# Incremental Service Deployment Using the Hop-By-Hop Multicast Routing Protocol

Luís Henrique M. K. Costa, *Member, IEEE*, Serge Fdida, *Senior Member, IEEE*, and Otto Carlos M. B. Duarte

***Abstract*—IP multicast is facing a slow take-off although it has been a hotly debated topic for more than a decade. Many reasons are responsible for this status. Hence, the Internet is likely to be organized with both unicast and multicast enabled networks. Thus, it is of utmost importance to design protocols that allow the progressive deployment of the multicast service by supporting unicast clouds. This paper presents HBH (hop-by-hop multicast routing protocol). HBH adopts the source-specific channel abstraction to simplify address allocation and implements data distribution using recursive unicast trees, which allow the transparent support of unicast-only routers. An important original feature of HBH is its tree construction algorithm that takes into account the unicast routing asymmetries. Since most multicast routing protocols rely on the unicast infrastructure, the unicast asymmetries impact the structure of the multicast trees. We show through simulation that HBH outperforms other multicast routing protocols in terms of the delay experienced by the receivers and the bandwidth consumption of the multicast trees. Additionally, we show that HBH can be incrementally deployed and that with a small fraction of HBH-enabled routers in the network HBH outperforms application-layer multicast.**

***Index Terms*—Multicast, routing, service deployment.**

## I. Introduction

IP MULTICAST is facing a slow take-off although it has been a hotly debated topic for more than a decade. Many reasons are responsible for this status. The IP Multicast architecture is composed of a service model that defines a group as an open conversation from $M$ sources to $N$ receivers, an addressing scheme based on IP class-D addresses, and routing protocols. In IP Multicast, any host can send to a multicast group and any host can join this group and receive data [1]. The IP Unicast service model is also completely open, but in multicast the potential burden caused by unauthorized senders is multiplied by the multicast group size.

The IP Multicast architecture is completed by group addressing and routing protocols. A multicast group is identified by a class-D IP address which is not related to any topological information, as opposed to the hierarchical unicast addressing model. Therefore, multicast address allocation is complex, and multicast forwarding state is difficult to aggregate. Currently, there is no scalable solution to inter-domain multicast routing. The approach used is to connect different domains through MBGP (multiprotocol extensions to BGP-4) [2] and MSDP (multicast source discovery protocol) [3]. MBGP is used to announce different unicast- and multicast-capable routes whereas MSDP exchanges active source information between the domains. The configuration complexity of this solution works against multicast deployment. On the other hand, backbone operators are currently over-provisioning their networks and thus have one reason less to use multicast. Nevertheless, ISPs (Internet Service Providers) have interest in multicast to face the increasing demand for network resources and content distribution. As a consequence, the Internet is likely to be organized with both unicast- and multicast-enabled networks. Therefore, it is of utmost importance to design protocols that allow the progressive deployment of the multicast service by supporting unicast clouds.

Different solutions that simplify the multicast service by reducing the distribution model were proposed [4]. EXPRESS [5] restricts the multicast conversation to 1 *to* $N$ (the *channel* abstraction), simplifying address allocation and data distribution, and still covering most of the current multicast applications. The source-specific multicast (SSM) [6] service is currently in final standardization phase at the Internet Engineering Task Force (IETF). The SSM service is implemented by Version 3 of Internet group management protocol (IGMP) [7] and by a modified version of protocol independent multicast—sparse mode (PIM-SM) [8], named PIM-SSM. Nevertheless, even if SSM largely simplifies the multicast model, it does not allow the *progressive* deployment of the multicast service. Currently, the only alternative is the use of tunnels to go through unicast-only networks.

Different multicast tunneling mechanisms have been proposed. One such mechanism is the UDP multicast tunneling protocol (UMTP) [9]. UMTP encapsulates UDP multicast datagrams inside UDP unicast datagrams, so it can be implemented as a user-level process at end-hosts. The work in [10] proposes a mechanism to automate the generation of UMTP tunnels. Automatic multicast tunneling (AMT) [11] is an alternative scheme that does not rely on UDP; it provides tunneling capability through pseudo network interfaces that act as default routes to multicast traffic. In this work, we do not propose a new automatic tunneling mechanism to connect the multicast-enabled parts of the Internet, but we propose instead a new multicast routing protocol that inherently supports unicast routers. Additionally, the protocol design takes into account the unicast routing asymmetries that may affect the structure of the multicast distribution tree, especially if unicast-only routers are present.

Manuscript received June 17, 2003; revised December 3, 2004, December 6, 2004, and June 20, 2005; approved by IEEE/ACM Transactions on Networking Editor R. Cohen. This work was sponsored by CNPq, CAPES/PRODOC, COFECUB, and RNP/FINEP/FUNTTEL. An earlier version of this paper appears in the Proceedings of ACM SIGCOMM'01.

L. H. M. K. Costa and O. C. M. B. Duarte are with the Grupo de Teleinformática e Automação-Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil (e-mail: luish@gta.ufrj.br; otto@gta.ufrj.br).

S. Fdida is with the Laboratoire d'Informatique de Paris 6-Université Pierre et Marie Curie, 75252 Paris Cedex 05, France (e-mail: Serge.Fdida@lip6.fr).

Digital Object Identifier 10.1109/TNET.2006.876157

The ability to transparently support unicast routers is the main motivation of the hop-by-hop multicast routing protocol (HBH) we present in this paper [12]. HBH implements multicast distribution through recursive unicast trees, an approach originally proposed in REcursive UNIcast TrEes (REUNITE) [13]. REUNITE does not use class-D IP addresses for group identification, abandoning the IP Multicast addressing model. Additionally, REUNITE faces some problems in the presence of asymmetric unicast routes.

HBH uses the unicast infrastructure to do packet forwarding with smaller routing tables, similarly to REUNITE, but uses the channel abstraction to identify the group. Thus, HBH preserves compatibility with IP Multicast as it uses class-D IP addresses for group identification. HBH constructs shortest-path trees (SPTs) instead of reverse SPTs as most routing protocols do [14]–[17]. Consequently, HBH has the potential to provide better routes in asymmetric networks. Additionally, the tree management algorithm of HBH provides enhanced tree stability in the presence of group dynamics and reduces tree bandwidth consumption in asymmetric networks, compared to most of the alternative solutions: shared trees, application-layer trees, and REUNITE. The results obtained through simulation support our statements.

This paper is organized as follows. Section II presents the related work, motivations, and basic ideas of HBH, Section III describes the HBH protocol, and Section IV presents the performance comparison of HBH and other multicast protocols through simulation. Finally, Section V concludes the paper.

## II. Basic Principles of HBH Multicast

This section presents previous works related to this paper, namely the EXPRESS and REUNITE protocols. We then introduce the basic principles of HBH, as well as the problems caused by asymmetric unicast routing that motivated the design of our protocol.

### A. Related Work

EXPRESS [5] provides a simple solution to the multicast address allocation problem, introducing the channel abstraction that reduces the multicast conversation from $M$ *to* $N$ to 1 *to* $N$. A channel is identified by the pair $\langle S, G\rangle$, where $S$ is the unicast address of the source and $G$ is a class-D multicast address. The concatenation of a unicast address with a class-D address solves the address allocation problem since the unicast address is unique. The channel model introduced by EXPRESS also simplifies group management issues such as sender access control. Nevertheless, the SSM service [6], which was inspired by EXPRESS and standardized by the IETF, does not change the group management support provided by IGMP [7].

REUNITE [13] implements multicast distribution based on the unicast routing infrastructure. The basic motivation of REUNITE is that, in typical multicast trees, the majority of routers simply forward packets from one incoming interface to only one outgoing interface, because only a few routers are branching nodes [18]. In other words, for the Internet topology, the multicast protocol acts as a unicast protocol in most of the nodes. Nevertheless, all multicast protocols keep per-group information in all routers of the multicast tree. Then, the key idea of REUNITE is to separate multicast routing information in two tables: a multicast control table (MCT), which is stored in the control plane, and a multicast forwarding table (MFT), which is installed in the data plane. Nonbranching routers simply keep group information in their MCT, whereas branching nodes keep MFT entries which are used to recursively create packet copies to reach all group members.

REUNITE identifies a conversation by a $\langle S, P\rangle$ pair, where $S$ is the unicast address of the source and $P$ is a port number. Class-D IP addresses are not used. As receivers join the group REUNITE populates its tables to construct the distribution tree, using two control messages: $join$ and $tree$. $Join$ messages travel upstream from the receivers to the source, whereas $tree$ messages are periodically multicast by the source to refresh the soft-state of the tree. Only the branching nodes for the group $\langle S_1, P_1\rangle$ keep $\langle S_1, P_1\rangle$ entries in their MFTs. The control table, MCT, is exclusively used for tree construction, not for packet forwarding. Nonbranching routers in the $\langle S_1, P_1\rangle$ tree have MCT entries for $\langle S_1, P_1\rangle$ but no MFT entry.

### B. Multicast Distribution Through Recursive Unicast

The basic idea of recursive unicast is that packets have *unicast* destination addresses. The routers that act as branching nodes for a specific multicast group are responsible for the creation of packet copies with *modified* destination address in such a way that all group members receive the information. Fig. 1(a) gives an example of the recursive unicast data distribution in REUNITE. In this figure, the source is $S$, $r_i$ is a receiver, and $R_j$ is a REUNITE router. The source sends data in unicast to the first receiver that joined the group. At a branching node, $R_B$, incoming packets are addressed to the first receiver, $r_i$, that joined the group in the subtree below $R_B$. The receiver $r_i$ is stored in a special MFT entry, $\mathrm{MFT}\langle S\rangle.dst$. The router $R_B$ creates one packet copy for each receiver in its MFT. The destination address of each packet copy is set to the unicast address of the receiver. The original packet is also forwarded to $r_i$. In the example of Fig. 1(a), $S$ produces data packets addressed to $r_1$, and these packets reach $r_1$ unchanged. The router $R_1$ creates one packet copy and sends it to $r_4$. Since $R_3$ is a nonbranching node, it simply forwards the packets without consulting its MFT. The router $R_5$ creates one packet copy to $r_8$, and finally $R_7$ creates copies to $r_5$ and $r_6$.

Fig. 1(b) shows how the recursive unicast data distribution works for our protocol, HBH.[1] In this figure, $H_j$ is an HBH router. The source $S$ sends data addressed to $H_1$. The router $H_1$ creates two packet copies and sends them to $H_4$ and $H_5$ (the next downstream branching nodes). The router $H_3$ simply forwards the packets in unicast. $H_5$ receives the data and sends a modified packet copy to $H_7$ and $r_8$. Finally, $H_7$ creates one packet copy to $r_4$, $r_5$, and $r_6$. Data distribution is symmetric on the other side of the tree.

The recursive unicast technique allows the progressive deployment of the multicast service because data forwarding is

[1]In the rest of the paper, we interchangeably use $\langle S\rangle$ and $\langle S, P\rangle$ to refer to REUNITE's multicast group and $\langle S\rangle$ and $\langle S, G\rangle$ to refer to HBH's multicast channel.

