Project Number:	**IST-2001-32603**
Project Title:	**6NET**
CEC Deliverable Number:	**32603/ULP/DS/4.1.5v1/A1**
Contractual Date of Delivery to the CEC:	December 31st 2003
Actual Date of Delivery to the CEC:	December 19th 2003
Title of Deliverable:	Multicast with mobile hosts: analysis and performance evaluation
Work package contributing to Deliverable:	WP4
Type of Deliverable*:	R
Deliverable Security Class**:	PU
Editors:	Christophe Jelger (ULP), Thomas Noel (ULP)
Contributors:	Christophe Jelger (ULP), Thomas Noel (ULP)
Reviewers:	Stig Venaas (UNINETT), Martin Dunmore (ULANC)

* Type: P - Prototype, R - Report, D - Demonstrator, O - Other

** Security Class: PU- Public, PP – Restricted to other programme participants (including the Commission), RE – Restricted to a group defined by the consortium (including the Commission), CO – Confidential, only for members of the consortium (including the Commission)

Abstract:

This document aims to provide an analysis of the issues relative to IP multicasting in the presence of mobile nodes. The objective is to make people aware of the specific concerns raised by the combination of multicasting and mobility. This deliverable also presents and evaluates some of the solutions that have been proposed to handle this particular situation. It is hoped that this document will provide a solid introduction for readers that may want to deploy such services.

This is the first version of this deliverable. A second version is due in month 36 of the project (December 2004). The second version of this deliverable will focus on practical issues when deploying multicast in the presence of mobile nodes.

Keywords:

Multicast, Mobile IPv6, Wireless LANs.

Executive Summary

This document proposes an introduction to the issues related to IP multicasting in the presence of mobile nodes using the Mobile IPv6 protocol. The motivation for providing multicasting to mobile devices is twofold. First, mobile communications are becoming extremely popular and efficient. Second, the wireless technology itself is not any longer the only criteria when it comes to make a choice between equivalent vendors or operators. Available services are indeed becoming of major importance. In the area of multimedia services, multicasting has a lot to offer, with services such as Internet TV and radio, network games, and video-conferencing applications. However, while multicasting has been designed to handle the dynamic registration of participants within a group communication, a number of issues arise when some of the participants are mobile.

In this document, we have therefore tried to provide a broad analysis of the issues raised by multicast operation in the presence of mobile nodes. Realistic proposals considered at the IETF are presented and evaluated. Some research-based extensions are then exposed. We then detail and evaluate a solution to the problem introduced by mobile SSM sources. SSM (Source-Specific Multicast) is a model in which only a single source is allowed to send data to a multicast group, i.e. it is a *one-to-many* communication. Finally, we discuss some concrete deployment issues and present our experience gained from experimentations with the currently available implementations of the latest version of Mobile IPv6. It is hoped that this document will provide a solid introduction for readers that may want to deploy multicast services in a mobile environment.

This is the first version of this deliverable. A second version is due in month 36 of the project (December 2004). The second version of this deliverable will focus on practical issues when deploying multicast in the presence of mobile nodes.

Table of Contents

1 Introduction and problem statement

IP multicasting has been extensively studied in the past twenty years. In contrast and for various reasons not detailed here, there have been very few native deployments of multicast in ISP networks. However, the increasing availability of broadband access technologies like ADSL, along with the transformation of personal computers into interactive multimedia desktops (for digital video, music, Internet TV and radio, network games …), are factors which call for a deployment of native multicast support in the next generation Internet. The transition/migration to IP version 6 can also benefit from the immediate adoption of native multicast support. In particular, services like Internet TV and radio are perfect examples of multicast-like applications, where one source broadcasts the same content to a (possibly) very large set of receivers.

Meanwhile, there has also been an increasing interest for wireless communications, especially since powerful handheld devices are now widely available. It is also becoming clear that mobile Internet users will expect to have access to the services and applications that are available in traditional wired networks and these services will surely include multimedia applications. Consequently, many efforts are being made to provide efficient mobility and multicasting support and to bring the two together in the next generation of IP networks.

Despite the fact that IP multicasting has been designed in order to support dynamic registration of group members and dynamic delivery tree maintenance, current multicast protocols have not been optimized to handle host mobility. One can think that dynamic membership and host mobility have similar requirements in the sense that the two problems can be solved by common mechanisms. In fact, most methods used to handle multicast group registration with static hosts need strong optimizations in order to be efficient when applied to mobile devices. Moreover, the performance of current multicast routing protocols (e.g. in terms of routing convergence efficiency and tree construction overhead) is in particular greatly worsened in the presence of mobile multicast senders.

The objective of this document is therefore to study the effects and implications of node mobility on multicast communications. Within the 6NET project, issues regarding multicast deployment with IP version 6 are studied by WP3, while issues regarding the deployment of Mobile IPv6 are tackled by WP4. ULP has man power in both work packages, and we are therefore in a good position to fill the gap between multicast and mobility.

This document will first present the two (realistic) solutions that have been proposed by the IETF MIP6 working group in order to support multicast in Mobile IPv6. We also present some simulation results comparing the two alternatives. Chapter three will then shortly present some research proposals whose objective is to enhance multicast communications in the presence of mobile nodes. In Chapter four, we present a solution developed by ULP in order to support mobile SSM sources. Simulations results are also presented. In the next chapter, we detail the functionalities of an MLD proxy, i.e. an application that can be used on a home agent in order to support bi-directional tunnelling of multicast data. The MLD proxy can be used if the home agent does not integrate the functionalities of a multicast router. Finally in Chapter 6, we present the first results obtained with available implementations of Mobile IPv6.

This is the first version of this document, a second and final version is due in M36 of the project (December 2004). The second version of this deliverable will focus on practical issues when deploying multicast in the presence of mobile nodes (it will mainly be a large extension of Chapter 6).

2 Summary of IETF proposals in MIPv6 draft 24

The IETF MIP6 (Mobility for IPv6) working group has proposed to use two straightforward solutions which can allow a mobile node to receive and send data to one or more multicast groups. These solutions, known in the literature as *remote subscription* and *bi-directional tunnelling* have originally been proposed in [1] for IP version 4 and in [2] for IP version 6. Actually, these mechanisms do not depend on any particular version of the IP protocol. The use of these two solutions is explicitly mentioned in the last version of the Mobile IPv6 draft [3]. These two solutions are illustrated in Figure 1 (HA = Home Agent, MN = Mobile Network).

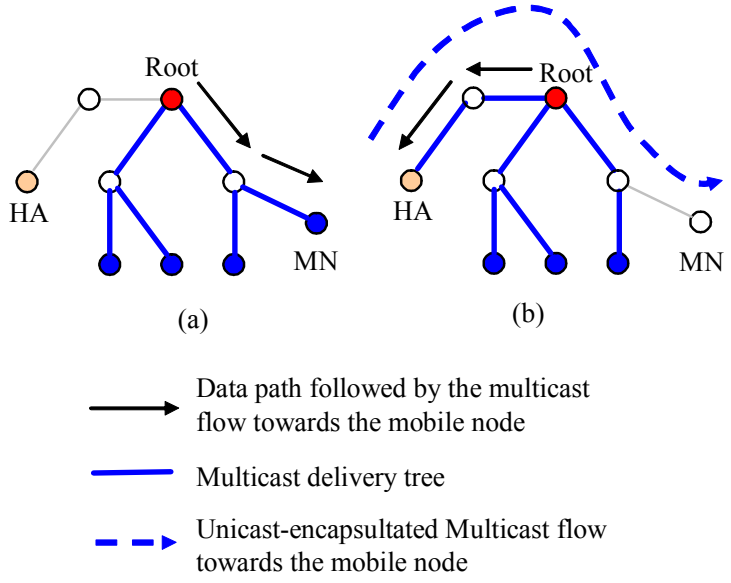

Figure 1. Bi-directional tunnelling vs. remote subscription

2.1 Remote subscription

With the solution known as remote subscription, a mobile node joins (or leaves) a multicast group via a (local) multicast router located on the foreign link being visited. Thus, the mobile node follows the traditional behaviour of any stationary node that wishes to join (or leave) a multicast group. In particular, the mobile node must use its (visited) link-local address when sending MLD messages (to the "all routers" multicast address). The mobile node must use its care-of address if it wants to send data to the group (if the mobile node is also a source) in order to avoid possible problems with ingress filtering. All packets/datagrams are sent directly on the foreign link. The main advantage of this method is that routing is optimal, in the sense that the multicast data follows a shortest path route between the mobile node and the root of the multicast delivery tree. The main drawback of this solution is that it can introduce frequent tree reconstructions when the mobile node is in movement and hand-offs between different subnets. Moreover, if source-rooted multicast is considered, a major problem occurs if the source is mobile (see Chapter 4).

