI [11]) membership knowledge.

Distributed approaches further differ in the way they create the overlay topology: some of them first create the tree topology, while others first create a mesh topology. The "mesh first" approaches are Narada [2], the proposals that assign an arbitrary coordinate to each member and then perform Delaunay triangulation [12], and Bayeux [13].

The "tree first" approaches include YOID, TBCP [10], HMTP [14], SHDC [15], NICE [16], Overcast [17], and ZIGZAG [18]. Some of them (TBCP, HMTP) rely on a recursive algorithm to build the tree: a newcomer first contacts the tree root, chooses the best node among the root's children, and repeats this top-down process until it finds an appropriate parent. The clustering solutions (NICE, SHDC, ZIGZAG) create a hierarchy of clusters (i.e., sets of nodes "close" to each other). Newcomers recursively cross this hierarchy to find the appropriate cluster.

Discussion — These proposals undoubtedly form a rich family that reflects the large diversity of objectives: high performance thanks to an optimized communication topology, adaptability and per-host profiling of the topology to take into account their features, robustness in the event of member departures and failures, and high scalability. Having a good level of congruence between the physical topology and the overlay is rather challenging and is often the key to good performance. However, being end to end, overall paths along the overlay can get rather long in terms of delays, and data can be replicated several times over some physical links. The bottom line here is that the right compromise should be achieved between growing an overlay multicast tree in length or width, to suit both application requirements and network conditions. Anyway, because of its very nature, an overlay will never match the efficiency of native IP multicast.

A major advantage is that most proposals do not require any special support from network routers and can therefore be deployed universally. As a result, they can be made available as libraries or built in application code, which reduces the need for standardization.

Proposals Based on Gossiping for Peer-to-Peer Communications

Principles — Gossiping is widely used for state and data distribution in distributed systems. Within the context of overlay networks, a gossiping technique like Scribe [19] can be used for *state distribution*. Each group member periodically sends an announcement message containing its list of neighbors to these neighbors themselves. Hence, each node increases the *group membership knowledge horizon* by one hop. Furthermore, each node adds to the list any nodes it has heard of to propagate knowledge about nodes not directly connected. Therefore, a node gains complete knowledge of the group just by connecting to any node already in the group. The identity of such a node can be learned from an RP that only maintains a list of a few members. Once a node has learned the identity of other nodes, it can periodically measure its distance to a randomly chosen set of these peers and replace its farthest neighbors with newly found closer ones to improve the overlay infrastructure efficiency.

Alternatively, gossiping can also be used to create a highly robust *data distribution service*. The difficulty is to estimate when to remove any given data item from the gossiping process (i.e., it is difficult to estimate when all group members

have seen the corresponding data item, especially within a dynamic group). This scheme is therefore usually limited to small data transfers or static group environments.

Discussion — Gossiping is a very robust state distribution mechanism. Indeed, the loss of any single node does not result in any knowledge loss. Furthermore, because an RP is only needed to bootstrap new nodes and only has limited membership knowledge, it does not represent a potential hot spot.

For very large groups, however, the periodic announcements result in a large overhead. The classical solution is to reduce the announcement frequency, which, in turn, reduces system responsiveness. Tuning such a system is therefore nontrivial.

Finally, nodes of a large group only have a reasonably accurate view of their vicinity. This is due to the time required to advertise new members (an announcement period per hop) and remove those who left (detected only after several silent announcement periods). The convergence of application-level multicast techniques based on gossiping can therefore be rather slow.

Proposals Based on a Specific Group Communication Routing Service

Principles — Several proposals rely on a dedicated new group communication routing service within routers. Therefore, they cannot be deployed on demand by the end users. However, they may one day be standardized and deployed; for example, the XCAST community is trying to create a new Internet Engineering Task Force (IETF) working group. Some proposals address some of the limitations of traditional multicast routing protocols:

- Low scalability in terms of the number of concurrent groups
- The need for a stable networking infrastructure

Two proposals try to solve the *scalability issue*. XCAST [20] adds an explicit list of destinations in each packet using either a new XCAST header (IPv4) or a new routing extension header (IPv6). Each router along the way parses the IP header and, in case of branching, creates and forwards a new packet with the appropriate subset of destinations reachable from each interface. When a single destination is left, the XCAST packet is turned into a normal unicast packet. XCAST derives its high scalability from the fact that no state information is kept within the backbone.