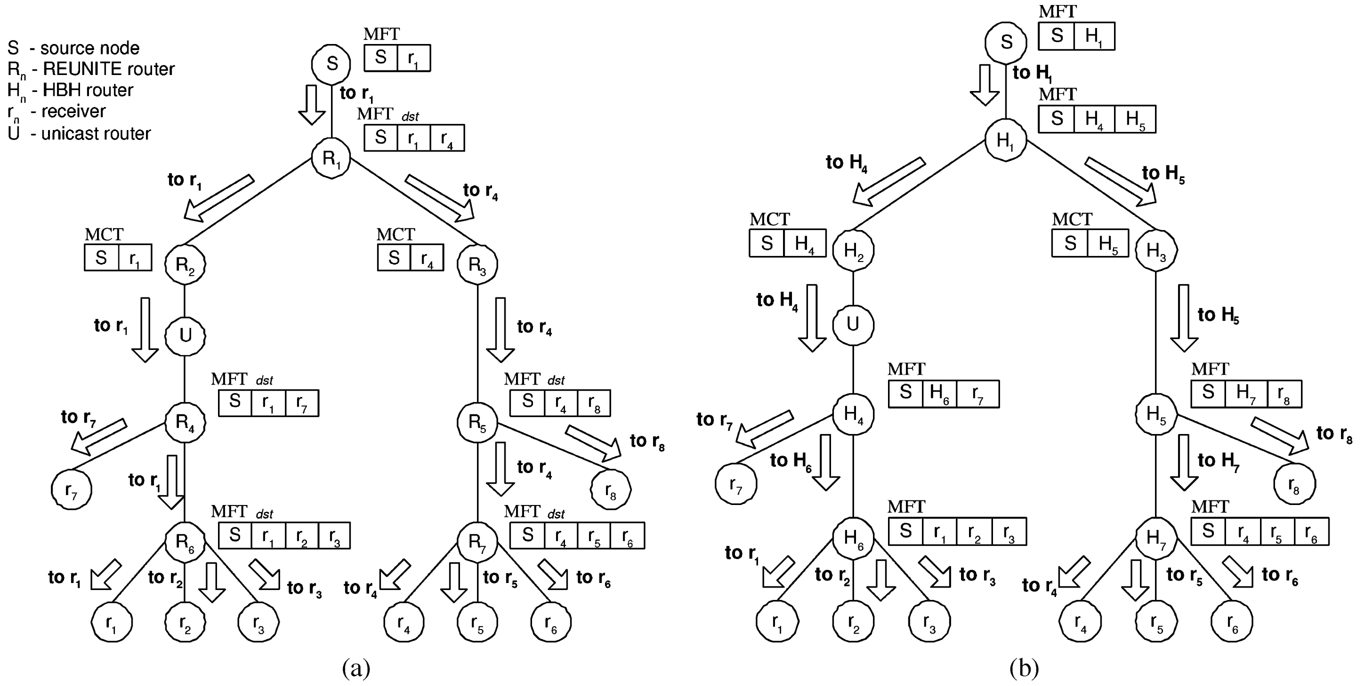


Fig. 1. Data distribution in the recursive unicast approach. (a) REUNITE tree. (b) HBH tree.

based on unicast addresses. Unicast-only routers in the distribution tree are transparently supported. These routers are unable to be branching nodes of the tree, but can still forward data since unicast destination addresses are used.

## C. Risks of Asymmetric Unicast Routing

Asymmetric routing means that the unicast path from $A$ to $B$ may differ from the path from $B$ to $A$. In the Internet, it may be due to different reasons [19]. The simplest case is that of asymmetric or unidirectional links (e.g., ADSL lines or satellite links). There are also less obvious sources of asymmetric routes: routing misconfiguration and routes "intentionally" configured asymmetric. One such phenomenon is known as "hot-potato routing" and occurs because of economical reasons. For example, suppose two ISPs, $\mathbf{A}$ and $\mathbf{B}$, both provide connectivity through the US territory. Traffic generated on the east coast in network $\mathbf{A}$ and destined to a customer on the west coast connected to $\mathbf{B}$ will be routed to network $\mathbf{B}$ as soon as possible, i.e., in a peering point located on the east coast. This way, $\mathbf{A}$ avoids using its own links to cross the country since these links are a more scarce resource. In the other direction, $\mathbf{B}$ uses the same strategy, causing routes between $\mathbf{A}$ and $\mathbf{B}$ to be asymmetric.

Measurements in the Internet have shown a high percentage of asymmetric routes. The analysis in [19] evaluated about 10 000 pairs of sites. Only major routing asymmetries were considered, where the virtual paths differ by one city or autonomous system (AS). About half of the measures revealed routes that differ by one city or more. In a different level of granularity, about 30% of the routes were asymmetric with at least one AS of difference, which is still high.

Asymmetric unicast routing affects multicast routing since most multicast routing protocols construct *reverse* SPTs [14], [16], [17]. Data packets from the source to a receiver follow the unicast route used to go from the receiver to the source. If these paths have different characteristics, the use of the reverse SPT may be a problem to any QoS mechanism which collects information about the path from the data source to the receiver, e.g. the path loss. The ability to construct SPTs is therefore advantageous for a multicast routing protocol.

REUNITE differs from previous protocols because it tries to construct SPTs. (multicast OSPF [20] is the only Internet protocol that constructs SPTs.) This is possible because the $tree$ messages that travel from the source to the destination nodes install the forwarding state and not the $join$ messages that follow the inverse direction. Nevertheless, REUNITE may fail to construct shortest-path branches in the presence of unicast routing asymmetries. A second undesirable behavior of REUNITE is that the route for one receiver may change after the departure of another receiver. This is undesirable if some QoS mechanism is to be implemented.

## D. Problems With the Operation of REUNITE

Fig. 2 illustrates the tree construction mechanism of REUNITE and gives an example where it fails to construct a SPT. Suppose the following unicast routes: $r_1 \rightarrow R_2 \rightarrow R_1 \rightarrow S$; $S \rightarrow R_1 \rightarrow R_3 \rightarrow r_1$; $r_2 \rightarrow R_3 \rightarrow R_1 \rightarrow S$; $S \rightarrow R_4 \rightarrow r_2$. Suppose the following events: $r_1$ joins $\langle S, P\rangle$, $r_2$ joins $\langle S, P\rangle$, and $r_1$ leaves the group.

Receiver $r_1$ joins the multicast group by sending a $join(S, r_1)$ message to $S$. This message reaches $S$ since there is no previous tree state for this group in the network. We say that $r_1$ *joined* the group $\langle S, P\rangle$ at $S$. The source $S$ then starts sending $tree(S, r_1)$ messages to $r_1$ (in unicast). These $tree$ messages install soft-state for $\langle S, P\rangle$ in the routers traversed downstream. The routers $R_1$ and $R_3$ create a $\langle S, r_1\rangle$ entry in their MCTs. Now $r_2$ joins the group. The $join(S, r_2)$ travels in the direction of $S$ reaching the tree at $R_3$. The router $R_3$ drops the $join(S, r_2)$ message, creates an $\mathrm{MFT}\langle S\rangle$ with $r_1$

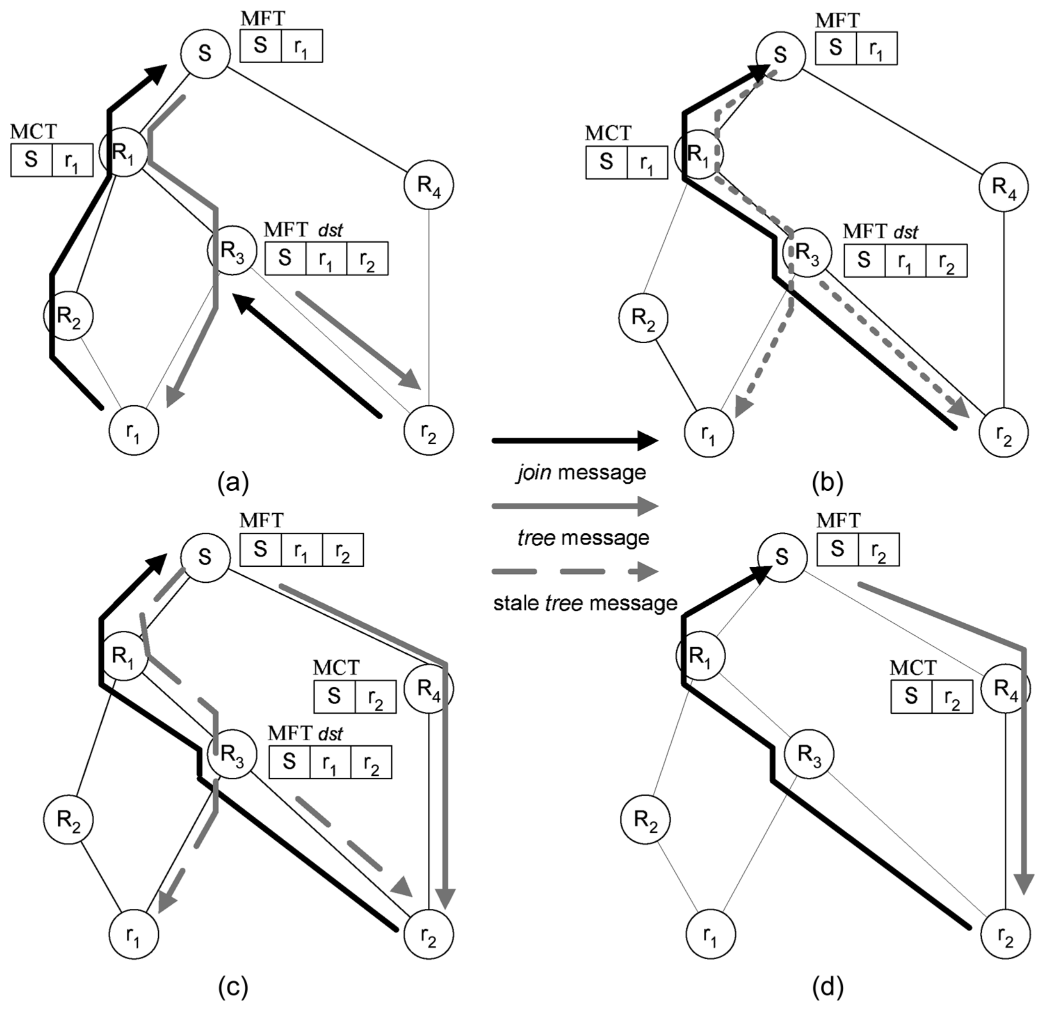


Fig. 2. REUNITE's tree construction mechanism. (a) $r_1$, then $r_2$ join the group. (b) $r_1$ leaves the group. (c) $r_2$ is connected to S. (d) Tree after $r_1$ left.

as $dst$, adds $r_2$ to $\mathrm{MFT}\langle S\rangle$, and removes $\langle S, r_1\rangle$ from its MCT. The router $R_3$ becomes a branching node and will consequently forward $tree(S, r_2)$ messages downstream, upon the reception of $tree(S, r_1)$ messages from upstream. We say that $r_2$ joined the group at $R_3$. Data packets sent to the multicast group, which are addressed to $r_1$, are now duplicated at $R_3$ and addressed to $r_2$. Subsequent $join$ messages sent by $r_1$ and $r_2$ refresh the MFT entries at $S$ and $R_3$, respectively.

