

Mobile Hop-By-Hop Multicast Routing

Rolland Vida^{a,b} Luís Henrique M. K. Costa^{a,c} Serge Fdida^a

^aLIP6 – Université Pierre et Marie Curie
8, rue du Capitaine Scott - 75015 - Paris - France

^bTMIT – Budapest University of Technology and Economics
Magyar Tudósok krt. 2 - H-1117 - Budapest - Hungary

^cGTA/COPPE/EE – Universidade Federal do Rio de Janeiro
P.O. Box 68504 - 21945-970 - Rio de Janeiro - RJ - Brazil¹

Abstract

This work focuses on mobility management in a multicast environment. Traditional solutions, based either on bi-directional tunneling or on remote subscription, have different drawbacks. We propose a different approach, which is based on a recursive unicast addressing scheme. The M-HBH protocol (Mobile Hop-By-Hop Multicast Routing) reduces triangular routing, avoids encapsulation and tunneling, supports asymmetric routes, and limits the frequency of tree reconstruction, for both source and receiver mobility. We evaluate the performance of the M-HBH protocol by comparing it with traditional solutions using mathematical models and simulations on a realistic Internet-like topology.

Key words: user mobility, group communication, multicast routing

1 Introduction

Recent technological advances resulted in the emergence of a variety of *mobile* devices, such as laptops, PDAs, and cellular phones. As a consequence, handling user mobility became a requirement for applications and services. On the other hand, an increasing number of applications, such as videoconferencing, Internet TV, and distributed games, need a simple and efficient multicast service. Providing such a service to mobile hosts is an open research issue.

Email addresses: Rolland.Vida@lip6.fr, Luis.Costa@lip6.fr, Serge.Fdida@lip6.fr.

¹ CAPES/Brazil and RNRT @IRS++/France.

Currently, there are two possible approaches to multicast user mobility. The first one is bi-directional tunneling, and consists on the creation of a tunnel between the home network (HN) and the foreign network (FN) of the mobile node (MN), similarly to unicast connectivity through Mobile IP [1]. The second one is remote subscription, and is based on the tree reconfiguration taking into account the new MN location. Hybrid solutions may switch from one technique to the other depending on specific criteria.

Supporting the multicast service in a mobile environment is difficult, since in traditional models the group is considered as a *global entity*, identified by its multicast address. Thus, the mobility of an *individual member*, source or receiver, is hard to handle. In this paper we propose a different approach, built on the HBH protocol [2]. In HBH nodes are considered as individual entities, not as group members; data is sent to the unicast addresses of consecutive branching nodes, instead of a multicast address. Therefore, mobility handling can represent a natural fit for this multicast distribution scheme.

The Mobile Hop By Hop multicast routing protocol (M-HBH) that we present handles source and receiver mobility, eliminates data tunneling, and reduces triangular routing. An M-HBH source that moves between networks can continue to send data without encapsulation, tunneling, or tree reconstruction. M-HBH receivers have no need to decapsulate tunneled packets or re-join the tree from their current location. Additionally, M-HBH inherits HBH properties. The multicast service can be progressively deployed, thanks to the recursive unicast technique. Additionally, HBH constructs direct shortest path trees (SPT) (even if unicast routes are asymmetric) as opposed to most routing protocols. We compare the performance of M-HBH with traditional solutions, using both mathematical models and simulations on a realistic Internet-like topology.

This paper is organized as follows. Section 2 presents previous work on multicast mobility handling. Section 3 presents the M-HBH protocol and shows how it handles source and receiver mobility. The mathematical evaluation of M-HBH is presented in Section 4, while simulation results are shown in Section 5. Finally, Section 7 concludes the paper.

2 Related Work

2.1 Columbia Mobile IP

The first solutions designed to extend multicast protocols to mobile hosts were based on Columbia's Mobile IP [3]. In this proposal, designed for unicast,

mobile nodes move between logical or geographical network segments, called cells. Cells contain special nodes, called Mobile Support Stations (MSS), that deliver packets to mobile nodes. Each mobile node has a permanent IP address, called its home address, regardless of its current point of attachment to the network. The MSSs in the cells of the same network exchange reachability information concerning mobile nodes. When the mobile node visits a foreign network, it obtains a *transient nonce address* from the mobile subnet address space of this network. The mobile node informs its home MSS about this address. From then on, the home MSS tunnels packets toward the foreign network. MSSs on the foreign network exchange local information between them; thus, they are able to deliver the packets to the mobile node.

Acharya et al. [4] extend the Columbia Mobile IP model to handle multicast communications. They introduce a multicast tunnel, that links all MSSs of the same network. When a mobile node joins a multicast group, incoming packets are transmitted through the multicast tunnel to all the MSSs of the network. Therefore, the mobile node receives these packets independently of the cell it currently resides in. When the mobile node moves to a foreign network, it informs its home MSS about its newly acquired transient address. From then on, when a multicast packet is received on the home network, for a group the mobile node is interested in, the home MSS tunnels it through a unicast connection to the foreign network. There, the packet is tunneled to all the MSSs.

2.2 IETF Mobile IP

The IETF Mobile IP proposal [1], shortly called Mobile IP, presents several differences in handling host mobility. Its design was focused on unicast communications; nevertheless, it has specific mechanisms to handle multicast distribution as well.

A mobile node (MN) is always identified by a long-term IP address, called home address (HoA), regardless of its current point of attachment to the Internet. When away from its home network, the MN is also associated to a transient IP address, called care-of address (CoA), that provides information about its current location. The care-of address can either be determined from a foreign agent's (FA) advertisement message (foreign agent care-of address), or obtained through an external address allocation mechanism, such as DHCP [5] (co-located care-of address). The node must register its current CoA with a home agent (HA), a router of its home network. The HA intercepts and tunnels to the CoA all data addressed to the MN. The tunnel endpoint can be either a FA or the mobile node itself, depending on the type of care-of address used.

Mobile IPv6 [6] introduces new mechanisms. For instance, it is no longer needed to deploy special routers as foreign agents; a mobile node acquires its care-of address through *stateless* [7] or *stateful* (e.g., DHCPv6 [8]) address auto-configuration mechanisms. Mobile IPv6 also integrates Route Optimization [6], a feature that was optional in Mobile IPv4. When a mobile node's HA intercepts a datagram sent by a correspondent node (CN) to the MN's home network, it assumes that the CN is not aware of the MN's current location. The home agent can then send a Binding Update (BU) message to the CN, informing it about the mobile node's current mobility binding. Any CN may maintain a binding cache to optimize its own communication with mobile nodes. If a CN has a binding cache entry for a specific MN, it can send its datagrams directly to the care-of address, bypassing the mobile node's HA.

Mobile IP introduces two basic mechanisms to handle multicast communications: *bi-directional tunneling* (BT) and *remote subscription* (RS). In the former, the HA forward multicast packets to the MN through a unicast Mobile IP tunnel. This approach has the advantage that the multicast delivery tree is not updated every time the MN moves. On the other hand, the data delivery path is not optimal because of triangular routing, i.e., packets have to pass through the HA, instead of being directly sent to the MN.

Using remote subscription, the MN re-joins the group each time it enters a foreign network. The main advantage of RS is that data are delivered on the shortest path. On the other hand, the multicast delivery tree must be frequently updated. This generates signaling overhead and might result in unnecessary bandwidth consumption. Indeed, a mobile node that asks its foreign agent to join the multicast tree on its behalf pulls the multicast traffic toward the visited network. If the MN travels at high speed, it leaves that network rapidly. Nevertheless, multicast traffic keeps flowing, until soft-state multicast group management information expires.

2.3 *Alternative proposals*

Both bi-directional tunneling and remote subscription have drawbacks. Thus, different hybrid solutions were proposed; they switch from one technique to the other depending on specific criteria. In the following, we present some of these approaches.

2.3.1 *Mobile Multicast (MoM)*

Standard bi-directional tunneling generates the so-called "tunnel convergence" problem. When several MNs from different home networks move in the same foreign network, all their respective HAs tunnel the same data in that network.

This leads to unnecessary bandwidth consumption and duplicate delivery of the multicast packets. To solve this problem, Harrison et al. propose the Mobile Multicast Protocol (MoM) [9]. A designated multicast service provider (DMSP) is elected for each network, among the HAs of the MNs that are currently visiting that foreign network. Only the DMSP tunnels multicast datagrams to the foreign network; thus, it provides at-most-once delivery, as desired.

2.3.2 Range-Based Mobile Multicast (RBMoM)

A drawback of MoM is that the tunnel established between the DMSP and the foreign network may be long. To alleviate this problem, the Range-Based Mobile Multicast (RBMoM) protocol [10] introduces multicast home agents (MHA) with a limited service range. This solution is a trade-off between the shortest delivery path and the frequency of the multicast tree reconfiguration.

Each MN has a MHA that tunnels multicast datagrams to the foreign network the MN currently resides in. Nevertheless, unlike in standard Mobile IP or MoM, the MHA serves only MNs that are roaming around the foreign networks which are in its service range. If the MN leaves this service range, MHA hand-off occurs, and a new MHA is assigned to the MN. Thus, the size of the tunnel is limited. Nevertheless, the tunnel convergence problem exists. To solve it, the RBMoM and MoM protocols can be combined.

2.3.3 Mobility Management in Hierarchical Multicast Architectures

Another solution that combines the advantages of bi-directional tunneling and remote subscription is presented in [11]. The authors propose a hierarchical multicast architecture based on multicast agents. A multicast agent provides the multicast service to mobile nodes in multiple (foreign) networks; it joins multicast groups on behalf of MNs in its service area. As opposed to service ranges in RBMoM, service areas of different multicast agents are disjoint. Thus, tunnel convergence is avoided. Several other proposals [12,13,14] also limit the frequency and the degree of multicast tree reconfiguration through different hierarchical schemes.

3 The Mobile Hop By Hop Multicast Routing Protocol

Most of the protocols presented in the previous section address only the mobility of multicast receivers. Nevertheless, multicast source mobility is also a non-trivial task to deal with, especially when multicast distribution is done

along source-specific trees. In the following, we present the Mobile Hop-By-Hop Multicast Routing protocol (M-HBH) that considers both source and receiver mobility; it offers transparent mobility handling, reduces routing triangles, eliminates encapsulation and tunneling. In addition, due to its recursive unicast addressing scheme, M-HBH supports incremental deployment over networks with unicast-only clouds. M-HBH extends the HBH protocol [2] to handle mobile multicast nodes. In order to understand its operation, we briefly describe first the original HBH proposal.

3.1 The HBH Protocol

One important problem of the current multicast service is the difficulty of incremental deployment. To address this issue, several techniques were proposed, such as tunneling through unicast-only clouds, or application layer multicast. Nevertheless, they present some drawbacks, like processing overhead or increased bandwidth consumption. Recent protocols such as REUNITE [15] and HBH [2] use a different approach, based on a recursive unicast addressing scheme. In HBH, packets have unicast destination addresses, instead of class-D addresses. Multicast routing information is divided into two tables: a Multicast Control Table (MCT) and a Multicast Forwarding Table (MFT). Non-branching routers keep group information only in their MCT, while branching nodes use the entries from their MFT to recursively create packet copies, and distribute them in order to reach all group members.