This proposal is illustrated in Figure 1(a): multicast data is sent from the root of the tree to the visited network that hosts the mobile node.

2.2 Bi-directional tunnelling

Alternatively, a mobile node can join (or leave) multicast groups via a bi-directional tunnel to its home agent: this method is known as bi-directional tunnelling. The mobile node tunnels all its MLD messages to the home agent, which will then forward data to the mobile node via the tunnel. If the mobile node wants to send data to the group it must also use the tunnel. To do so, the source address of the multicast datagrams sent by the mobile node must be the home address of the mobile node(in order to avoid possible problems with ingress filtering). In order to support bi-directional tunnelling, the home agent must either have the functionalities of a multicast router (routing and forwarding), or have the functionalities of an MLD proxy (as detailed in Chapter 5). The main advantage of this solution is that the mobility of the mobile node does not have any influence on the multicast tree(s), since the mobility is hidden. Another advantage is that multicast only needs to be supported at the home network. The main drawback of this proposal is that routing becomes sub-optimal because the solution involves triangular routing via the home agent. Another problem introduced by this method is known in the literature as *tunnel convergence*. For a given multicast group, a home agent may indeed forward the same multicast flow to two mobile nodes which are located in the same visited network. This results in a waste of bandwidth and resources.

Bi-directional tunnelling is illustrated in Figure 1(b): multicast data is forwarded from the root of the tree to the home agent (HA) which forwards it (encapsulated) to the current position of the mobile node.

2.3 Simulation results comparing the IETF proposals

Remote subscription clearly exhibits better routing performance than bi-directional tunnelling in terms of data delivery path (i.e. it builds a shortest-path tree from the root of the multicast tree to the receivers). However, it is not trivial to intuitively guess which of the two solutions can offer the smallest delay to start receiving the multicast flow (as witnessed by the mobile subscribers). This is indeed strongly related to the relative positions of the receivers (including mobile nodes), of the home agent serving the mobile members and of the root of the multicast delivery tree.

This section therefore aims to evaluate and compare the two IETF solutions in terms of both data delivery path length and delay (i.e. the time and length in hops it takes for a multicast datagram to reach a mobile receiver). Moreover, we also propose to use a new extension of the Multicast Listener Delivery (MLD) protocol that we call "MLD Hold". With such a message, a mobile node can ask its home agent to temporarily stop forwarding data for a given multicast group G but that it should remain a member of G. This option can be useful in a number of situations. Consider a mobile node (MN) which is subscribed to a group G via its home agent. The MN detects an imminent handoff, but does not know if native multicast is supported in the network it is about to get connected to. It therefore sends a MLD hold to its home agent, so that multicast delivery can be rapidly resumed after the handoff (in a faster way than the case where the home agent has to re-subscribe to the group) if native multicast is not supported in the visited network.

This study has lead to a publication in an international conference [4]. A footnote in this paper gives credit to the 6NET project. Details of the simulation methodology can be found in [4]. They are intentionally omitted in this report which primarily focuses on the results. However, note that we have performed a very large amount of simulations with different Internet-like topologies, simulating different group sizes and configurations (i.e. different positions of the receivers, the sources, and the root of the trees). The results presented here are significant with respect to all the simulations carried out, in the sense that the statistical error of the results presented in this document is very low.

Figure 2 illustrates the main conclusion obtained from the simulations. We have measured the efficiency of the two IETF solutions in terms of both delivery delay and delivery path length (in hops). Because remote subscription is always equal or better than bi-directional tunnelling with respect to route length, we have presented the gain obtained with this solution when compared to tunnelling via the home agent. This is presented in the form of a survival function, i.e. it gives the amount of receivers that see a gain strictly superior to a given value.

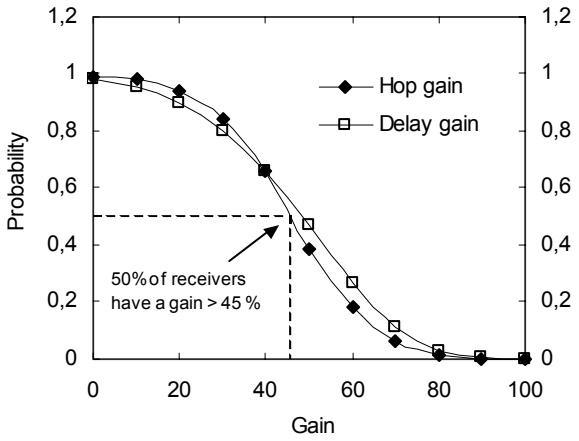

Figure 2. Gain distribution across receivers

From what is seen on Figure 2, it is clear that remote subscription gives better performance than bi-directional tunnelling. On average (not shown on the figure), multicast datagrams sent directly to the visited network travel via a path that is 46.5% shorter than the alternate path via the home agent. This is consistent with the nature of triangular routing: if the average distance (in hops) between any two nodes in a graph is equal to d, then the length of the path between (any) two nodes A and B via a node C is equal to $2d$ (the distance between A and C is d, and the distance between C and B is also d). Therefore the gain (in hops) of using the direct path between A and B, instead of going via the node C, is equal to 50%. The simulations that we have carried out have permitted to extend this result to multicast communications. Note that the same conclusions apply to the measures of the delay.

We have shown that remote subscription gives better performance than bi-directional tunnelling in terms of datagrams delivery. The length and delay of the path followed by the multicast datagrams are indeed shorter when remote subscription is used. Moreover, we have also measured if it was more interesting (or not) to join a group via the visited network than via the home agent. We have also considered the case in which the home agent was already a member of the group(s) of interest (or, strictly talking, a router in the multicast tree). We have considered that the home agent has stopped forwarding the multicast flow to the mobile node (of interest) because the latter had sent a *MLD Hold* message to its home agent. With such a message proposed in [5], a mobile node can ask its home agent to stop forwarding data for a given multicast group, but that the home agent must remain a member of the group. When the mobile node wants to ask its home agent to resume the forwarding of data, it simply sends a *MLD Report* message to its home agent. Because the home agent is already a member of the group, the resumption of the data delivery is faster than if the home agent had to rejoin the group. The results obtained during this second set of measurements are illustrated by Figure 3.

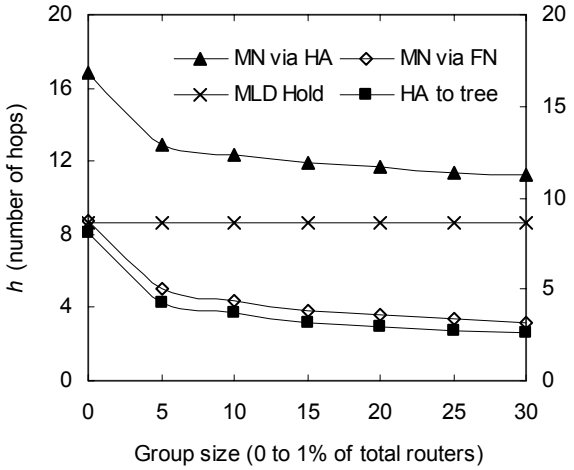

Figure 3. Distance *h* between receivers and tree members

With remote subscription, *h* represents the length (in hops) between the mobile receiver and the closest tree member on the path towards the root of the tree, e.g. the length that a PIM Join request has to travel to reach a tree member. With bi-directional tunneling, *h* represents the length between the home agent and the closest tree member on the path towards the root of the tree, plus the distance between the mobile receiver and its home agent. That is, the length that the unicast-encapsulated *MLD Report* sent by the mobile receiver has to travel to reach the home agent, plus the length traveled by the PIM Join request (sent by the home agent towards the root of the tree) before it reaches a tree member (or eventually the root). This indicator (*h*) shows "how quickly" (in hops) the mobile receiver managed to graft to the existing multicast tree, and it therefore also shows how quickly the mobile receiver will start to receive the multicast data flow.

The curve labeled "MN via HA" shows *h* for bi-directional tunneling, and the curve labeled "MN via FN" (for Foreign Network) shows *h* for remote subscription. We have also considered the case in which the home agent was already subscribed to the multicast group but was not forwarding data (i.e. the home agent had received a MLD hold message sent by the mobile node). In this case, *h* would represent the distance between the home agent and the mobile receiver (i.e. the length that the unicast-encapsulated MLD report sent by the mobile receiver has to travel to reach the home agent). This is also shown in Figure 3 under the label "MLD hold". Finally, the fourth curve labeled "HA to tree" shows the distance between the home agent and the closest member of the tree in the case of bi-directional tunneling.