In this configuration, $r_1$ receives data from $S$ through the shortest path, but not the receiver $r_2$. Because the unicast routes between $S$ and $r_2$ are asymmetric, and because $R_3$ intercepts the $join(S, r_2)$ message, data follows the path $S \rightarrow R_1 \rightarrow R_3 \rightarrow r_2$, the same as $tree$ messages from $S$ down to $r_2$ [Fig. 2(a)].

MCT and MFT states are soft. Receivers periodically send $join(S, r_i)$ messages and the source periodically multicasts a $tree(S, r_i)$ message. A receiver simply stops sending $join$ messages to leave the multicast group. When the tree structure is stable, a $tree(S, r_i)$ message refreshes the $r_i$ MCT entries and the $\mathrm{MFT}.dst = r_i$ entries down the tree. The $join(S, r_j)$ messages refresh the $r_j$ entry in the MFT of the node where $r_j$ joined $\langle S\rangle$. For example, in Fig. 2, $join(S, r_1)$ refreshes the $r_1$ entry in the MFT of $S$ and $join(S, r_2)$ refreshes the $r_2$ entry in the MFT of $R_3$.

Now, $r_1$ leaves the group: it stops sending $join(S, r_1)$ messages. As the $r_1$ entry in $S$'s MFT is not refreshed, after the expiration of timer $t1$, the $r_1$ entry becomes stale. A second timer, $t2$, is created and will eventually destroy the $r_1$ entry. As $r_1$ is stale, $S$ now sends $stale\ tree(S, r_1)$ messages [Fig. 2(b)]. The stale $tree(S, r_1)$ messages indicate that the data flow addressed to $r_1$ will soon stop, thus the tree portion based on $r_1$ has to be reconfigured. At the branching nodes, MFT tables that have $\mathrm{MFT}\langle S\rangle.dst = r_1$ become stale as the stale $tree$ travels down the tree. At nonbranching nodes, the reception of a stale $tree(S, r_1)$ causes the destruction of any $r_1$ MCT entries. Consequently, $join(S, r_2)$ messages are no more intercepted by $R_3$ (as its $\mathrm{MFT}\langle S\rangle$ is stale) and reach $S$. The receiver $r_2$ now joins $\langle S, P\rangle$ at $S$ [Fig. 2(c)]. Eventually $t2$ times out resulting in the deletion of $r_1$ from $S$'s and $R_3$'s MFTs. As $R_3$ stops receiving $tree$ messages, its $\mathrm{MFT}\langle S\rangle$ is destroyed [Fig. 2(d)]. Now, $r_2$ receives data through the shortest path from $S$.

Asymmetric routing may also lead REUNITE to unneeded packet duplications on certain links. In fact, this possibility also exists when the network has pure unicast routers or when the REUNITE router is overloaded. In both cases, the branching node must migrate to another router and may cause packet duplications in some links. For a more detailed description, the reader is referred to [13].

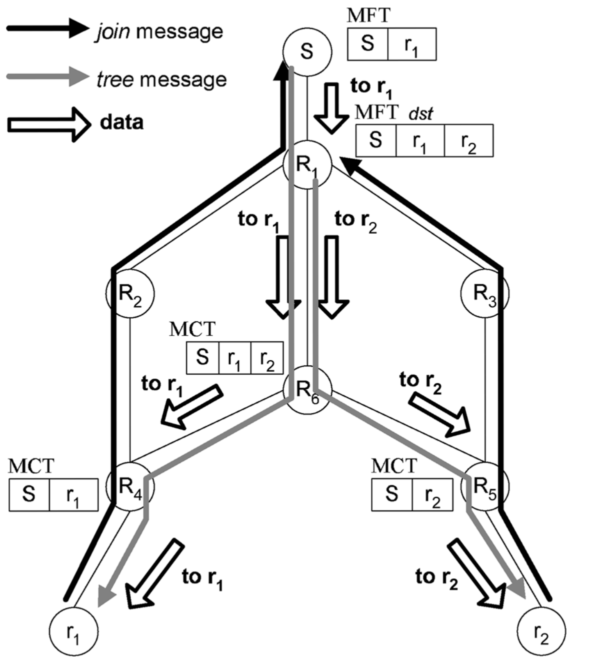


Fig. 3. Packet duplication due to asymmetric routes in REUNITE.

Fig. 3 gives an example of packet duplication due to asymmetric routes. The first receiver $r_1$ sends a $join(S, r_1)$ that follows the path $r_1 \rightarrow R_4 \rightarrow R_2 \rightarrow R_1 \rightarrow S$. The $tree(S, r_1)$ messages follow the route $S \rightarrow R_1 \rightarrow R_6 \rightarrow R_4 \rightarrow r_1$. Suppose now that $r_2$ joins and that $join(S, r_2)$ follows $r_2 \rightarrow R_5 \rightarrow R_3 \rightarrow R_1 \rightarrow S$. The $tree(S, r_1)$ (produced by $S$) and the $tree(S, r_2)$ (created at $R_1$) both traverse the link $R_1 - R_6$. As $R_6$ does not receive $join$ messages from these receivers, it is not identified as a branching node. The source $S$ creates data packets to $r_1$ and $R_1$ creates packets to $r_2$. Therefore, there are two packet copies on the link $R_1 - R_6$.

We define the cost of the distribution tree as the number of copies of the same packet that need to be generated so that all receivers get the packet. Therefore, the cost of a regular multicast tree, such as the PIM-SSM tree, is equal to the number of links in the tree. On the other hand, for application-layer multicast protocols, REUNITE, HBH, and when tunneling is used, there may be more than one instance of the same packet being transmitted on a link. Our cost definition takes the additional copies into account. The number of copies of the same packet transmitted on a single link is often called link stress. As a consequence, the cost of a REUNITE tree may be larger than that of a source tree constructed by a classic multicast routing protocol such as PIM-SM [16], because the reverse path forwarding (RPF) algorithm ensures that only one packet copy is ever transmitted over each network link.

The first simulations with REUNITE showed that in some cases its algorithm uses a huge number of control messages. The investigation revealed a temporary loop creation, which needs to be solved by an additional check, e.g. the list of routers traversed by a control message. HBH, as shown in the next section, does not suffer from temporary loops.

## III. HBH Multicast Protocol

The HBH multicast protocol has a tree construction algorithm that is able to better deal with the pathological cases due to asymmetric unicast routes. HBH uses two tables, an MCT and an MFT, that have nearly the same function as in REUNITE. The difference is that one entry table in HBH stores the address of a *next branching node* instead of the address of a *receiver*, except the branching router nearest the receiver. The MFT has no $dst$ entry. Data received by a branching router, $H_B$, has unicast destination address set to $H_B$ (in REUNITE, data are addressed to $\text{MFT}\langle S\rangle.dst$). This choice makes the tree structure more stable than in REUNITE.

A multicast channel in HBH is identified by $\langle S, G\rangle$, where $S$ is the unicast address of the source and $G$ is a class-D IP address allocated by the source. This definition solves the address allocation problem while being compatible with SSM's channel definition. Therefore, HBH can support IP Multicast clouds as leaves of the distribution tree.

The tree structure of HBH has the advantage of an enhanced stability of the table entries when compared with REUNITE. The tradeoff is that in HBH each data packet received by a branching node produces $n$ modified packet copies while in REUNITE it produces $n - 1$ modified packets, assuming that $n$ is the number of outgoing multicast links. The tree management scheme of HBH minimizes the impact of member departures in the tree structure. This is possible because the MFT receiver entry is located at the branching node nearest the receiver. For example, the departure of $r_1$ in Fig. 1(a) has a greater impact on the tree structure of REUNITE than on that of HBH. For REUNITE, the routing tables of $R_6$, $R_4$, $R_2$, and $R_1$ are modified, whereas only the table of $H_6$ is modified for HBH. In the worst case, HBH may need one more change than REUNITE, when a branching node becomes a nonbranching one (e.g., after the departure of $r_8$). In this example, there is no route changes for other members when a member leaves the group because the unicast routes are symmetric. Nevertheless, tree reconfiguration in REUNITE may cause route changes to the remaining receivers, as for $r_2$ in the example of Fig. 2. This is avoided in HBH.

### A. Tree Management in HBH

HBH has three control messages: $join$, $tree$, and $fusion$. $Join$ messages are periodically unicast by the receivers in the direction of the source and are used to refresh the forwarding state (MFT entry) at the router where the receiver was connected to the tree. A branching router itself "joins" the multicast channel at the next upstream branching router. The $join$ messages may be intercepted by the branching routers, which must sign themselves $join$ messages, and filter the $join$ messages received from downstream nodes. The source periodically multicasts a $tree$ message that refreshes the tree structure. $Fusion$ messages are sent by potential branching routers and construct the distribution tree together with the $tree$ messages.

The Appendix gives a detailed description of the control messages processing rules of HBH. The basic ideas are: the first $join$ issued by a receiver is never intercepted, reaching the source, and the $tree$ messages are periodically multicast by the source. These $tree$ messages are combined with $fusion$

messages, sent by potential branching nodes, to construct and refine the tree structure.

### B. Routing Tables

The HBH protocol uses two routing tables, an MCT and an MFT. Consider a multicast channel $\langle S, G\rangle$ implemented by the source $S$. Each HBH router in the distribution tree of the source $S$ has either a $\mathrm{MCT}\langle S, G\rangle$ or an $\mathrm{MFT}\langle S, G\rangle$.

The state kept in HBH's multicast routing tables is soft and periodically refreshed by the protocol control messages. Each receiver $r$ periodically sends a $join(S, r)$ in unicast to the source. The source periodically multicasts a $tree$ message for each $\langle S, G\rangle$ channel.

Routers implementing HBH which are in the distribution tree of $S$ have $\langle S, G\rangle$ entries in their routing tables. Normally, nonbranching routers have $\langle S, G\rangle$ entries in their MCT. Branching routers have $\langle S, G\rangle$ entries in their MFT. Nevertheless, a nonbranching router may also have an MFT, in some configurations, to cope with asymmetric routes (see Section III-B3).

*1) MCT Entries:* An HBH router in $S$'s distribution tree may have an $\mathrm{MCT}\langle S, G\rangle$. The $\mathrm{MCT}\langle S, G\rangle$ has a single entry, which can be *fresh*} or *stale*. Therefore, two timers are associated to the MCT entry, $t1$ and $t2$. At the expiration of $t1$ the MCT becomes stale and at the expiration of $t2$ the MCT is destroyed.