To construct the HBH tree, three control messages are used: *join*, *tree*, and *fusion*. Intermediate HBH routers always forward the first *join* message of a new receiver up to the source. Upon reception of a new *join*, the source adds the receiver to its MFT, and sends a *tree* message downstream. As intermediate routers forward the *tree* message toward the receiver, they add its address to their MCT. From now on, they act as non-branching nodes of the tree. *Tree* messages are also sent to the other nodes registered in the source's MFT. If a router receives *tree* messages for two different receivers, it sends a *fusion* upstream, announcing itself as a better branching node for these receivers. Upon reception of the *fusion* message, the source updates its MFT.

HBH is a soft-state protocol. Receivers periodically send *join* messages that update the MFTs of branching nodes. As mentioned before, the first *join* message of a new receiver always reaches the source. Following messages are intercepted by branching nodes, that regularly issue their own *join* message. The *tree* messages, sent by the source in answer to the periodic *joins*, refresh the MCTs of non-branching nodes. HBH sends all control and data packets with unicast destination addresses. To send a data packet to the group, the source creates a copy of it for each entry present in its MFT. Upon reception of

the packet, an HBH branching node creates copies of it for each entry present in its own MFT. Non-branching HBH routers simply forward data packets toward their unicast destinations. Using this recursive unicast addressing scheme, data is finally delivered to all the receivers in the HBH tree.

An HBH branching node necessitates a larger "routing" table (MFT) than a traditional multicast router, as it has to keep the unicast addresses of all its downstream branching nodes, instead of the single multicast address of the group. Nevertheless, HBH, as well as REUNITE, scales better than traditional source-tree based routing protocols, because, by using the recursive unicast approach, non-branching nodes do not store entries in the forwarding table. Moreover, node degree distribution in real multicast trees over the Internet follows a power law [28]; there are very few branching nodes with high degree. More details on HBH tree construction and data delivery can be found in [2].

3.2 Handling Multicast Source Mobility with M-HBH

Most studies on multicast mobility only address mobile receivers. Nevertheless, source mobility is critical for multicast routing protocols that construct source-specific delivery trees (e.g., DVMRP [16], PIM-SSM [17]). Handling the mobile root of a source-specific multicast tree is more complex than the case of a shared tree (e.g., PIM-SM [18]). In the latter, a mobile source can send data without interruption, while moving; data reaches the Rendezvous Point (RP), and it is then forwarded on the shared tree. No special mobility mechanism is needed. On the other hand, source mobility is also more complicated than the case of receiver mobility. A receiver can simply leave the tree and re-join from its new location. For a mobile source, the entire tree is reconstructed.

Suppose a group of receivers that listen to a source, S , by joining the multicast channel $\langle S, G \rangle$. A source-specific multicast tree rooted at S is created. Only S is allowed to send data on this tree. The problem is how to deliver data when the mobile source leaves its home network. Different issues arise:

- the foreign network the source is visiting might not contain a multicast router; thus, a packet sent to a multicast channel is dropped;
- if the source uses its home address as the source address of its packets, ingress filtering routers [19] drop them because of security considerations;
- the first router that receives a packet might not be connected to the original tree; thus, it does not have state for the channel, and drops the packet;
- most routing protocols perform an RPF (Reverse Path Forwarding) check [20]; if the mobile source uses its home address to send data, packets are dropped.

The two classical approaches to multicast mobility, bi-directional tunneling (BT) and remote subscription (RS), have problems to handle SSM. First,

handling a mobile SSM source using remote subscription means that the distribution tree must be entirely reconstructed. On the other hand, BT introduces triangular routing. Moreover, packets have to be encapsulated and decapsulated at the two ends of the tunnel, increasing the processing time.

M-HBH extends the original HBH protocol to handle user mobility. First, we define the following two entities: the M-HBH control tree, formed by routers that exchange M-HBH control messages, and the M-HBH data tree, formed by routers that forward multicast data. For a multicast channel without mobile hosts the M-HBH control and data tree are identical; we simply call it M-HBH tree. M-HBH uses the same mechanisms as HBH to create this tree and deliver multicast data. Suppose an M-HBH tree rooted at S . The source S is in its home network and sends *tree* messages, as well as data packets, using its home address. The only difference to HBH is that M-HBH routers keep the CoA of the source in addition to its HoA in their MFT and MCT. At the beginning, while S is in its home network, the CoA and the HoA coincide.

Figure 1(a) presents the case of S leaving its home network. We consider that mobile nodes use Mobile IP to keep unicast connectivity; the M-HBH protocol offers an additional mechanism that handles multicast communications. When the mobile source S enters a new network, it obtains a CoA. In the following we use the notation S' to refer to this new location. Mobile IP is used to keep unicast connectivity, establishing a tunnel between the home agent of the source and the visited network. Receivers, as well as intermediate routers on the M-HBH (control) tree, keep sending their *join* messages to the source's HoA. The HA then tunnels these messages to S' . Meanwhile, the source tunnels its *tree* messages to the HA, which forwards them on the M-HBH (control) tree. Thus, the control tree remains unchanged while the source moves.

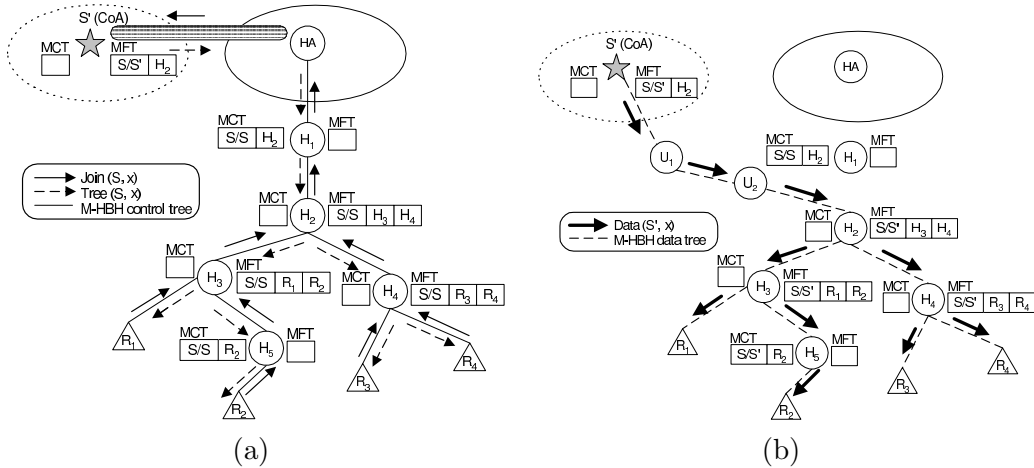


Figure 1. Handling source mobility with M-HBH.

Nevertheless, unlike in bi-directional tunneling, M-HBH data packets do not have to be encapsulated and tunneled. Once the source obtains its new ad-

dress S' in the foreign network, it immediately starts to send data directly, without tunneling. The source sends data to nodes whose unicast addresses are in its MFT. As these packets are sent in unicast, there is no need to have intermediate multicast routers with routing state for this channel. Data is delivered from S' to H_2 , and then forwarded on the rest of the tree (Figure 1(b)). The data delivery path is improved (compared to bi-directional tunneling) because triangular routing is reduced. On the other hand, the control tree is unchanged. Thus, extra signaling is avoided.

To pass through ingress filtering routers of the foreign network, the source should send data with the source address set to S' . Nevertheless, in order to comply with the “single-source” restriction of the SSM service, the home address S is needed. A possible solution is to use a mechanism similar to the Home Address destination option proposed in Mobile IPv6. By analyzing the headers, receivers notice that the packets were sent by the same, although mobile, source S . All M-HBH routers that receive such a packet update their MFT and MCT with the current CoA of the source. Receivers are also informed about the address change. Nevertheless, they do not redirect their *join* messages immediately, because of efficiency reasons that we explain later. The receivers keep sending *join* messages towards S (Figure 1(a)).

Note that the mobile source uses S' as source address only to send data. *Tree* messages continue to be sent using the HoA. They are tunneled to the HA; once decapsulated in the home network, they are forwarded towards the destination. Ingress filtering routers in the home network let the packets pass, as the source address belongs to its address space. By sending *tree* messages with the HoA, nodes that are on the control tree but not on the data tree are not aware of the source’s movements (e.g., router H_1 in Figure 1(c)). Nevertheless, the protocol operation is not modified. A timing diagram that describes the message exchange in case of source mobility is presented in Figure 2.

One may argue that the first branching node in M-HBH has similar functionality to the RP in PIM-SM [18]. Nevertheless, the RP is relatively static, and is independent of the source’s location. On the other hand, the position of the first branching node in M-HBH is related to the location of the source; passing through this node provides a path shorter than the path that crosses the RP. In M-HBH, if the group membership changes, the first branching node is updated accordingly, *if needed*. This is the reason why we considered the M-HBH control tree dynamic, whereas the RP location is much more static.

Join messages of new receivers must also be taken into account. The join process is not affected by the absence of the source from its home network. A new receiver sends the *join* message to the source’s home address, S , the only address it is aware of. Being the first *join* message, it reaches the home network of the source. Then, the message is tunneled by the HA to the source’s current

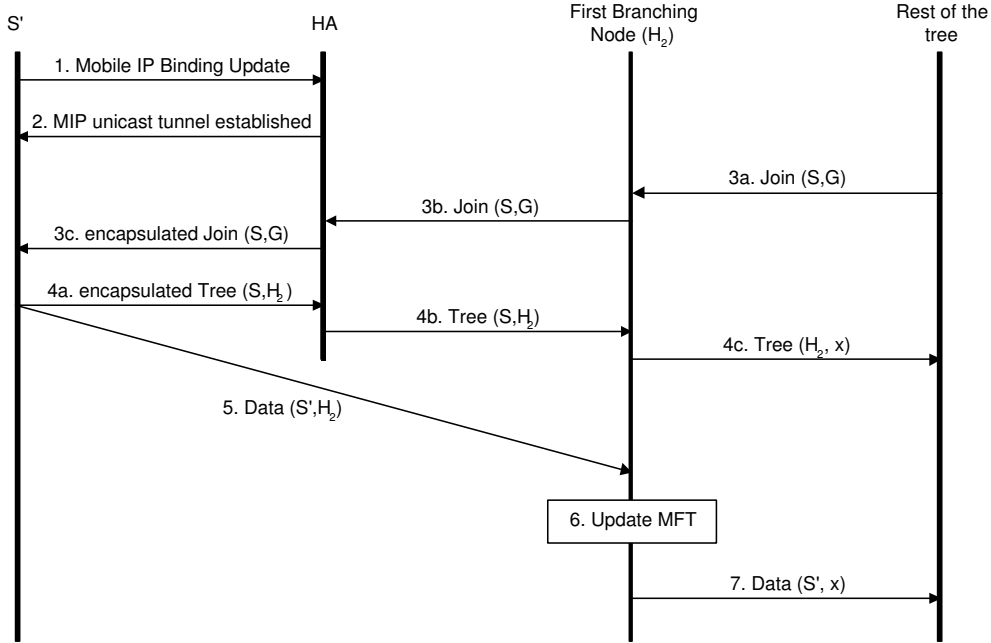


Figure 2. Protocol operation for source mobility.

location, S' . The source adds the receiver to its MFT. If subsequent *fusion* messages result in a change of the first branching node, the source updates its MFT accordingly. The resulting control tree is the same that would be obtained if S stayed in its home network. On the other hand, the data tree is updated, as the first branching node has changed.

