Matthew Cheng
John Lee
Mario Joa-Ng
Telcordia Technologies, Inc.
December 2000

Hierarchical Level-based IP Multicast (HLIM)

STATUS OF THIS MEMO

ABSTRACT

This document proposes and describes a new IP multicast routing protocol, called "Hierarchical Level-based IP Multicast" (HLIM). It is designed for a hierarchical IP network and leverages the hierarchical structure to provide efficient IP multicasting. It is originally designed for tactical networks in battlefields but some of its functions could be applied to commercial networks as well. Besides the basic support for multicast in static environments, HLIM supports host mobility (movements of IP multicast hosts) and network mobility (movements of IP multicast routers with/without hosts). Since HLIM is designed for harsh environments, reliability and survivability functions are also built into HLIM. Unlike other existing IP multicasting schemes, HLIM can provide a shared and shortest-path multicast tree without using a concentration point like a core or rendezvous point. The shortest paths between multicast sources and receivers can still be maintained after a host or network movement. HLIM also allows better control of multicast session establishments by requiring both sources and receivers to "join" a multicast group before they can send or receive any multicast packets.

TABLE OF CONTENTS

1. INTRODUCTION

Existing multicasting schemes are basically differentiated by how a tree (called "spanning tree" or "multicast tree") is set up to forward multicast packets from a sender to the receivers within a multicast group. Such schemes can be classified into two main categories: source-based tree (e.g., distance vector multicast routing protocol (DVMRP), multicast open shortest path first (MOSPF), and protocol independent multicast-dense mode (PIM-DM) [1]-[3]) and shared tree (e.g., core based tree (CBT) and protocol independent multicast-sparse mode (PIM-DM) [4]-[6]). In the source-based tree approach, a multicast tree is created from each sender to all the receivers of a multicast group. This approach maintains the shortest paths but is less scalable, especially when there are many sources sending to the same multicast group. The shared tree approach mitigates the scalability problem by establishing a single multicast tree shared by all sources and receivers of a group. A tree rooted at a known concentration point (such as a core in CBT or a rendezvous point in PIM-SM) is created to all the receivers of a group. The sources send their packets to the root, which then disseminates the packets to the receivers. The drawback, however, becomes apparent when optimal routing is emphasized because the shared tree scheme provides the shortest path only between the root and receivers, not between a source and receivers. In addition, this approach requires complex mechanisms to advertise the addresses of the roots to every multicast host.

In addition to these shortcomings, existing IP multicasting schemes have no explicit provision for host and network mobility since the schemes have been mainly derived for fixed IP networks. Recently, some ideas for supporting host mobility have been proposed. The basic idea is to extend the current IP multicasting schemes to adapt to mobile environments with the help of mobile IP [7],[8]. However, this extended approach cannot completely resolve the mobility problems incurred by the movements of multicasting hosts. For example, in the source-based approach, the shortest path between a sender and receivers would not be maintained as the sender moves. To date, the network mobility issue has been addressed only rarely. Therefore, a new IP multicasting scheme that can support both host and network mobility efficiently and reliably is highly desirable.

Our multicasting protocol design, HLIM, can overcome the shortcomings of the conventional IP multicasting schemes and also support both host and network mobility. HLIM assumes that the IP multicast routers are arranged hierarchically as in tactical networks in battlefields. Each IP multicast router is assigned a level number corresponding to its level in the hierarchy. In addition to the conventional group identifier, a HLIM multicast address carries a scope information, which defines the highest and lowest levels in the hierarchy that a multicast packet can be forwarded to. HLIM establishes a multicast tree along the shortest paths from the sources to the receivers. It is also shared by all the sources and receivers of the same group. HLIM requires both sources and receivers to "join" a multicast group before they can send or receive multicast packets. This allows better control of multicast session establishments.

As a source or receiver moves, any ongoing multicast session involved is maintained by extending the tree to the new locations of the host. When a network of routers and hosts move, all the ongoing multicast sessions in the network are maintained. Moreover, only the parent router of the mobile network is aware of the movement and invokes appropriate operations. The movement is transparent to all other routers and hosts inside the mobile network. It should be noted that the resulting tree after a host or network movement is still shortest-path and shared. The mobility functions only apply to "ongoing" sessions. A mobile host or network with no session established does not need to do anything. When the mobile host or a host in the mobile network wants to participate a group at the new location, the baseline functions are invoked to join the group within the scope of the new location. There is no concept of "home" and "foreign" networks used in Mobile IP since we assume the users are always interested in "local" information.

Although HLIM is designed for hierarchical tactical networks, it could be applied to any other commercial networks arranged in a hierarchical manner. It could also be extended to an arbitrary network with the help of a clustering protocol to form a logical hierarchy. For example, the MMWN (multimedia support for mobile wireless networks) protocol discussed in [9] can be used to establish a logical hierarchical network in an ad-hoc network before applying HLIM to the ad-hoc network.

2. TACTICAL NETWORK ARCHITECTURE

A tactical IP network is a multi-level hierarchical mobile network where IP routers are hierarchically connected as shown in Fig. 1 [10]. Both IP hosts and routers are mobile in the network. The highest level is the brigade level (level 4) and the lowest level is the platoon level (level 1). An IP router at one level (e.g., brigade) interconnects directly with multiple (e.g., 3 to 5) IP routers at the next lower level (e.g., battalion). We refer to such a direct interconnection between one higher-level and multiple lower-level IP routers as a "core" tactical IP subnet. The tactical IP network can be pictured as an integration of the "core" subnets in a hierarchical manner.

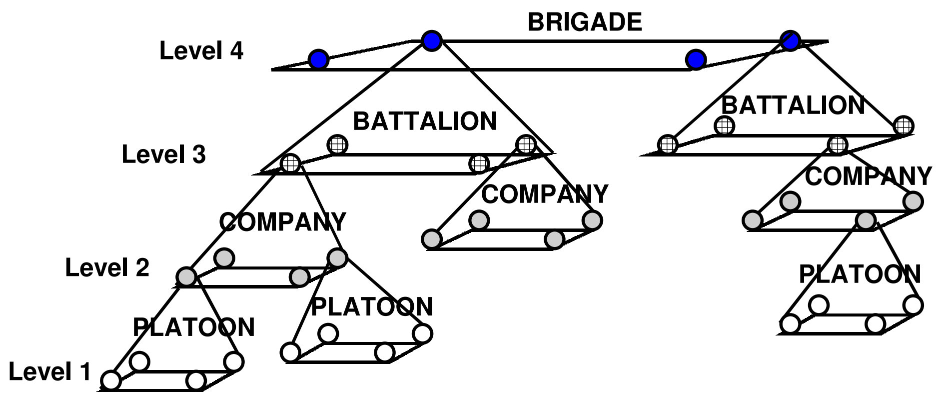

Fig. 1. Tactical IP Network.

Fig. 2 illustrates a core tactical IP subnet with local subnet attached at each router. The router at the higher level is labeled H_1, while the four routers at the lower level are labeled L_1 - L_4. Each router (except those located at the lowest or highest level) has up to three interfaces: local interface (L_IF), down interface (D_IF), and up interface (U_IF). A router uses its L_IF to handle traffic from and to its local subnet, D_IF for its descendant routers, and U_IF for its ascendant router and its adjacent routers at the same level. In each subnet (core or local), some wireless multiple access technology with broadcast/multicast capabilities (e.g., wireless LAN) is used to interconnect the IP hosts and/or routers. If the wireless multiple access technology has no broadcast/multicast capabilities (e.g., TDMA or CDMA), emulation for such capabilities may be required (e.g., multiple unicast transmissions).

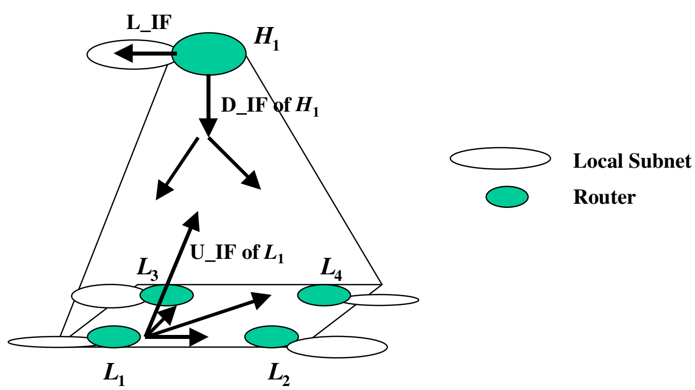

Fig. 2. Three possible interfaces in tactical IP subnets.

Though only one router is shown in a local subnet (or at the higher level in a core subnet) in the previous figures, it is possible to have more than one IP router to handle IP traffic between the local subnet and the ascendant/descendant routers as shown in Fig. 3. In unicast, this would not produce any duplicate messages because one preferred IP router among them is selected by the unicast routing protocol. However in multicasting, if a designated IP router is not assigned in advance, unnecessary duplicate multicast traffic may result. To avoid this situation, the HLIM protocol elects only one of the routers to handle the multicast traffic. The elected router is denoted as a hierarchical designated router (HDR). The HDR which handles multicast traffic for its local subnet is referred as a local HDR. The HDR which handles multicast traffic flowing between core subnets of adjacent levels is referred as a global HDR. The other non-HDR routers within a subnet ignore any multicast control or data packets received. Note that the other non-HDR routers may serve as backups for the elected HDRs, in case that a HDR moves away, has a malfunction, or is destroyed.

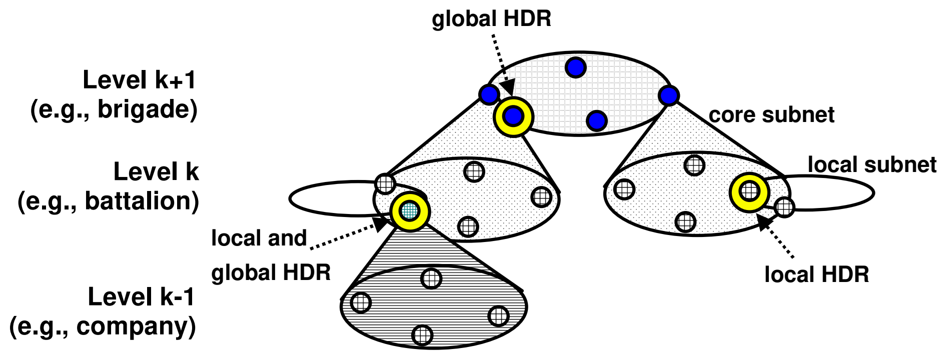

Fig. 3. Concept of hierarchical designated routers (HDR).

3. THE BASIC CONCEPT OF HLIM

The HLIM protocol provides optimal and efficient IP multicasting in a hierarchical IP network by taking advantage of the hierarchical structure. HLIM demonstrates the benefits of both a source-based algorithm (providing the shortest multicasting paths) and a shared tree algorithm (providing one shared tree per group for efficient utilization of network resources). The basic ideas of HLIM are:

1) to assign a hierarchical level for each IP router,
2) to create scope regions bounded by two hierarchical levels (one for the highest level and the other for the lowest level that a multicast packet for a group can reach), and
3) to include this scope information in a multicast group address.

Each multicast packet carries its own scope information that indicates how far it should traverse, and each IP router determines whether or not it should discard a multicast packet received by comparing its own level with the scope levels specified in the packet. In addition to packet forwarding, HLIM includes mechanisms to establish a multicast tree for a multicast session and maintain such a tree dynamically against host mobility, network mobility and network failure. Reliability of the control messages is also taken into account throughout the HLIM design.

3.1 HIERARCHICAL LEVELS OF ROUTERS

Each router has its own hierarchical level. If there are a total of N hierarchical levels in the whole network, routers at the highest level are assigned to level N and routers at the lowest level are labeled as level 1.

3.2 SCOPE REGIONS

A HLIM scope region is uniquely identified by a scope boundary, SB(L,H), and the root identity of the region, "root ID" (RID). The SB(L,H) is comprised of two hierarchical levels, where L sets the lower boundary and H sets the upper boundary that a multicast packet generated within the region can reach. The RID for a given SB(L,H) is the identity of the HDR at level H+1 in that region (denoted as HDR(H+1)). It can be some identifier assigned to the root HDR of the region (which should be unique among all HDRs at the same level of the hierarchical IP network), or the IP address of the HDR.