The *x* axis of Figure 3 gives the number of static receivers (for a given group). It can be seen that when the size of the group is equal to 0, the value for "MN via HA" is equal to 16.8, which is twice the value of the average distance in the Internet-like topologies considered during the simulations (i.e. 8.77). In contrast, the three other curves show a value of around 8. These values are very coherent with the expectations (remind that the group size is 0). As the number of static receivers increases, a few observations can be made. First, the curve "MLD hold" remains constant because it is only depends on the average distance of the topologies. On the other hand, the curves "MN via HA" and "MN via FN" decrease in a very similar way. This is again coherent with the experimentation : in one case the home agent grafts to the tree, and in the other case the mobile node grafts to the tree. Again due to the large number of simulated combinations, we find similar average results. The decrease is explained by the fact that, when the number of members in the tree increases, there is a higher probability to reach a tree member in fewer hops.

To conclude, it is clear from this set of results that it is more interesting for a mobile node to join a multicast group via the visited network. It is however worth noting that if the home agent is already a member of the group, the distance *h* becomes constant (on average among all simulations).

2.4 Conclusions and discussions

We have shown in the previous sub-section that remote subscription exhibits better performance than bi-directional tunnelling, in the sense that datagrams will follow a shorter path with a smaller end-to-end delay. Moreover, it is more interesting for a mobile node to join a group via the current visited network.

While these results are purely based on performance estimations, there are a number of reasons which are in favour of bi-directional tunnelling. First, remote subscription implies that a network visited by a mobile node must support multicast. If we look at the state of multicast deployment with IP version 4, this condition would not be satisfied in the vast majority of cases. However, we expect multicast to be more widely deployed with IP version 6 in order for ISPs to support popular services such as Internet TV and radio. In spite of this, bi-directional tunnelling remains a very attractive solution, as only the home network of a mobile node has to support multicast. Second, bi-directional tunnelling is the only solution for a mobile node to subscribe to a group that is local to the home network. If we consider, for example, a video-conference which is local to the network of a company (the network itself may span multiple sites, thanks to VPN technology), bi-directional tunnelling is the only solution that can allow a mobile user to join the session. Third, the centralization imposed by bi-directional tunnelling (i.e. all datagrams are forwarded by the home agent) permits to use transcoding mechanisms: for example, the home agent can modify the original multicast flow, e.g. by changing the video codec in order to support the size of the screen of the mobile equipment(s) to which it forwards the flow.

In the mean time, remote subscription has a number of disadvantages compared to bi-directional tunnelling. First, each time a mobile node performs a hand-off it must re-join the group(s) to which it is subscribed. We show in Chapter 6 that in practise this constraint induces a large disruption time in the delivery of the multicast flow. In contrast and with bi-directional tunnelling, a mobile node simply sends a BU message to its home agent after the hand-off and subsequently the home agent will forward the multicast data to the new location of the mobile node. Second, if a mobile node is the source of a source-rooted tree (e.g. the source of an SSM group), a handoff of the mobile node would result in the interruption of the multicast flow (i.e. all the tree branches are rooted to a network that the source has just left). However, a solution presented in Chapter 4 can handle this situation. In contrast and with bi-directional tunnelling, the source-rooted tree would be rooted at the home network of the mobile source, and the mobility of the source has therefore no effect on the multicast tree. Details of this technique are given in Chapter 4.

To conclude, we believe that the two solutions are in fact complimentary since they provide different levels of services. Remote subscription provides better performance in terms of datagrams delivery, while bi-directional tunnelling offers more flexibility and adaptability. The main features of these two solutions are summarized in Table 1.

	Advantages	**Drawbacks**
Remote subscription	▪ Routing is optimal ▪ More scalable	▪ Visited network(s) must be multicast capable ▪ Requires native multicast support at the foreign network
Bi-directional tunnelling	▪ Allows the forwarding of data local to the home network ▪ Supports source-rooted trees with mobile sources ▪ Transcoding is possible ▪ Faster recovery after handoff	▪ Introduces triangular routing ▪ Limited scalability ▪ Processing overhead at the HA

Table 1. Advantages and drawbacks of MIPv6 proposals to support multicast

3 Summary of research based proposals

There have been a few research proposals which try to optimize the delivery of multicast data to mobile nodes. Most of these proposals have however been proposed for IP version 4, but they could be extended to IP version 6. We decided to include this section mainly to shortly show that alternate solutions (to the IETF proposals) have been proposed. In a general comment, while all these solutions use innovative mechanisms in order to optimize the delivery of multicast data to mobile nodes, all of them suffer from severe deployment problems. In particular, all these solutions will not scale with the size of the Internet. Moreover, these protocols require a collaborative effort between routers that may belong to different entities. The nature of this collaboration is such that these entities may not be willing to cooperate.

The Mobile Multicast (MoM) protocol [6] has been proposed to solve the *tunnel convergence* problem already mentioned in the previous section. It has been proposed for IP version 4 and cannot be directly extended to IP version 6 as it makes use of the foreign agent (FA) which does not exist in Mobile IPv6. In MoM, the FA appoints one home agent as the *designated multicast service provider* (DMSP) for a given multicast group. The DMSP forwards only one multicast datagram to the FA (and not to the mobile nodes it serves) which delivers the data in native multicast over its local link. Home agents that are not the DMSP must stop forwarding packets to their respective mobile nodes.

The Range-Based Mobile Multicast (RBMoM) [7] protocol has been proposed in order to trade off between the shortest delivery path and the overhead induced by the multicast delivery tree reconfiguration. In particular, remote subscription and bi-directional tunneling have been shown to be extreme cases of RBMoM. In this proposal, a router called the *Multicast Home Agent* (MHA) is responsible for tunneling multicast data to foreign networks where mobile hosts reside. The initial MHA of a mobile node is set to be its home agent (i.e. RBMoM is then similar to bi-directional tunneling) and the tunnel convergence problem is solved by using a DMSP selection algorithm as in MoM. Each MHA has a specific service range and a MHA only forwards packets to mobile hosts that are located within the service range. When a mobile node moves to a new foreign network, its new foreign agent (FA) is made aware of the mobile node's previous MHA and it subsequently calculates the distance to the MHA (e.g. in number of hops). If it is greater than the MHA's range, a new MHA must be selected and in the current version of RBMoM, the FA becomes the new MHA. If the new MHA is not already part of the appropriate multicast tree(s) it must initiate the corresponding join procedure. The performance of RBMoM is mainly controlled by the choice of the service range and it has been shown how this value can be chosen in order to adapt to mobility changes and group sizes. Also, RBMoM has so far been considered for use with IPv4 and, because it makes use of a foreign agent, it cannot be directly extended to IPv6.

A similar protocol has also been proposed in [8]. It is called Multicast by Multicast Agent (MMA) and it introduces a Multicast Agent (MA) and a Multicast Forwarder (MF). As the MHA in RBMoM, a MF is responsible for forwarding multicast packets to the MA of the foreign network (which forwards it in native multicast on its local link) but in MMA the range of the MF is unlimited. Initially, the MF of a network is the MA itself. When a mobile host reaches a network whom MA is not served by a MF, the MF that served the network where the mobile comes from becomes the MF of the current MA. In contrast, if the new MA was already served by a MF, then a MF selection occurs between the current MF and the MF that served the network from which the mobile arrived. This proposal can be used with both IP versions 4 and 6.

4 Mobile SSM Sources for IPv6 (MSSMSv6)

4.1 Introduction

In a very near future, multicast groups with mobile sources will become a reality. For example, wireless video cameras (using IEEE 802.11b) are already available. One can imagine a city equipped with wireless LAN access, in which a reporter wants to interview people in the streets. The reporter is equipped with a wireless video camera and is multicasting the video flow on the Internet directly from its current position. One can also imagine the same scenario at a larger scale, i.e. the same reporter hopping between the base stations of a cellular network like the future UMTS network.

To support a mobile multicast source, two solutions are available. If a multicast shared tree is used, the mobile source simply sends its data to the root of the tree (e.g. the RendezVous point in PIM-SM), which forwards it to the receivers. The mobility of the source is (partially) hidden to the receivers. If the source performs a handoff and acquires a new care-of address, the source will appear as a new source to the receivers. Higher layers protocols may *hide* this to the user, but the network layer will see the mobile source as a new source. If a source-rooted tree is used, the multicast tree will be rooted at the current position of the mobile source. If the source performs a handoff, the reception of the multicast flow is interrupted because the tree remains rooted at the old location of the source. To overcome this situation, we have proposed a protocol called Mobile SSM Sources for IPv6 (MSSMSv6). SSM (Source-Specific Multicast) is multicast model in which only a single source is allowed to send data to a multicast group. In SSM, the association between a source (S) and a multicast group address (G) is defined as a channel (S,G), i.e. a source S sending to a multicast group address G. Our proposal extends the Mobile IPv6 support to handle the mobility of such multicast sources. This protocol is presented and evaluated in the next two sections.