Depending on the topology, it is possible that the shortest path from the source to the first branching node crosses the previous M-HBH data tree (Figure 3). When the source obtains its new address S' , a tunnel is established (Figure 3(a)). As *join* and *tree* messages are tunneled, the M-HBH control tree remains unchanged. On the other hand, the source S' sends data to the first branching node, H_2 , on the direct shortest path, which passes through H_3 (Figure 3(b)); H_2 then forwards data downstream. Thus, the same data packet crosses the link between nodes H_2 and H_3 twice, causing extra bandwidth consumption.

This problem can be solved through a reconfiguration of the M-HBH data and control trees: receivers start sending *join* messages to S' , routing table entries along the old control tree gradually expire, while a new control tree is constructed. Note that a source implosion problem can arise, if all receivers decide to redirect their *join* messages toward the new location of the source at the same time. To avoid this, M-HBH receivers have to implement a back-off timer to spread the *join* messages in time.

M-HBH reduces routing triangles, as it shortcuts the path between the source and the first branching node of the control tree, but it does not assure shortest path delivery. Thus, periodical tree reconfiguration can be considered. Nev-

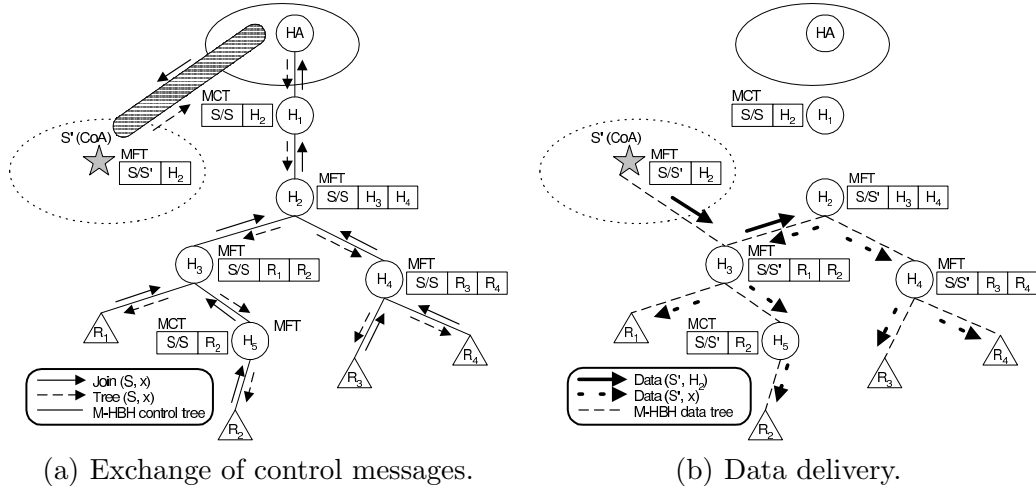


Figure 3. Data crosses the link between routers H_2 and H_3 twice.

ertheless, entirely reshaping a large control tree is costly and therefore the reconfiguration frequency must be limited. Different criteria can be used to trigger tree reshaping. Detecting double crossed links in the M-HBH data tree is a possible reason. The source can also initiate a reconfiguration if the length of the direct path to the first branching node is too large.

The "stability" of the source may be a criterion as well. When the source leaves its home network, receivers are aware of it as they receive packets sent from a new address S' . Nevertheless, they do not change immediately the destination address of their Join messages. Instead, they wait for the source to stabilize. A stability threshold can be fixed for data packets received from the same CoA of the source. When a host receives a packet from a new CoA, it initializes a stability counter. Each time a new packet arrives with the same source address, the counter is incremented. When it reaches the fixed threshold, the receiver assumes that the source is stable enough, and redirects its Join messages towards this CoA. If instead it receives a packet from a new address, it concludes that the source moved again, and reinitializes the stability counter.

M-HBH represents a trade-off between shortest path delivery and tree reconstruction. It continuously optimizes bi-directional tunneling, as it shortcuts routing triangles. On the other hand, it periodically operates like remote subscription, as it reconstructs a shortest path tree. At a given moment, whether the M-HBH data tree looks more like a BT tree, or a shortest path RS tree, depends largely on the underlying topology. On the other hand, the shape of the M-HBH data tree is also influenced by the tuning of the stability threshold. For small values, M-HBH operates more like remote subscription, with frequent tree reconfiguration. For larger values of the stability threshold, the reshaping frequency is reduced.

3.3 Handling Multicast Receiver Mobility with M-HBH

This section presents how M-HBH handles mobile receivers. There are two possible scenarios: receivers can either be themselves M-HBH enabled nodes, or they can use a group management protocol (IGMP [21] or MLD [22]) to connect to an M-HBH enabled node. In the following, we consider the first scenario. A detailed analysis of the other scenario can be found in [23].

An M-HBH receiver simply sends *join* messages toward the source; on the other hand, it receives *tree* messages from the source, or the last branching node on the control tree. The receiver does not behave as a router, it does not keep forwarding and control tables. Figure 4 presents an example. Suppose that receiver R_2 is visiting a foreign network where it acquired a CoA, R'_2 . R_2 uses Mobile IP to keep unicast connectivity (there is a tunnel between R'_2 and its HA). At the beginning, this tunnel is also used to maintain connectivity with the M-HBH control and data trees. *Join* messages sent by R'_2 pass through it; then, they are forwarded by the HA upstream on the control tree (Figure 4(a)). At the same time, the last branching router (H_1) sends data to the home address of R_2 ; the HA tunnels the data to R'_2 .

Nevertheless, R'_2 can announce its new location and update the data tree (Figure 4(b)). The source address of the *join* message must be set to R_2 ; otherwise, the packet, once decapsulated and forwarded by the HA, would be dropped by the home network's ingress filtering router. Therefore, to announce the new CoA of the receiver, a mechanism similar to the Home Address destination option must be used. We call it Care-of Address destination option (CAO).

Receiver R_2 places its current CoA in the Care-of Address destination option of the *join* message. When H_1 , the branching router responsible of R_2 , receives this message from the HA, it notices that R_2 , present in its MFT, has moved. In order to handle mobile receivers, the MFT has to be modified: for each node in the MFT the router stores its HoA as well as its current CoA. This modification is similar to the one introduced previously for mobile sources. If the receiver is in its home network, the current CoA and the HoA are identical.

Upon reception of the *join* message from R'_2 , H_1 updates its MFT. From now on, it forwards data to the mobile receiver's new address, R'_2 . Thus, the data tree is updated. The control tree remains unchanged; *tree* and *join* messages are still sent on the old path, through the HA and the bi-directional tunnel.

A problem can occur when the mobile receiver moves again, and obtains a new CoA, R''_2 . Until router H_1 receives the node's new address, data packets are sent to R'_2 , and consequently lost. This happens because the Mobile IP tunnel is established between R''_2 and the HA, not between R''_2 and R'_2 . Nevertheless, the mobile receiver remains connected to the M-HBH control tree, as its *join*

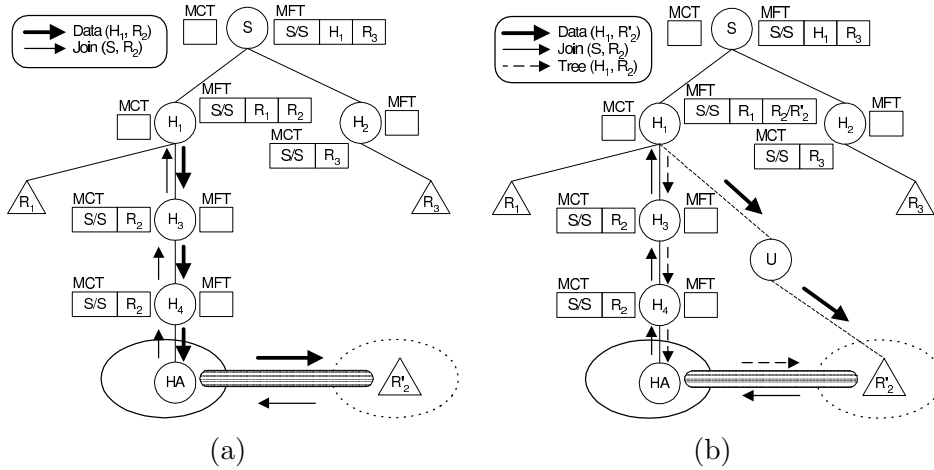


Figure 4. Handling a mobile M-HBH enabled receiver.

messages still reach the HA, and router H_1 . Thus, data delivery can resume. A timing diagram that describes the message exchange in case of receiver mobility is presented in Figure 5.

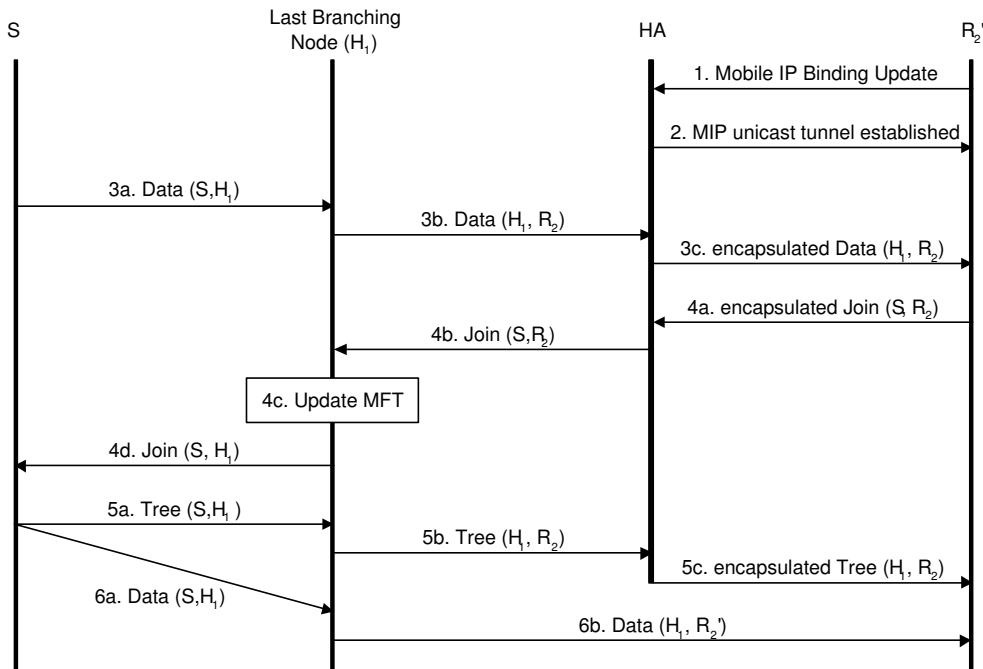


Figure 5. Protocol operation for receiver mobility.

