TD(05)012

2nd COST290 MC Meeting
Colmar, France,3-4 Feb 2005

MoSReM: Mobile Multicast Protocol Using Dynamic Branching Node-Based Tree

Khalid Al-Begain and Yewen Cao
School of Computing
University of Glamorgan
Pontypridd, Cardiff, Wales
CF37, 1DL, UK

{kbegain, ycao}@glam.ac.uk

Abstract

Providing multicast support for mobile hosts in an IP inter-network must face many challenging problems, such as compatibility with existing multicast protocols (implicitly assume static hosts), mobility management, scalability, etc. This paper, we propose a new mobile multicast protocol, called Mobile Scalable Recursive Multicast (MoSReM). MoSReM is based on the concept of dynamic ranching node-based tree (DBT), setting up multicast tree gradually and dynamically. In MoSReM, only branching nodes router (BNRs) keep the multicast state about their next BNRs and mobility information (if any) about destinations, and the process of join/leave and mobility management of members of a multicast session is always carried out locally. MoSReM has many positive features such as fixed size control message, being scalable and low join/leave latency.

Keywords

Multicast, Mobile Multicast, Multicast Protocols

Working Group 3

I. INTRODUCTION

Recently, more attention has been drawn to providing multicast service for mobile nodes (MNs) in an IP internet-network [1-3]. However, there are several challenging issues regarding mobile multicasting. First, IETF Mobile IP [4] concentrates on unicast delivery to mobile host (or MNs). In order to efficiently support multicast for mobile host, additional mechanisms must be added within or on top of Mobile IP. Secondly, the addition of mobility to the host group model implies that both dynamic group membership and dynamic member location are needed to be tackled. Thirdly, existing multicast routing protocols, such as DVMRP [5], MOSPF [6] and PIM [7], implicitly assume static hosts when setting up a multicast tree.

The current version of Mobile IP [4] proposes two approaches to support mobile multicast, i.e., remote subscription (RS) and bi-directional tunneling (BT). In remote subscription, mobile hosts are needed to re-subscribe to its desired multicast groups while at a foreign network. The remote subscription approach offers shortest routes for delivery of multicast datagram to mobile hosts and suitable for applications where the mobile host spends a relatively long time at each foreign network (compared to the join and graft latencies). However, the approach implicitly assumes that mobile hosts are only recipients of multicast messages or that they have a co-located address on the foreign network. If the mobile host sends a multicast datagram with its home address as the source, the incoming interface check of most multicast routing algorithms may discard datagram intended for the multicast group. Finally, the approach assumes the existence of a multicast router at the visited network, an assumption that may not always hold in an IP inter-network. Without such a multicast router, multicast message delivery can be achieved only by using some form of tunneling.

With bidirectional tunneling, mobile hosts send and receive all multicast datagram by way of their home net-work, using unicast Mobile IP tunnels from their Home Agents. This approach handles source mobility as well as recipient mobility, and in fact hides host mobility from all other members of the group. The drawbacks, however, are two-fold. First, the routing path for multicast delivery can be far from optimal, i.e., triangular routing problem. Second, the approach offers limited scalability. Home agents with multiple mobile host group members away from home must replicate and deliver tunneled multicast datagram to each of them, regardless of at which foreign networks they reside.

In this paper, following up the success of our previously proposed multicast protocol SReM [8], we propose a new scalable mobile multicast protocol, called Mobile Scalable Recursive Multicast (MoSReM). The basic idea of MoSReM is to use a novel multicast mechanism, i.e., the concept of dynamic ranching node-based tree (DBT) to set up and update multicast tree gradually, dynamically and locally. In MoSReM, only branching nodes router (BNRs) keep the multicast state about their next BNRs and mobility information (if any) about destinations, and the process of join/leave and mobility management of members of a multicast session is always carried out locally. MoSReM aims to address both scalability problem and join /leave latency in mobile IP networks.

II. RELATED WORKS

2.1. EXPRESS, REUNITE, Xcast and SEM

EXPRESS [9] have been proposed recently to tackle the address allocation and the sender access control problems. EPRESS proposes an extension to IP multicast to support the channel model of multicast and describes a specific realization called EXPlicitly REquested Single-Source (EXPRESS) multicast. In these schemes, there is a special node (sender or core) associated with each group and the group is identified by a two-tuple <special node's unicast IP address, class D multicast address>. In this model, a multi-cast channel has exactly one explicitly designated source, and zero or more channel subscribers. A single protocol supports both channel subscription and efficient collection of channel information such as subscriber count. EXPRESS addressed problems such as lacks a basis for charging, lacks access control, etc.

The idea of REUNITE [10] is to use recursive unicast trees to implement multicast service. REUNITE does not use class D IP addresses. Instead, both group identification and data forwarding are based on unicast IP addresses. Compared with existing IP multicast protocols, REUNITE has several unique properties. First, only routers that are acting as multicast tree branching points for a group need to keep the multicast forwarding state of the group. All other non-branching-point routers simply forward data packets by unicast routing. In addition, REUNITE can be incrementally deployed in the sense that it works even if only a subset of the routers implement the protocol. Furthermore, REUNITE supports load balancing and graceful degradation such that when a router does not have resources (forwarding table entry, buffer space, processing power) to support additional multicast groups, the branching can be automatically migrated to other less loaded routers. Finally, sender access control can be easily supported in REUNITE.