The other alternative that can be used to support mobile multicast sources is bi-directional tunneling. With both shared trees and source-rooted trees, the mobile source sends its data encapsulated to its home agent. The source address of the multicast datagrams (i.e. inner header of the encapsulated packet) must be the home address of the mobile node. Therefore the source address of datagrams sent by the mobile source remains unchanged and the current location of the source is hidden to the receivers. The main drawback of this solution is that it introduces triangular routing via the home agent. Moreover, in the particular case of a source-rooted tree, the tree would not be rooted at the real and current position of the mobile source. This is somehow in contradiction with the nature of a source-rooted tree. This particular example is illustrated on Figure 4 with a mobile SSM source (acronyms on figure 4 are: cCoA = current care-of address ; HA = home agent (and home agent address) ; H@ = mobile node home address ; G = multicast group address ; the pair (H@,G) represents the SSM channel).

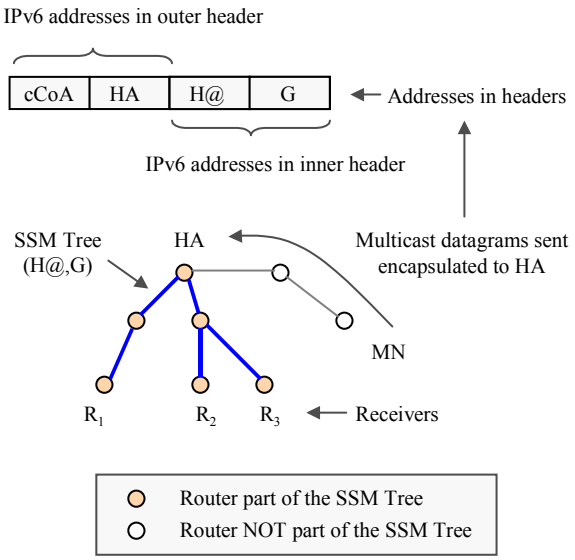

Figure 4. Bi-directional tunnelling with a mobile SSM source

4.2 Protocol specifications

First, it is important to introduce the concept of channel and session announcements. In the current existing multicast backbone (Mbone), the existence of a multicast session is usually announced by using a specific tool implementing both the Session Announcement Protocol [10] (SAP) and the Session Description Protocol [11] (SDP). Session announcement is performed by multicasting the session description to a common well-known group (as defined in [10]) that multicast listeners join if they wish to receive session announcements. The most popular tool for session announcements is called *SDR* (for Session Directory) and it is currently being maintained at the University College of London. In this document, we will often refer to both channel and session announcements, and will assume that this functionality is achieved by using a *SDR*-like tool.

The main terminology introduced by our proposal is related to SSM. In particular, a SSM channel (S,G) as defined in [12] identifies the multicast delivery tree associated with a source address S and a SSM destination address G. In addition we introduce the notion of a SSM session (S',G'), which identifies a SSM communication associated with a source address S' and a SSM destination address G'. In the case of a mobile SSM source, the session may be different than the channel used to identify the multicast delivery tree, mainly because S can be different than S'.

In practice, and in order to maintain consistency at the transport layer, the SSM session must remain constant and, for a mobile source, will be identified by the pair (H@,G), with H@ being the home address of the mobile source and G being the SSM destination address. Because we want the channel to be rooted at the current location of the mobile source, the channel will be identified by the pair (cCoA,G), with cCoA being the current care-of address of the mobile source. With MSSMSv6, the mobile source must announce the SSM channel along with its home address. Interested receivers must join the SSM channel (cCoA,G) that is rooted at the current position of the mobile, and use the SSM session (H@,G) to identify the communication at the transport layer. The source must use its current care-of address as the source address of the multicast datagrams it sends. This condition ensures correct processing by the routers on the tree. Moreover, these datagrams must also contain an IPv6 Home Address Option (as defined in Mobile IPv6), set to the home

address (H@) of the mobile source. This is used by the receivers' transport layers to identify the SSM session.

4.2.1 Source handover

When a mobile SSM source moves into a new subnet, it must inform the multicast receivers about its new care-of address (nCoA). Receivers subsequently join the new multicast delivery tree (nCoA,G). If the nCoA is known prior to the handover, by using a protocol such as Fast Handovers for Mobile IPv6 [13], the MN can send a Binding Update (BU) message onto the old delivery tree (oCoA,G) to inform the receivers about the nCoA. A new BU sub-option is required and its format is described in the next sub-section. This new sub-option is called *SSM-Source Handover Notification*. If the nCoA is known after completion of the handover (i.e. when the MN is in the new subnet), the BU message must be encapsulated towards the old Access Router (oAR), which will remove the outer header and then send the multicast datagram onto the old channel (oCoA,G). Whether the nCoA is known prior to the handover or not, the source address in the header of the multicast datagram that contains the BU must be oCoA. If the multicast datagram that contains the BU is sent after the handover completes (i.e. encapsulated in a unicast packet sent to the oAR), the source address in the header of the unicast packet must be nCoA.

Upon reception of a BU with the *SSM-Source Handover Notification* sub-option set to nCoA, a receiver initiates a join operation towards the new channel (nCoA,G). In practice, the receiver will send a *MLD Report* message for the channel. However, the receiver must not initiate the pruning of the old channel (oCoA,G) before it starts receiving datagrams on the (nCoA,G) channel. Only when the receiver starts receiving datagrams on the new channel (nCoA,G), it should initiate the pruning of the old channel (oCoA,G). In practice, the receiver will wend a *MLD Done* message for the old channel. If a receiver is unable to join the new channel, it should maintain its subscription to the old channel.

After the handover, the mobile source continues to send datagrams onto the old channel (oCoA,G) until it is notified by the oAR that there are no receivers listening to the old channel (oCoA,G) any more. This notification could be done via the Multicast Source Notification of Interest Protocol [14] (MSNIP) but details are out of the scope of the MSSMSv6 protocol itself. Note that to avoid maintaining two parallel trees if some receivers do not join the new delivery tree, the source can decide to stop forwarding data to the old tree after a fixed period of time.

When the MN is in the new subnet, it encapsulates the multicast datagrams to the oAR, for further delivery onto the old channel (oCoA,G). The source address in the header of the multicast datagram must be oCoA and the source address of the unicast packet must be nCoA. Once in the new subnet, the MN must also send the multicast datagrams in native multicast onto the new channel (nCoA,G). The source address of the native multicast datagrams must be nCoA.

If all of the above recommendations are followed, group members will continue receiving multicast datagrams from the old delivery tree (oCoA,G) until they manage to join the new SSM channel (nCoA,G). Our proposed protocol therefore allows receivers to perform a smooth migration from the old tree to the new delivery tree with minimal delivery disruption. When all the receivers have joined the new delivery tree, the source is notified by the oAR to stop sending packets to the old channel and the protocol reaches a stable state.

The whole operation described above is illustrated in Figure 5. Between (a) and (b), the source has performed a handover and has obtained a new Care-of-Address. The source has started to send encapsulated multicast datagrams to the oAR, the first datagram being a BU with the "SSM-Source Handover Notification" sub-option set to nCoA. Group members continue to receive datagrams from the old tree (oCoA,G). From (b) to (c), receivers R_2 and R_3 have joined the new delivery tree (nCoA,G), and they subsequently pruned the old branch. At that point, the two trees co-exist, but it is important to note that for a given receiver, only one branch is active at a time. From (c) to (d), receiver R_1 has joined the new delivery tree. At that point, oAR did not have any multicast state for (oCoA,G), and it notified the source to stop encapsulating data to the old tree. All receivers have joined the new delivery tree and the oAR is not any longer involved in the multicast delivery.