Fig. 4 shows some possible HLIM scope regions. Note that there is no RID for SB(L=1, 2, 3 or 4, H=4). In this case, null is used as the RID in the group address. Multicast packets associated with, say SB(2,3), are not allowed to move beyond the HDRs at level 3 (e.g., battalion level) and below the HDRs at level 2 (e.g., company level) in normal cases. The packets will be forwarded beyond the region only when a mobile user who has already established a multicast session at its original location moves outside the scope region and is willing to continue that multicast session. As shown in the figure, a scope region is defined in a way that the multicast packets from any host within a scope region are delivered to any other host within the same scope region without traversing other scope regions. Each HDR should keep a list of the IDs of its ascendant HDRs (e.g., a HDR(2) keeps the IDs of HDR(3) and HDR(4)). Such list is used by the HDR to determine the appropriate HLIM operation when a mobile host/network moves to this HDR and wants to maintain its on-going multicast sessions.

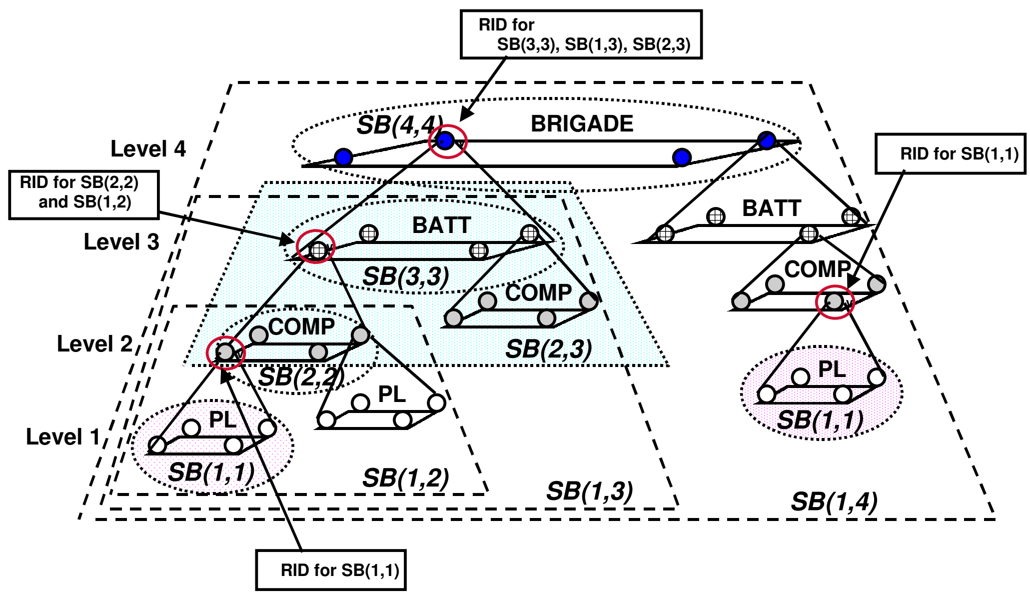

Fig. 4 HLIM scope regions

3.3 HLIM MULTICAST ADDRESS

A complete HLIM group address is comprised of the scope boundary (SB(L,H)), the root ID (RID) of the scope region, and the application ID (APL_ID) identifying a specific multicast application/service. It is denoted as G(L,H,RID,APL_ID) in this document. Any multicast packet will carry this complete HLIM group address and the routers will forward the packets only within the specified scope region. In IPv4, the destination address field in the IP header is not long enough to carry all the HLIM address information. The SB(L,H) and APL_ID can be placed as the 28-bit group ID in an IP class D address. For example, to support a hierarchical network of 4 levels, only 4 bits are required for the scope boundary and there are still 24 bits remaining for the APL_ID. The HLIM scope will work independently of the TTL (Time To Live) field in the IP header. The RID can be the IP address of the root HDR (i.e., the HDR right above the highest level of the scope region) or some other ID assigned. If some other IDs are used, they need to be unique among the routers at the same level and require more administration and management efforts. Currently, we consider to use the IP address as the RID and define a new IP option to carry the RID in the IP header. In IPv6, we can put the complete HLIM group address into the IPv6 destination address field using the unicast address prefix as the RID.

Note that the RID is needed because HLIM allows the same APL_ID and SB(L,H) to be reused at different places in the hierarchical IP network (so that the same type of applications/services with the same scope can be identified by the same number no matter where the applications/services are delivered) and supports mobile hosts. As long as a host remains in the scope region where a multicast session is started, the packets to/from the host will not interfere with the packets bearing the same SB(L,H) and APL_ID information at another place. However, when the host moves outside that scope region, the packets to/from the host will interfere with the packets carrying the same SB(L,H) and APL_ID information at the new place if the RID is not included.

4. HLIM BASELINE FUNCTIONS

New IGMP messages, such as IGMP_RID_Request, IGMP_RID_Reply, IGMP_M_Report, IGMP_M_Reply, IGMP_Hello and IGMP_Flush, are added into the current IGMP protocol [11] to support the new features in HLIM. Other existing IGMP messages are modified to carry the new multicast group address defined in HLIM and handle both multicast sources and receivers. The baseline functions in HLIM for "fixed" hosts and routers are described in this section. These are the operations for a mobile host to start a new multicast session. The HLIM and IGMP operations to handle host and router mobility (i.e., handover of on-going multicast sessions) and network survivability are described later. To facilitate the explanation of the HLIM operations, we assume a HDR is elected to be both a local and global HDR.

4.1 LOCAL HOST JOINING A GROUP

When a multicast application is started at a receiver host at level k, the host will obtain the scope boundary (denoted as SB(L,H)) and application identity (denoted as APL_ID) information of the multicast session being joined from the application. The application may obtain such information through some advertising mechanisms such as SAP (session announcement protocol), SDP (session description protocol), or web. Once the host obtains the SB(L,H) and APL_ID for the multicast session, it will join the multicast session by sending an IGMP_RID_Request to its parent HDR. After the HDR completes the HLIM regular receiver join process, it sends an IGMP_RID_Reply with the RID of the scope region back to the host. When a multicast application is started at a source host, the same procedure is followed except that the host indicates it is a source instead of a receiver and the HDR invokes the HLIM source join operation instead of the receiver join operation. Similar to the existing IGMP, a HDR periodically issues an IGMP_Query message to its local subnet, and the local hosts (sources and receivers) respond to the query by sending IGMP_Report. Note that the operations for establishing a multicast tree is now initiated by IGMP_RID_Request/IGMP_RID_Reply, instead of IGMP_Report. IGMP_Report is now used to maintain but not create a multicast session, and is applied to both sources and receivers. The detailed procedure is described below:

1. When a host at level k wants to start a multicast session, it first asks its parent HDR (HDR(k)) for the RID for the session by sending a IGMP_RID_Request[SB(L,H), APL_ID, R] if it is a receiver, or IGMP_RID_Request[SB(L,H), APL_ID, S] if it is a source, to its local subnet.
2. Upon receiving a IGMP_RID_Request(R/S), the HDR(k) determines whether the requesting host is located at the right scope region or not.
3. If $L \leq k \leq H$ (i.e., the host is within the scope requested), the HDR(k)
    - finds the RID from its HDR list,
    - invokes the regular receiver or source join operation (HLIM_Join(R/S)), and then
    - replies to the host with an IGMP_RID_Reply[GA, R/S], where GA is the HLIM group address and comprised of SB(L,H), APL_ID, and RID.

Note that the regular join operation is initiated by IGMP_RID_Request(R/S), not by IGMP_Report as in the existing IGMP paradigm. In HLIM, IGMP_Report is used only for maintaining a local group membership.

4. Otherwise (i.e., the host is outside the scope requested), the HDR(k) ignores the IGMP_RID_Request(R/S).
5. Once the host gets an IGMP_RID_Reply(R/S), it records the GA in its receiver or source GA list. When it receives an IGMP_Query later, it issues IGMP_Report(R/S) with the GA in response to the IGMP_Query.
6. If the host is a source, it sets up the host forwarding cache for the GA after receiving the IGMP_RID_Reply(S). This cache is used to insert the RID as an IP option in the IP header of outgoing multicast packets and is transparent to the applications.

4.2 ESTABLISHING THE HLIM MULTICAST TREE

Once a HDR receives an IGMP_RID_Request from a multicast receiver or source (distinguished by a flag inside the message) and the host is within the scope requested, it invokes the HLIM_Join(R) operation for receiver and HLIM_Join(S) operation for source to establish the multicast tree. Note that the HLIM control messages are multicast to a well-known "all-HDR" multicast address with IP TTL of 1. The detailed procedure is described below:

1. When the HDR(k) receives the IGMP_RID_Request(R/S) and the host is within the scope requested, it initiates the regular join operation by sending HLIM_Join(R) for a receiver, or HLIM_Join(S) for a source, to its parent HDR (HDR(k+1)) via its U_IF.
2. When the HDR(k+1) receives a HLIM_Join(R/S), it checks the followings:
   - If it is outside the scope region (k > H or k < L or RID ≠ HDR(H+1)) or the message is received from its U_IF (i.e., from a peer, not child, HDR), it ignores the message.
   - Otherwise, if it is an on-tree HDR (i.e., with an existing receiver/source forwarding cache for the group), it issues a HLIM_ACK(R/S) back to its child HDR(k) via its D_IF.
   - Otherwise, if it is located at the highest level of the scope region (i.e., k=H), it sets up a "confirmed" forwarding cache for the group and sends a HLIM_ACK(R/S) back to its child HDR(k) via its D_IF.
   - Otherwise, it sets up a "transient" forwarding cache for the group and relays the HLIM_Join(R/S) to its parent HDR (HDR(k+2)) via its U_IF.
3. The HDR(k+2) repeats the last step. Eventually, the HLIM_Join(R/S) message reaches an on-tree HDR or the highest HDR for the GA, and then a HLIM_ACK(R/S) is returned.
4. When a HDR receives a HLIM_ACK(R/S) massage, it will check the followings:
   - If it has a "transient" forwarding cache for the GA specified in the HLIM_ACK(R/S) message and receives the message from the U_IF, it switches the "transient" forwarding cache to the "confirmed" forwarding cache and becomes an on-tree HDR for the group. It then forwards the message to the D_IF.
   - Otherwise, the message is ignored.
5. Eventually, the HDR(k) receives a HLIM_ACK(R/S) message and establishes the "confirmed" forwarding cache for the group. It then sends an IGMP_RID_Reply with the RID back to the host.

Fig. 4-1 shows examples of the HLIM receiver join operation. The IGMP_RID_Request from the invalid receiver is discarded since the receiver is located outside of the scope region. The join message from the *receiver 1* is forwarded up to the highest HDR on the scope region, HDR(4), and this HDR issues a HLIM_ACK down to the HDR to which the receiver is attaching. On the other hand, the join message from the *receiver 2* traverses only up to the HDR(3) which is an on-tree HDR and a HLIM_ACK is returned by this on-tree HDR. The operations are the same for a source except that the messages are marked "S" instead of "R".