The connectivity loss during the network changes is independent of the routing protocol. The receiver must obtain its new CoA and announce it to the HA, before being reachable again. For M-HBH, the announcement of the new CoA must reach the last branching node, instead of the HA. Thus, the disconnected period is larger than that of bi-directional tunneling. Nevertheless, depending on the network topology and the reliability requirements of the application, M-HBH's delay and bandwidth gains compensate this larger period.

Under certain conditions (e.g., a receiver joining or leaving the multicast group), a receiver’s last branching node may change. If the new branching node does not have up to date information about the current CoA of the mobile receiver, data starts flowing again through the tunnel. Nevertheless, as soon as the next *join* message of the MN reaches the branching node, data uses the shortest path. Thus, the protocol handles receiver mobility correctly. It regularly shortcuts the routing triangle between the MN and its last branching node, even if membership changes reshape the M-HBH control tree.

The MFT table of an M-HBH branching node has to be larger than that of a traditional HBH branching node, as the current CoA of the mobile nodes has to be kept as well. Nevertheless, this does not affect scalability, as there is at most one additional address to be kept for each entry. Thus, the table size can be at most the double of that of an HBH node. The mobility of the multicast nodes does not introduce additional scalability concerns.

4 Mathematical Analysis

This section presents two mathematical models to compare M-HBH with the multicast-related mechanisms of Mobile IP, i.e., bi-directional tunneling (BT) and remote subscription (RS). As Mobile IP, M-HBH does not eliminate tunnel convergence either; two different last branching nodes can send the same packet to their respective mobile receivers that visit the same foreign network. Nevertheless, hybrid mechanisms that address this issue, such as MoM [9] and RBMoM [10] can be applied to M-HBH. Therefore, the following analysis does not consider these proposals. The pure M-HBH and Mobile IP approaches are compared. We assume that group membership is stable and that unicast routes are symmetric. Thus, an HBH shortest path tree and a PIM-SSM reverse shortest path tree are identical. In the case of asymmetric routing, the performance of M-HBH is improved; direct shortest paths used in M-HBH are shorter than reverse shortest paths [2].

First, we define the evaluation metrics. Then, we compare the three approaches using k-ary and self-similar trees. Let $d(a, b)$ be the distance, in number of links, between nodes a and b . Let T_S denote the source-specific multicast tree rooted at S ; L_S is the tree size in number of links. Finally, let fbn denote the first branching node of an M-HBH control tree; and $lbn(r)$ denote the last branching node before receiver r .

4.1 Source Mobility

Suppose a source S and a group of m receivers $\{r_i \mid i = 1..m\}$. The following metrics are used to analyze the effect of source mobility on data delivery.

Total delivery cost

The total delivery cost $Cost(S, S')$ is the number of links traversed by data packets to reach all receivers, when the source moves from S to S' . Thus,

$$Cost_{MHBH}(S, S') = d(S', fbn) + L_S - d(S, fbn) = d(S', fbn) + L_{fbn}, \quad (1)$$

$$Cost_{BT}(S, S') = d(S', S) + L_S, \quad (2)$$

$$Cost_{RS} = L_S. \quad (3)$$

Therefore, the following inequality holds:

$$Cost_{RS}(S, S') \leq Cost_{MHBH}(S, S') \leq Cost_{BT}(S, S'), \quad \forall S, S'. \quad (4)$$

RS has the smallest total delivery cost, equal to the total number of links on a shortest path tree. On the other hand, M-HBH performs better than BT, as routing triangles are reduced.

Average delivery delay

We define the delivery delay as the time needed to deliver a packet from the source to a receiver. We assume that all links have the same propagation delay D_{Prop} ; we ignore processing and queuing delays, encapsulation/decapsulation time in BT, as well as copy-creation time in M-HBH. In these conditions, the average delivery delay is obtained as follows:

$$Del_{MHBH}(S, S') = D_{Prop} [d(S', fbn) + \sum_{i=1}^m d(fbn, r_i)/m] \quad (5)$$

$$Del_{BT}(S, S') = D_{Prop} [d(S', S) + \sum_{i=1}^m d(S, r_i)/m] \quad (6)$$

$$Del_{RS}(S, S') = D_{Prop} \sum_{i=1}^m d(S', r_i)/m \quad (7)$$

Signaling overhead

We define the signaling overhead $Sig(S, S')$ as the number of links that are traversed by routing messages. *Join* messages are periodically sent, therefore

the signaling overhead is computed over n time intervals. Additionally, the source moves c times during the evaluation period. As we assumed symmetric routes and stable group membership, M-HBH nodes do not send *fusion* messages (only *join* and *tree* messages).

We define the following average values:

$$\bar{d}(S, S') = \sum_{i=1}^c d(s, s_i)/c \quad \text{- average tunnel size} \quad (8)$$

$$\bar{L}_{S'} = \sum_{i=1}^c L_{S_i}/c \quad \text{- average size of the new tree} \quad (9)$$

In M-HBH, *join* and *tree* messages are tunneled; only data packets are sent directly to the first branching node. Thus, M-HBH's signaling overhead is:

$$Sig_{MHBH}(S, S') = 2n(\bar{d}(S, S') + L_S) \quad (10)$$

To evaluate BT's signaling overhead, we assume that a traditional multicast protocol (e.g., PIM-SM) constructs a source-specific tree rooted at S . PIM *join* messages are sent regularly toward the source; each link of the tree is traversed by a single *join* message. Thus, BT's signaling overhead is:

$$Sig_{BT}(S, S') = n(\bar{d}(S, S') + L_S) \quad (11)$$

In the case of RS, we suppose that a tree rooted at S is used to distribute binding updates (BU). Thus, receivers send *join* messages both on the BU tree and on the current tree rooted at S' . Moreover, each time the source moves, a BU is tunneled to S , and forwarded on the BU tree. The receivers then immediately send *join* messages toward S' , constructing a new tree.

As in the case of BT, we consider that during the construction (or the regular refreshment) of the source-specific tree, each link is traversed by a single *join* message. Thus, the number of control messages is equal to the size of the tree. For the sake of simplicity, we assume that the movements of the source do not affect the periodicity of the regular *join* messages sent by a receiver; a *join* message sent in response to a BU does not cancel a scheduled regular *join*.

In these conditions, the signaling overhead of RS is:

$$Sig_{RS}(S, S') = n(L_S + \bar{L}_{S'}) + c(\bar{d}(S, S') + L_S + \bar{L}_{S'}) \quad (12)$$

Note that, for the sake of simplicity, we considered only *join* messages in both BT and RS. Nevertheless, routing protocols may use other control messages.

PIM routers regularly send *hello* messages to discover neighbors and may send advertisements to choose RPs [18]. PIM-SSM reduces signaling, as it eliminates RPs. Nevertheless, periodical *hello* messages are still used. Therefore, the signaling overhead of BT and RS are underestimated. On the other hand, M-HBH only uses *join* and *tree* messages (*fusion* messages are not needed in a stable tree with symmetric unicast routing). Note also that, for both M-HBH and BT, the signaling overhead does not depend directly on c (apart the variations of the average tunnel size). This is not true for RS, as $Sig_{RS}(S, S')$ increases with c . Thus, RS is inappropriate for highly mobile sources.

4.2 Receiver Mobility

This section presents the metrics to evaluate the protocol efficiency in the case of mobile receivers. We suppose that mobile receivers implement M-HBH.

Delivery delay

Suppose that receiver r enters a foreign network and obtains a CoA r' . The delivery delay $Del(r, r')$ is the time needed to deliver a multicast packet from the source to the mobile receiver. As for source mobility, we assume that all links have the same propagation delay D_{Prop} ; we neglect processing and queuing delays, copy-creation time in M-HBH, and encapsulation and decapsulation times in BT. Under these conditions, the delivery delay is obtained as:

$$Del_{MHBH}(r, r') = D_{Prop}[d(S, lbn(r)) + D(lbn(r), r')] \quad (13)$$

$$Del_{BT}(r, r') = D_{Prop}[d(S, r) + d(r, r')] \quad (14)$$

$$Del_{RS}(r, r') = D_{Prop}d(S, r') \quad (15)$$

Service interruption

We define the service interruption $Int(r, r')$ as the time during which the receiver loses multicast connectivity, when it moves. We neglect agent discovery and tunnel setup time, and we suppose that the propagation delay D_{Prop} of a control message is the same on every link.

In M-HBH, the mobile nodes receives data directly from its last branching node. If it moves into a new network, it should inform this branching node about the address change. Thus, service interruption is obtained as:

$$Int_{MHBH}(r, r') = D_{Prop}[d(r, r') + d(r, lbn(r))] \quad (16)$$

In BT, the binding update must reach only the home agent, instead of the last branching node. Thus, the service interruption is obtained as:

$$Int_{BT}(r, r') = D_{Prop}d(r, r') \quad (17)$$

In RS, the service interruption is the time needed by the receiver to join the tree from its new location. Let $d(r, T_S)$ denote the *distance*² between receiver r and the tree T_S rooted at S . Then, the service interruption is obtained as:

$$Int_{RS}(r, r') = D_{Prop}d(r', T_S) \quad (18)$$

Other metrics can be defined. Receiver mobility generates lower signaling overhead for RS than source mobility because only one branch is reconstructed, not the entire tree. On the other hand, the total delivery cost is not influenced by receiver mobility. While the movement of the source affects the delivery cost for every receiver, the movement of one receiver only affects its own delivery cost, the total delivery cost is hardly modified, especially for large groups.

4.3 Theoretical Analysis

The above metrics show that the performance of M-HBH depends on the network topology, and, thus, on the shape of the multicast tree. Compared to BT, M-HBH reduces triangular routing. M-HBH shortcuts the path between the mobile source and the first branching node of the tree, or between a mobile receiver and its last branching node. Thus, the performance of M-HBH depends on the location of these specific branching nodes.

For a mobile source (Figure 6(a)), we define the *relative gains in total delivery cost of M-HBH over BT* as follows:

$$\Delta Cost(S, S') = \frac{Cost_{BT}(S, S') - Cost_{MHBH}(S, S')}{Cost_{BT}(S, S')} = \frac{x_S + y_S - z_S}{y_S + L_S} \quad (19)$$

where: x_S is the distance between the source S and the first branching node of the tree; y_S is the length of the BT tunnel between S and S' ; z_S is the length of the direct path between S' and the first branching node.

Note that x_S depends on the group size: the larger the group, the higher the probability of having an early branching in the tree, especially if receivers

² The number of hops between receiver r and node x , where x is the first node on the tree T_S in the shortest path from r to S .