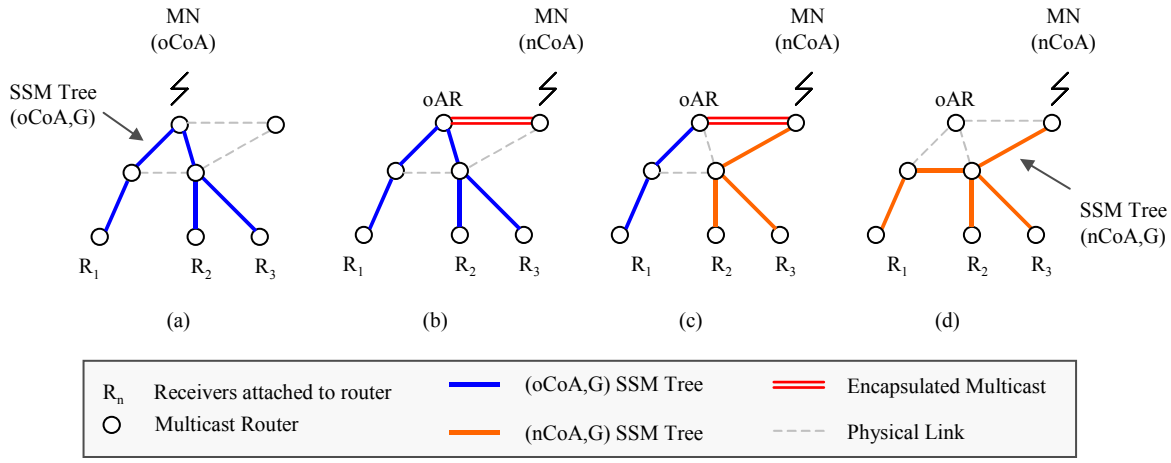

Figure 5. MSSMSv6 mode of operation

4.2.2 SSM-Source Handover Notification sub-option

We have defined a new binding update sub-option called SSM-Source Handover Notification. It is illustrated in Figure 6 (using the usual IETF format). It is used by a mobile source to notify multicast receivers about the new SSM channel associated with the mobile source after a handover. The field "new source address (nSA)" contains the new address to be used by receivers to join the new multicast delivery tree, i.e. the channel (nSA,G). The sub-option type/identifier is to be attributed (TBA).

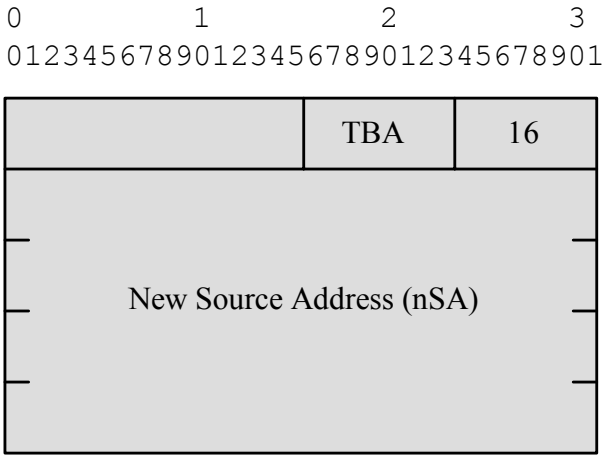

Figure 6. SSM-Source Handover Notification sub-option

4.2.3 Multiple handovers of the source

We now consider the case of a fast moving source performing multiple handovers. The initial care-of address of the source is noted CoA_1, the second CoA_2, the third CoA_3 and so on after further handovers. After the first handover, some receivers may still be in the process of joining the SSM channel (CoA_2,G) when the source performs a second handover and acquires CoA_3. To ensure proper multicast delivery to all receivers, the source must tunnel datagrams to both oAR_1 and oAR_2, the respective access routers associated with CoA_1 and CoA_2. Moreover, each time a handover is performed, a BU with the *SSM-Source Handover Notification* sub-option set to nCoA must be sent onto all the old trees that still have receivers. In addition, such a BU should be sent at a frequent rate (e.g. 1s) onto all the old trees to make sure that the receivers are aware of the new delivery tree to be joined.

4.3 Simulation results

In order to evaluate the gain obtained by using MSSMSv6 when compared to bi-directional tunnelling, we have carried out a large number of simulations with the well-known NS-2 simulator. We have implemented a light version of the PIM-SSM protocol and a complete version of the MSSMSv6 protocol in NS-2.

This study has lead to a publication in an international conference [15]. A footnote in this paper gives credit to the 6NET project. Details of the simulation methodology can be found in [15]. They are intentionally omitted in this report which primarily focuses on the results. However, note that we have performed a very large amount of simulations with different Internet-like topologies, simulating different group sizes and configurations (i.e. different positions of the receivers, the mobile sources, and many combinations for the handovers). The results presented here are significant with respect to all the simulations carried out, in the sense that the statistical error of the results presented in this document is very low.

Figures 7, 8, and 9 illustrate the main conclusion obtained from the simulations. We have measured the efficiency of MSSMSv6 compared to bi-directional tunnelling in terms of both delivery delay and delivery path length (in hops). Because we wanted to evaluate the performance of the MSSMSv6 protocol, we have presented the gain obtained with this solution when compared to bi-directional tunnelling via the home agent.

Figure 7 shows the average gain observed by receivers vs. the handoff distance *d* (i.e. the distance between the home agent and the new position of the source). For example, for a given handoff distance of 1, packets received when using MSSMSv6 have traveled along a path 15% shorter than packets received via the home agent. It can be seen on Figure 7 that both gains increase with the handoff distance, a direct and evident consequence of the considered protocols: MSSMSv6 builds shortest-path trees while packets traveling via the home agent have to go through *d* extra hops (i.e. the length of the tunnel) before "reaching" a shortest-path tree. The gain seems to reach a maximum of 45% for $d = 8$. Results for $d > 8$ are ignored because they represent less than 5% of all simulated handoffs and there is therefore not enough data to consider these results. It is however worth mentioning that the small amount of data analysed for $d > 8$ present an excellent performance.

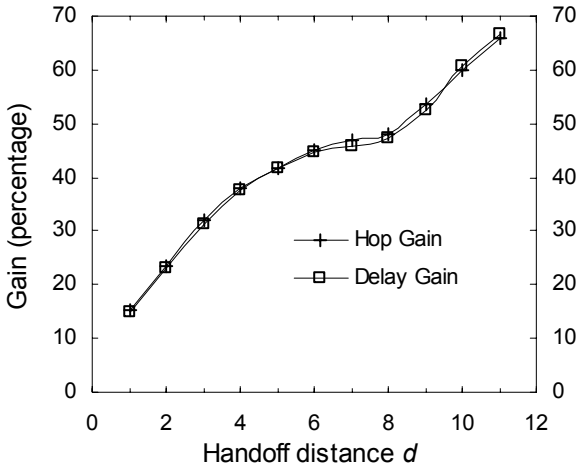

Figure 7. Average gain measured (MSSMSv6 vs. tunneling)

Figures 8 and 9 show the receivers gain estimation (in terms of hops) in the form of a survival function, i.e. it gives the amount of receivers that have a gain strictly superior to a given value. Figure 8 gives an average picture of the gain for all *d*. Figure 9 gives a more detailed measure of the gain for a number of values for *d*. For example and for $d = 3$, more than 80% of receivers observe a gain superior to 20%.

We have also measured that it takes less than one second for data to be received on (nCoA,G), and less than 2 seconds for the entire new tree to be constructed when MSSMSv6 is considered. In comparison, data is received via the home agent in less than 730 ms and in both cases (tunnelling and MSSMSv6) the data delivery disruption time was less than 370 ms (we did not consider the time required to complete the layer 2 and the layer 3 handoffs). It means that the MSSMSv6 protocol is only profitable if the mobile source remains at least more than 2 seconds in the visited network (a hypothesis which is more than likely plausible). The MSSMSv6 protocol therefore appears to be very profitable in terms of performance of multicast delivery with mobile SSM sources.