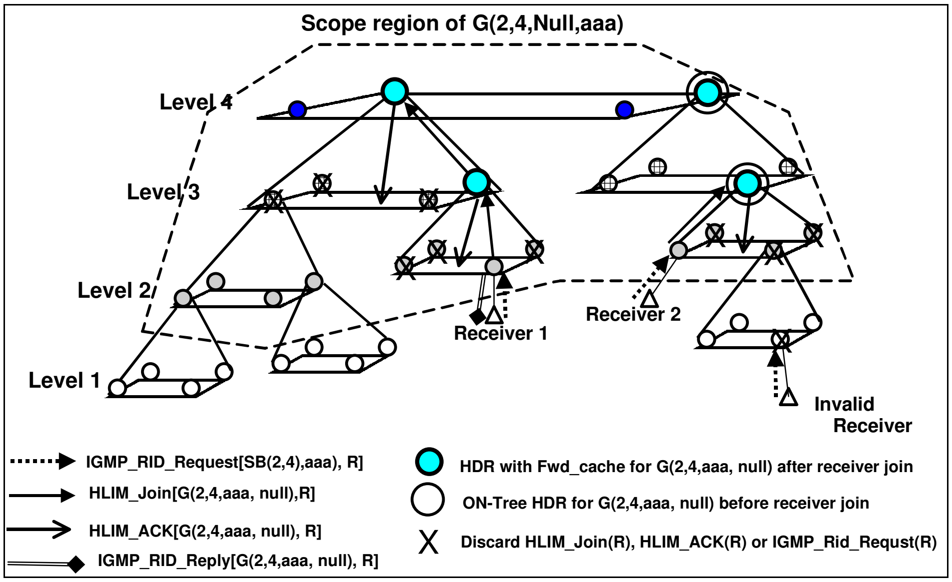

Fig. 4-1 HLIM operations for a receiver joining a group

4.3 FORWARDING MULTICAST PACKETS

When a HDR receives a multicast packet (denoted as MP[G(L,H,APL_ID,RID), S]), it simply forwards the packet to the outgoing interfaces indicated in the "confirmed" forwarding cache for the addressed group (if any), except the incoming interface of the packet. If the cache does not exist, the packet is discarded.

5. HLIM MOBILITY FUNCTIONS

The basic concept to support mobility in HLIM is to figure out whether a mobile node (host or network) has moved within or outside the original scope region of an on-going multicast session. If the mobile node has moved within the scope region, the baseline operations described in the previous section are applied. If the mobile node has moved outside the scope region, the mobility operations are invoked. The main idea for the mobility operations is to extend the established multicast tree for a mobile host or network through a binding point (BPT). A BPT is the HDR that provides linkage between the original scope region and the new location of a mobile node through the shortest path. It is located at the bottom of the original scope region for the mobile node which has moved beneath the scope region. It is the root of the original scope region if the mobile node has moved outside (but not below) the scope region. Through the BPT, a mobile node outside its original scope region can continue to receive or send multicast packets from the sources or to the receivers in the same multicast session. It should be noted that all the multicast packets are still delivered to/from the new location of the mobile node through the shortest path. Two examples of BPTs are shown Fig. 5-1.

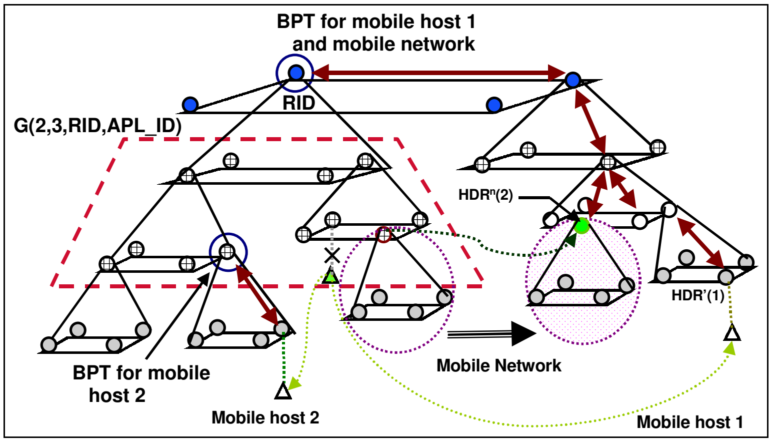

Fig. 5-1 Example BPTs for mobile nodes

Each HDR keeps a list of all its ascendant HDRs (referred as HDR list) and updates its HDR list whenever any of its ascendant HDRs is changed. If a HDR moves to a new location, it will get a new HDR list. When the movement of a mobile host (or network) engaged in a multicast session is detected, the new parent HDR of the mobile host or the root of the mobile network will determine the movement type (i.e., inside or outside the scope region) with respect to its on-going multicast sessions. For example, the determination of movement type for the *mobile host 1* and the *mobile network* shown in Fig. 5-1 takes place at HDR'(1) and HDR^n(2), respectively. Similar to Mobile IP, mobility detection mechanism is built into HLIM. This is to eliminate the dependence of HLIM on other protocols (e.g., the lower layers, or Mobile IP). However, if other mobility detection mechanism is available, HLIM could interact with the other scheme (e.g., the radio layer) instead to provide a more efficient mobility detection.

5.1 FORWARDING CACHES IN HDR

In order to forward multicast packets and support mobility, each HDR maintains R_MFC (multicast forwarding cache for receivers) and S_MFC (multicast forwarding cache for sources) lists. The followings are possible entries in the forwarding cache lists.

- In the R_MFC list:
  For R_MFC located within the scope region of the multicast group,
  {GA, R, Out_IF[L_IF]},
  {GA, R, Out_IF[D_IF]},
  {GA, R, Out_IF[L_IF;D_IF]}
  For R_MFC located outside the scope region of the multicast group,
  {GA, MR, Out_IF[L_IF]},
  {GA, MR, Out_IF[D_IF]},
  {GA, MR, Out_IF[L_IF;D_IF]},
  {GA, MR, Out_IF[U_IF]},
  {GA, MR, Out_IF[L_IF;U_IF]}.
- In the S_MFC list:
  For S_MFC located within the scope region of the multicast group,
  {GA, S, Out_IF[D_IF]},
  For S_MFC located outside the scope region of the multicast group,
  {GA, MS, Out_IF[D_IF]},
  {GA, MS, Out_IF[U_IF]}.

GA is a HLIM group address which contains SB(L,H), APL_ID, and RID. R/S/MR/MS indicates the type of the cache: R for receiver-created cache located within the scope region of the multicast group, S for source-created cache located within the scope region of the multicast group, MR for receiver-created cache located outside the scope region of the multicast group, and MS for source-created cache located outside the scope region of the multicast group. In software implementation, a flag will be used in the cache to differentiate between source and receiver. Whether a cache is located inside or outside a scope region will be determined by a HDR by comparing the RID in its cache and the HDR(H+1)'s ID in its HDR list when the cache is processed by an operation, rather than using a flag. If the two IDs are the same and k is between L and H, the cache is inside the scope region; otherwise, it is outside the scope region. In the rest of the document, we will also use these notations (R/S/MR/MS) to refer to receivers or sources inside or outside the scope region. Out_IF[L_IF/D_IF/U_IF] indicates the outgoing interface(s) that a multicast packet of the GA should be forwarded to.

5.2 HLIM HOST MOBILITY

Hosts in the hierarchical IP network are allowed to move to anywhere in the network. Note that the "original" scope region refers to the scope region in which a multicast group is started by a mobile host through the HLIM baseline operations. New messages added into the current IGMP protocol to handle host mobility explicitly are IGMP_M_Report, IGMP_M_Reply and IGMP_Hello.

5.2.1 Host Mobility Detection

The IGMP_Hello message is used to detect host mobility. A HDR periodically sends IGMP_Hello to its local subnet at the rate of 1 message per 1 to 5 seconds (the rate can be adaptive to the degree of mobility). Note that IGMP_Hello and IGMP_Query are independent. The transmission rate of the former message is higher and its size is smaller for efficiency. The IGMP_Hello message informs local hosts of the change of their parent HDRs. If a host does not receive the IGMP_Hello from its current parent HDR for a certain period (one or more transmission periods of the IGMP_Hello) but receives an IGMP_Hello from another HDR, it considers itself attached to the other HDR. This could happen when a host moves and affiliates to a new HDR at a new location, or when a new HDR is elected.

Once a host detects its mobility through the IGMP_Hello message, it issues an IGMP_M_Report(R/S) message (R for receiver and S for source) to its new parent HDR for every multicast session it has.

5.2.2 IGMP Operations for a Mobile Host

When a mobile host engaged in a multicast session detects its movement to a new location at level k, the following procedure will be invoked:

1. The mobile host issues an IGMP_M_Report[GA, R/S] message to its new parent HDR(k) at the new location, where GA is G[L,H,RID,APL_ID].
2. Upon receiving the IGMP_M_Report, the HDR(k) finds out whether the mobile host moved within or outside the original scope region by the following comparisons:
   - $H = N$ (where N is the number of levels in the hierarchy) and $k \geq L$
   - $H < N$ and $L \leq k \leq H$ and RID = HDR(H+1) in the HDR list

   If one of these conditions satisfies, the HDR(k) determines that the mobile host has moved within the original scope region. Otherwise, the mobile host has moved out of the original scope region.
3. If the HDR(k) has already a membership record for the requested GA, it returns an IGMP_M_Reply[GA, R/S] to the host. Otherwise, it invokes the HLIM regular join operation by sending a HLIM_Join[GA, R/S] to its U_IF if the host moved within the original scope region, or

starts the mobile join operation by sending a HLIM_M_Join[GA, R/S] toward the RID. After completing the join procedure and tree adjustment, the HDR(k) sends an IGMP_M_Reply[GA, R/S] back to the mobile host.

5.2.3 Adjusting the HLIM Tree for a Mobile Host

The new HDR to which a source or receiver has moved invokes the mobile join operation by sending a HLIM_M_Join(R/S) toward the RID of the affected multicast session. On the way toward the RID, if the message reaches an on-tree HDR outside the scope region or the BPT, a HLIM_M_ACK(R/S) message is returned and the multicast tree will be extended along the path. The procedure is described below:

1. When the new HDR receives and accepts an IGMP_M_Report(R/S) from its local subnet and has no existing member for the requested GA, it adds the GA into its membership database. It then looks up the unicast routing table to find out the outgoing interface toward the RID. If an IGMP_M_Report(R) is received, the HDR sets up a "transient" forwarding cache with L_IF set in the Out_IF. If an IGMP_M_Report(S) is received, the HDR sets up a "transient" forwarding cache with the outgoing interface toward the RID set in the Out_IF. The HDR then sends a HLIM_M_Join(R/S) message toward the RID. Note that the HLIM_M_Join(R/S) message is sent in multicast so that all HDRs in the subnet attached to the outgoing interface will receive it.
2. When a HDR receives a HLIM_M_Join(R/S) message, it determines if it is the BPT for the mobility. If its own address = RID or the address of its ascendant HDR(H+1) = RID and its level = L, it is the BPT and then go to step 4. Otherwise, proceed to step 3.
3. If a HDR is not the BPT, it first determines the outgoing interface toward the RID from the unicast routing table. If the outgoing interface is the same as the incoming interface of the HLIM_M_Join(R/S) message, the HDR discards the message. Otherwise, it processes the message as follows:
   - If the HDR is on-tree for the group (i.e., it already has the receiver/source forwarding cache that the received HLIM_M_Join(R/S) tries to set up), it stops forwarding the HLIM_M_Join(R/S) and replies with a HLIM_M_ACK(R/S) message to the incoming interface of the HLIM_M_Join.
   - Otherwise, the HDR sets up a "transient" forwarding cache for the group with the incoming interface of the HLIM_M_Join (for receiver) or the outgoing interface toward the RID (for source) set in the Out_IF. It then forwards the HLIM_M_Join(R/S) to the outgoing interface toward the RID.
4. When the BPT receives the HLIM_M_Join(R/S) message:
   - If the level of the BPT = L (i.e., at the bottom of the scope region) and the message is received from the D_IF, it returns a HLIM_M_ACK(R/S) to the D_IF. If it is not yet an on-tree HDR for the group, it sets up a "transient" forwarding cache with the D_IF set in the Out_IF and issues a regular HLIM_Join(R/S) message to its U_IF, which is then processed as in the regular case.
   - If the BPT = RID and the message is received from the U_IF, it returns a HLIM_M_ACK(R/S) to the U_IF. If the required receiver/source forwarding cache does not exist, it sets up a "confirmed" forwarding cache for the group with the U_IF (for receiver) or D_IF (for source) set in the Out_IF.
   - Otherwise, the message is ignored.
5. When a HDR receives a HLIM_M_ACK(R/S) for a group, if it has the corresponding "transient" forwarding cache and the message is received from the outgoing interface toward the RID, it switches the cache to a "confirmed" cache and forwards the HLIM_M_ACK(R/S) to the Out_IF in the cache for receiver, or to the interface opposite to the Out_IF in the cache for source. Otherwise, the message is ignored.