The distributed core multicast (DCM) [21] proposal has similar goals but follows a different method. DCM uses several distributed core routers (DCRs), located at the edge of the backbone and synchronized with a dedicated membership distribution protocol. Each site contains one or more DCRs that forward traffic to/from other sites. The scalability asset of DCM derives from the fact that group state information is kept in the DCR routers rather than disseminated across the backbone routers.

The second issue is typical in *mobile ad hoc networks* characterized by rapidly changing multihop topologies composed of several wireless links with no fixed infrastructure. Traditional multicast routing protocols cannot be deployed then, since they rely on well identified multicast routers. Therefore, a number of proposals have been proposed:

- AMRoute [22], a protocol that first creates an overlay mesh and then a shared multicast tree on top of it. It shares many similarities with the protocols mentioned earlier.
- ODMRP [23] is a protocol where a source creates on demand and for a limited lapse of time a mesh of hosts in which data is flooded.
- MAODV [24], a multicast routing protocol building a shared tree. This process is purely on demand and follows a route request/route reply discovery cycle, where the request is broadcast to neighborhoods and forwarded until it reaches the destination or a node having a route to the destination.

These protocols largely differ in the way the distribution topology is created and maintained, some of them leading to mesh-based distribution, others to tree-based distribution. In each case the topology is regularly updated to take into account possible topology changes.

Finally, the REUNITE [25] and HBH [26] proposals follow a recursive unicast approach to solve the multicast deployment issue. The idea is to have some REUNITE/HBH-capable routers that act as branching nodes and create copies with modified unicast destination address between two hops. It is similar to XCAST except that packets do not carry the list of destinations. Branching nodes thus need to keep some state for each group.

Discussion — The efficiency of many of these proposals (e.g., XCAST, REUNITE, HBH), not surprisingly, depends on the number and location of routers offering the service, even if partial deployment is still possible. If routers at the natural branching points (usually within the backbone) support it, efficiency can be high. If only the routers close to end nodes support it, the link stress is significantly higher.

Concerning the proposals dedicated to ad hoc networks, performance largely differs and depends on host mobility (how many of them are mobiles and how fast they move). For instance, [27] shows that in highly mobile environments, mesh-based protocols outperform tree-based protocols, essentially because of the presence of alternate routes.

Discussion and Open Points

We have so far described and compared a large variety of proposals. We now discuss several key points that were raised and classify them in decreasing order of importance (which can be modified according to the exact application requirements).

Ease of Deployment

An AGCS should be easy to deploy to offer a viable alternative to native multicast routing. Manual deployment is only realistic if the procedure is straightforward, for instance, to bootstrap the AGCS system (e.g., to specify a reflector address or an RP address).

Several proposals [28, 29] suggest using an active networking approach to provide an AGCS. It is clear that if an active networking service is available in each potential node, because of the flexibility it offers, deploying an AGCS becomes an easy task. But this is largely dependent on the availability of the active networking service in a sufficient number of nodes.

Also, Network Address Translation (NAT) devices and firewalls present nontrivial issues within the context of some of the AGCS techniques discussed above. Some of these issues require the use of *application-level gateways* (e.g., to ensure correct translation of addresses contained in protocol messages).

Robustness

Interdomain multicast routing is often said to be fragile. If an AGCS offers a way to alleviate this problem, it also creates other instability problems. For instance, a solution based on end hosts (usually PCs or workstations) is intrinsically less robust than one based on dedicated and well administered commercial routers. There is a high risk, as the group size

increases, that the topology will be partitioned after a single node failure. Some proposals address this robustness aspect by using some level of flooding (e.g., gossiping approaches). Other proposals [20] suggest adding explicit redundancy in the overlay topology and a learning mechanism whereby less reliable hosts are identified and the topology is created taking this feature into account. Finally, solutions for ad hoc networks all address this aspect, which is then fundamental.

In any case, having a fast detection and repair mechanism is required but, in our opinion, not sufficient. For instance, some applications may require that partitions be avoided altogether (e.g., cooperative work or a high-quality multimedia-on-demand session).

Security

A point that is usually neglected in the above proposals is security. The AGCS service provided does not offer any additional security (e.g., there is no authentication of the nodes), nor is it itself secure (e.g., control mechanisms are not secured). This is paradoxical since most proposals are based on unicast communications, either among group members and/or members and an RP. Indeed, offering security for point-to-point communications (which is the basis of many AGCS proposals) or when there is a central RP collecting information on group members is much simpler than in the general case of multicast communications where many additional hard problems must be solved (see the MSEC multicast security IETF working group charter).