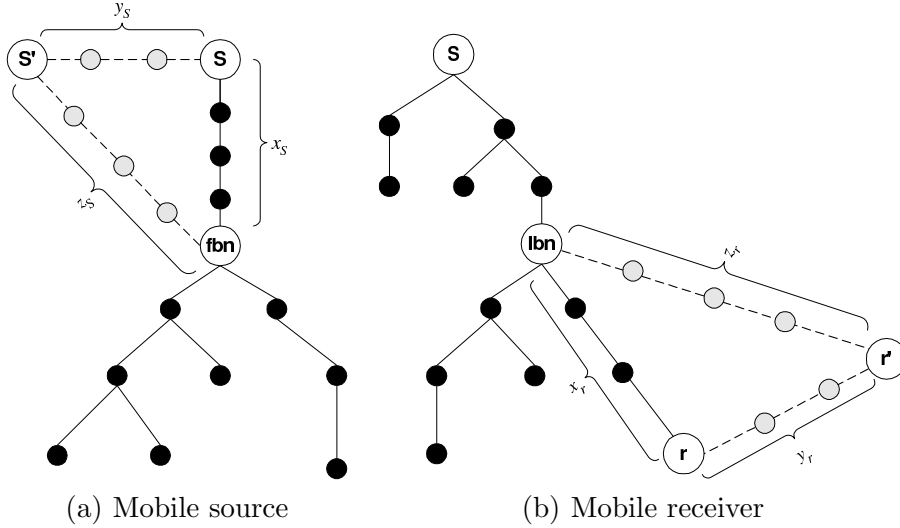


Figure 6. Eliminating triangular routing.

are uniformly distributed in the network. In practice, though, the distribution may not be uniform. Nevertheless, this scenario provides a worst case bound.

The other two sides of the triangle, y_S and z_S , depend on S' . Intuitively, the farther the source moves from its home network, the smaller the difference $y_S - z_S$. Moreover, y_S and z_S tend to be close, even if S stays near its home network. This is because the source moves between edge networks, which are usually not directly connected. Therefore, the tunnel from S' to S includes some core routers. On the other hand, the first branching node of the tree is usually in the core; thus z_S is less influenced by the shape of the underlying topology.

Therefore, for a group of m receivers, a fixed tunnel size $y_S = c$, and considering that $y_S \approx z_S$, we obtain:

$$\Delta Cost(S, S') \approx \frac{x_S}{c + \theta(m^{0.8})} \quad (20)$$

where we used the Chuang-Sirbu scaling law [24] to estimate the size of the tree with m receivers ($L_S(m) = \theta(m^{0.8})$).

Similarly, the *relative gains in average delivery delay* are defined as:

$$\Delta Del(S, S') = \frac{Del_{BT}(S, S') - Del_{MHBH}(S, S')}{Del_{BT}(S, S')} = \frac{x_S + y_S - z_S}{y_S + \sum_{i=1}^m d(S, r_i)/m} \quad (21)$$

where the propagation delay, D_{Prop} , was simplified. Thus, for a fixed tunnel size $y_S = c_1$, a fixed average distance c_2 between the source and the receivers,

and considering that $y_S \approx z_S$, we obtain:

$$\Delta Del(S, S') \approx \alpha x_S \quad (22)$$

where $\alpha = (c_1 + c_2)^{-1}$. Note that the relative gains in both total delivery cost and average delivery delay are dependent of x_S , i.e., the number of links between the source and the first branching node of the multicast tree.

The case of receiver mobility can be treated similarly (Figure 6(b)). We define *relative gains in delivery delay* for a mobile multicast receiver as:

$$\Delta Del(r, r') = \frac{Del_{BT}(r, r') - Del_{MHBH}(r, r')}{Del_{BT}(r, r')} = \frac{x_r + y_r - z_r}{d(S, r) + y_r} \quad (23)$$

where the propagation delay, D_{Prop} , was simplified. x_r is the distance between the home network of receiver r and its last branching node; y_r is the length of the BT tunnel between r and r' ; z_r is the length of the direct path between r' and the last branching node.

Similarly to source mobility, x_r depends on the group size; the larger the group, the higher the probability of a late branching in the tree. The other two sides of the triangle, y_r and z_r , can be estimated as in the case of mobile sources. Thus, for a fixed tunnel size $y_r = c_1$, a fixed distance c_2 from the source to the receiver's home network, and supposing that $y_r \approx z_r$, we have:

$$\Delta Del(r, r') \approx \beta x_r \quad (24)$$

where $\beta = (c_1 + c_2)^{-1}$. Thus, the relative gains in delivery delay for a mobile receiver depend on x_r , i.e., the number of links between its home network and its last branching node on the multicast tree.

In the following, we evaluate x_S and x_r for two different mathematical models, *k-ary* and *self-similar* trees. Then, we compare these theoretical models with results obtained through simulation.

4.3.1 Evaluation for K-ary Trees

Multicast trees are often modeled as subsets of perfect k-ary trees [25,26]. Consider a k-ary tree of depth D , rooted at the source S , and with all receivers placed on leaves. We randomly choose m distinct leaves, and construct the corresponding tree. Then, we evaluate x_S and x_r (assuming that $m > 1$).

Consider node N at level j in the k-ary tree; k^j such nodes exist. For each receiver r_i , there is a branch $b(r_i)$ in the tree that connects r_i to S . Each $b(r_i)$

includes one of the k^j nodes from level j . Thus, $N \in b(r_1)$ if r_1 is placed at one of the leaves of the sub-tree below N . As this sub-tree has k^{D-j} leaves, the probability that r_1 is placed at one of these nodes is k^{D-j}/k^D . Therefore, $P(N \in b(r_1)) = k^{D-j}/k^D$. Once r_1 is placed, $(k^{D-j} - 1)$ leaves remain available in the sub-tree. Then, $P(N \in b(r_2)|N \in b(r_1)) = (k^{D-j} - 1)/(k^D - 1)$. Thus, the probability of N being included in all $b(r_i)$, $i = 1..m$, is:

$$P\left(\bigcap_{i=1}^m b(r_i)\right) = \prod_{i=0}^{m-1} \frac{k^{D-j} - i}{k^D - i} \quad (25)$$

Proposition 1: For a given tree, the *fbn* is situated at level j , ($j < D$), if the following events occur simultaneously:

- $e_1(j)$: “All branches, corresponding to the m receivers, include the same node at level j ”;
- $e_2(j)$: “Among the k child nodes of the common node at level j , no node is included in all $b(r_i)$, $i = 1..m$ ”.

As $m > 1$, *fbn* cannot be located at level D . To obtain the probability of event e_1 , we use Eq. 25. There are k^j nodes at level j , and the events $e(N_h) = "N_h \in b(r_i), \forall i = 1..m"$, $h = 1..k^j$, are mutually exclusive. Thus,

$$P(e_1(j)) = P\left(\bigcup_{h=1}^{k^j} e(N_h)\right) = \sum_{h=1}^{k^j} P\left(N_h \in \bigcap_{i=1}^m b(r_i)\right) = k^j \prod_{i=0}^{m-1} \frac{k^{D-j} - i}{k^D - i} \quad (26)$$

The probability of e_2 is also obtained from Eq. 25. We choose a node N at level j so that there is a receiver r for which $N \in b(r)$. If e_1 occurs, then all the m receivers are included in the sub-tree below N . We are interested in the probability of having no node at level 1 of this sub-tree that is included in all $b(r_i)$, $i = 1..m$. As the sub-tree is a k -ary tree of depth $D - j$, we obtain:

$$P(e_2(j) | e_1(j)) = 1 - k \prod_{i=0}^{m-1} \frac{k^{D-j-1} - i}{k^{D-j} - i} \quad (27)$$

Some special cases exist. It is impossible to place more than k^{D-j} receivers on the leaves of a k -ary sub-tree of depth $D - j$. Thus, if $m > k^{D-j}$, then $P(e_1(i)) = 0$, $\forall i = j..D$. In other words, $P(e_1(j)) = 0$, $\forall j > D - \ln m / \ln k$.

Finally, from Proposition 1, if $P(e_1(j)) = 0$, then $P(e_2(j - 1)) = 1$, $\forall j > 0$. Therefore, the probability that *fbn* is situated at level j is:

$$P(x_S = j) = P(e_1(j))P(e_2(j) | e_1(j)) = \quad (28)$$

$$\begin{cases} k^j \left(1 - k \prod_{i=0}^{m-1} \frac{k^{D-j-1-i}}{k^{D-j-i}}\right) \prod_{i=0}^{m-1} \frac{k^{D-j-i}}{k^{D-i}}, & 0 \leq j \leq D - 1 - \frac{\ln m}{\ln k} \\ k^j \prod_{i=0}^{m-1} \frac{k^{D-j-i}}{k^{D-i}}, & D - 1 - \frac{\ln m}{\ln k} < j \leq D - \frac{\ln m}{\ln k} \\ 0, & D - \frac{\ln m}{\ln k} < j \leq D \end{cases}$$

Then, we compute x_S as a weighted average:

$$x_S = \sum_{j=0}^{D-1} j P(x_S = j) \quad (29)$$

We proceed similarly for the calculus of x_r . Consider that receiver r is placed at one of the m distinct leaf nodes selected on the k -ary tree. Let N_i denote the node that is included in the branch $b(r)$ at level i .

Proposition 2. $d(r, lbn(r)) = j$ ($j > 0$) if both events occur:

- $e_1(j)$: “There is no receiver, apart from r , in the sub-tree below node N_{D-j+1} ($j > 1$)”;
- $e_2(j)$: “There is at least one receiver other than r in the subtree below node N_{D-j} ”.

As $m > 1$, lbn cannot be at level D . There are $k^{j-1} - 1$ free leaves in the sub-tree below N_{D-j+1} (r is already on a leaf). Thus, the probability that receiver r_1 is not placed at one of these free nodes is $1 - (k^{j-1} - 1)/(k^D - 1)$.

Once r_1 is placed outside the sub-tree, there are $k^{j-1} - 1$ free leaves in the sub-tree. Nevertheless, the total number of available leaves in the k -ary tree has been reduced to $k^D - 2$. As we have to place $m - 1$ receivers outside the subtree, the probability of event e_1 is:

$$P(e_1(j)) = \prod_{i=1}^{m-1} \left(1 - \frac{k^{j-1} - 1}{k^D - i}\right) \quad (30)$$

Given that event e_1 occurred, event e_2 occurs if at least one of the $m - 1$ receivers on the $k^j - k^{j-1}$ remaining leaves is placed on the sub-tree below node N_{D-j} . Therefore, we obtain:

$$P(e_2(j) | e_1(j)) = 1 - \prod_{i=0}^{m-2} \left(1 - \frac{k^j - k^{j-1}}{k^D - k^{j-1} - i}\right) \quad (31)$$

Thus, $P(x_r = j) = P(e_1(j))P(e_2(j) | e_1(j)) =$

$$\left[1 - \prod_{i=0}^{m-2} \left(1 - \frac{k^j - k^{j-1}}{k^D - k^{j-1} - i} \right) \right] \prod_{i=1}^{m-1} \left(1 - \frac{k^{j-1} - 1}{k^D - i} \right) \quad (32)$$

We then calculate x_r as a weighted average

$$x_r = \sum_{j=1}^D j P(x_r = j) \quad (33)$$

4.3.2 Self-Similar Trees

Adjih et al. [27] argue that self-similar trees better approximate Internet trees than k-ary trees. Suppose a k-ary tree of depth D , with all receivers located on the leaves. Suppose that between a node at level $l - 1$ and a node at level l there are a random number of concatenated links. Let t_l denote the average number of these links. Then, a self-similar tree with similarity factor θ , is a tree for which $t_{l-1} = \theta t_l$, $\theta = 1$, $l = 2..D$. Thus, $t_l = \theta^{D-l} t_D$. For simplicity, we assume that $t_D = 1$. Note that if $\theta = 1$, we have a regular k-ary tree.