6. Eventually, the new HDR receives a HLIM_M_ACK(R/S) from the outgoing interface toward the RID. It then switches the "transient" forwarding cache for the group into the "confirmed" state and returns a IGMP_M_Reply to the mobile host.

Some example operations for host mobility are shown in Fig. 5-2. Since *mh1* has moved within the scope region, only the regular join operations are invoked. Since *mh2* and *mh3* have moved out of the scope region, the mobile join operations are performed between the new HDR serving *mh2* and the BPT_L, and between the new HDR serving *mh3* and the BPT_H. The prune operation is not shown in the figure.

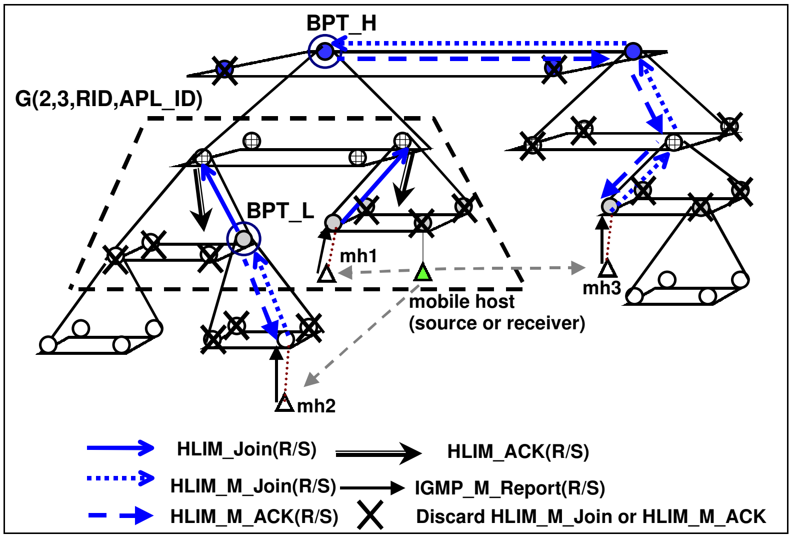

Fig. 5-2 Join operations for a mobile host after movement

Fig. 5-3 depicts how multicast packets are forwarded from mobile sources and to mobile receivers after their movements.

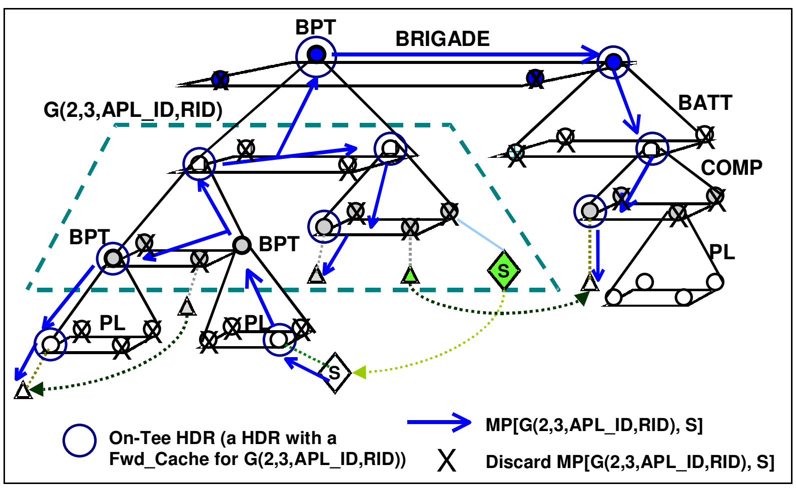

Fig. 5-3 Example multicast forwarding for mobile sources and receivers

After the source or receiver moves away, the previous parent HDR at the old location will invoke the prune operation if no other sources or receivers of the affected multicast sessions exist in its local subnet. The prune operation eliminates unnecessary branches of the multicast trees. The details of the prune operation will be described later.

5.3 HLIM NETWORK MOBILITY

We assume that a network movement in the hierarchical IP network is allowed only within the same level. A router at level k can only move to the same level at another branch of the hierarchy. It can move with all of its descendant networks (including its local subnet), its local subnet only, or alone. Our design goal is to hide any network mobility from the hosts or routers within the moving network. In other words, no operations are required from the hosts or routers inside the moving network. Only the "topmost" router interacts with the outside of the moving network to hand over any on-going multicast sessions to the new location. Only the movement of an on-tree HDR (i.e., the HDR with one or more forwarding caches established by HLIM_Join or HLIM_M_Join) will invoke mobility operations to hand over all the on-going multicast sessions through that router.

5.3.1 Network Mobility Detection

A child HDR detects its own mobility and a new parent HDR through periodic HLIM_Parent_Hello messages. A parent HDR periodically multicasts the HLIM_Parent_Hello message to its child HDRs. If a child HDR has not received the HLIM_Parent_Hello from its current parent for a certain number of the message period and receives the message from a new parent HDR, it considers itself moved to the new parent. It then exchanges HLIM_HDR_List_Solict and HLIM_HDR_List_Update with the new parent HDR to obtain the HDR list at the new location. If it has ongoing multicast sessions, it invokes the appropriate network mobility operations to extend the multicast trees.

When a child HDR attaches to a parent HDR, the parent keeps a record of the new child. When a parent HDR receives a HLIM_HDR_List_Solicit or HLIM_Child_Hello (whichever comes first) from a new child HDR, it adds the new child's address into its children list. Each child HDR periodically multicasts a HLIM_Child_Hello message to its parent HDR and a parent HDR detects a child moved away through the periodic HLIM_Child_Hello messages. If a parent HDR has not received the HLIM_Child_Hello from a child HDR for a certain number of the message period, it considers the child has moved to another location and invokes the appropriate network mobility operations to prune the unnecessary branches of the multicast trees.

5.3.2 Network Mobility Operations

When a network moves to a new location, the root HDR updates its HDR list at the new location through the HLIM_HDR_List_Solicit and HLIM_HDR_List_Update operations. It then forwards the updated HDR list to its descendent HDRs through the HLIM_HDR_List_Update operation. It also determines the movement type of each on-going multicast session it has by the comparison described for the host mobility operation. Once the movement types are determined, the root HDR invokes appropriate operations for every entry in its R_MFC and S_MFC lists.

- For entries with (R:D_IF), (S:U_IF), (MR:D_IF), or (MS:U_IF) for a GA, the regular or mobile join operation should be invoked because a regular/mobile source/receiver for the GA is located inside the moving network, thus the multicast tree of the GA needs to be extended to/from the moving network. The following join operations are invoked:
  - HLIM_Join(R) for (R:D_IF)
  - HLIM_Join(S) for (S:U_IF)
  - HLIM_M_Join(R) for (MR:D_IF)
  - HLIM_M_Join(S) for (MS:U_IF)

  All the join messages mentioned above are sent toward the RID of the GA and are processed as in the baseline and host mobility cases.

- For entries with (MR:U_IF) or (MS:D_IF) for a GA, which imply that the moving network contains the RID of the GA, HLIM_Net_Join(R) or HLIM_Net_Join(S) operation is invoked. These network join messages are sent toward the old parent HDR at the old location. The operations are described later.

At the old location, once the movement of a child HDR is detected, the old parent HDR initiates the HLIM_Compare(M) operation to its D_IF (i.e., into its lower core subnet) for the group addresses whose RIDs are not located inside the moving network. Through the operation, the old parent HDR finds out the unmatched forwarding caches that are no longer needed. It then removes the caches and invokes the appropriate prune operation to its U_IF:

- HLIM_Prune(R) for (R:D_IF)
- HLIM_Prune(S) for (S:U_IF)
- HLIM_M_Prune(R) for (MR:D_IF)
- HLIM_M_Prune(S) for (MS:U_IF)

All the prune operations mentioned above are sent toward the RID. Note that the unwanted caches of the group addresses whose RIDs are located inside the moving network are removed by the HLIM_Net_Join(R/S) operation instead.

5.4 HDR List Update Operations

As mentioned earlier, after a HDR moves to a new location, it will obtain a new HDR list from its new parent HDR through the HLIM_HDR_List_Solicit and HLIM_HDR_List_Update operations. It will then update its own HDR list and forward the list to its descendant HDRs through the HLIM_HDR_List_Update operation. The procedure is described below:

1. After a HDR moves to a new parent HDR, it sends a HLIM_HDR_List_Solicit message to the new parent HDR through its U_IF and then waits for a HLIM_HDR_List_Update message from the parent HDR.
2. When the parent HDR receives the HLIM_HDR_List_Solicit message from its D_IF, it checks if the message is sent from a new child HDR, and if so, adds the new child's address into its children list. It then sends back a HLIM_HDR_List_Update message including its HDR list and its own address.
3. When a HDR receives a HLIM_HDR_List_Update message from its parent HDR, it will update its own HDR list. If it has child HDRs attached to its D_IF, it sends a HLIM_HDR_List_Update including its HDR list and its own address to the child HDRs.

5.5 HLIM Compare Operations for Mobility - HLIM_COMPARE(M)

The HLIM compare for mobility operations are used by the parent and child HDRs at the old location of a moving network to remove any unwanted branches of the multicast trees affected by the movement.

1. When the parent HDR at the old location detects the movement of one of its child HDRs, it sends a HLIM_Compare(M) message to its child HDRs through its D_IF (i.e., toward its lower core subnet). It sets a Compare_Repeat_Timer during which the HLIM_Compare_Reply(M) messages from its child HDRs will be collected. A HLIM_Compare(M) message contains the inbound GAs for sources and outbound GAs for receivers at the HDR. "Inbound GAs" here refer to the group addresses of the forwarding caches with D_IF as an Out_IF (packets of the groups are sent "into" the core subnet through the D_IF). Similarly, "outbound GAs" refer to the group addresses of the forwarding caches with L_IF or U_IF as an Out_IF (packets of the groups are sent "out" of the core subnet through the L_IF and U_IF).

2. When a child HDR receives a HLIM_Compare(M) from its parent HDR from its U_IF, it sets two timers: a Compare_RandTimer which is a random timer (its maximum value should be less than the Compare_Repeat_Timer mentioned above) to wait before sending a HLIM_Compare_Reply(M), and a Compare_Timer (its value should be longer than the maximum time for the parent HDR to finish all the retransmissions of the HLIM_Compare(M)) during which the HLIM_Compare_Reply(M) messages from its neighboring HDRs will be collected. A HLIM_Compare_Reply(M) is *multicast* and contains the inbound GAs for sources and outbound GAs for receivers. "Inbound GAs" here refer to the group addresses of the forwarding caches with U_IF as an Out_IF (packets of the group addresses are sent "into" the core subnet through the U_IF). Similarly, "outbound GAs" refer to the group addresses of the forwarding caches with L_IF or D_IF as an Out_IF (packets of the group addresses are sent "out" of the core subnet through the L_IF and D_IF).
3. When the Compare_Repeat_Timer expires, the parent HDR checks if it has received the reply from all of its child HDRs. If not, it retransmits the HLIM_Compare(M) message and starts the Compare_Repeat_Timer again. This operation is repeated until the maximum number of trials. The repeated transmissions are to increase the reliability of this operation.
4. When a child HDR has its Compare_Timer expire or the parent HDR receives replies from all of its children or finishes the maximum number of HLIM_Compare(M) transmissions, the HDR will perform the following comparison:
   - The HDR compares its own outbound GAs for sources and the inbound GAs for sources in all the HLIM_Compare_Reply(M) messages received. It deletes the caches of its outbound GAs with no matching inbound GAs and then invokes an appropriate prune operation to its U_IF.
   - The HDR compares its own inbound GAs for receivers and the outbound GAs for receivers in all the HLIM_Compare_Reply(M) messages received. It deletes the caches of its inbound GAs with no matching outbound GAs and then invokes an appropriate prune operation to its U_IF.