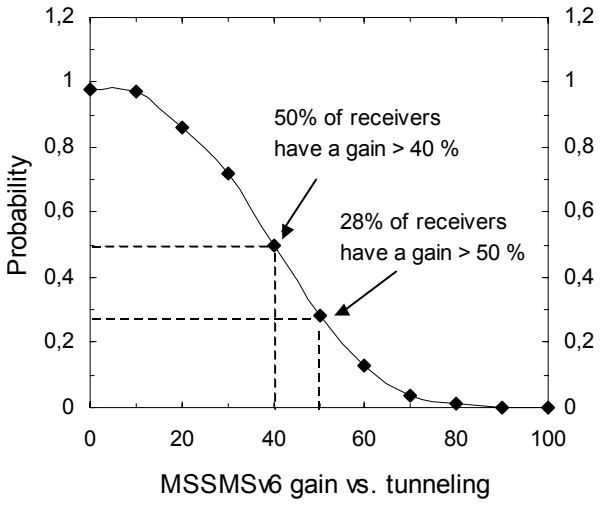

Figure 8. Average gain distribution across receivers

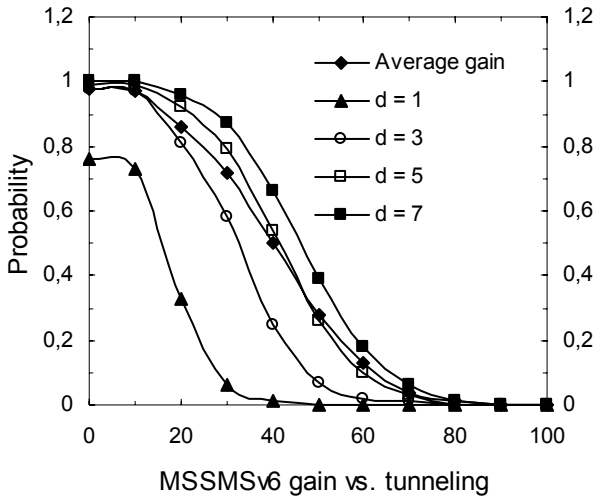

Figure 9. Gain distribution across receivers for different values of *d*

With bi-directional tunnelling, if a home agent serves a large number of mobile SSM sources, the consequence of having all trees rooted at the home agent network is that routers in the periphery of the home agent will have to support a large amount of forwarding states. In contrast, with MSSMSv6 trees are rooted at the current positions of the mobile sources and forwarding states are distributed among all the routers of the topology. This property is a positive *side-effect* of MSSMSv6, in the sense that it eliminates the aggregation problem but was not initially designed for that.

During the simulations we have therefore also measured the number of forwarding states in routers at a given distance *d* from the home agent for the two cases (bi-directional tunnelling and MSSMSv6). We then calculated, for each possible distance *d*, the ratio between the number of forwarding states for bi-directional tunnelling and the number of forwarding states for MSSMSv6. Figures 10 and 11 show this ratio for two different number of channels, respectively 30 and 150. It can be seen on Figure 10 that the neighbour-routers of the home agent (i.e. $d = 1$) have up to 9.5 times more forwarding states with bi-directional tunnelling than when compared to MSSMSv6. This ratio decreases with both the number of channels and the number of receivers. This can be understood as follows. When both the number of channels and receivers increase, the superposition of the source-rooted trees (MSSMSv6) is such that almost all routers (around the home agent or not) have an increasing amount of forwarding states. This is to be compared with the tunnel-based approach in which usually half the neighbour-routers have all forwarding states: the total number of forwarding states increases linearly with the number of channels (the number of receivers does not have an influence), while the growth for MSSMSv6 is faster and therefore the ratio reduces. An important point is that usually, with the tunnel-based solution, at least half the neighbour-routers of the home agent have all forwarding states (i.e. one state for each channel). In contrast with MSSMSv6, and over the entire data of our simulations, we have never found a neighbour-router that had all forwarding states.

This property is very interesting in the sense that, with MSSMSv6, routers that are close to a home agent are not overloaded by extra multicast forwarding states. This is particularly attractive if routers close to a home agent are not part of the same administrative domain (from a network management point of you).

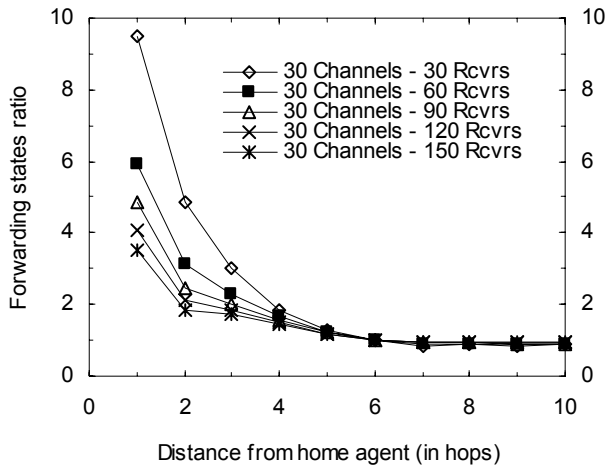

Figure 10. Forwarding states ratio vs. distance from HA

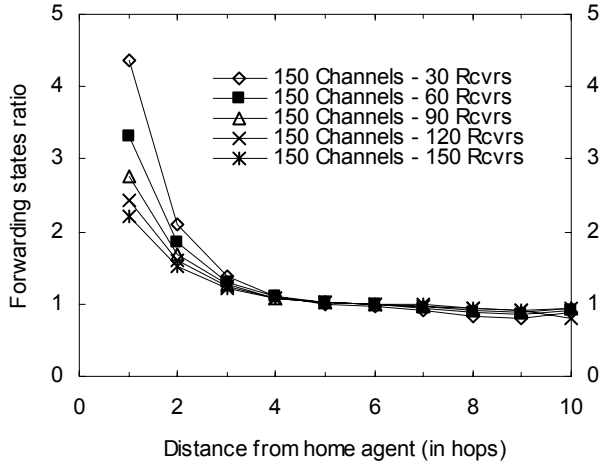

Figure 11. Forwarding states ratio vs. distance from HA

5 MLD proxy

5.1 Introduction and specifications

In the particular case of bi-directional tunnelling, the home agent must have the functionalities of a multicast router. These functionalities can be separated in three fundamental modules: receivers subscription, multicast tree grafting/pruning, and data forwarding. Receivers subscription is the ability to receive MLD messages, i.e. the ability for the home agent to learn that a receiver wants to subscribe (or unsubscribe) to a given multicast group (or channel for SSM). Multicast tree grafting/pruning is the capability to join (or leave) the multicast delivery tree, i.e. the core operation of multicast routing. Finally, data forwarding is the aptitude to forward multicast data to interested receivers. For example, the PIM-SM multicast routing daemon available on the FreeBSD operating system complies with all these requirements. For IP version 6, this is specifically possible because the kernel of this operating system is capable of forwarding IPv6 multicast datagrams. In other words, FreeBSD implements an IP version 6 multicast forwarding cache (MFC). On the other hand, the current version of the Linux operating system does not support an IPv6 MFC.

In a strict sense, a home agent should be able to satisfy all of the three functionalities of a multicast router. However in practice, and especially in the particular case of bi-directional tunnelling, it is possible to separate these functionalities. The home agent does not indeed strictly require to perform the multicast routing operation. Removing this functionality from the home agent is of high interest for two reasons. First, the home agent can focus on its main task: forwarding (unicast and multicast) data to the current locations of the mobile nodes it serves. The second reason is mainly a deployment issue: in the first phase of the deployment of Mobile IPv6 services, network operators may be reluctant to integrate the multicast routing operation in their home agents. The main motivation for that would be the ease of administration.

This possibility leaves the home agent with two tasks. Receivers subscription and data forwarding. In the literature, these functionalities are known as *MLD proxying*. The idea had already been presented in [5] and it is also considered at the IETF within the MAGMA working group [16]. An MLD proxy must of course be associated with a router capable of multicast routing, but the MLD router itself is simply seen as a multicast receiver by the multicast router. In short, the MLD proxy subscribes to multicast groups on behalf of the mobile nodes it serves. To do so, it simply sends MLD messages on the link of the network to which it belongs. The multicast router subsequently joins the corresponding multicast trees and forwards the data on the network where the MLD proxy resides. The final task of the MLD proxy is then to forward the multicast flows to the mobile nodes. To do so, it sends the multicast datagrams in the tunnels towards the mobile nodes. A mobile node can also wish to send datagrams to a given group. To do so, it uses its bi-directional tunnel to send multicast datagrams to the home agent, which removes the unicast header and forwards the multicast datagrams on the home network. It is then the task of the multicast router to forward this multicast flow onto the multicast tree. The operation of an MDL proxy is illustrated by Figure 12.