In such a self-similar tree, if fbn is at level j ($j > 0$), then the number of links between S and fbn is:

$$\sum_{i=1}^j t_i = \sum_{i=1}^j \theta^{D-i} = \theta^{D-j} \left(\frac{\theta^j - 1}{\theta - 1} \right) \quad (34)$$

If Eq. 29 is adapted to these conditions, we obtain:

$$x_S = \sum_{j=1}^{D-1} \theta^{D-j} \left(\frac{\theta^j - 1}{\theta - 1} \right) P(x_S = j) \quad (35)$$

where $P(x_S = j)$ is obtained from Eq. 28. Similarly, the number of intermediate links between receiver r and $lbn(r)$, situated j levels above, is:

$$\sum_{i=0}^{j-1} t_{D-i} = \sum_{i=0}^{j-1} \theta^i = \frac{\theta^j - 1}{\theta - 1} \quad (36)$$

If Eq. 33 is adapted to these conditions, we obtain:

$$x_r = \sum_{j=1}^D \left(\frac{\theta^j - 1}{\theta - 1} \right) P(x_r = j) \quad (37)$$

where $P(x_r = j)$ is obtained from Eq. 32.

5 Simulation Results

In the previous section we proposed a theoretical analysis of the gains provided by the M-HBH protocol, for the specific cases of k-ary and self-similar trees. Nevertheless, these theoretical models are restricted to simplified multicast trees. For example, to ease the calculus, we assumed that all receivers were situated at leaf nodes. Nevertheless, Chalmers and Almeroth argue that in realistic multicast trees receivers are not situated only at the bottom of the tree [28]. Thus, to improve our model, we can allow receivers to be spread uniformly over the entire tree.

We also assumed that receivers were uniformly distributed in the network. Nevertheless, in practice, receivers are more or less clustered (i.e., grouped together) in the topology. Taking into account receiver clustering could also improve the quality of our theoretical models. A more detailed analysis on the impact of these additional requirements can be found in [23].

In the following, to complement the theoretical results, we evaluate the efficiency of M-HBH using a realistic Internet-like topology.

5.1 Topology Generation

Magoni and Pansiot [29] compared the main topology generators currently available. We chose to use *nem* [30], a topology generator based on Internet map sampling. *Nem* extracts a random sub-graph with properties (such as node degree, mean distance, mean eccentricity, and topology diameter) similar to the original map.

We generated a topology with 2000 nodes and 2600 edges, a mean degree of 2.6, a maximum degree of 33, a mean distance of 8.08 hops, a mean eccentricity of 14.07, and a diameter of 20 hops. The reduced value of the mean degree (real Internet maps have a mean degree of around 3.1) was chosen in order to guarantee a realistic value for the mean distance between any two nodes in the topology (around 9 hops [30]).

The shape of the multicast tree depends on the underlying network. Chalmers and Almeroth [28] observed that the skewed distributions and the power laws that characterize the Internet also hold for multicast trees. As a consequence, realistic multicast trees on the Internet also have a high percentage of leaf and

relay nodes. Moreover, receivers are not situated only in the deepest part of the tree, and branching occurs throughout the whole height of the tree.

5.2 Distances to the First and Last Branching Nodes

On the 2000-node topology, we vary the group size and randomly select the source and receivers. Then, using Dijkstra’s algorithm, we compute the shortest path tree connecting the source and receivers. We repeat the experience 50 times for each group size, and we compute the average values of x_S and x_r on the resulting trees.

Figure 7(a) shows that both distances decrease exponentially, and that x_r is almost twice as large as x_S . Thus, M-HBH gains are expected to be larger for receiver mobility. Note that the x_r curve is smoother, as it uses average values of distances between receivers and their last branching nodes. The larger the group, the lower the influence of an “extreme” case (a receiver too far from, or too close to its *lbn*) on the average. On the other hand, the source is unique. Thus, an “extreme” case (the source too far from, or too close to the *fbn*) biases the values measured. Nevertheless, by increasing the number of generated trees from 50 to 200, we obtain a smoother x_S curves (Figure 7(b)).

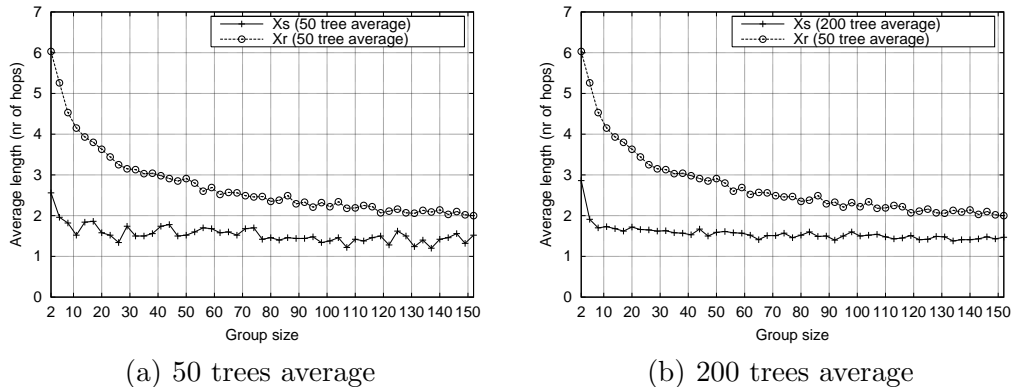


Figure 7. Evaluation of x_S and x_r for different group sizes.

Figure 8 compares theoretical and simulation results. First, the k-ary tree model is analyzed (Figures 8(a) and 8(c)). The values of k and D were chosen to obtain trees of size comparable to the generated topology (around 2000 nodes). We studied two “extreme” cases: a “deep” ten-level binary tree ($k = 2$, $D = 10$), and a “wide” three-level tree of degree 12 ($k = 12$, $D = 3$). Note that the mathematical model gives a reasonable approximation for x_r (Figure 8(c)). Nevertheless, Figure 8(a) shows that the k-ary model is less precise for x_S ; the k-ary curves rapidly converge to zero, as opposed to the one obtained through simulation. This is because x_r is an average value computed over all receivers. On the other hand, there is only one source, thus a “badly” placed receiver

(placed in a part of the topology that does not contain other receivers) can result in the fbn being the source itself. In uniformly connected mathematical models the probability of a “bad” placement is high, therefore x_S rapidly converges to zero. Nevertheless, for realistic topologies, with variable connectivity and unevenly populated regions, the convergence to zero is slower.

Figures 8(b) and 8(d) show the results for self-similar trees. We observe that the differences between theoretical and simulation results are reduced by tuning the similarity factor θ . Note that higher values of θ increase the topology size. In the case of x_S , the introduction of the similarity factor improves the approximation; nevertheless, the rapid convergence to zero persists (Figure 8(b)). Therefore, simulation results are particularly useful to evaluate source mobility. Indeed, the performance of M-HBH obtained through simulation is better than that predicted by both k-ary and self-similar models.

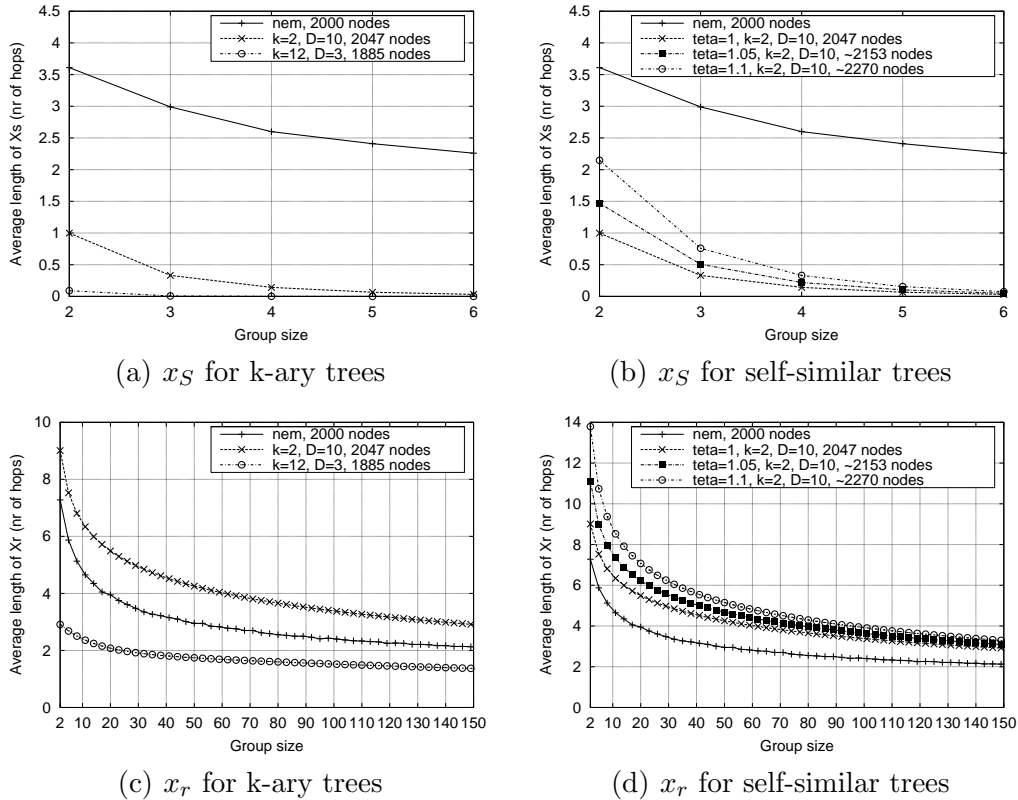


Figure 8. Comparing simulation results with theoretical models.

In order to check the correctness of our mathematical model, we generated two sample k-ary trees, for $(k = 2, D = 10)$ and $(k = 12, D = 3)$ respectively. We varied the number of receivers and randomly generated 50 multicast trees for each group size. Each multicast tree was computed as a sub-tree of the global k-ary tree; the root of the k-ary tree represented the multicast source, while the receivers were randomly chosen among its leaf nodes. Then, we computed x_S and x_r for these trees. Results are shown in Figure 9. Solid lines represent

the different curves corresponding to Eq. 29 and Eq. 33, while dashed lines represent the corresponding simulation results. We observe that differences are minimal, confirming the correctness of our mathematical models.