5.6 HLIM PRUNE OPERATIONS

The following events result in an invocation of a prune operation at a HDR:
- when a local source moves away or leaves a multicast group,
- when the R_MFC for a GA is deleted (e.g., after a multicast receiver moves away or leaves the multicast group),
- when an unmatched GA is found by the HLIM_Compare(M) operation (e.g., after a network movement).

5.6.1 HLIM_Prune(S) and HLIM_M_Prune(S)

1. When a GA is registered in the source group membership list in a HDR, a prune timer is set for the GA. The timer is refreshed by IGMP_Report, which is sent in response to the periodic IGMP_Query.
2. If the prune timer expires, the HDR (e.g., HDR1 in Fig. 5-4 and Fig. 5-5) deletes the GA from the membership list and is ready to prune the associating outgoing interface (D_IF or U_IF) for the GA listed in the S_MFC.
3. It sets a prune timer (Prune_Expire_Timer) for the outgoing interface and issues a number (≥ 2) of source prune messages (HLIM_Prune(S) or HLIM_M_Prune(S)) to the interface opposite to the listed outgoing interface (one after another separated by a certain time interval). For instance, the message is sent to D_IF if U_IF is listed in S_MFC. The repeated transmissions of the source prune message are to increase the reliability of the message especially in a harsh environment.
4. When the prune timer expires, if no prune reply (HLIM_Prune_Reply(S) or HLIM_M_Prune_Reply(S)) arrives from the interface to which the source prune message has been sent, the S_MFC is deleted and a source prune message is issued through the outgoing interface of the cache just deleted (i.e., toward the RID of the GA). Otherwise, it keeps the S_MFC.

5. When a HDR receives a source prune message, it checks whether the GA is listed in its S_MFC list or not. If the GA is not listed, it (e.g., HDR4 in Fig. 5-5) will discard the prune message. If the GA is listed, it will compare the Out_IF of the GA listed in the S_MFC with the incoming interface of the prune message received.
   - If the listed Out_IF is the same as the incoming interface, the HDR (e.g., HDR4 in Fig. 5-4, HDR2, HDR5 in Fig. 5-5) sets a prune reply random timer and then returns a prune reply (HLIM_Prune_Reply(S) or HLIM_M_Prune_Reply(S)) when the random timer expires. If it receives a prune reply message while the random timer is running, it stops the timer.
   - If the listed Out_IF is not the same as the incoming interface, the HDR (e.g., HDR3 in Fig. 5-5) checks if there is a member of the GA in its source membership list.
     - If yes, it ignores the source prune message received.
     - Otherwise, it sets a prune timer on the S_MFC for the GA and waits for a prune reply. If it doesn't receive a prune reply before the prune timer expires, it deletes the S_MFC and sends a number of source prune messages through the outgoing interface of the deleted cache (i.e., toward the RID of the GA). If it receives a prune reply before the prune timer expires, it keeps the S_MFC. If it receives the same source prune message again while the timer is running, it ignores the message.

This operation proceeds to the HDR(H) for HLIM_Prune(S) and the RID for HLIM_M_Prune(S). Note that the total time needed to transmit all the prune messages should be shorter than the prune timer, and the interval between two consecutive prune messages should be longer than the maximum value of the prune reply random timer.

The following two figures show some examples of the source prune operation and the operational details are presented as well. In these examples, the prune messages are assumed to be sent twice.

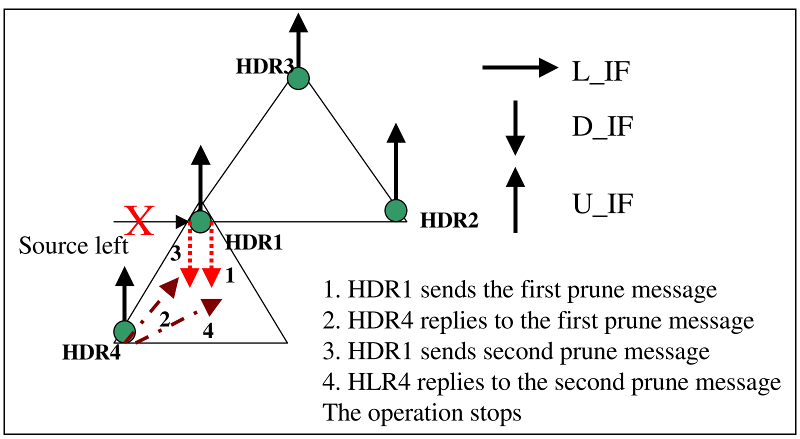

Fig. 5-4 First example of Source Prune Operation

Once HDR1 detects no source for a GA on its local subnet, it sets two timers, a prune timer and a prune tramsission timer, and a prune-reply counter. The prune timer is set to prune the outgoing interface listed in its S_MRF (i.e., U_IF) and the prune transmission timer is set for issuing the second prune message. The prune reply counter counts the number of received prune-reply messages. The HDR sends the first prune message to the D_IF. As the prune transmission timer expires, it sends the second prune message.

When HDR4 receives the first prune message through its U_IF which is the same as that listed in its S_MFC for the GA, it sets a prune random timer first and then sends a prune reply message after the random timer expires. It does the same for the second prune message received.

The counter on HDR1 keeps counting the number of the received prune reply messages until its prune timer expries. When the prune timer expires, HDR1 checks the number indicated in the counter. If the number is greater than 0, HDR1 keeps the S_MFC.

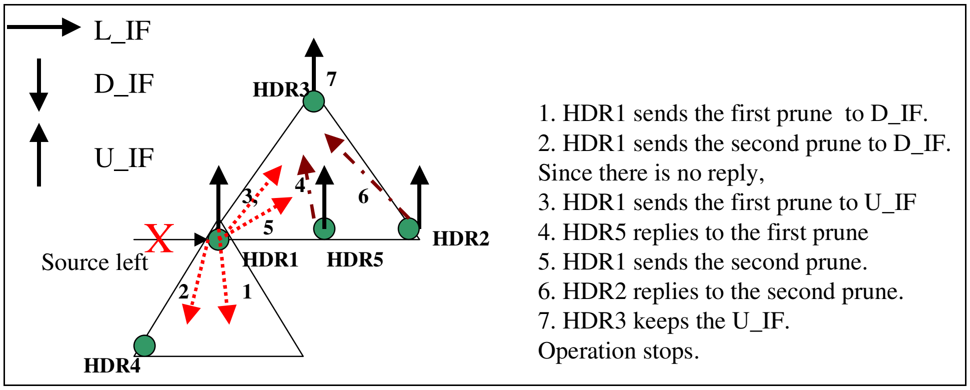

Fig. 5-5 Second example of Source Prune Operation

In the case of Fig. 5-5, HDR1 first sends out the prune messages through its D_IF as described in the previous example. Since HDR1 doesn't receive any prune reply message, the prune reply counter at HDR1 is zero. It then deletes the S_MFC for the GA and sends the first prune message through the outgoing interface of the deleted cache.

Once HDR2 and HDR5 receives a prune message, each of them sets a prune reply random timer since both of them have S_MFC for the GA with the outgoing interface the same as the incoming interface of the prune message received. As one of the timers expires while the other one is still running, a prune reply message is sent out by the HDR with the expired timer (e.g., HDR5). When the other HDR with a running timer (e.g., HDR2) receives the prune reply message, it will suppress its prune reply message by stopping its timer.

When HDR3 receives a prune message from HDR1, it sets a prune timer and prune reply counter, and waits for prune reply messages. If it receives a prune reply message from either HDR2 and HDR5, it will increment the counter. In this case, it will receive two prune reply messages, each one from HDR2 and HDR5, unless the messages are lost due to bad wireless channel condition during the transmission. Since the number on the counter (2) is greater than 0, it keeps the S_MFC for the GA. If it receives the second prune message from HDR1 while its prune timer is running (which has been initiated by the first prune message), it will discard the second prune message.

5.6.2 HLIM_Prune(R) and HLIM_M_Prune(R)

1. When a GA is recorded in the receiver group membership list in a HDR, a prune timer is set for the GA. If the prune timer expires, the HDR deletes the GA from the membership list and the L_IF from the Out_IF list of the R_MFC. The timer is refreshed by IGMP_Report, which is sent in response to the periodic IGMP_Query.
2. When the Out_IF list of the R_MFC for a GA becomes empty, the HDR (e.g., HDR1 in Fig. 5-6) deletes the R_MFC and sends a receiver prune message (HLIM_Prune(R) or HLIM_M_Prune(R)) toward the RID of the GA. The receiver prune message can be sent multiple times (≥ 2) to increase the reliability of the message in a harsh environment.
3. Upon receiving a receiver prune message, a HDR finds a match for the GA in its R_MFC list.

4. If it finds a match, it compares the incoming interface of the prune message with the Out_IFs listed in its R_MFC of the GA.
   - If it has an Out_IF which is the same as the incoming interface, it (e.g., HDR2, HDR4 in Fig. 5-6) sets a prune timer for that Out_IF and a prune reply counter, and waits for a prune reply message (HLIM_Prune_Reply(R) or HLIM_M_Prune_Reply(R)).
     - ➢ As the HDR receives a prune reply message(s) before the prune timer expires, it increments the counter. When the prune timer expires, it checks the counter and it will keep the Out_IF if the number in the counter is greater than 0. If it receives the same receiver prune message again while the timer is running, it ignores the message.
     - ➢ If it doesn't receive any prune reply before the prune timer expires (i.e., the number in the counter is zero), it deletes the Out_IF and checks whether the Out_IF list for the GA is empty or not (since L_IF may still be listed). If the list is empty, it removes the R_MFC and sends a receiver prune message through the interface opposite to the deleted interface (i.e., toward the RID).
   - If the Out_IF list of the R_MFC includes the L_IF or the interface opposite to the incoming interface, it (e.g., HDR3 in Fig. 5-6) sets a prune reply random timer and returns a prune reply message (HLIM_Prune_Reply(R) or HLIM_M_Prune_Reply(R)) if the timer expires. If it receives a prune reply message from the incoming interface of the prune message before the timer expires, it stops the timer.

This operation proceeds to the HDR(H) for HLIM_Prune(R) and the RID for HLIM_M_Prune(R). The following figure shows an example of the receiver prune operation, assuming the prune message is sent twice.

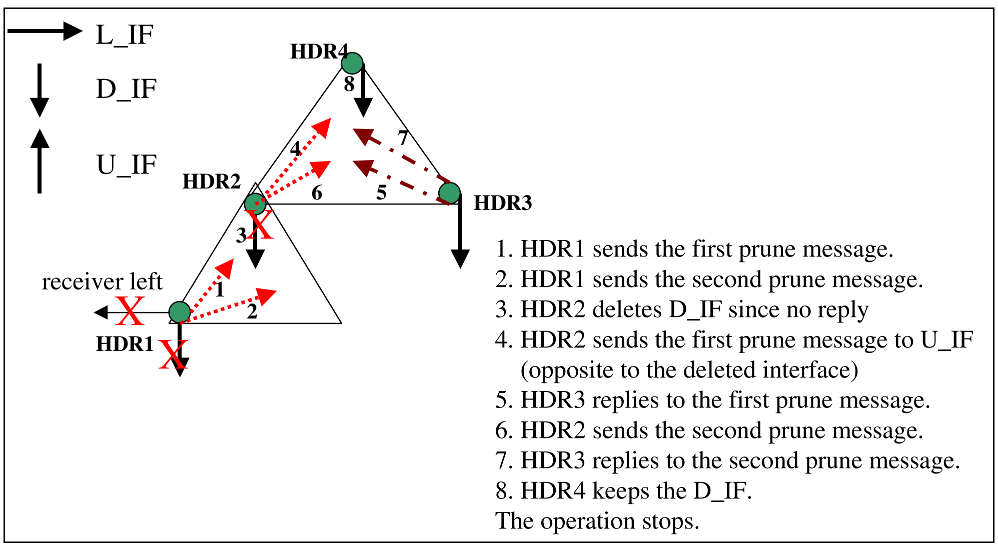

Fig. 5-6 Receiver Prune Operation

Once HDR1 finds that the R_MFC for a GA is empty, it deletes the cache and issues the first receiver prune message toward the RID (i.e., via the U_IF) and sets a prune transmission timer for sending the second prune message.