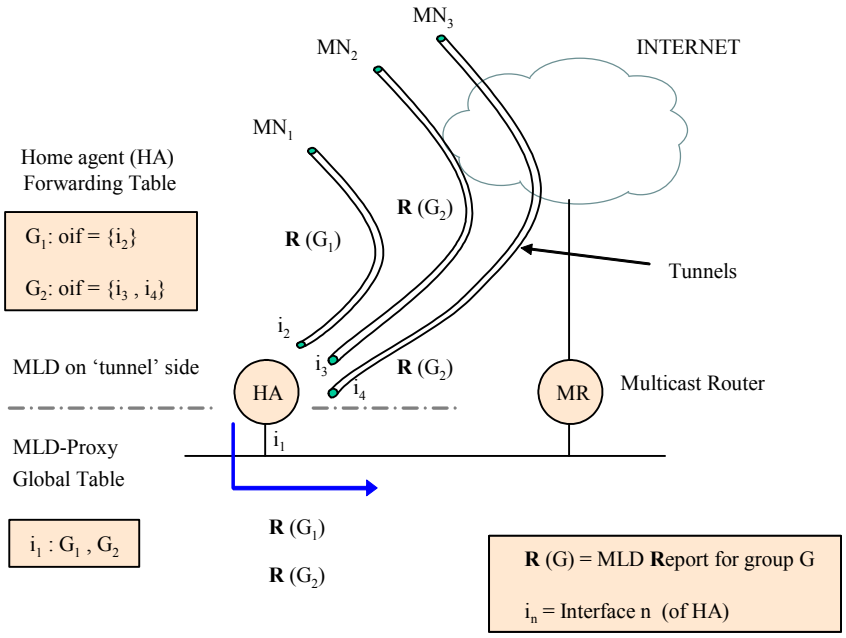

Figure 12. MLD proxy operation

On Figure 12, a home agent is also an MLD proxy. Three mobile nodes have registered with the home agent. One mobile node (MN_1) has sent an *MLD report* message for the multicast group G_1. The two other mobile nodes (MN_2 and MN_3) have sent *MLD report* messages for the multicast group G_2. The home agent, when receiving the reports, subscribes to the corresponding groups. To do so, it sends its own *MLD report* messages onto the network link. The multicast router subsequently joins the two groups, and starts forwarding the multicast flows onto the network. When the home agent receives a datagram destined for a multicast group for which at least one of the mobile it serves is subscribed, it forwards the multicast datagram to all the mobile nodes which are subscribed to this group. To do so, a copy of the datagram is sent to each interested mobile node via the corresponding Mobile IPv6 tunnel. The MLD proxy must therefore maintain a list of multicast groups to which it is subscribed, and a forwarding table that keeps a record of the outgoing interfaces (oif on Figure 12) to which datagrams must be forwarded. Moreover, and in order to be compatible with the multicast model, the home agent must forward multicast datagrams sent by the mobile nodes. It is the task of the multicast router to forward (or not) these datagrams onto the appropriate multicast trees.

5.2 MLD proxy implementation

In the context of the 6NET project, ULP has implemented a first version of an MLD proxy. This proxy will soon be made available to other partners within 6NET, as it is still in a development / debugging phase. This first version has been developed on the Linux operating system, with the MIPL 1.0 implementation of the Mobile IPv6 protocol. This MLD proxy has been used in the tests that are presented in Chapter 6.

The particularity of this MLD proxy is that it forwards datagrams in *user land*, i.e. the MLD proxy itself is responsible for forwarding the multicast datagrams. The main reason for that is that the current version of the Linux operating system does not implement an IPv6 MFC. We plan to do a porting on FreeBSD, but with the current state of the KAME implementation of Mobile IPv6, it is unfortunately not possible to use it in the context of this deliverable. With the KAME implementation, a tunnel towards a mobile node is indeed not

represented by an active interface by the operating system. This therefore prevents the MLD proxy from being able to use the tunnels. We are having contacts with the KAME development team to try to overcome this situation. They seem to be quite interested to get some feedback from our experience with the MLD proxy implementation on Linux (using the MIPL 1.0 Mobile IPv6 implementation). Moreover, and as shown in Chapter 6, the current state of the KAME implementation does not permit to use bi-directional tunnelling for multicast data. For all these reasons, the implementation of the MLD proxy is currently limited to the Linux operating system. The details and experiences gained from this development will be presented in the second version of this deliverable, which is due at month 36 of the project (December 2004).

6 Results of experiments with MIPv6 implementations

In this section, we present the initial results of the experiments carried out with implementations of Mobile IPv6. In December 2003, only two implementations of Mobile IPv6 are (partially) compliant with the latest Internet Draft (i.e. draft 24) [3].

The MIPL 1.0 implementation (see http://www.mipl.mediapoli.com) is available on the Linux operating system, and requires a 2.4.22 kernel version. This Linux kernel version implements the version 2 of the MLD protocol, which is defined in [17]. Note that this kernel version has a number of bugs that affect the operation of MLD. These bugs can be removed by applying a specific patch (developed by David Stevens from IBM).

The second Mobile IPv6 implementation that has been used for our experiments is available via the KAME project (see http://www.kame.net). The KAME implementation requires a BSD-like operating system. We have used the latest stable version of the FreeBSD operating system, i.e. FreeBSD 4.9, and we have used the KAME SNAP kit 20031124, i.e. of November the 24th. FreeBSD 4.9 also implements the version 2 of the MLD protocol.

In the following sub-sections, we will use the words MIPL and KAME to designate the two implementations of Mobile IPv6.

6.1 Remote subscription

The first set of tests was used to check if remote subscription was supported by MIPL and KAME. We expected this functionality to be supported by both implementations, mainly because it follows the normal behaviour of any stationary host. The results of the experiments carried out with both KAME and MIPL have indeed shown that remote subscription is supported by both implementations. A mobile node can thus receive and send multicast data via the visited network. In particular, the mobile node will use its care-of address when sending multicast datagrams.

On the other hand, the operation of MLD is affected in the presence of mobility and this can result in a very long disruption time in the delivery of multicast data to a mobile receiver. Consequently, we have also shown that this problem affects both KAME and MIPL. The MLD protocol has indeed not been specified in order to support the mobility of nodes. This does not prevent a mobile node to subscribe to multicast groups, but it introduces a serious inefficiency when a mobile node performs a handoff. With MLD, when a node wants to subscribe to a specific multicast group it sends an *MLD Report* message. Initially, this message is sent when the node wants to subscribe to the multicast group, e.g. when a multicast application is started. Subsequent reports, which are sent in order to refresh the subscription, are sent upon reception of an *MLD query* message, which is sent by multicast routers at periodical intervals. In the MLDv2 specifications [17], this period is (by default) set to 125 seconds. As a direct consequence, report messages will also be sent with a similar frequency.

Now consider a mobile node which is attached to a network N_1. This mobile node subscribes to the multicast group G, and soon starts to receive multicast datagrams destined for G. At some point in time, the mobile node performs a handoff and get attached to the network N_2. With the current specifications of the MLD protocol, this handoff will not trigger the immediate sending of an *MLD report* message on N_2. In the worst case, the mobile node attaches to N_2 a few milliseconds after the sending of the *MLD Query* message on N_2. This means that the mobile node will not send an *MLD Report* before receiving the next query message, i.e. it can wait up to almost 125 seconds. During this period, the mobile node will not receive any data for G (if there is no other member of G in N_2). On average, the disruption period is theoretically equal to 125/2 = 62.5 seconds. This problem has been observed with a small testbed as presented by Figure 13.

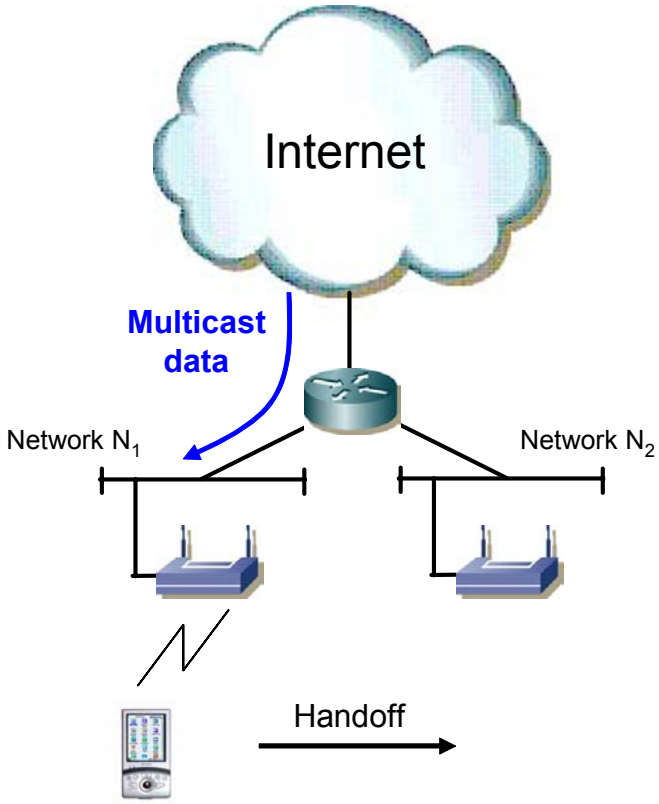

Figure 13. Problem after handoff with remote subscription

It is important to note that, whereas the value of the period can be changed in order to better handle mobile receivers, reducing this value will induce an increase in the overhead caused by the sending of *MLD Report* messages, which are sent upon reception of an *MLD Query* message. With IPv6, any node will respond to a *MLD Query* message by sending at least one *MLD Report* message in order to advertise its solicited multicast address, i.e. the address used by a node to find the MAC address of a correspondent. Therefore, if the period at which *MLD Query* messages are sent is too low, it will trigger the sending of a non-negligible amount of data.