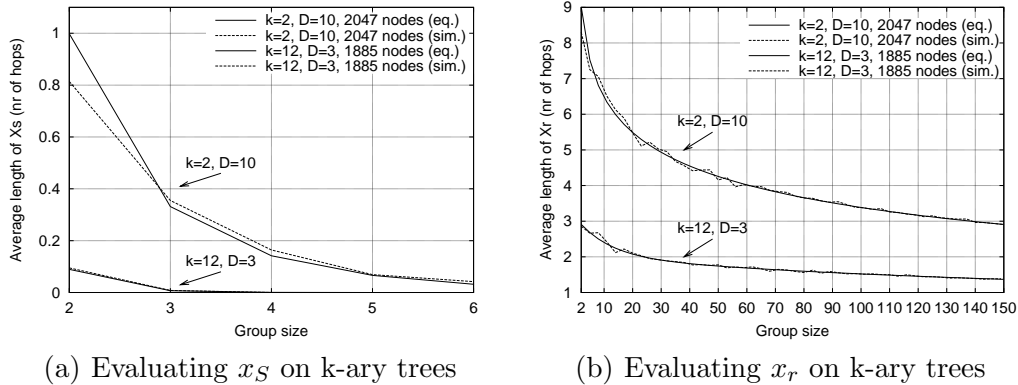


Figure 9. Verifying the correctness of the mathematical model.

5.3 Simulation Results for Source Mobility

In the following, we present a comparison of the simulation results obtained for the three different approaches: M-HBH, BT, and RS. To analyze source mobility, we considered different group sizes, and generated 200 trees for each size; the source and the receivers are leaf nodes of the topology. The source moves 200 times, always choosing the current location among leaf nodes. Then, we calculated average values for our metrics. These values do not correspond to continuous consecutive movements of the source, as the topology generator we used does not provide geographical location information for nodes. Thus, we could not simulate movement between geographic-neighbor networks. Rather, we evaluated each case as a snapshot during the source movement.

The relative gains in total delivery cost for M-HBH over BT decrease exponentially with the group size (Figure 10(b)), confirming Eq. 20. For small groups (2 to 4 members), the relative gain is about 15-20%, but as the group size increases, the gains decrease exponentially (about 0.7% for groups with 150 members). Note that Figure 10(a) presents results for small groups (up to 20 members). As the total delivery cost increases linearly with the group size the differences are small for large groups.

On the other hand, relative gains in average delivery delay decrease exponentially with group size (Figure 10(d)). For small groups, we have a relative gain of about 25-30%, while for groups as large as 150 members, the gains are around 15%. In the same time, by plotting x_S on the same graph (using the right hand y axis), we observe that the two curves have similar shapes; they decrease proportionally, which confirms Eq. 22.

To evaluate signaling overhead, we used 10 reporting cycles ($n = 10$). The receivers send one *join* message per cycle. During the 10 cycles, the source moves once ($c = 1$). Simulation results are presented in Figure 10(e). As network changes are not frequent, the remote subscription solution does not require frequent tree reconstructions. Thus, there is no significant difference in signaling overhead between M-HBH and RS.

The difference between M-HBH and BT signaling overhead is due to M-HBH's *tree* messages; the difference between RS and BT, to the additional BU tree used by RS. Nevertheless, BT and RS signaling are underestimated, as traditional multicast protocols that are used with either BT or RS use additional control messages (Section 4.1).

Figure 10(f) shows results for a higher source mobility; we assumed that during the evaluation period the source moves five times ($c = 5$). We observe that RS signaling overhead increases significantly, due to the frequent tree reconstructions. On the other hand, neither M-HBH nor BT performances are affected by this high mobility.

5.4 Simulation Results for Receiver Mobility

To analyze receiver mobility, we proceeded in a similar way as for source mobility. We considered different group sizes, and generated 50 trees for each size. Then, we chose 10 receivers randomly (or all the receivers, if group size was smaller than 10), and moved them 10 times, always among leaf nodes.

Simulation results are presented in Figure 11. We observe that, for small groups, average delivery delay in M-HBH approximates average delivery delay in RS (Figure 11(a)); thus, M-HBH delivers data on a nearly-shortest path tree, without any tree reconstruction. M-HBH average delivery delay increases logarithmically with the group size. Nevertheless, for groups as large as 150 members, M-HBH still performs better than BT.

The relative gains in average delivery delay decrease exponentially with the group size (Figure 11(b)). For small groups, gains are as large as 40%, while for large groups, the gains are around 20%. At the same time, by plotting x_r on the same graph (using the right hand y axis), we observe that the two curves have similar shapes; they decrease proportionally, which confirms Eq. 24.

Simulation results for service interruption are shown in Figure 11(c). We observe that RS service interruption decreases exponentially with the group size; the larger the group, the faster a remote subscription *join* message reaches the multicast tree. The average value of BT service interruption is approximately constant; it corresponds to the average tunnel size between two edge networks

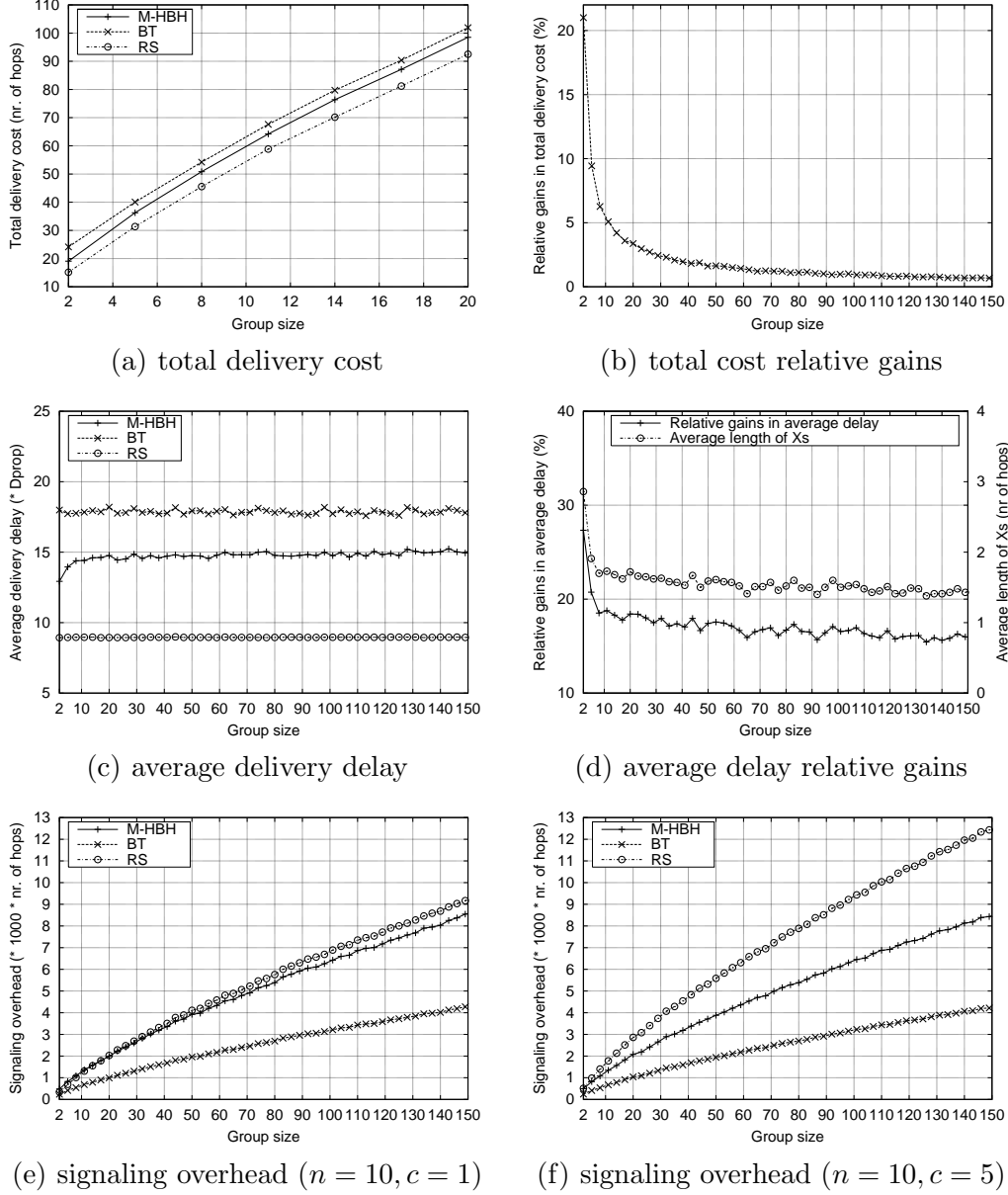


Figure 10. Simulation results for source mobility.

of a realistic Internet-like topology. Finally, M-HBH service interruption decreases exponentially with group size; the larger the group, the closer the last branching node of the mobile receiver to its home agent. Thus, the *join* message containing the binding update, tunneled to the HA and then forwarded on the M-HBH control tree, reaches the last branching node faster. For large groups, M-HBH service interruption approximates BT-service interruption.

If source and receiver mobility occur simultaneously, M-HBH performance should further improve, when compared to traditional bi-directional tunnelling; the higher the number of mobile nodes, the higher the number of M-HBH routing triangle shortcuts as well. Nevertheless, to keep concise, we did not present

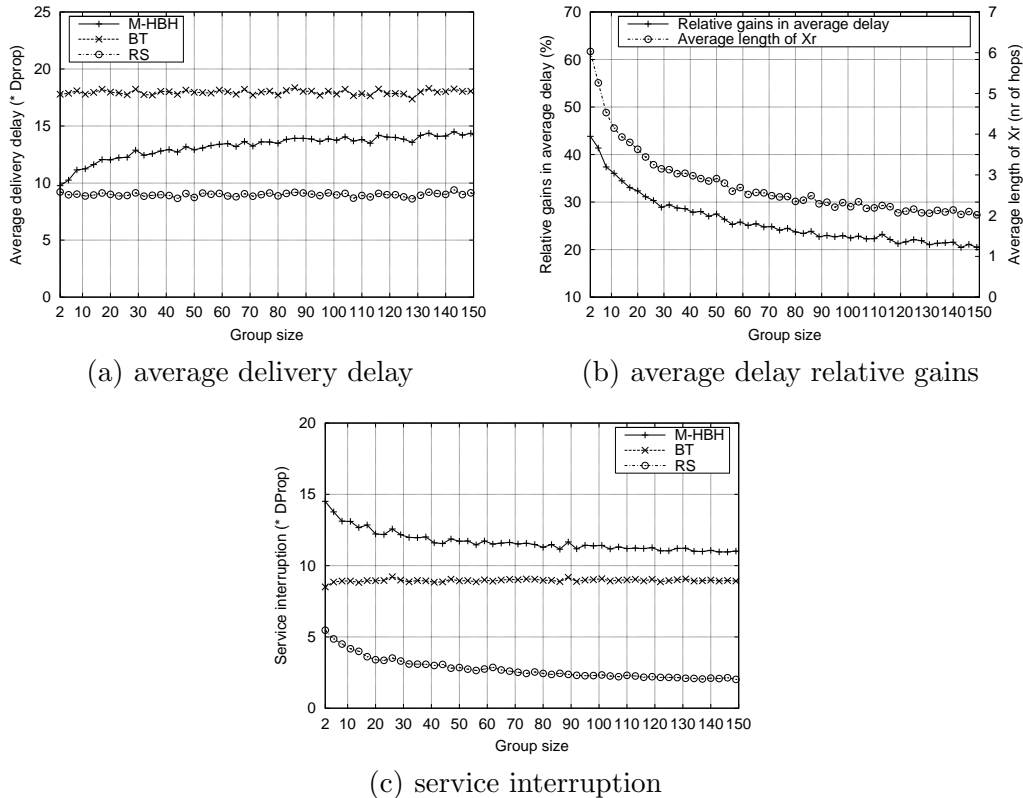


Figure 11. Simulation results for receiver mobility.

numerical results for such a combined scenario.