If HDR2 receives a prune message, it sets a prune timer for the Out_IF (i.e., D_IF) listed in the R_MFC and a prune reply counter for counting incoming prune reply messages. As the prune timer expires, HDR2 checks the number in the counter. In this case, the number is zero since no prune reply message is received. Thus, HDR2 deletes the Out_IF and the R_MFC, and sends the first prune message to its U_IF and sets a prune transmission timer as the HDR1 did.

Upon receiving a prune message, HDR3 sets a prune random timer since the incoming interface of the prune message is opposite to the Out_IF listed in its R_MFC for the GA. It responds a prune reply message when the prune random timer expires. In this case, it responds twice since it receives two prune messages from HDR2.

When HDR4 receives a prune message from HDR2, it sets a prune timer and a prune reply counter since the incoming interface of the prune message is the same as the Out_IF listed in its R_MFC for the GA. Whenever HDR4 receives a prune reply message, it increments the counter. In this case, the number on the counter (2) is greater than 0 when the prune timer expires. Thus, it keeps the Out_IF in its R_MFC.

5.7 NETWORK JOIN OPERATIONS: HLIM_NET_JOIN(R/S)

The network join operation is used to handle a special case of network mobility in which the moving network contains the RID of an on-going multicast session. It is invoked by the root HDR of the moving network when the root HDR has the forwarding cache of (MR:U_IF) or (MS:D_IF). A HLIM_Net_Join(R/S) message is forwarded toward the old parent HDR at the old location and processed by the HDRs on the path to the old parent. The other HDRs (i.e., HDRs not on the path) will discard the message.

1. The root HDR sets the forwarding cache with U_IF for MR or D_IF for MS as outgoing interface to the transient state and sends a HLIM_Net_Join[GA, R/S] toward the old parent HDR.
2. When a HDR on the shortest path between the new location and the old parent HDR receives the HLIM_Net_Join message (i.e., the incoming interface of the message is opposite to the outgoing interface toward the old parent HDR), it checks if it has a R_MFC (if HLIM_Net_Join(R) is received) or S_MFC (if HLIM_Net_Join(S) is received) for the GA.
   - If the corresponding MFC is not found, the HDR will create a transient forwarding cache for the GA and relay the message toward the old parent as in the mobile join operation. The direction of the outgoing interface set in the forwarding cache is opposite to the mobile join operation: the outgoing interface for a HLIM_Net_Join[GA, R] message is listed as an Out_IF in the R_MFC for the GA and the incoming interface for a HLIM_Net_Join[GA, S] is listed as an Out_IF in the S_MFC for the GA.
   - Otherwise, it compares the incoming interface of the message with the Out_IF listed in its MFC. Note that L_IF is excluded in this comparison.
     For a HLIM_Net_Join[GA, R],
     - If the incoming interface is opposite to the listed Out_IF, the HDR ignores the message.
     - Otherwise, the listed Out_IF is switched to a "transient deleted" state and a new transient R_MFC pointing to the opposite interface is set. The HDR then forwards the HLIM_Net_Join toward the old parent HDR.

     For a HLIM_Net_Join[GA, S],
     - If the incoming interface and the listed Out_IF are the same, the HDR ignores the message.
     - Otherwise, the listed Out_IF is switched to a "transient deleted" state and a new transient R_MFC pointing to the opposite interface is set. The HDR then forwards the HLIM_Net_Join toward the old parent HDR.
3. The HLIM_Net_Join operation will proceed to the old parent HDR at the old location. When the old parent receives a HLIM_Net_Join[GA, R/S] from its U_IF, it sets up a confirmed forwarding cache for the GA pointing to the incoming interface (for source) or the interface opposite to the incoming

interface (for receiver), and deletes the old cache. It then returns a HLIM_Net_Join_ACK toward the root HDR of the moving network and invokes the HLIM_Compare(M) operation to its D_IF.

4. When a HDR with a transient MFC for the GA receives a HLIM_Net_Join_ACK from the right interface (i.e., the outgoing interface toward the old parent HDR), it switches the MFC into a confirmed state and deletes the transient deleted cache. It then forwards the HLIM_Net_Join_ACK toward the root HDR of the moving network.

Examples of the HLIM_Net_Join operation are presented below to help explain how the operation works.

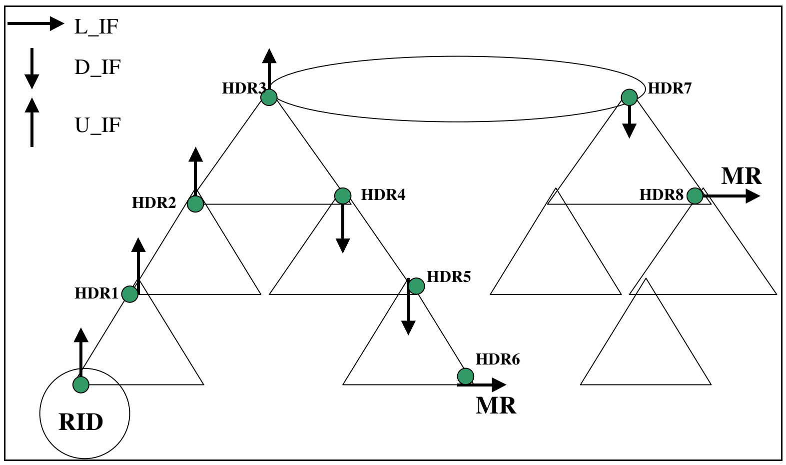

Fig. 5-7 Multicast Tree for HLIM_Net_Join Operation before Network Movement (Example 1)

In the diagram shown above, two receivers have moved out of the scope region and continue to receive multicast packets of the session from the RID. The multicast tree of the session is represented by the arrows of U_IF, D_IF and L_IF.

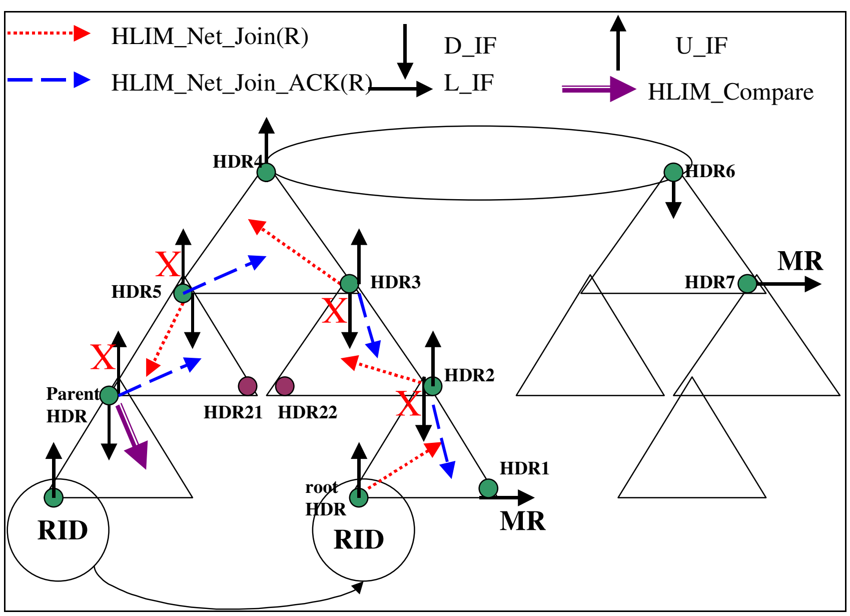

Fig. 5-8 HLIM_Net_Join(R) Operation after Network Movement (Example 1)

In this first example, the network containing the RID of a GA (illustrated as the big circle in the figure) moves from the parent HDR at the old location to the new parent HDR, HDR2, at the new location. In the course of the HLIM_Net_Join(R) operation, each HDR proceeds as follows:

Root HDR at a new location:

- Once it detects its movement to a new location and finds a R_MFC with (MR:U_IF), it issues a HLIM_Net_Join[GA, R] toward the old parent HDR.

HDR2 and HDR3:

- Upon receiving a HLIM_Net_Join[GA, R], since it has the forwarding cache for the GA, it compares the incoming interface of the message with the Out_IF listed in the R_MFC (i.e., D_IF).
- Because they are the same (i.e., both are D_IF), the listed Out_IF (D_IF) is switched to the "transient deleted" state.
- Since it is on the path toward the old parent HDR, it sets up a transient forwarding cache, (MR:U_IF), for GA and relays the HLIM_Net_Join[GA, R] message toward the old parent HDR.
- If it receives a HLIM_Net_Join_ACK , it will switch the forwarding cache with the transient state to the confirmed state and remove the forwarding cache with the "transient deleted" state.

HDR5:

- Since it has the forwarding cache for the GA, it compares the incoming interface of the message with the Out_IF listed in the R_MFC (i.e., U_IF). Because they are the same (i.e., both are U_IF), the listed Out_IF (U_IF) is switched to the "transient deleted" state.
- It sets up a transient forwarding cache, (MR:D_IF), for GA and relays the HLIM_Net_Join[GA, R] message toward the old parent HDR.
- When it receives a HLIM_Net_Join_ACK, it will switch the forwarding cache with the transient state to a confirmed state and remove the forwarding cache with the "transient deleted" state.

Parent HDR:

- Since it has the forwarding cache for the GA, it compares the incoming interface of the message with the Out_IF listed in its R_MFC (i.e., U_IF). Because they are the same (i.e., both are U_IF), the listed Out_IF (U_IF) is deleted.
- It sets up the confirmed Out_IF of the GA opposite to the incoming interface of the message and sends a HLIM_Net_Join_ACK to the incoming interface.
- It invokes the HLIM_Compare(M) operation.

HDR1, HDR4, HDR21 and HDR22 ignore the HLIM_Net_Join/HLIM_Net_Join_ACK messages received.

The diagram below shows the adjusted multicast tree resulted from the HLIM_Net_Join(R) operation. As we can see, shortest path multicasting is preserved after the network movement. The Out_IF (D_IF) listed in HDR5 and Parent HDR will be deleted through the HLIM_Compare(M) and HLIM_M_Prune operations at the end since they are not needed.

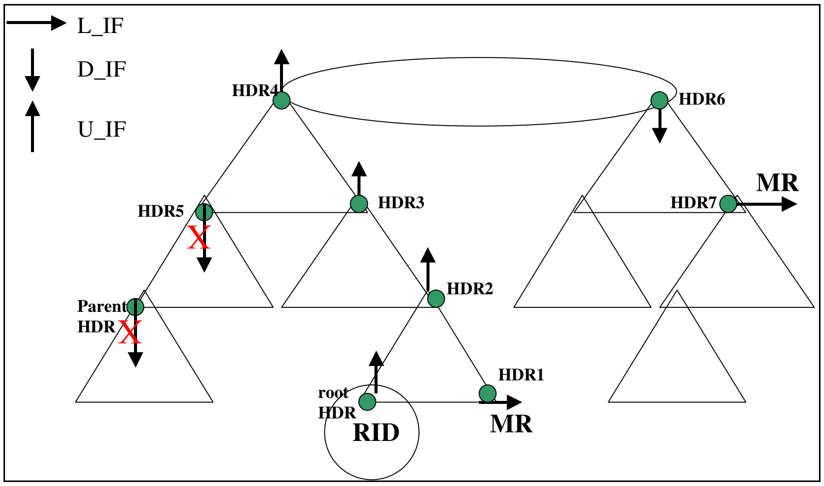

Fig. 5-9 Resultant Multicast Tree after the HLIM_Net_Join(R) Operation (Example 1)

The next two diagrams show another example of the HLIM_Net_Join(R) operation. The same logic and operations described in the previous example are applied.