MCT entries are not used to replicate data. Depending on its state, different actions are taken upon reception of control messages. These actions are described in detail in the Appendix.

*2) MFT Entries:* An HBH router in the distribution tree of the source $S$ may also have an $\mathrm{MFT}\langle S, G\rangle$. The MFT entries are also soft-state. Two timers, $t1$ and $t2$, are associated with each entry in $\mathrm{MFT}\langle S, G\rangle$. When $t1$ times out, the MFT entry becomes stale and it is destroyed when $t2$ expires.

The MFT entries stored for channel $\langle S, G\rangle$ may be in one of three states:

- *default*—The default state of an MFT entry is to be *fresh* (not stale) and *unmarked*. In default state, the MFT entry is used for both data and control forwarding.
- *stale*—If the MFT entry is *stale*, it is used to forward data (avoiding service disruption during tree reconfiguration) but it is no longer used to forward control messages (there is no notion of stale $tree$ message in HBH).
- *marked*—The MFT entry may also be *marked*. As opposed to a *stale* entry, a *marked* entry is used to produce $tree$ control messages but does not participate in data replication.

*3) Branching Versus Nonbranching Nodes:* There is one peculiarity of HBH regarding branching nodes. A branching node is a router that has more than one outgoing multicast link, which is the common definition of a branching node. Therefore, an HBH branching router has MFT state. Nevertheless, a nonbranching router may also keep MFT state, in certain scenarios, as a consequence of the algorithm we devised to cope with asymmetric routes. The difference between a "regular" nonbranching router, which only has MCT state, and a nonbranching router which has an MFT is that the first one simply forwards packets in unicast, whereas the second one modifies the destination addresses of the data packets it forwards. The example in Fig. 5 illustrates one such scenario.

### C. Data Forwarding

HBH uses IP unicast destination addresses for data packets. Data packets are replicated at HBH branching routers, in such a way that all the receivers connected to the multicast channel receive the data.

Data forwarding in HBH works as follows. Suppose a source, $S$, and the source's first-hop router $H_1$. The source produces IP packets which have the IP source address set to $S$'s IP address and the IP destination address set to $H_1$'s IP address. The router $H_1$ creates copies of the data packet, according to its MFT. The copies have the IP destination address set to the unicast address stored in the MFT (typically, that is the address of the next-hop HBH router). Packets are recursively replicated according to the distribution tree, until they get to the tree leaves.

There are two possibilities for the tree leaves. The leaf may be the receiver's first-hop router (the last-hop router from the source) or the receiver itself. In the first case, the receiver use IGMP to tell the first-hop router which multicast channel it is willing to listen to, just as in classical IP Multicast. HBH requires IGMPv3 because the multicast group and the source IP addresses must be passed to the local HBH router. The other possibility is for the receiver to connect itself to the multicast tree. In that case, its local HBH router would have one entry in its MFT for each receiver. The first solution is the preferred configuration, because it reduces the number of MFT forwarding entries in the local router and because it does not need any modification in the receiver's operating system—the only requirement is version 3 of IGMP.

Similarly, there are two choices for the source. The first one is for the source, $S$ to produce data packets with unicast destination address set to $H_1$. In that case, there is no modification in the operating system of the source station, but the application may need modification to allow two unicast destination addresses for the channel (normally, the application would use a multicast destination address, say, $M$, and the source's unicast address for the channel). The other possibility is to make no modifications at all in the station. In that case, the application produces data addressed to $\langle M, S\rangle$, just as it would for SSM. The first-hop HBH router is responsible for modifying the data packets with the next hop's unicast destination address. This configuration is probably better, since it requires no modification in the station (there are no changes in the service interface) but only in the first-hop router, which anyway is modified to implement HBH.

### D. Operation of HBH

To illustrate the operation of HBH, we repeat the examples of Section II-C. First, Fig. 4 repeats the scenario of Fig. 2. The receiver $r_1$ joins the multicast channel at the source $S$, which starts producing $tree(S, r_1)$ messages. These messages create a $\mathrm{MCT}\langle S\rangle$ containing $r_1$ at routers $H_1$ and $H_3$ [Fig. 2(a)]. When $r_2$ joins the group, by sending the first $join(S, r_2)$, this message is not intercepted and reaches $S$ (the first $join$ message is never intercepted). The $tree(S, r_2)$ produced by the source create $\mathrm{MCT}\langle S\rangle$ state at router $H_4$ [Fig. 4(b)]. Both receivers are connected to the source through the shortest path. This can only

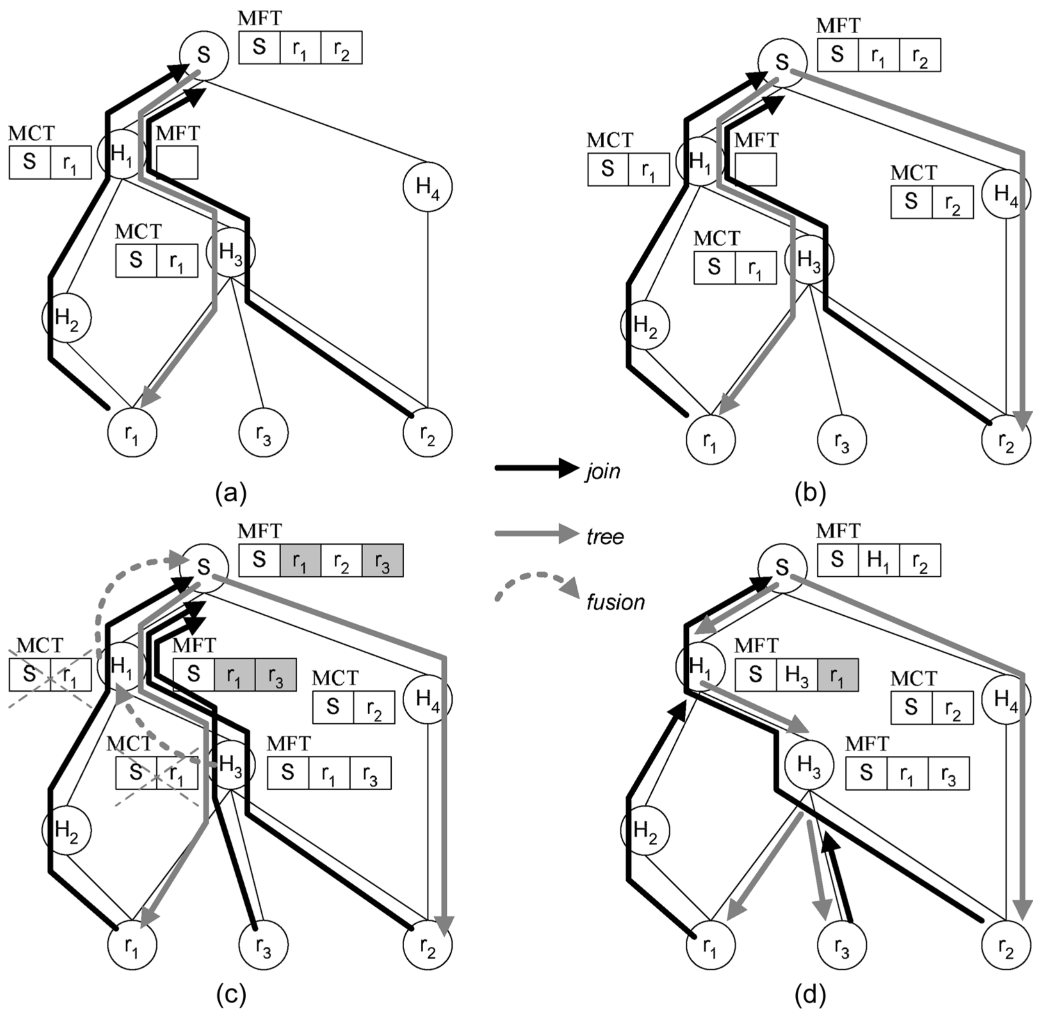


Fig. 4. HBH's tree construction mechanism. (a) $r_1$, then $r_2$, subscribe the channel. (b) $r_2$ is connected to $S$. (c) $r_3$ subscribes the channel. (d) $r_1$ leaves the channel.

be guaranteed because the *first join* message always reaches the source $S$.

Suppose now that the receiver $r_3$ (unicast routes: $S \rightarrow H_1 \rightarrow H_3 \rightarrow r_3$ and $r_3 \rightarrow H_3 \rightarrow H_1 \rightarrow S$) joins the channel. It sends a $join(S, r_3)$ to $S$, which starts sending $tree(S, r_3)$ messages. As $H_1$ receives two different $tree$ messages, it sends a $fusion(S, r_1, r_3)$ to the source. The reception of the $fusion$ message causes $S$ to mark the $r_1$ and $r_3$ entries in its MFT and to add $H_1$ to it. In the same way as $H_1$, $H_3$ receives $tree(S, r_1)$ and $tree(S, r_3)$ messages and thus sends a $fusion(S, r_1, r_3)$ to the source [Fig. 4(c)]. $H_3$'s MFT now contains $r_1$ and $r_3$. Subsequent $join(S, r_1)$ messages are intercepted by $H_1$ and refresh the $r_1$ marked entry in $H_1$'s MFT. The $join(S, r_3)$ messages refresh the $r_3$ MFT entry at $H_3$. The source $S$ sends data addressed to $H_1$ that forwards it addressed to $H_3$. The router $H_3$ sends copies to $r_1$ and $r_3$. Afterwards, as $S$ receives no more $join(S, r_1)$ or $join(S, r_3)$ messages, its corresponding MFT entries time out and are destroyed. The final structure is shown in Fig. 4(d). In this way, HBH discovers the ideal branching point for the distribution tree.

The $fusion$ messages used by HBH cope with the problem of Fig. 3. The first $join(S, r_2)$ will reach the source. After receiving different $tree$ messages for $r_1$ and $r_2$, router $H_1$ sends a $fusion(S, \{r_1, r_2\})$ to $S$ (Fig. 5). Subsequent $join(S, r_1)$ and $join(S, r_2)$ messages will be intercepted by $H_1$. At its turn, router $H_6$ receives two different $tree$'s and sends a $fusion(S, \{r_1, r_2\})$ upstream. In this case, however, it will never receive $join$ messages issued by receivers $r_1$ and $r_2$. The consequence is that $H_6$'s entry in $H_1$ will be kept stale and $r_1$ and $r_2$ entries will be fresh, but *marked*. Thus, data will be produced to $H_6$ as control will be addressed to $r_1$ and $r_2$. The design choice of HBH imposes some control overhead but minimizes data duplication.

Temporary loop creation is avoided in HBH because the creation of forwarding state is always a consequence of messages that travel downstream, the $tree$ messages sent by the source to the receivers. $Fusion$ messages can also instantiate MFT state, but as the production of $fusion$ messages is itself consequence of different $tree$ message reception, *and* because the "fusion" process is done hop by hop, the produced forwarding structure is always a SPT.