An exception is [9], which explains how a group communication service can be set up in a fully secure VPN environment. Here, point-to-point IPSec tunnels are dynamically created between the sites that host group members, and removed when all members have left. The architecture proposed relies on a centralized approach around a network operation center that is in any case required for security control purposes. We expect that many future proposals will continue to address this aspect, which is of the utmost importance for many applications.

Performance

The performance of most AGCS is unsurprisingly lower than that of native multicast routing protocols because traffic forwarding at the end host or a limited number of hosts (e.g., with reflectors) is necessarily less efficient than using multicast routers in the backbone.

However, performance is not necessarily the primary target of these proposals. For instance, solutions dedicated to ad hoc networks and gossiping techniques used in large dynamic peer-to-peer communities deliberately lay more importance on robustness and scalability. Consequently, the topology created deliberately includes many redundant paths that affect final performance. Likewise, reflector-based solutions set more importance on ease of deployment than performance.

On the contrary, proposals creating an automatic overlay topology are more concerned with creating good data delivery. Many aspects will affect performance:

The type of topology created: A per-source shortest path tree is undoubtedly more efficient than a single shared tree used by many different sources; but managing several trees, one per source, also has a higher cost.

The possibility of dynamic topology adaptation: The topology must reflect the dynamic networking conditions, so network monitoring is required. Passive monitoring is sometimes possible, taking advantage of ongoing data flow reception statistics; otherwise, active monitoring is required, adding some overhead.

The performance metrics considered: Much work only considers communication delays, assuming that all paths are symmetric (which enables the use of simple tools like `ping`). Reference [26] reminds us that performance can be affected by the presence of asymmetric unicast routing, which is not so uncommon in the Internet. Finally, [30] argues that the delay and bandwidth metrics should both be considered.

Per-host profiling: Nodes can widely differ, and network-related metrics cannot catch all of their specifics. For instance, lightweight nodes (e.g., PDAs), with limited processing power and battery should not become transit nodes, even if they benefit from small communication delays. Likewise, a history of all nodes should be kept, so the nodes that turn out to be unstable (e.g., because of a wireless nomad connection) can be moved to the leaves of a tree. Overlay multicast proposals using a centralized algorithm, because of their node database [4], can easily take this into account during the topology creation process.

The price to pay for higher performance is more complex topology maintenance algorithms and a higher signaling load to perform network monitoring and dynamic topology adaptation. There is clearly a trade-off to find between performance and management costs.

Scalability

If scalability in terms of the group size is an explicit target of traditional multicast routing protocols, it is not required in all situations. Consequently, some of the AGCS discussed previously (e.g., reflectors or [4]) are specifically designed to handle small groups, which is sufficient in many situations (e.g., collaborative work). Better control mechanisms (e.g., to adapt more frequently to networking conditions) or a simpler architecture compensate for the lack of scalability.

Many other proposals (e.g., [15]), on the contrary, target high scalability, which is required, for instance, with large peer-to-peer applications. This scalability is usually achieved with a hierarchical overlay topology (e.g., based on clustering) and distributed partial membership knowledge.

Since intradomain multicast routing is often available, a frequent assumption is that the AGCS is only used between sites, not within a site. A representative in each site locally multicasts the traffic received. Doing so increases global scalability since all the local members are hidden behind their representative.

In some cases, the scalability is not in terms of number of members but number of concurrent groups, which is a totally different issue, as discussed earlier. This kind of scalability can be important for deploying new services and protocols over the Internet (e.g., there are proposals to improve Mobile-IP handoff thanks to XCAST, which can be valuable in a cellular IP environment with a very high number of mobile nodes).

Finally, the idea of aggregated multicast with intergroup tree sharing [31] can easily be applied to AGCS. For instance, any collaborative work session is composed of several audio/video/whiteboard tools with approximately the same set of end users. Sharing a single overlay topology would help reduce the global control overhead.

Dynamic Discovery of Sources and Receivers

An AGCS system must be informed of the presence of sources and receivers. Many AGCSs solve this problem by using a static configuration where the local administrator/user decides beforehand which group(s) to distribute. This can lead to useless traffic distribution, say, after the last interested host has left from a site. A more elaborate solution is thus required.

In the simplest case, the AGCS is implemented as a library linked to the application. Direct communication is then possible (e.g., through a dedicated API), and source/receiver discovery is immediate.