6.2 Bi-directional tunnelling

In a second set of experiments, we have tried to use bi-directional tunnelling with both KAME and MIPL. It quickly appeared that the current version of KAME does not support this functionality. In particular, it is not possible to specify that a multicast application must use the bi-directional tunnel to send and receive multicast data. We have contacts with the KAME development team to try to overcome this situation.

In contrast to KAME, the MIPL implementation allows a multicast application to specify that it must use the bi-directional tunnel to send and receive multicast data. With MIPL, this is indeed possible because the tunnel to the home agent is represented as an active network interface by the operating system. Any multicast application that allows to specify the interface to be used for multicast operations can use the tunnel. On another hand, the main drawback with Linux is that it does not implement an IPv6 multicast forwarding cache. The main consequence is that the PIM routing daemon is not available in its IPv6 version on Linux.

Therefore, the only alternative was to develop an IPv6 MLD proxy, which has been presented in the previous chapter.

We have successfully validated our MLD proxy, which extends MIPL with bi-directional tunnelling capabilities, in the sense that MIPL itself cannot forward multicast data to mobile receivers. Of particular interest, we have validated that, with a scenario similar to the one described in the previous sub-section, a mobile node can receive multicast data almost immediately after a handoff. The disruption time is equal to the time it takes for the *binding update* message to reach the home agent. Moreover, we have shown that a mobile node, while away from its home network, can receive multicast data that is local to its home network.

6.3 Summary

The results obtained during our various experiments with both KAME and MIPL are summarised in Table 2.

	MIPL	KAME
Remote subscription	✓	✓
Bi-directional tunnelling	✓ (with MLD Proxy)	✘
Optimized MLD support with remote subscription	✘	✘

Table 2. Summary of available features in MIPL and KAME

7 Conclusions

At the time of writing, the integration of multicast and mobility is clearly still in an experimental stage. In particular, the KAME and MIPL implementations of the latest version of Mobile IPv6 do not natively support bi-directional tunnelling of multicast data, nor do they provide optimizations to overcome the problem introduced by MLD with remote subscription.

However, ULP expects to share its experience by the development and the distribution of our MLD proxy, which can promote the use of multicast communications with mobile devices. To fulfil this objective, we are willing to accelerate the development of the MLD proxy in order to soon be able to make it widely available to the IPv6 networking community. Our aim is to provide a flexible tool that is easy to install and configure, and which exhibits a satisfactory performance in terms of data forwarding.

In practise, we believe that the two solutions advocated by the MIP6 IETF working group are complimentary. They could for example be used "in parallel", in the sense that a mobile node could benefit from the advantages of both solutions: better performance of data delivery with remote subscription, and smaller disruption time of data delivery during a handoff with bi-directional tunnelling. In the second version of this deliverable, we will try to present different scenarios to show how the two solutions can be combined.

Moreover, we believe that the integration of multicast and mobility will be accelerated with the future deployment of native multicast routing. This deployment is already being accelerated in order to provide multimedia services efficiently. For example, some ADSL network operators in France start deploying IPv4 multicast in order to propose an Internet TV service to their customers, i.e. traditional TV channels will be available via the ADSL network. In the mean time, the transition to IP version 6 is an adequate moment to deploy multicast routing at a very large scale, thus turning the future next generation Internet into a next generation *multimedia-ready* Internet. This would allow the rapid development of multicast services for mobile nodes, such as Internet TV and radio. One can imagine a next generation mobile phone that is also a TV and radio receiver. Such a revolution could trigger a large attraction towards the next generation of mobile phone technologies. Such services could be the "killer applications" that would accelerate the deployment of IP version 6 along with the next generation of mobile phone networks.

References

[1] G. Xylomenos, and G. Polyzos, "**IP Multicast for Mobile Hosts**," IEEE Communications Magazine, vol. 35, no. 1, pp. 54-58, Jan. 1997.

[2] C. Bettstetter, A. Riedl, and G. Geßler, "**Interoperation of Mobile IPv6 and Protocol Independent Multicast Dense Mode**," *In Proceedings* of Workshop on Wireless Networks and Mobile Computing (held in conjunction with ICPP2000, 29th International Conference on Parallel Processing), Toronto, Canada, August 2000.

[3] D. Johnson, C. Perkins, and J. Arkko, "**Mobility Support in IPv6**," Internet Draft, draft-ietf-mobileip-ipv6-24.txt, June 2003.

[4] C. Jelger, and T. Noel, "**An Analysis of Multicast Delivery with Mobile Receivers**," *In Proceedings of* IEEE PIMRC'03, Beijing, China, Sept. 2003.

[5] C. Jelger, and T. Noel, "**Multicast for Mobile Hosts in IP Networks: Progress and Challenges**," IEEE Wireless Communications Magazine, vol. 9, no. 5, pp. 58-64, Oct. 2002.

[6] T. Harrison, C. Williamson, W. Mackrell, and R. Bunt, "**Mobile Multicast (MoM) Protocol: Multicast Support for Mobile Hosts**," *In Proceedings of* ACM Mobicom'97, pp. 151-160, Budapest, Hungary, Sept. 1997.

[7] C. Lin, and K. Wang, "**Mobile Multicast Support in IP Networks**," *In Proceedings of* Infocom'2000, vol. 3, pp. 1664-1672, Tel Aviv, Israel, March 2000.

[8] Y. Suh, H. Shin, and D. Kwon, "**An Efficient Multicast Routing Protocol in Wireless Mobile Networks**," Wireless Networks, vol.7, no.5, pp.443-453, Sept. 2001.

[9] C. Jelger, and T. Noel, "**Supporting Mobile SSM Sources for IPv6**," *In Proceedings of* IEEE Globecom'02, Taipei, Taiwan, Nov. 2002.

[10] M. Handley, C. Perkins, and E. Whelan, "**Session Announcement Protocol**," IETF RFC 2974, Oct. 2000.

[11] M. Handley, and V. Jacobson, "**Session Description Protocol**," IETF RFC 2327, April 1998.

[12] B. Fenner, M. Handley, H. Holbrook, and I. Kouvelas, "**Protocol Independent Multicast - Sparse Mode (PIM-SM) : Protocol Specification (Revised)**," Internet Draft, draft-ietf-pim-sm-v2-new-08.txt, Oct. 2003, work in progress.

[13] R. Koodli (Editor), "**Fast Handovers for Mobile IPv6**," Internet Draft, draft-ietf-mipshop-fast-mipv6-00.txt, Oct. 2003, work in progress.

[14] B. Fenner, B. Haberman, H. Holbrook, and I. Kouvelas, "**Multicast Source Notification of Interest Protocol**," Internet Draft, draft-ietf-idmr-msnip-04.txt, June 2003, work in progress.

[15] C. Jelger, and T. Noel, "**Performance Evaluation of Multicast Transmissions with Mobile Sources**," *In Proceedings of* IEEE ICON'03, Sydney, Australia, Sept.-Oct. 2003.

[16] B. Fenner, H. He, B. Haberman, and H. Sandick, "**IGMP/MLD-based Multicast Forwarding ("IGMP/MLD Proxying")**," Internet Draft, draft-ietf-magma-igmp-proxy-04.txt, Sept. 2003, work in progress.

[17] R. Vida, and L. Costa (Editors), "**Multicast Listener Discovery Version 2 (MLDv2) for IPv6**," Internet Draft, draft-vida-mld-v2-08.txt, work in progress.

Abbreviations / Glossary

ADSL	Asymmetric Digital Subscriber Line
BU	Binding Update
CoA	Care-of Address
HA	Home Agent
IETF	Internet Engineering Task Force
ISP	Internet Service Provider
LAN	Local Area Network
MAC	Medium Access Control
MFC	Multicast Forwarding Cache
MIPv6	Mobile IP version 6
MLD	Multicast Listener Discovery
MMA	Multicast by Multicast Agent
MN	Mobile Node
MoM	Mobile Multicast
MSSMSv6	Mobile Source-Specific Multicast Sources for IPv6
PIM-SM	Protocol Independent Multicast – Sparse Mode
RBMoM	Range-Based Mobile Multicast
SSM	Source-Specific Multicast
ULP	Université Louis Pasteur
VPN	Virtual Private Network