## IV. Performance Analysis

We used Network Simulator (NS) [21] to compare HBH to other routing algorithms from the viewpoint of the constructed trees. We analyzed the average delay experienced by all of the

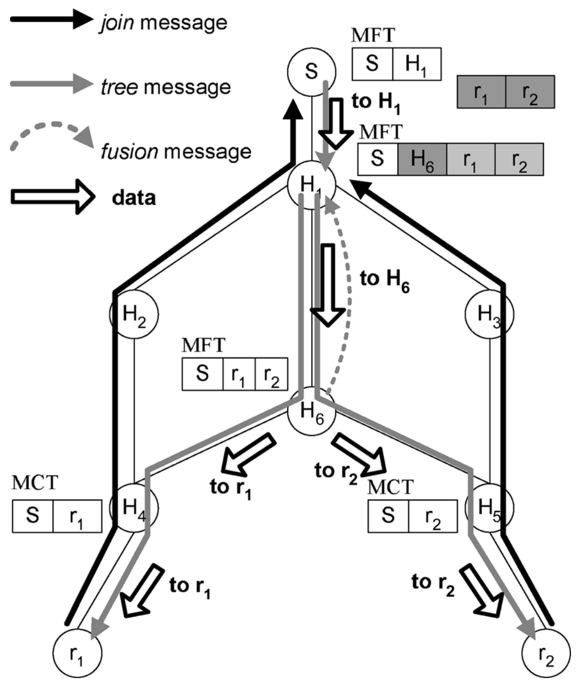


Fig. 5. Avoiding packet duplication in HBH.

receivers of the group and the number of copies of the same packet that are transmitted to reach all of the receivers.

## A. Simulation Scenario

The first topology used in our simulations has 18 nodes and is typical of a large ISP network [22]. Without loss of generality, we suppose that only one receiver is connected to each node in the topology. The presence of one or many receivers attached to a border router through IGMP [7] does not influence the cost of the tree, so we do not consider the aggregation provided by the multicast service at the local network level. Thus, in the 18-node topology, nodes 0–17 are routers, whereas nodes 18–35 are potential receivers of the multicast channel.[2] The second topology was randomly generated, with 50 nodes and higher connectivity (8.6 versus 3.3). Additionally, we used *nem* [23], which is a topology generator based on Internet map sampling. *Nem* randomly extracts a subgraph of a network map, keeping its original properties, such as: node degree, mean distance, mean eccentricity, and topology diameter. Magoni and Pansiot [23] show that the topologies generated by *nem* respect the power laws established on real Internet maps. The third topology we used has 500 nodes and was produced by *nem*.

As the NS does not implement BGP, we produced asymmetric routes by the use of different link costs. We associate with every link a cost in each direction. Hence, for the link $L_{i,j}$ composed by the nodes $n_i$ and $n_j$, we associate costs $c(n_i, n_j)$ and $c(n_j, n_i)$ that are integers randomly chosen in the interval [1], [10]. Simulations consider one multicast group from one source to $N$ receivers. The source node is fixed. A variable number of randomly chosen receivers join the channel. The plots show the average of 500 simulation runs per protocol per group size.

[2]REUNITE uses the term multicast group whereas HBH and SSM use multicast channel. In this section, we also use channel for REUNITE.

We compare HBH to REUNITE and two classical multicast protocols. NS has a routing protocol which is able to construct shared trees and source trees with the same structure as the trees constructed by the PIM-SM protocol. The difference is that the NS implementation is centralized, and the change from the shared tree to the source tree is triggered through an explicit command and not automatically as in the original PIM-SM [16]. Therefore, PIM-SM in our simulations is a protocol similar to the real implementation of PIM-SM but that exclusively constructs shared trees, whereas PIM-SSM is a protocol that only constructs source trees. The tree structure PIM-SSM is a reverse SPT. In addition to HBH, we implemented REUNITE according to [13]. All routers implement the multicast service in the first set of experiments, i.e., all routers implement HBH or REUNITE. No tunneling through unicast clouds was needed.

## B. Tree Cost Analysis

We first evaluated the cost of the trees constructed by the different multicast routing protocols. We define the cost of a tree as the number of copies of the same packet that are transmitted in the network links. This cost definition takes the additional copies, or link stress, into account. The cost of a PIM-SM or PIM-SSM tree is equal to the number of links in the tree. Nevertheless, the tree cost for a REUNITE, HBH, or application-layer multicast tree may be different from the number of links in the tree, because one packet may be sent more than once on the same link. The recursive unicast technique may send more than one copy of the same packet over a specific link. This may be due to routing asymmetries (as shown is Section II-C) but also to unicast routers inside the network that are not able to be branching nodes. In this case, the location of a branching node may not be ideal. The same is valid when we use automatic tunneling or application-layer multicast, because the physical network topology is not known by the end systems. In our experiments all routers are multicast capable, therefore extra packet copies produced by REUNITE or HBH are a consequence of routing asymmetries.

Fig. 6 shows the average cost of the multicast trees constructed by the different protocols as the number of receivers varies. For the 18-node ISP topology, PIM-SM constructs the trees with the highest cost in most cases. This result was expected since PIM-SM constructs shared trees. As we simulated the distribution from one source to many receivers, the utilization of a shared tree is disadvantageous since the tree is centered on a *rendezvous* point (RP). With a high probability, this tree has a higher cost than the equivalent source tree. HBH and PIM-SSM construct the cheapest trees. This result is expected since PIM-SSM constructs source trees based on the RPF algorithm, which guarantees that, at the maximum, one copy of the same packet is transmitted over each link, and that each receiver is connected to the source through the *reverse* shortest-path. HBH performs similar to PIM-SSM because in HBH each receiver is connected to the source through the shortest path. Using this path or the reverse shortest path does not influence tree cost.

The REUNITE curves in Fig. 6 demonstrate that the tree construction mechanism of REUNITE suffers from the pathologies produced by asymmetric unicast routing, investigated in

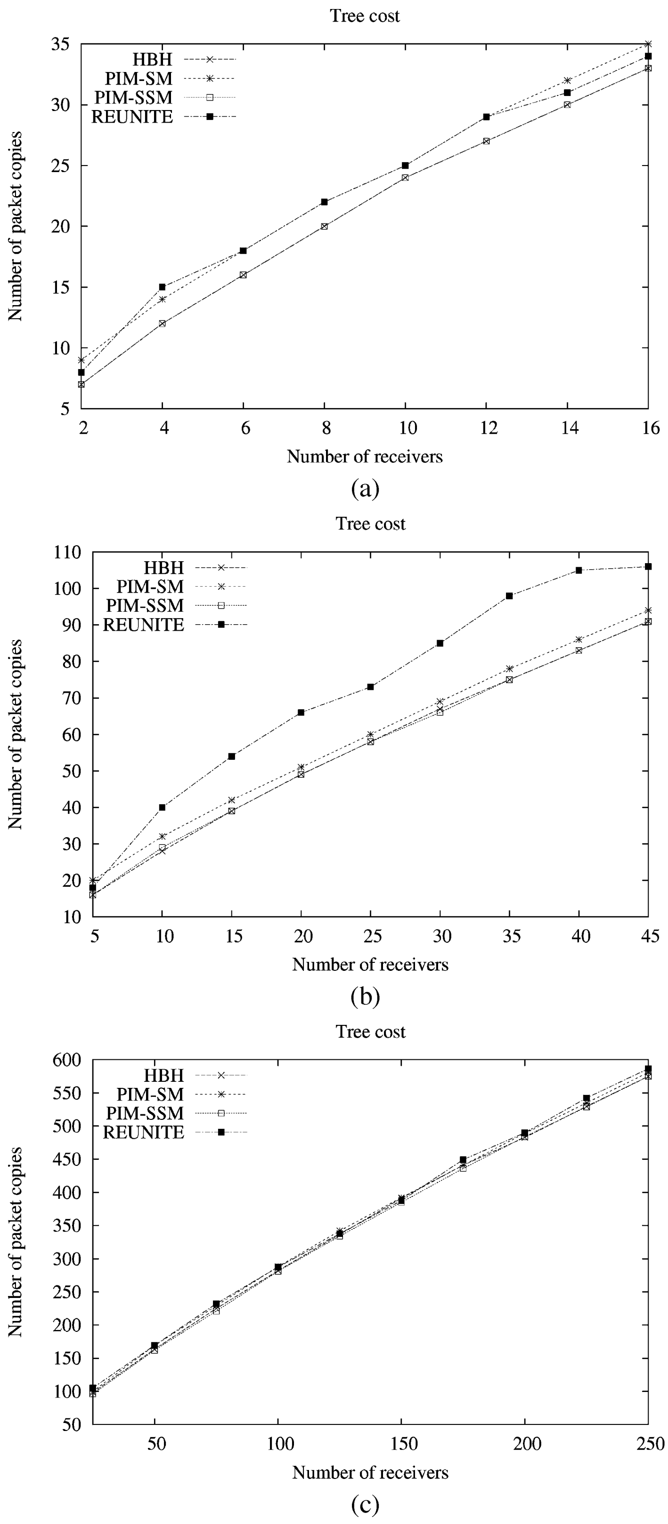


Fig. 6. Tree cost. (a) 18-node IS topology. (b) 50-node random topology. (c) 500-node Internet-like topology.

Section II-C. The phenomenon is less frequent with a small number of receivers, since the probability that two receivers share the same link in the multicast tree is smaller. For the ISP topology, the problem is also less severe when the number of receivers is huge (receiver distribution is dense) since a large percentage of network links is used in the distribution tree anyway. Nevertheless, this is not the case for the 50-node topology. This topology has a much higher connectivity, which means that a smaller percentage of network links is used. In this topology, the advantage of HBH increases with the group size. The PIM-SM shared trees also outperform REUNITE. This is a consequence of badly placed branching nodes, which lead REUNITE to useless packet duplication. With the 500-node topology, the gains of HBH over REUNITE are approximately constant and around 5%. The difference is almost constant because the number of receivers represents a smaller percentage of the total number of nodes than for the other two topologies. Furthermore, the connectivity of the 500-node Internet-like topology is similar to the 18-node ISP topology, giving percent gains in the same order of magnitude.

The analysis of the HBH curve shows the enhanced efficiency of the tree construction mechanism of HBH. In terms of tree cost, the advantage of HBH over REUNITE, averaged gain over all group sizes, is as large as 5% for the 18-node ISP topology, 18% for the 50-node topology, and 5% for the 500-node Internet-like topology. We conclude that HBH potentially provides a better bandwidth utilization than REUNITE for asymmetric networks.