But when the AGCS tool runs on a different host than applications, two cases are possible: the AGCS tool is either

on the same LAN as group members (sources or receivers) or on a different LAN. The first case is easily addressed by listening to IGMP traffic and multicast traffic. But this solution no longer works for the second case, since the top multicast router by default isolates the various LANs. Reference [9] gives some insight on how to address the source and receiver discovery problems that arise in such a case.

This discussion shows that local source and receiver discovery is not so simple and is rarely considered. However, this is the price to pay for the AGCS to follow the dynamic behavior of group members and limit useless traffic.

Conclusions

This article has introduced and discussed several proposals for building an alternative group communication service. The motivation is usually to offer an alternative to the lack of deployment of interdomain multicast routing. Another motivation is sometimes to go beyond the limitations of multicast routing protocols: some proposals try to improve the scalability in terms of concurrent number of groups; others are designed for group communications in ad hoc networks; finally, some are used to create a robust communication system in large dynamic communities, such as for peer-to-peer applications.

We classify the proposals into several categories: based on a reflector approach, relying on permanent tunneling, creating an automatic overlay topology, or relying on a specific routing service. We show that these proposals can differ widely and are still the subject of important research efforts. We expect this trend to continue, since AGCSs fulfill many important needs.

Finally, it should be noted that many of the techniques discussed in this article can complement each other as well as IP multicast.

References

[1] C. Diot *et al.*, "Deployment Issues for the IP Multicast Service and Architecture," *IEEE Network*, Jan. 2000, pp. 78–88.

[2] Y. Chu, S. Rao, and H. Zhang, "A Case for End System Multicast," *ACM SIGMETRICS*, June 2000.

[3] H. Deshpande, M. Bawa, and H. Garcia-Molina, "Streaming Live Media Over Peers," Stanford Univ., Database Group, Submitted for publication, 2002.

[4] V. Roca and A. El-sayed, "A host-based Multicast (hbm) Solution for Group Communications," *1st IEEE Int'l. Conf. Networking*, Colmar, France, July 2001.

[5] A. El-Sayed and V. Roca, "Improving the Scalability of an Application-level Group Communication Protocol," *10th Int'l. Conf. Telecommun.*, Papeete, French Polynesia, Feb. 2003.

[6] R. Finlayson, "The UDP Multicast Tunneling Protocol," work in progress, draft-finlayson-umtp-07.txt, Sept. 2002.

[7] P. Parnes, K. Synnes, and D. Schefstrom, "Lightweight Application Level Multicast Tunneling Using Mtunnel," *Comp. Commun.*, vol. 21, no. 15, Apr. 1998, pp. 1295–1301.

[8] D. Thaler *et al.*, "IPv4 Automatic Multicast without Explicit Tunnels (AMT)," work in progress: draft-ietf-mboned-auto-multicast-01.txt, Apr. 2002.

[9] L. Al-Chaal, V. Roca, and M. Habert, "Offering A Multicast Delivery Service in a Programmable Secure IP VPN Environment," *4th Int'l. Wksp. Networked Group Commun.*, Boston, MA, Oct. 2002.

[10] L. Mathy, R. Canonico, and D. Hutchison, "An Overlay Tree Building Control Protocol," *3rd Int'l. Wksp. Networked Group Commun.*, London, U.K., Nov. 2001.

[11] D. Pendarakis *et al.*, "ALMI: An Application Level Multicast Infrastructure," *3rd USNIX Symp. Internet Tech. and Sys.*, Mar. 2001.

[12] J. Liebeherr, M. Nahas, and W. Si, "Application-Layer Multicast with Delaunay Triangulations," *IEEE GLOBECOM '01*, also tech. rep. CS-2001-26, Nov. 2001.

[13] S. Zhuang *et al.*, "Bayeux: An Architecture for Scalable and Fault-Tolerant Wide-Area Data Dissemination," *11th Int'l. Wksp. Net. and Op. Sys. Support for Digital Audio and Video*, June 2001.

[14] B. Zhang, S. Jamin, and L. Zhang, "Host Multicast: A Framework for Delivering Multicast to End Users," *IEEE INFOCOM '02*, New York, NY, June 2002.

[15] L. Mathy *et al.*, "Scalable Adaptive Hierarchical Clustering," *IEEE Commun. Lett.*, vol. 6, Mar. 2002, pp. 117–19.

[16] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable Application Layer Multicast," *ACM SIGCOMM '02*, Pittsburgh, PA, Aug. 2002.