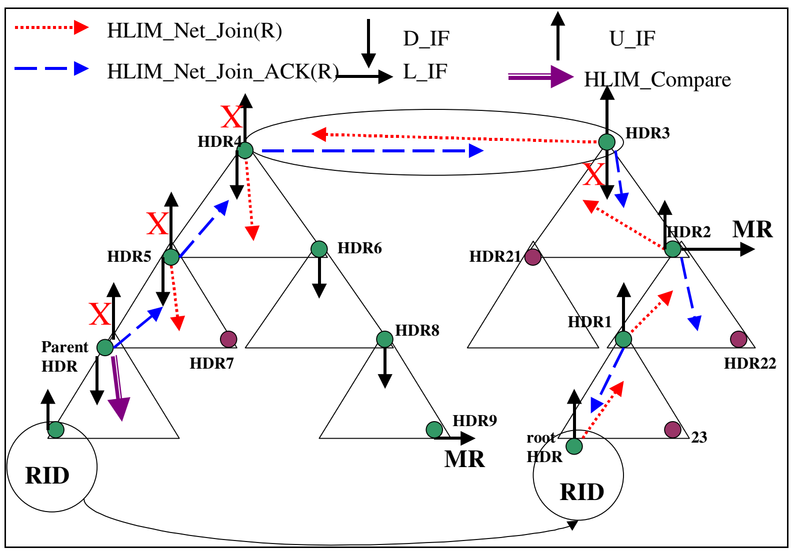

Fig. 5-10 HLIM_Net_Join(R) Operation after Network Movement (Example 2)

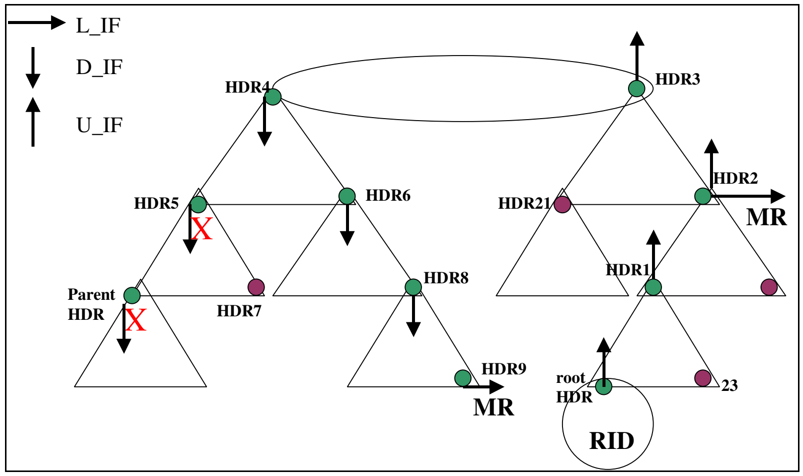

Fig. 5-11 Resultant Multicast Tree after the HLIM_Net_Join(R) Operation (Example 2)

The diagrams below show the examples for the HLIM_Net_Join(S) operation. The HLIM_Net_Join(S) operation is very similar to the HLIM_Net_Join(R), except that the logic of setting up forwarding caches and the comparison between the incoming interface of the HLIM_Net_Join message and the Out_IF listed in the S_MFC is reversed.

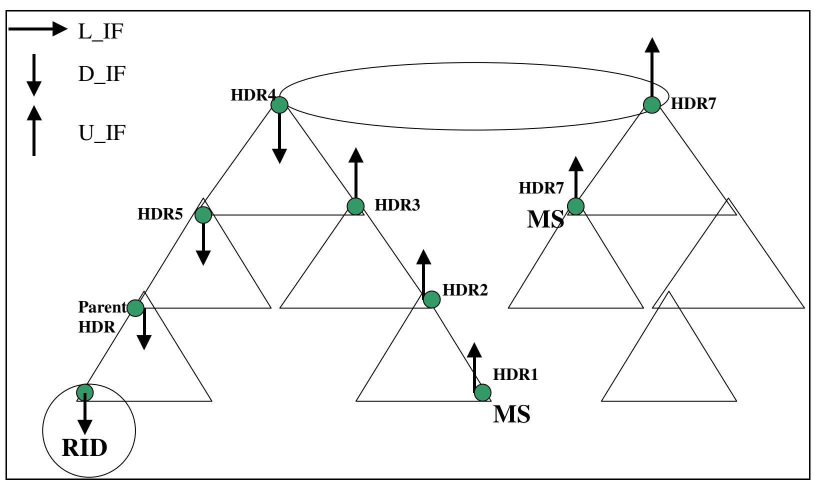

Fig. 5-12 Multicast Tree for HLIM_Net_Join(S) Operation before Network Movement (Example 1)

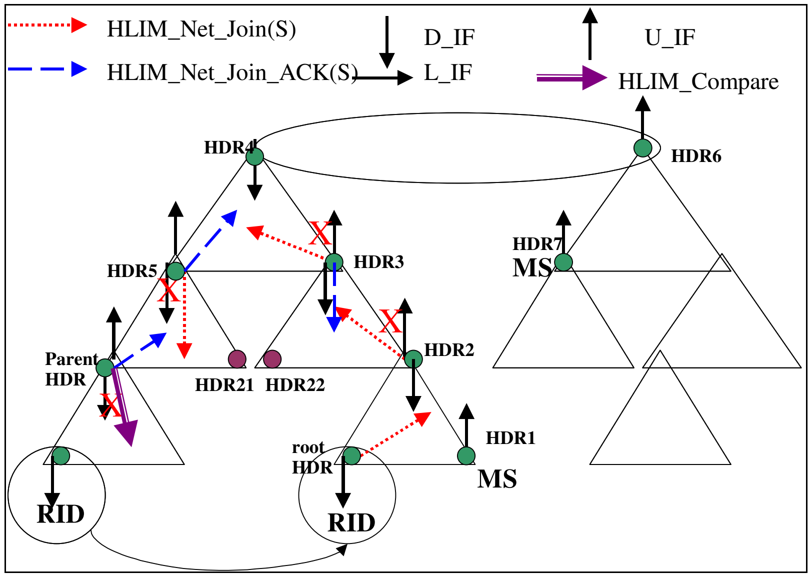

Fig. 5-13 HLIM_Net_Join(S) Operation after Network Movement (Example 1)

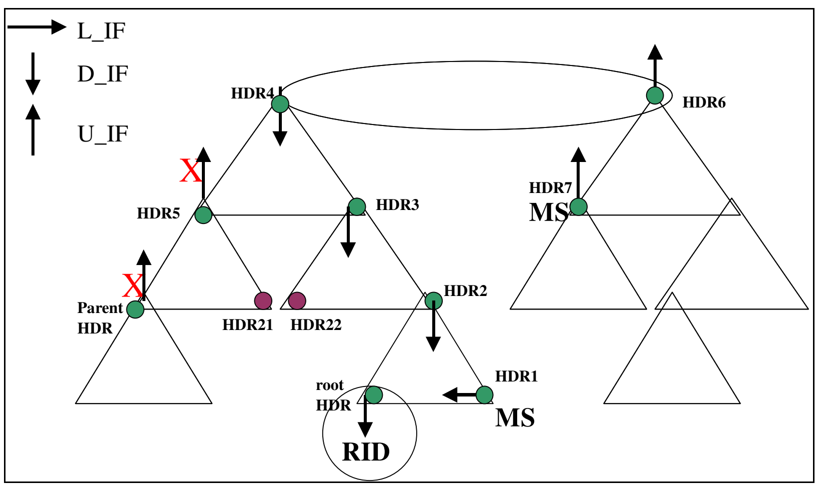

Fig. 5-14 Resultant Multicast Tree after the HLIM_Net_Join(S) Operation (Example 1)

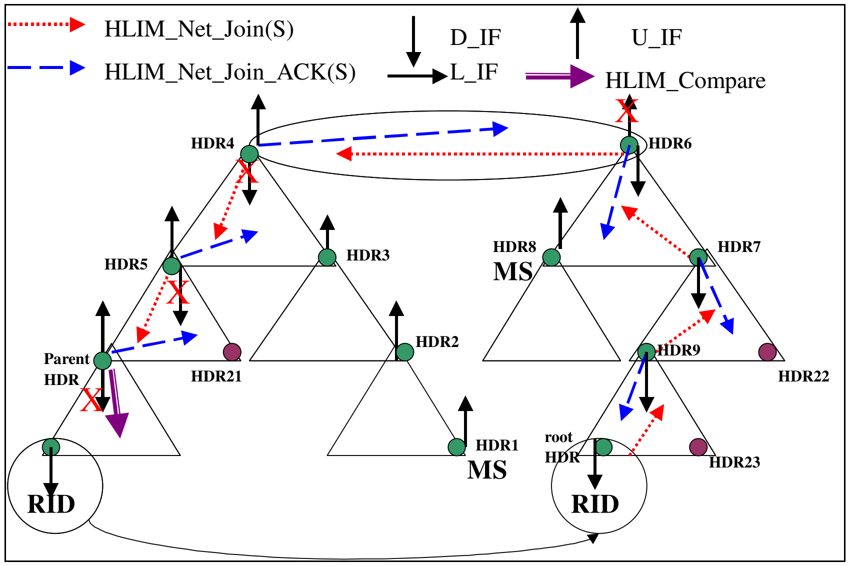

Fig. 5-15 HLIM_Net_Join(S) Operation after Network Movement (Example 2)

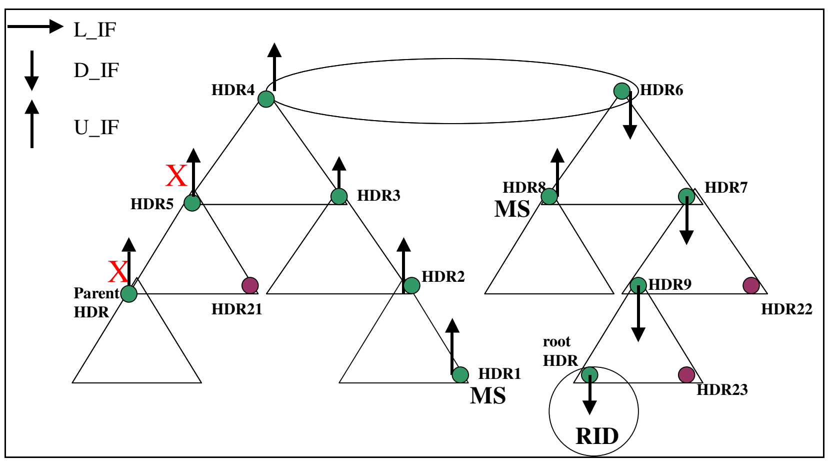

Fig. 5-16 Resultant Multicast Tree after the HLIM_Net_Join(S) Operation (Example 2)

6. HLIM SURVIVABILITY FUNCTIONS

As described earlier, there may be more than one router in a subnet (local or core) for backup purpose. A HDR will be elected to handle all HLIM/IGMP control messages and forward multicast traffic. The HDR election is done through the IGMP_Hello message in a local subnet and the HLIM_Parent_Hello message in a core subnet. When the elected HDR moves away, malfunctions or is destroyed, a new HDR will be elected again among the backup routers and then recover the multicast forwarding caches and membership information in the previous HDR by some local operations.

6.1 HLIM ELECTION PROCESS

Though the current IGMP specification defines a finite state machine for the IGMP_Query that could be used for election, the procedure would switch the status of a router back and forth between "elected" and "not elected" due to message loss, timing difference or new router coming. We therefore define a different finite state machine to make the election process more stable. The process at a HLIM router is described below with reference to the state diagram shown:

1. When a HLIM router is initialized, it sets its timer (T1) to d, where d is a random value between 0 and HoldTime and moves to the idle state.
2. If the HLIM router gets a HLIM_Parent_Hello[pref] through its D_IF before T1 expires and the pref is lower than its own pref (or if the pref is equal to its own pref but the source IP address is lower than its own IP address), it resets its timer (T1) to one or more PeriodTime + d and moves back to the idle state.
3. If T1 expires, the HLIM router sends a HLIM_Parent_Hello[pref] into the core subnet via its D_IF, and sets its timer (T2) to HoldTime and its counter (N) to 1. The pref is the preference value of the HLIM router. A lower pref value indicates that the router is more preferred to be a HDR. A HLIM_Parent_Hello[pref] is multicast to all the other HLIM routers in the same core subnet. The router then moves to the transient state.
4. If the HLIM router gets a HLIM_Parent_Hello[pref] through its D_IF before T2 expires and the pref is lower than its own pref (or if the pref is equal to its own pref but the source IP address is lower than its own IP address), it sets its timer (T1) to one or more PeriodTime + d and moves back to the idle state.
5. If T2 expires and N is below a threshold (which is set to allow the HDR to send the HLIM_Parent_Hello multiple times before becoming elected in case the message is lost), the HLIM router sends out the HLIM_Parent_Hello[pref] again, increments its counter (N), and restart its timer (T2) to HoldTime.
6. If T2 expires and N reaches the threshold, the HLIM router will:
   - Assume itself to be elected, set its pref to zero, and move to the elected state.
   - Send a HLIM_Parent_Hello[0] to its D_IF.
   - Reset its timer (T3) to PeriodTime.
7. When T3 expires, the HLIM router sends the HLIM_Parent_Hello[0] to its D_IF again and restarts its timer (T3) to PeriodTime.