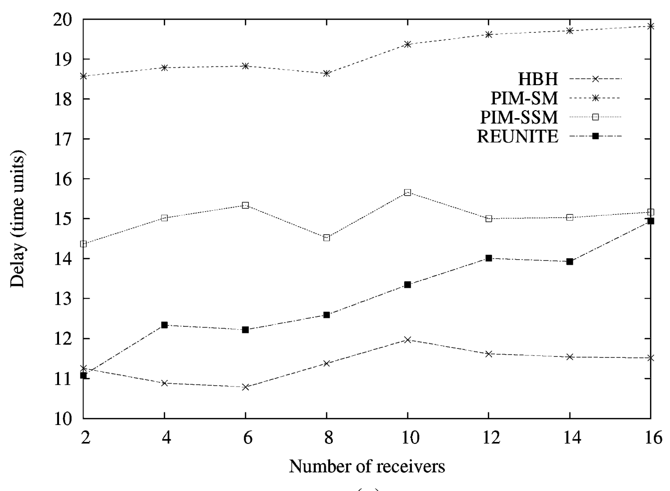
## C. Delay Analysis

Fig. 7 presents the average delay experienced by the receivers in the multicast channel for the same set of simulations. The curves show that HBH is effectively able to generate better quality routes than REUNITE in the presence of asymmetric unicast routing.

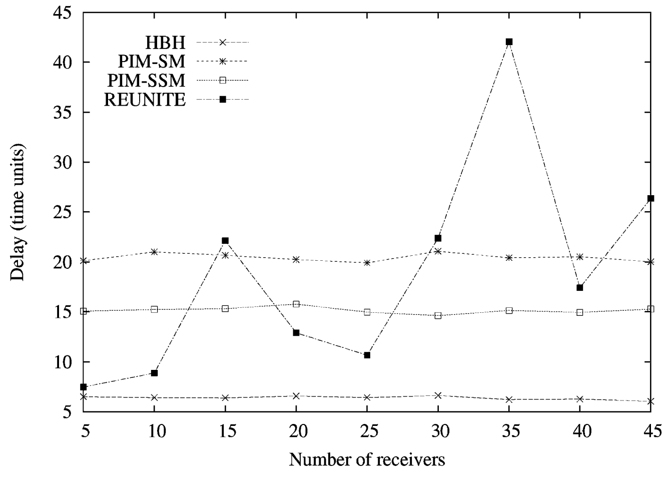
Fig. 7(a) shows two unexpected results. First, PIM-SM performs better that PIM-SSM in terms of delay for the ISP topology. In other words, this result means that the shared trees have a better delay performance than the source trees. This fact is explained because PIM-SSM tree is a *reverse* SPT and not a SPT and, as a consequence, delay is not minimized. Delay is not minimized either in the shared tree constructed by PIM-SM. But, in the shared tree, all of the paths from the source to the different receivers have one common portion, namely, the path between the source and the RP. As data are encapsulated in unicast between the source and the RP, delay is minimized between the source and the RP. Hence, paths in the PIM-SM tree have two parts: one from the source to the RP, where the delay is minimized, and another from the RP to the receiver, where the delay is not minimized since it is a *reverse* shortest path. This explains the advantage of PIM-SM over PIM-SSM. The same was not observed for the 50-node topology because this topology is larger and has a higher connectivity. Therefore, going through the RP is likely to result in a longer path than going directly from the source to the receiver. As expected, the shared tree has the worst delay performance in this case.

The second important remark for the 18-node ISP topology is that the effect of the network asymmetries in the quality of REUNITE trees may be strong, as REUNITE performed worse than PIM-SM when the receiver set is large. REUNITE performs better than PIM-SM in the 50-node topology because this topology has a higher connectivity.

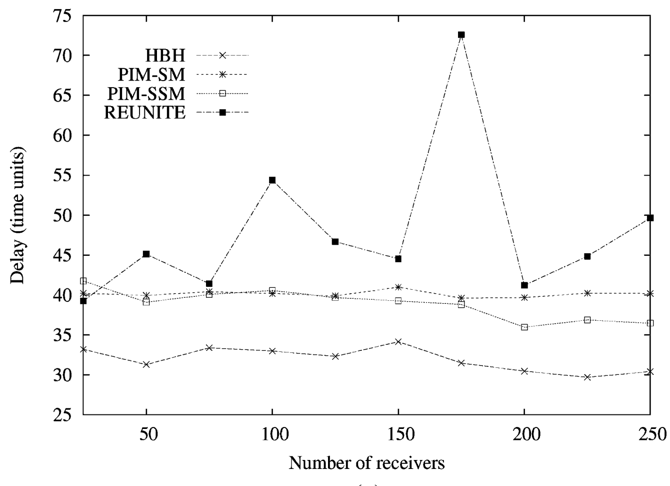
The delay performance of HBH is better than that of REUNITE for all group sizes, and for all topologies. The advantage



(a)



(b)



(c)

Fig. 7. Receiver average delay. (a) 18-node ISP topology. (b) 50-node random topology. (c) 500-node Internet-like topology.

becomes larger as the number of receivers grows, being of 14% in average for the 18-node ISP topology. The absolute values for the 50-node random topology are smaller as this topology has a higher connectivity. On the other hand, the advantage obtained by HBH over REUNITE for this topology is larger, 30% in average. This is a consequence of its richer connectivity, as the routing protocol has more choices to construct the distribution tree, and is consequently more vulnerable to unicast routing asymmetries. For the 500-node Internet-like topology, the advantage of HBH over REUNITE is of 8% in average. The difference is smaller because the connectivity of this topology is similar to the 18-node ISP topology and because the maximum number of receivers is a small fraction of the total number of nodes in this topology. The delay gains increase with the group size.

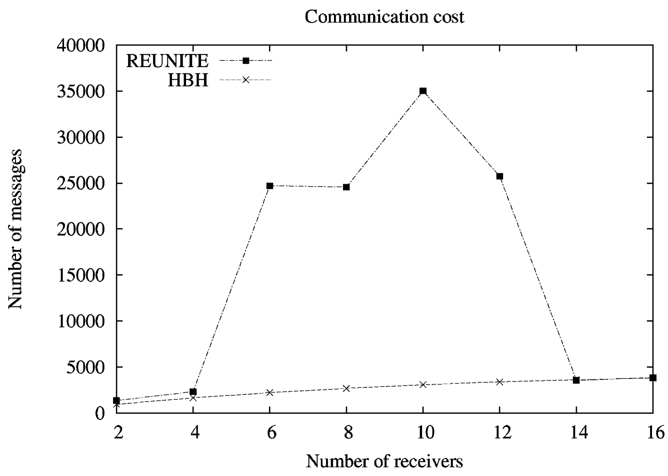
## *D. Message Cost Analysis*

In this section, we compare the message processing overheads of HBH and REUNITE, i.e., the amount of control messages used by the protocol to keep the multicast distribution tree. To make a fair evaluation, both protocols use the same join refresh periods (the interval between the production of consecutive $join$ messages by a receiver), as well as the same period for $tree$ message production and timeout values (used to “timeout” the table entries and eventually destroy them). Additionally, two sets of experiments were made. In the first one, routes are asymmetric as in all previous experiments. In this situation, REUNITE may produce temporary loops for certain scenarios and create a huge number of messages. In such scenarios HBH largely outperforms REUNITE because the average processing overhead of REUNITE is increased. Even though the actual Internet scenario is of asymmetric routes, we forced a second type of scenario, where all unicast routes are symmetric, to measure the overhead of the algorithm of HBH over REUNITE.

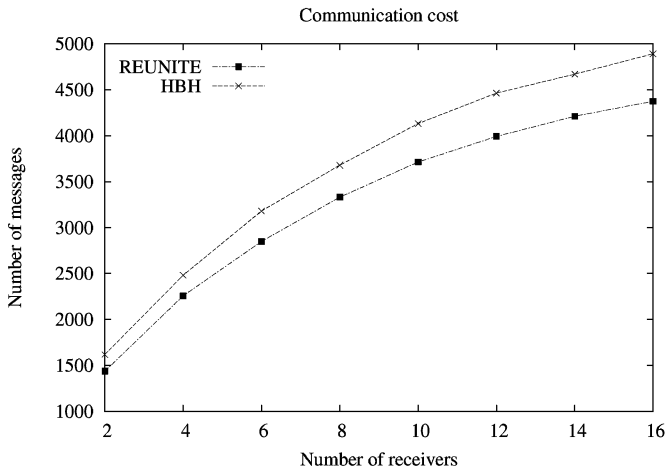
These experiments were conducted with the ISP topology. Fig. 8(a) shows the results obtained with asymmetric routes. They confirm our statements on the problems REUNITE may incur. As we did not add any loop avoidance mechanism to REUNITE, the number of useless messages produced during the temporary loop is a function of the link bandwidth and propagation delay. This is because a $tree$ message received by a router is immediately forwarded in our implementation, and, if two routers form a “$tree$ loop,” the number of messages produced is only limited by the physical path connecting them. For small groups, REUNITE performs the same as HBH, because the possibility of loop creation is low. The same holds for large groups, because almost every router in the network is part of the tree.

Fig. 8(b) shows the results obtained with symmetric routes. In this case, HBH does show a message processing cost larger than REUNITE. This is explained by the more complex algorithm of HBH. To guarantee the construction of a SPT, the *first join* message issued by a receiver always reaches the source. Furthermore, in some cases HBH will continue to produce $fusion$ messages as a way to avoid useless data-packet duplication, for example, router $H_6$ in Fig. 5. Fig. 8(b) shows that the control overhead of HBH compared to REUNITE is about 11%.

We can conclude that HBH is a promising approach. The HBH algorithm outperformed REUNITE by 5% in tree cost and 14% in delay paying 11% in control overhead, considering the unfavorable scenario of symmetric routes.

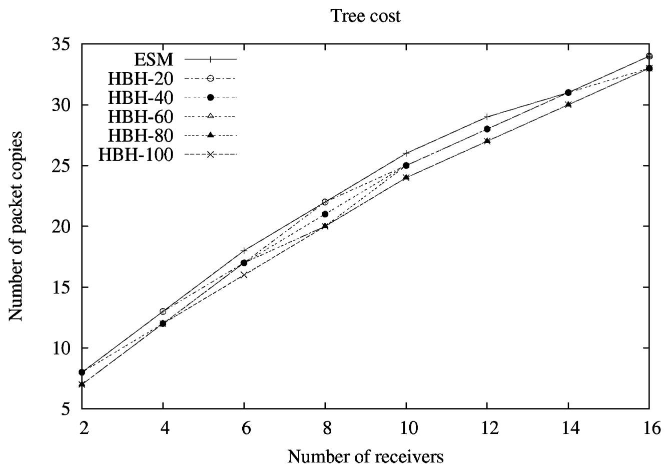


(a)

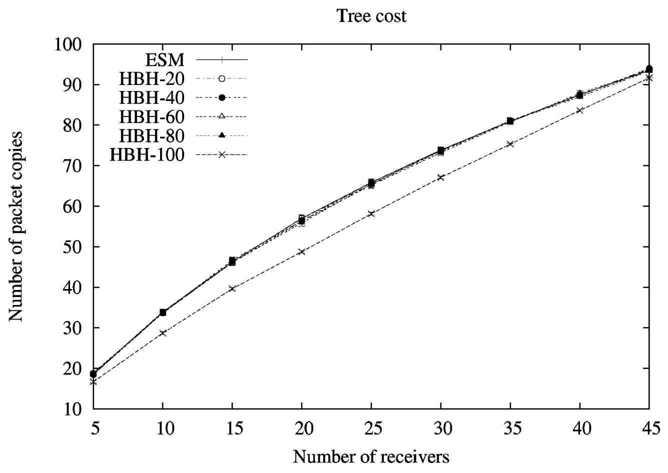


(b)

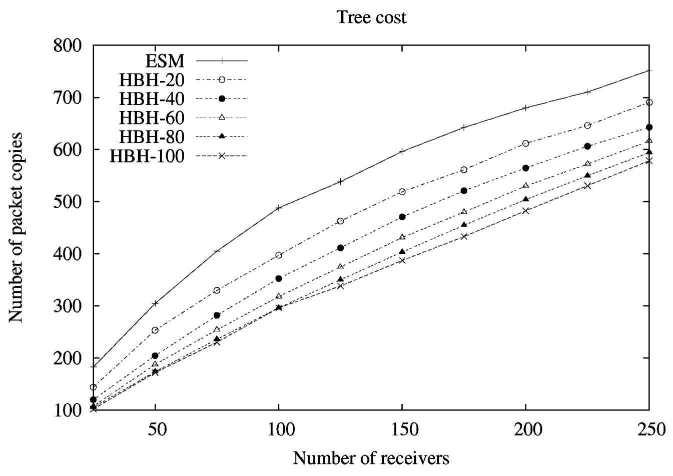
Fig. 8. Average number of control messages—18-node ISP topology. (a) Asymmetric routes. (b) Symmetric routes.