[17] J. Jannotti *et al.*, "Overcast: Reliable Multicasting with an Overlay Network," USENIX OSDI, Oct.2000.

[18] D. Tran, K. Hua, and T. Do, "ZIGZAG: An Efficient Peer-to-Peer Scheme for Media Streaming," Univ. Central FL, Orlondo, tech. rep. CS-UCF 2002, 2002.

[19] M. Castro *et al.*, "Scribe: A Large-Scale and Decentralized Application-level Multicast Infrastructure," *IEEE JSAC*, 2002.

[20] R. Boivie *et al.*, "Explicit Multicast (Xcast) Basic Specification," work in progress, draft-ooms-xcast-basic-spec-03.txt, June 2002.

[21] L. Blazevic and J.-Y. Le Boudec, "Distributed Core Multicast (dcm): A Multicast Routing Protocol for Many Groups with Few Receivers," *ACM SIGCOMM Comp. Commun. Rev.*, vol. 29, no. 5, Oct. 1999.

[22] M. Liu, R. Talpade, and A. McAuley, "Amroute: Adhoc Multicast Routing Protocol," Tech. rep. TR 99-8, CSHCN, 1999.

[23] S-J. Lee, W. Su, and M Gerla, "On-Demand Multicast Routing Protocol (ODMRP) for Ad Hoc Networks," work in progress: draft-ietf-manet-odmrp-01.txt, June 1999.

[24] E. Royer and C. Perkins, "Multicast Operation of the Ad-Hoc On-Demand Distance Vector Routing Protocol," *MobiCom '99*, Seattle, WA, Aug. 1999.

[25] I. Stoica, T. S. Eugene Ng, and H. Zhang, "Reunite: A Recursive Unicast Approach to Multicast," *IEEE INFOCOM 2000*, Mar. 2000.

[26] L. Costa, S. Fdida, and O. Duarte, "Hop by Hop Multicast Routing Protocol," *ACM SIGCOMM '01*, San Diego, CA, Aug. 2001.

[27] S.-J. Lee *et al.*, "A Performance Comparison Study of Ad Hoc Wireless Multicast Protocols," *IEEE INFOCOM 2000*, Mar. 2000.

[28] A. Ghosh, M. Fry, and J. Crowcroft, "An Architecture for Application Layer Routing," *2nd Int'l. Working Conf. Active Nets.*, Oct. 2000.

[29] L. Yamamoto and G. Leduc, "Autonmous Multicast Reflectors over Active Network," *Symp. Software Mobility and Adaptive Behavior*, Mar. 2001.

[30] Y. Chu *et al.*, "Enabling Conferencing Applications on the Internet Using an Overlay Multicast Architecture," *ACM SIGCOMM '01*, San Diego, CA, Aug. 2001.

[31] A. Fei *et al.*, "Aggregated Multicast with Inter-group Tree Sharing," *3rd Int'l. Wksp. Networked Group Commun.*, London, U.K., Nov. 2001.

Biographies

AYMAN EL-SAYED (ayman.elsayed@inrialpes.fr) received his B.Sc. degree in computer science and engineering in 1994 and his Master's degree in computer networks in 2000 from the University of Menoufia, Egypt. He is now working in the Planete team of INRIA Rhone-Alpes Research Center where he is working toward a Ph.D. in computer science from the University of Grenoble (INPG). His research interests include multicast routing and application-level multicast techniques.

VINCENT ROCA (vincent.roca@inrialpes.fr) obtained his Ph.D. in computer science in 1996 from INPG, France. From 1997 to 1999 he worked as an associate professor at the University of Paris 6 (LIP6). Since October 2000 he has been a researcher in the Planete team of the INRIA Rhone-Alpes research institute. His research interests include group communication techniques, reliable multicast, overlay multicast, security, and multimedia communications.

LAURENT MATHY is a lecturer in the Distributed Multimedia Research Group (DMRG), Computing Department at Lancaster University, England. He spent the 1995–1996 academic year at the Center for Integrated Computer Systems Research (CICSR), the University of British Columbia, Vancouver, Canada, as a visiting scholar. He was also a research engineer in the Research Unit in Networking (RUN) of the University of Liege, Belgium, from 1993 to 1995. He graduated in electrical engineering from the University of Liege, Belgium, in June 1993, and was awarded his Ph.D. in computer science from Lancaster University in January 2000. His research interests include multimedia communications, group communication support, programmable networks, overlay structures, and e-commerce security.