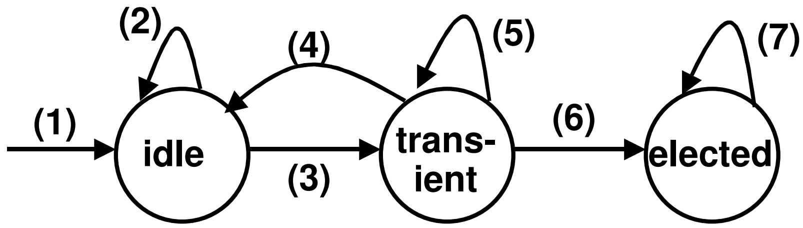

Figure 6-1. State Diagram of HLIM election process

If the elected HDR keeps sending the periodic HLIM_Parent_Hello[0] message, the backup routers will refresh its timer and remain in the idle state (transition 2). If the elected HDR has moved away, malfunctioned, or been destroyed, no periodic HLIM_Parent_Hello[0] message will be received and the timer (T1) of one of the backup routers will eventually expire first and then send out a HLIM_Parent_Hello[pref] (transition 3). The election of a local HDR is elected similarly using the IGMP_Hello message. After a new HDR is elected, it will perform the followings:

1. Similar to a network mobility case, the newly elected HDR exchanges HLIM_HDR_List_Solicit and HLIM_HDR_List_Update with its parent HDR to obtain the HDR list of its ascendant HDRs, updates the list, and forwards it in a HLIM_HDR_List_Update to its descendant HDRs.

2. It then invokes the HLIM compare for election (HLIM_Compare(E)) operations as described below to recover the caches originally in the previous HDR. The local membership will be recovered by the IGMP_Report or IGMP_M_Report sent by the local hosts after the new HDR is elected.

6.2 HLIM COMPARE OPERATIONS FOR ELECTION - HLIM_COMPARE(E)

1. When a new HDR is elected, it issues a HLIM_Compare(E) message to both its U_IF (i.e., toward its upper core subnet) and D_IF (i.e., toward its lower core subnet). The transmission of the HLIM_Compare(E) message will be repeated after a certain time interval (Compare_Repeat_Timer) for a number of times (≥ 1) to increase the reliability of this operation.
2. When a HDR receives a HLIM_Compare(E) from an interface (U_IF or D_IF), it sets two timers: a Compare_RandTimer which is a random timer (its maximum value should be less than the Compare_Repeat_Timer mentioned above) to wait before sending a HLIM_Compare_Reply(E), and a Compare_Timer (its value should be longer than the maximum time for the new HDR to finish all the retransmissions of the HLIM_Compare(E)) during which the HLIM_Compare_Reply(E) messages from its neighboring HDRs will be collected. A HLIM_Compare_Reply(E) is *multicast* and contains the inbound GAs for receivers and outbound GAs for sources. Inbound GAs refer to those GAs whose caches have the incoming interface of the HLIM_Compare(E) as an outgoing interface. Outbound GAs refer to those GAs whose caches have an outgoing interface opposite to the incoming interface of the HLIM_Compare(E).
3. When a HDR has its Compare_Timer expire, the HDR will perform the following comparison:
   - The HDR compares its own outbound GAs for receivers and the inbound GAs for receivers in all the HLIM_Compare_Reply(E) messages received. It invokes HLIM_Join(R) (if the HDR is inside the scope region of the GA) or HLIM_M_Join(R) (if the HDR is outside the scope region of the GA).
   - The HDR compares its own inbound GAs for sources and the outbound GAs for sources in all the HLIM_Compare_Reply(E) messages received. It invokes HLIM_Join(S) (if the HDR is inside the scope region of the GA) or HLIM_M_Join(S) (if the HDR is outside the scope region of the GA).
4. If the new HDR replaces the RID of the GA of an on-going multicast session, it will invoke the HLIM_Flush and IGMP_Flush operations to update the forwarding caches of the established multicast tree. The HLIM_Flush is sent through the whole multicast tree to update the caches with the affected RID.
   - The new HDR elected will send a HLIM_Flush[old RID, new RID] to any D_IF and U_IF listed in the affected caches in its R_MFC and S_MFC lists, and an IGMP_Flush[old RID, new RID] to the L_IF if it is listed in the caches.
   - When a HDR receives a HLIM_Flush, it will check if it has any forwarding caches with the old RID.
     - ➢ If matches are found, it will update the caches with the new RID. It will then forward the HLIM_Flush to the Out_IF (U_IF or D_IF) indicated in the caches, except the incoming interface of the message, and IGMP_Flush to the L_IF if it is listed in the caches.
     - ➢ Otherwise, it will ignore the message.
   - When a host receives an IGMP_Flush, it will update its own records for any active multicast sessions of the old RID.

   Note that the new HDR elected may send the HLIM_Flush and IGMP_Flush for several times in case there is any message loss.

7. HLIM RELIABILITY CONSIDERATIONS

Throughout the design of the HLIM protocol design, control message reliability has always been taken into account. Reliability measures are built into the design. Three main schemes are used to handle control message loss and increase reliability at the HLIM protocol level. Additional reliability can be provided by the lower layers (such as forward error correction, automatic repeat request, etc.) but they are outside the scope of the HLIM design. The three reliability schemes used in HLIM are described below:

1. Periodic transmission - Some of the HLIM/IGMP messages are transmitted periodically, or sent in response to periodic messages. If they are lost, another one will always be sent at the next period or in response to the next periodic message.
2. Acknowledgment and retransmission - Two-way handshake is applied to some of the HLIM/IGMP messages. If the acknowledgment of a protocol message is not received by the message originator within a timeout period (i.e., either the message or the acknowledgment is lost), the message, and thus the acknowledgment, will be retransmitted until the maximum number of trials (the maximum number is chosen according to the target operating environments).
3. Multiple transmissions - In some cases, two-way handshake is difficult to implement. Multiple transmissions will be used instead. A protocol message will be sent consecutively for multiple times (the number of transmissions is chosen according to the target operating environments). Some reply messages may be sent in response to these protocol messages. Such a reply message will be returned for each of the repeated messages received. In the case that a message/reply is lost, there are other chances for the target recipient to receive the message/reply.

The following table lists the reliability scheme used for each control message.

Table 1 The Methods Applied for Increasing Reliability in HLIM Operations

❑ **HLIM Control Operations**	❑ **Methods of Reliable Transport for Control Message**
Regular Membership Operation • IGMP_Query • IGMP_Report	Periodic transmission
Mobile Membership Operation • IGMP_M_Report • IGMP_M_Reply	Acknowledgment and retransmission
RID Retrieval Operation • IGMP_RID_Request • IGMP_RID_Reply	Acknowledgment and retransmission
Host Mobility Detection • IGMP_Hello	Periodic transmission
RID Update Operation (in local) • IGMP_Flush	Multiple transmissions
Join Operation • HLIM_Join • HLIM_ACK • HLIM_M_Join • HLIM_M_ACK	Acknowledgment and retransmission
Network Mobility Detection	Periodic transmission

• HLIM_Parent_Hello • HLIM_Child_Hello	
HDR List Update Operation (solicited) • HLIM_HDR_List_Solicit • HLIM_HDR_List_Update	Acknowledgment and retransmission
HDR List Update Operation (unsolicited) • HLIM_HDR_List_Update	Multiple transmissions
HLIM_Compare(E) Operation • HLIM_Compare(E) • HLIM_Compare_Reply(E)	Multiple transmissions
HLIM_Compare(M) Operation • HLIM_Compare(M) • HLIM_Compare_Reply(M)	Acknowledgment and retransmission
Prune Operation • HLIM_Prune • HLIM_Prune_Reply • HLIM_M_Prune • HLIM_M_Prune_Reply	Multiple transmissions
Network Join Operation • HLIM_Net_Join • HLIM_Net_Join_ACK	Acknowledgment and retransmission
RID Update Operation (in global) • HLIM_Flush	Multiple transmissions

8. SECURITY CONSIDERATIONS

Security issues are not discussed in this document.

9. ACKNOWLEDGEMENTS

The work on HLIM was funded by the U.S. Army CECOM under the contract DAAB07-98-C-D007. The authors would also like to thank Dr. Charles Graff and Mr. Michael Bereschinsky (of U.S. Army CECOM) for their continued support and active involvement in this work.

10. INTELLECTUAL PROPERTY CONSIDERATIONS

Telcordia Technologies, Inc. may seek patent or other intellectual property protection for some of the technologies specified in this document. If any standards arising from this disclosure are or become protected by one or more patents assigned to Telcordia Technologies, Telcordia Technologies, Inc. intends to disclose those technologies and license them on reasonable and non-discriminatory terms. Future revisions of this draft may contain additional information on intellectual property protection sought or received.

11. AUTHORS' ADDRESSES

Matthew Cheng
Telcordia Technologies

331 Newman Springs Road
Red Bank, NJ 07701-5699
Phone: (732) 758-2897
Email: mcheng@telcordia.com

John Lee
Telcordia Technologies
331 Newman Springs Road
Red Bank, NJ 07701-5699
Phone: (732) 758-2896
Email: jolee@research.telcordia.com

Mario Joa-Ng
Telcordia Technologies
331 Newman Springs Road
Red Bank, NJ 07701-5699
Phone: (732) 758-3377
Email: mjoang@research.telcordia.com

REFERENCES

1. T. Pusateri, "Distance Vector Multicast Routing Protocol", Internet Draft. ftp://ftp.isi.edu/internet-drafts/draft-ietf-idmr-dvmrp-v3-08.txt, Feb. 1999.
2. J. Moy, "Multicast Extensions to OSPF", RFC 1584, March 1994.
3. S. Dearing, et. al., "Protocol Independent Multicast Version 2, Dense Mode Specification," Internet Draft, ftp://ftp.isi.edu/internt-drafts/draft-ietf-pimv2-dm-0.txt, Nov. 1998.
4. D. Estrin, et. al., "Protocol Independent Multicast-Sparse Mode (PIM-SM) : Protocol Specification", RFC 2362, June 1998.
5. A. Ballardie, "Core Based Trees (CBT) Multicast Routing Architecture", RFC 2201, September 1997.
6. A. Ballardie, "Core Based Trees (CBT version2) Multicast Routing - Protocol Specification," RFC 2189, Sep. 1997.
7. G. Xylomenos and G. C. Pdyzos, "IP Multicast for Mobile Hosts", IEEE Comm. Mag., Jan. 1997.
8. C. E. Perkins, "Mobile IP Design Principles and Practices", Addison Wesley, 1998.
9. R. Ramanathan and M. Steenstrup. "Hierarchically-organized, multi-hop mobile wireless networks for quality-of-service support", ACM Mobile Networks & Applications (MONET), June 1998.
10. C. Graff, "Future Network Architecture for Tactical Army Networks", IEEE talk, Viewgraphs available from graff@doim6.monmouth.army.mil.
11. W. Fenner, "Internet Group Management Protocol, version 2 (IGMPv2)," RFC 2236, Nov. 1997.