(a)



(b)



(c)

Fig. 9. Tree cost for different HBH deployment levels. (a) 18-node ISP topology. (b) 50-node random topology. (c) 500-node Internet-like topology.

### *E. Incremental Deployment Analysis*

As a consequence of the difficulties involved in deploying a router-based multicast service in the Internet, different approaches were proposed to implement the multicast service at the application layer [24], [25]. Implementing the multicast service at the application layer has the advantage of not needing to modify router implementations. On the other hand, the points where packets are replicated are limited to the stations, normally located at the boundaries of the network. Therefore, application-layer multicast (ALM) is a contender to router-based multicast, as implemented by IP Multicast routing protocols, or alternative routing protocols such as EXPRESS, REUNITE, or HBH.

A differential characteristic of HBH is its ability to be incrementally deployed. Routers have to be modified to implement HBH, but not all routers in the network have to implement HBH. The protocol can be partially deployed because of the recursive unicast technique. In this section, we compare HBH with ALM, with different degrees of HBH deployment.

Different ALM systems exist. They have in common the fact that replication is done at the stations (end-points of the network). Therefore, we decided to compare HBH to what we called end system multicast (ESM), not to any specific ALM system. Hence, ESM is not a specific application-layer protocol, but a protocol which simulates a generic application-layer protocol. The characteristic we analyze to compare HBH with ESM is the cost of the multicast tree, which is the same metric used in the previous analysis. To be fair, ESM is a

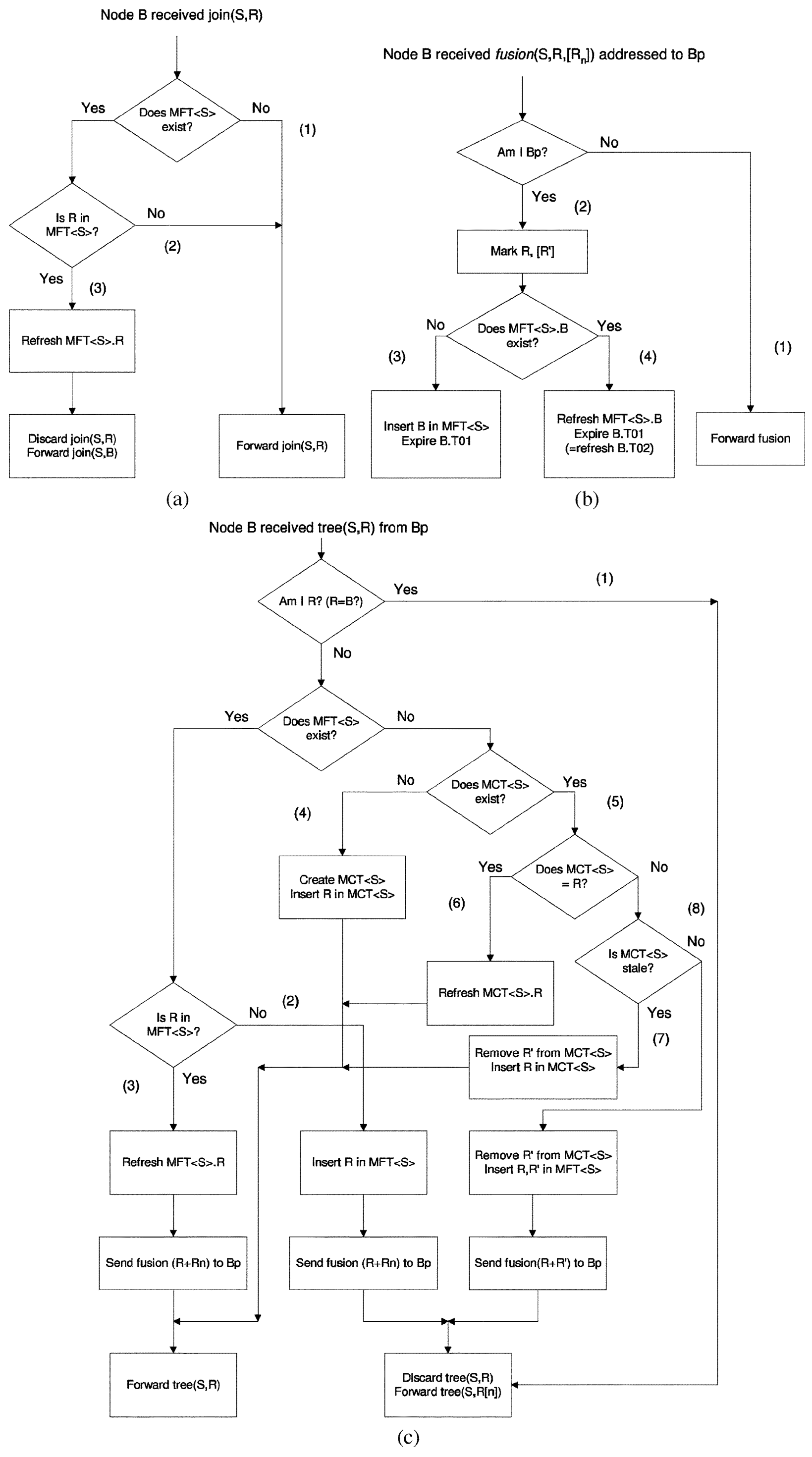


Fig. 10. Message processing in HBH. (a) $Join$ message. (b) $Fusion$ message. (c) $Tree$ message.

protocol which simply builds SPTs, with the restriction that the only possible branching nodes of the multicast distribution tree are the end systems. Routers do not participate in the multicast distribution. This is the main characteristic of the different application-layer protocols, therefore the one we chose to model by implementing ESM.

The simulation scenario uses the same topologies of the previous analysis. Nevertheless, all links are symmetric in this scenario because we aim to isolate the effect of different degrees of deployment over the cost of HBH trees. We varied the percentage of HBH routers in the network from 20% up to 100% of the total number of routers. We vary the group size and measure the cost of the distribution tree, as the total number of packets sent.

Fig. 9(a)–(c) show the results. In the figures, HBH-$n$ means that $n\%$ of the routers in the network implement HBH, whereas the other $(100 - n)\%$ of the routers are unicast-only. HBH routers were randomly placed in the network. In all figures, we can see that the advantage of HBH over ESM increases with the percentage of HBH routers deployed. Fig. 9(a) shows the results for the smallest topology, with 18 nodes. With symmetric links, HBH is 7% better than ESM when all routers are HBH-enabled, and with only 20% of HBH routers, HBH already performs 3% better than ESM. For the 50-node random topology, on the other hand, HBH was 10% better than ESM with 100% of HBH-enabled routers, but with other percentages of deployment, HBH performed almost the same as ESM [Fig. 9(b)]. The most promising results were obtained with the 500-node topology, generated from a real Internet map (see Section IV-A). With only 20% of HBH routers deployed in the network, HBH performed 14% better than ESM. With 40% of HBH-enabled routers, the gain is as large as 30%, and on the other hand, HBH is 35% better than ESM when all routers implement HBH. Therefore, it is not needed to implement HBH in all routers, and, in close to 100% of HBH routers in the network, the relative gains are small.

## V. Conclusion

In this paper, we analyzed HBH, a multicast routing protocol that implements multicast distribution through recursive unicast trees. HBH allows the incremental deployment of the multicast service because unicast routers inside the network are transparently supported. The main goals of HBH are: to support unicast clouds, allowing incremental deployment; to have a stable tree structure, by minimizing the impact of receiver departures; and to construct low-cost trees.

The tree management algorithm of HBH uses three control messages to construct an SPT. $Join$ messages are periodically sent to the source by the receivers. The source periodically produces $tree$ messages that are multicast to the receivers. As the $tree$ messages travels in the tree, the intermediate nodes may generate $fusion$ messages that are responsible of refining the tree structure.

HBH is able to construct SPTs even if unicast routing is asymmetric. HBH also provides better network utilization because it constructs recursive unicast trees minimizing packet duplication. This is a great advantage when the network bandwidth is scarce. Moreover, HBH provides better delay performance than other routing protocols, favoring interactive applications.

The simulation results show that HBH is a promising approach. Its tree construction algorithm outperformed REUNITE in terms of the tree cost and the delay experienced by the receivers. The advantage of HBH grows with larger and more connected networks.

We also analyzed the incremental deployment of HBH. The performance of HBH improves with the number of HBH-enabled routers in the network, as should be expected. Nevertheless, the interesting results are that with only 20% of HBH routers in the network, HBH already presents significant gains. Additionally, with 60% of HBH routers, the performance of HBH is close to the one obtained with 100% of HBH routers. Therefore, there is no need to implement HBH at all routers in the network to take advantage of the protocol. Thus, a point that deserves investigation is the placement of HBH routers in the topology.

## Appendix
## Message Processing in HBH

Fig. 10 presents the processing rules of the three message types used by HBH to construct the distribution tree. Each receiver $r$ periodically sends a $join(S, r)$ in unicast to the source. The source periodically multicasts a $tree$ message for each $\langle S, G\rangle$ channel.

$Join$ **message** [Fig. 10(a)]—When router $B$ receives a $join(S, R)$, it should forward this message unchanged if $B$ has no MFT (1) or if $R$ is not in $B$'s MFT (2). Only if $B$'s MFT has a $R$ entry, does $B$ intercept the $join(S, R)$ and send a $join(S, B)$ afterwards. This means that $B$ is a branching node of the channel $\langle S, G\rangle$ (3).