On the other hand, the higher the number of mobile nodes, the larger the impact of a realistic mobility pattern on the quality of the results. Therefore, in the case of more complex mobility scenarios the geographical location of the nodes and the physical vicinity of the access networks the nodes can possibly move through should be taken into account. The extension of this study through the use of a topology generator able to provide such location information is subject of future work.

6 Applicability Analysis

In this paper we presented a new solution to handle the mobility of both the source and the receivers of an ongoing multicast session. The proposed approach is based on a recursive unicast addressing scheme to provide the multicast service. This section presents some applications and scenarios that can benefit from M-HBH.

From the theoretical analysis we can conclude that M-HBH offers performance gains mainly for small source-specific multicast groups, with highly mobile

members. For small multicast groups there is a high probability that the first branching node is many hops away from the mobile source. In such cases short-cutting the routing triangle significantly improves the performance. Among the possible applications that use such groups we can mention video-conferences, chat, and collaborative work tools.

On the other hand, M-HBH performs well even for large groups, if the receivers are all located in the same geographical area. E-learning can be such an application. A mobile professor, located in a distant country, presents a lecture to hundreds of students, which are located in the same city or region; due to receiver clustering, the first branching node is situated far from the mobile source.

As far as receiver mobility is concerned, M-HBH performance improves if the last branching node is far from the mobile receiver. This usually happens in the case of small multicast groups; the smaller the group, the higher the probability to have receivers located far from each other.

In all the cases mentioned above, a mobile source (or receiver) located far from its corresponding first (or last) branching node makes M-HBH a significantly better choice than traditional bi-directional tunnelling. On the other hand, the mobility pattern also affects M-HBH performance. M-HBH is well suited for scenarios where the mobile nodes move frequently, but not too far from the home network. Frequent updates of the multicast tree are avoided with low performance loss, compared to the shortest path tree.

Finally, we stress that M-HBH supports the multicast service even over unicast-only clouds. Therefore, M-HBH can be easily applied over different access networks or bearer technologies that do not support native multicast; no encapsulation or tunnelling is needed to pass through unicast segments.

6.1 Application Example

We now present in detail a typical application scenario for M-HBH. We consider the case of the CEO of a Hungarian software company that wants to hold a confidential video-conference with the company's Board of Directors (e.g., five members). Suppose that the board members sit in their respective offices, which are spread across different Hungarian cities, while the CEO is travelling on a high-speed train in Japan.

In the traditional case, this scenario would be handled through six SSM channels, where the CEO and each additional board member would be the root of a SSM tree. A participant should join each of the channels it is not the root of, to receive traffic from all the other participants. (A restricted scenario can be

assumed if some board members are only listeners; in this case these members do not have their own SSM channel).

Consider now that the CEO's SSM channel is replaced by an M-HBH tree. As all the board members are grouped together in a restricted geographical area, the first branching node in the tree is situated somewhere in Hungary, many hops away from the CEO's access server. As the CEO moves around rapidly, it changes its attachment points frequently. Nevertheless, it continues to send its audio and video stream to the same branching point, without interruption. On the other hand, if one of the board members answers to the CEO, the CEO becomes a mobile receiver. If the board member's SSM channel is replaced by an M-HBH tree, the last branching node that handles the mobile CEO will be situated somewhere in Hungary, many hops away from the mobile receiver. In both cases, M-HBH provides performance gains in handling the mobile nodes.

7 Conclusions

Efficient handling of mobile nodes in a multicast environment is a non-trivial task. Bi-directional tunnelling (BT) introduces encapsulation and triangular routing. Remote subscription (RS) needs tree reconstruction, which is inefficient for rapidly moving nodes. In this paper we proposed the M-HBH multicast routing protocol, an approach based on a recursive unicast addressing scheme. M-HBH reduces routing triangles, limits tree reconstruction, avoids encapsulation and tunnelling.

M-HBH provides a trade-off between shortest path delivery and tree reconstruction; its performance depends largely on the shape of the multicast tree. First, we analyzed the protocol behavior with two mathematical models, k-ary and self-similar trees. Then, we used *nem*, an Internet-like topology generator based on map sampling, to provide more realistic models. We simulated the behavior of M-HBH on a topology with 2000 nodes.

M-HBH was compared with BT and RS using different metrics: the total delivery cost, the average delivery delay, the signaling overhead, and the service interruption. Simulation results showed the efficiency of M-HBH, especially in the case of relatively small multicast groups. As an example, relative gains in average delivery delay, for M-HBH over BT, varied from 30 to 40% for small groups (5 to 10 members), and from 15 to 20% for groups with 150 members.

Besides the performance gains, M-HBH has two major advantages which are intrinsic to the protocol and hardly quantifiable. On the one hand, it provides a simple and transparent method to handle the mobility of the root of a source-specific multicast tree, which is the most complex case of multicast

node mobility. On the other hand, M-HBH provides an incrementally deployable solution, where intermediate unicast nodes participate in both mobility handling and multicast data delivery.

References

- [1] C. Perkins, *IP Mobility Support for IPv4*, RFC 3220, Jan. 2002.
- [2] L. H. M. K. Costa, S. Fdida, and O. C. M. B. Duarte, "Hop by hop multicast routing protocol," in *ACM SIGCOMM'2001*, pp. 249–259, Aug. 2001.
- [3] J. Ioannidis, D. Duchamp, and G. Maguire, "IP-based protocols for mobile internetworking," in *ACM SIGCOMM'91*, pp. 235–245, Sept. 1991.
- [4] A. Acharya, A. Bakre, , and B. Badrinath, "IP multicast extensions for mobile internetworking," in *IEEE INFOCOM'96*, pp. 67–74, Mar. 1996.
- [5] R. Droms, *Dynamic Host Configuration Protocol*, RFC 2131, Mar. 1997.
- [6] D. B. Johnson, C. E. Perkins, and J. Arkko, *Mobility Support in IPv6*, June 2003. Work in progress, <draft-ietf-mobileip-ipv6-24.txt>.
- [7] S. Thomson and T. Narten, *IPv6 Stateless Address Autoconfiguration*, RFC 2462, Dec. 1998.
- [8] R. Droms, J. Bound, B. Volz, T. Lemon, C. Perkins, and M. Carney, *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*, RFC 3315, July 2003.
- [9] T. G. Harrison, C. L. Williamson, W. L. Mackrell, and R. Bunt, "Mobile multicast (MoM) protocol: Multicast support for mobile hosts," in *ACM/IEEE Mobicom'97*, pp. 151–160, Sept. 1997.
- [10] C. R. Lin and K. M. Wang, "Mobile multicast support in IP networks," in *IEEE INFOCOM'00*, Mar. 2000.
- [11] Y. Wang and W. Chen, "Supporting IP multicast for mobile hosts," *ACM/Kluwer Mobile Networks and Applications*, vol. 6, no. 1, pp. 57–66, Jan. 2001.
- [12] J. McAuley, E. Bommaiah, A. Misra, R. Talpade, S. Thomson, and K. C. Y. Jr., "Mobile multicast proxy," in *IEEE Military Communications Conference (MILCOM'99)*, vol. 1, pp. 631–635, Oct. 1999.
- [13] K. H. Chi, C. C. Tseng, and T. L. Huang, "A framework for mobile multicast using dynamic route reconstructions," *The Computer Journal*, vol. 42, no. 6, no. 6, pp. 522–533, 1999.
- [14] H. Omar, T. Saadawi, and M. Lee, "Multicast support for Mobile-IP with the hierarchical local registration approach," in *ACM International Workshop on Wireless and Mobile Multimedia (WoWMoM'00)*, pp. 55–64, Aug. 2000.

- [15] I. Stoica, T. S. E. Ng, and H. Zhang, “REUNITE: A recursive unicast approach to multicast,” in *IEEE INFOCOM’2000*, Mar. 2000.
- [16] D. Waitzman, C. Partridge, and S. Deering, *Distance Vector Multicast Routing Protocol*. RFC 1075, Nov. 1988.
- [17] S. Bhattacharyya, Ed., *An Overview of Source-Specific Multicast (SSM)*, RFC 3569, July 2003.
- [18] 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, June 1998.
- [19] P. Ferguson and D. Senie, *Network Ingress Filtering: Defeating Denial of Service Attacks which Employ IP Source Address Spoofing*, RFC 2827, May 2000.
- [20] 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 Transactions on Networking*, vol. 4, no. 2, pp. 153–162, Apr. 1996.
- [21] B. Cain, S. Deering, I. Kouvelas, B. Fenner, and A. Thyagarajan, *Internet Group Management Protocol, Version 3*, RFC 3376, Oct. 2002.
- [22] R. Vida, Ed., L. Costa, Ed., *Multicast Listener Discovery Version 2 (MLDv2) for IPv6*, June 2003. Work in progress, <draft-vida-mld-v2-07.txt>.
- [23] R. Vida, *Protocol Design for Group and Mobility Management in a Multicast Environment*. Ph.D. thesis, Université Pierre et Marie Curie (Paris VI), Paris, France, Dec. 2002.
- [24] J. Chuang and M. Sirbu, “Pricing multicast communications: A cost-based approach,” in *Internet Society’s Annual Networking Conference (INET’98)*, July 1998.
- [25] G. Phillips, S. Shenker, and H. Tangmunarunkit, “Scaling of multicast trees: Comments on the Chuang-Sirbu scaling law,” in *ACM SIGCOMM’99*, pp. 41–51, Sept. 1999.
- [26] P. V. Mieghem, G. Hooghiemstra, and R. V. der Hofstad, “On the efficiency of multicast,” *IEEE/ACM Transactions on Networking*, vol. 9, no. 6, pp. 719–732, Dec. 2001.
- [27] C. Adjih, L. Georgiadis, P. Jacquet, and W. Szpankowski, *Is the Internet Fractal?*, 2001. Research Report 4157.
- [28] R. C. Chalmers and K. C. Almeroth, “Modeling the branching characteristics and efficiency gains in global multicast trees,” in *IEEE INFOCOM’2001*, Apr. 2001.
- [29] D. Magoni and J.-J. Pansiot, “Analysis and comparison of Internet topology generators,” in *IFIP/TC6 Networking’02*, May 2002.
- [30] D. Magoni and J.-J. Pansiot, “Internet topology modeler based on map sampling,” in *IEEE International Symposium on Computer Communications*, July 2002.