$Tree$ **message** [Fig. 10(c)]—A $tree(S, R)$ message received by router $B$ is treated and forwarded in multicast. If $B$ is a branching node, it may receive a $tree$ message addressed to $B$. In this case, $B$ discards the message and sends a $tree$ message to each node present in its MFT (1). If $B$ is a branching node and $tree(S, R)$ is not addressed to $B$, there are two possibilities: $R$ is a new receiver (in which case $B$ inserts $R$ in its MFT and sends a $fusion$ message upstream) (2) or $R$ is present in the MFT of $B$—which means that $B$ does not receive $join(S, R)$ messages from $R$—therefore $B$ simply has to refresh the $R$ entry in its MFT and to send a $fusion$ message upstream (3). If $B$ is not a branching node, there are two possibilities: $B$ was not in $S$'s distribution tree in which case $B$ creates an MCT containing $R$ (4) or $B$ was already in the tree but was not a branching node (in which case $B$ has an $\mathrm{MCT}\langle S\rangle$) (5). If $R$ was already in $B$'s MCT, nothing has changed and $B$ simply refreshes its MCT (6). If $R$ is not present in the MCT, then maybe the MCT is stale in which case $R$ replaces the previous MCT entry, or the MCT is up to date which means that there is a second receiver that will receive data through $B$, so $B$ becomes a branching node. This implies the creation of an MFT, the destruction of the MCT, and a $fusion$ message to be sent upstream (8). The $fusion$ messages produced by $B$ contain all the nodes that $B$ has in its MFT—the nodes for which $B$ is branching node.

$Fusion$ **message** [Fig. 10(b)]—Suppose router $B$ receives a $fusion(S, R, \ldots R_n)$ from node $B_p$. If the message is not addressed to $B$, then $B$ simply forwards it upstream (1). If the message is addressed to $B$, then $B$ marks the receiver entries in its MFT that are listed in the $fusion$ message (2). A marked entry in the MFT is used to $tree$ message forwarding but not for data distribution. $B_p$ is added to $B$'s MFT if it was not previously present. In addition, $B_p$'s $t1$ timer is expired—$B_p$ becomes stale (3). Consequently, $B_p$ will be used for data forwarding, but not for $tree$ message forwarding. If $B_p$ was already present in $B$'s MFT, then $B_p$'s $t2$ timer is refreshed (it avoids the destruction of the $B_p$ entry), but its $t1$ timer is kept expired (4).

If afterwards $B_p$ (the node that produced the $fusion$ for $R, \ldots R_n$) receives the $join$ messages produced by any of $R, \ldots R_n$, it intercepts them and send a $join(S, B_p)$ upstream. In this case the $R, \ldots R_n$ entries in $B$'s MFT will eventually timeout and be destroyed, while the $B_p$ entry in $B$'s MFT is refreshed by the $join(S, B_p)$. If $B_p$ does not receive $join$ messages from $R, \ldots R_n$, the emission of $fusion$ messages continues since there is another node upper in the tree that receives the $join(S, R, \dots R_n)$ and periodically produce the $tree(S, R, \dots R_n)$ messages to these receivers. Nevertheless, this node will not forward data to these receivers, but to $B_p$ instead since the receivers' entries are marked. The node $B_p$ is responsible for data duplication. In this way, HBH solves the second asymmetry problem presented in Section II-C.

## References

[1] S. Deering, "Host Extensions for IP Multicasting," RFC 1112, Aug. 1989.

[2] T. Bates, R. Chandra, D. Katz, and Y. Rekhter, "Multiprotocol Extensions for BGP-4," RFC 2283, Feb. 1998.

[3] D. Meyer and B. Fenner, "Multicast Source Discovery Protocol (MSDP)," RFC 3618, Oct. 2003.

[4] C. Diot, B. N. Levine, B. Liles, H. Kassem, and D. Balensiefen, "Deployment issues for the IP multicast service and architecture," *IEEE Network*, vol. 14, no. 1, pp. 78–88, Jan. 2000.

[5] H. W. Holbrook and D. R. Cheriton, "IP multicast channels: EXPRESS support for large-scale single-source applications," in *ACM SIGCOMM*, Sep. 1999, pp. 65–78.

[6] S. Bhattacharyya, C. Diot, L. Giuliano, R. Rockell, J. Meylor, D. Meyer, G. Shepherd, and B. Haberman, "An Overview of Source-Specific Multicast (SSM)," RFC 3569, July 2003.

[7] B. Cain, S. Deering, I. Kouvelas, B. Fenner, and A. Thyagarajan, "Internet Group Management Protocol, Version 3," RFC 3376, Oct. 2002.

[8] B. Fenner, M. Handley, H. Holbrook, and I. Kouvelas, "Protocol Independent Multicast—Sparse Mode (PIM-SM): Protocol Specification (Revised)," Work in progress, <draft-ietf-pim-sm-v2-new-10.txt>, Jul. 2004.

[9] R. Finlayson, "The UDP Multicast Tunneling Protocol," Work in progress, <draft-finlayson-umtp-09.txt>, Nov. 2003.

[10] P. Liefooghe and M. Goossens, "An architecture for seamless access to multicast content," in *Proc. IEEE Conf. Local Computer Networks*, Nov. 2000, pp. 488–494.

[11] D. Thaler, M. Talwar, A. Aggarwal, L. Vicisano, and D. Ooms, "IPv4 Automatic Multicast Without Explicit Tunnels (AMT)," Work in progress, <draft-ietf-mboned-auto-multicast-02.txt>, Feb. 2004.

[12] L. H. M. K. Costa, S. Fdida, and O. C. M. B. Duarte, "Hop by hop multicast routing protocol," in *Proc. ACM SIGCOMM*, Aug. 2001, pp. 249–259.

[13] I. Stoica, T. S. E. Ng, and H. Zhang, "REUNITE: A recursive unicast approach to multicast," in *Proc. IEEE INFOCOM*, Mar. 2000, vol. 3, pp. 1644–1653.

[14] S. Deering, D. L. Estrin, D. Farinacci, V. Jacobson, C.-G. Liu, and L. Mei, "The PIM architecture for wide-area multicast routing," *IEEE/ACM Trans. Netw.*, vol. 4, no. 2, pp. 153–162, Apr. 1996.

[15] C. Diot, W. Dabbous, and J. Crowcroft, "Multipoint communication: A survey of protocols, functions and mechanisms," *IEEE J. Sel. Areas Commun.*, vol. 15, no. 3, pp. 277–290, Apr. 1997.

[16] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, and L. Wei, "Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification," RFC 2362, Jun. 1998.

[17] D. Waitzman, C. Partridge, and S. Deering, "Distance Vector Multicast Routing Protocol," RFC 1075, Nov. 1988.

[18] R. C. Chalmers and K. C. Almeroth, "On the topology of multicast trees," *IEEE/ACM Trans. Netw.*, vol. 11, no. 1, pp. 153–165, Feb. 2003.

[19] V. Paxson, "End-to-end routing behavior in the Internet," *IEEE/ACM Trans. Netw.*, vol. 5, no. 5, pp. 601–615, Oct. 1997.

[20] J. Moy, "Multicast Extensions to OSPF," RFC 1584, Mar. 1994.

[21] K. Fall and K. Varadhan, "The ns Manual," UC Berkeley, LBL, USC/ISI, and Xerox PARC, Dec. 2003 [Online]. Available: http://www.isi.edu/nsnam/ns/ns-documentation.html

[22] G. Apostolopoulos, R. Guerin, S. Kamat, and S. K. Tripathi, "Quality of service based routing: a performance perspective," in *Proc. ACM SIGCOMM*, Sep. 1998, pp. 17–28.

[23] D. Magoni and J.-J. Pansiot, "Internet topology modeler based on map sampling," in *Proc. IEEE Int. Symp. Comput. Commun.*, Jul. 2002, pp. 1021–1027.

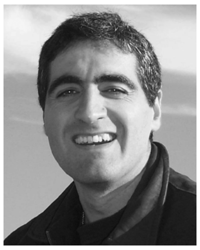
[24] A. El-Sayed, V. Roca, and L. Mathy, "A survey of proposals for an alternative group communication service," *IEEE Network*, vol. 17, no. 1, pp. 46–51, Jan. 2003.

[25] P. Radoslavov, C. Papadopoulos, R. Govindan, and D. Estrin, "A comparison of application-level and router-assisted hierarchical schemes for reliable multicast," *IEEE/ACM Trans. Netw.*, vol. 12, no. 3, pp. 469–482, Jun. 2004.

**Luís Henrique M. K. Costa** (M'99) received the Electronic Engineer degree and the M.Sc. degree in electrical engineering from the Federal University of Rio de Janeiro (UFRJ), Rio de Janeiro, Brazil, in 1997 and 1998, respectively, and the D.Sc. degree from the University Pierre et Marie Curie (Paris 6), Paris, France, in 2001.

In 2002, he was a Post-Doctoral Researcher with Laboratoire d'Informatique de Paris 6, Paris. Then, he was awarded a research grant from CAPES (an agency of the Brazilian Ministry of Education) and joined COPPE/UFRJ. Since August 2004, he has been an Associate Professor with UFRJ. His major research interests are in the area of routing, especially on wireless networks, group communication, quality of service, multicast, and large-scale routing.

Dr. Costa has been a member of the Association for Computing Machinery since 2001.



**Serge Fdida** (M'88–SM'98) received the Doctorat de 3eme Cycle and the Habilitation a Diriger des Recherches from the University Pierre et Marie Curie, Paris, France, in 1984 and 1989, respectively.

He has been a Professor with the University Pierre et Marie Curie since 1991. From 1989 to 1995, he was a Full Professor with the University Rene Descartes, Paris, France. His research interests are in the area of high-speed networking, multicast communication, resource management, and performance analysis. He heads the Network and Performance Group of the Laboratoire d'Informatique de Paris 6, Paris. He is also involved in two IFIP working groups on networking. He is also the Co-Director of EURONETLAB, which is a joint laboratory established in 2001 between the University Paris 6, CNRS, THALES, and 6WIND, developing research and development work on QoS routers, radio routers, and security.

Dr. Fdida is a member of the Association for Computing Machinery.

**Otto Carlos M. B. Duarte** received the Electronic Engineer degree and the M.Sc. degree in electrical engineering from the Federal University of Rio de Janeiro (UFRJ), Rio de Janeiro, Brazil, in 1976 and 1981, respectively, and the Dr. Ing. degree from ENST/Paris, Paris, France, in 1985.

Since 1978, he has been a Professor with UFRJ. From January 1992 to June 1993, he was with MASI Laboratory, University Paris 6, Paris. In 1995, he spent three months with the International Computer Science Institute (ICSI), University of California, Berkeley. In 1999 and 2001, he was an Invited Professor with the University Paris 6. He heads the computer network group (Grupo de Teleinformática e Automação-GTA), UFRJ. His major research interests are in multicast, QoS garantees, security, and mobile communications.