Explicit Multicast (Xcast) [11] is a newly proposed multicast scheme to support a very large number of small multicast groups. Xcast+ [12] is an enhanced scheme for the support of receiver initiated join in explicit multicast which complements the existing Xcast. This is achieved by adding an IGMP join at receiver side and sending the join request through source-specific join to the sender, and then by explicitly encoding the list of addresses of the multicast routers, instead of receiver addresses. Xcast+ encoding of the destination list in IPv4 and IPv6 are the same as Xcast. Whereas Xcast can support a very large number of small multicast groups, Xcast+ can support a very large number of medium size multicast groups. The main advantage of Xcast or Xcast+ is that both of them eliminate the need for forwarding states. However, there is still the scalability problem.

SEM [13] uses an efficient method to construct the multicast tree and deliver multicast packets. In order to construct the multicast tree, this protocol uses the same mechanism as Xcast+[12]. For the delivery of multicast packets it uses the mechanism of branching routers similar to the mechanism used in REUNITE [10]. The simple explicit multicast (SEM) [13], the source uses unicast encoding for multicast packets and sends them to its next hop branching node routers. Each branching node router acts as a source and packets travel from a branching node router to another. In SEM, the mechanism of using Xcast is introduced to build a whole multicast tree in advance and delivery of multicast packets is done between branching nodes. However, SEM has still the scalability problem, as all of destinations addresses (of receivers for Xcast, or designated routers for Xcast+) are needed to be encoded in the branching message. In SEM, another major problem is the join/leave latency.

2.2. SReM

In SReM [8], we proposed the idea of using a dynamic branching node-based multicast tree (DBT) to deliver packets. In SReM, there are two types of signaling messages, i.e., join / leave signaling messages (JS/JL) and branching node messages (BNMs). Each receiver who wants to join /leave a multicast group (*G*) sends a join /leave message (JoinM/LeavM) to its local multicast router (LMR) by IGMPv.3, and this LMR then sends a registration request message (RqM) to the multicast source (*S*) on behalf of this receiver. In response to the receiving of the RqM, the source will send a registration replying message (RpM) to confirm the registration. In the initial stage of the multicast tree setup, the multicast source will be in charge of searching for the first branching node router (BNR) by using of a pair of BNMs upon receiving these registrations. Following that both the multicast source and BNRs will perform this function (searching for BNRs). As a result, a dynamic branching node-based multicast tree (DBT) will then be created. Each of BNRs in DBT has a multicast forward table (MFT) with the multicast tree identity (MTI) of (*S*,*G*), which includes the address of its previous BNR as well as a list of its next branching nodes' addresses. Using the BDT, SReM deliver multicast via BNRs by unicast.

In Table 1, a comparison between SReM and others is made. As a result of the cost analysis, both SReM and SEM have some control plane against the Xcast and Xcast+, their cost of packet header processing is minimized. SReM has all of the advantages of SEM. Furthermore, compared to SEM, SReM has less control overhead and lower join and leave latency.

III. MoSReM DETEALS

3.1 . Dynamic Branching-node based Tree (DBT)

Table 1 Cost analysis of SReM, Xcast, Xcast+ and SEM

Evaluations\ Protocols	Xcast	Xcast+	SEM	SReM
Multicast address allocation	none	medium	medium	medium
Multicast routing state management	none	low	low	low
Control overhead	none	medium	medium	low
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

As in SReM [8], MoSReM uses a dynamic branching node-based tree (DBT) to set up the multicast tree, where multicast tree is processed gradually and dynamically between branching nodes as the joining of members of a multicast group, not in advance. The DBT is set up by the use of a pair of BNMs, i.e., enquiring BNM (eBNM) and replying BNM (eBNM). BNMs carry the information such as the previous BNR, one next BNR, and new LMR, as detailed in the following section (see Section 3.2).

The key point of building DBT is that each intermediate multicast-enabled router (IMR) is aware of whether or not it is one of BNRs for the source or its next BNR. In MoSReM, IMRs have three separate functions: i) identify existence of new BNR in the route to destinations; ii) know itself the sate of being a BNR or not; iii) inform one level upper (i.e. previous) BNR of its state. These functions are carried out as follows.

On receiving a new join signaling (JS) message (say, from a receiver *a*), one BNR (can be the multicast source at the initial stage) in the DBT, say *N*, will check the route (where *a*'s JS message is coming) against all existing routes in its MFT. If it does not match any one, then *N* will be aware of this route is a new branch of DBT and the address of *a*'s LMR will then be added to the *N*'s MFT. Otherwise, the next hop address for the new JS message is equal to one of entries in *N*'s MFT, and then the BNR will start a new branching node search by sending enquiring BNM (eBNM) to the next IMR with an address couple of an existing BNR (say *M*) and the *a*'s LMR.

When the IMR, say *L*, receives the eBNM from the upper level (previous) BNR (at the moment, *N*), it will check the next-hop address for both IP address in the message. If these are the same, then forward to the next IMR. If not, then this IMR is a new BNR and it sends a replying BNM (rBNM) to the upper level (previous) BNR, i.e., *N*.

When *N* receives this rBNM, it will replace the entries for existing BNR (*M*) and the new LMR (*a*'s LMR) with the address of the new BNR (*N*).

Based on this mechanism, a DBT will be built gradually. In order to help to understand the set-up process of DBT, three examples are given below. This first one is about the initial process of DBT, shown in Fig 1. At the start, a receiver or destination, say *a*, will trigger off the process of a multicast session towards the source, say, *S*, by sending a join signaling (JS) message. A source specific group, named (*S*,*G*), then is formed. This JS message will arrive at *S* and *S* then creates an entry for *a* in its MFT, which includes MTI and the IP address of *a* , denoted by MTI | IP_*a*. And immediately, *S* can send multicast packets to *a* along the route, named route I, by which *a*'s JS message is coming. This route I is the first part of DBT.

As joining of a second member *b* of *G*, its JS message will also arrive at *S*, along a route, named route II. Upon receiving this message, *S* creates an entry for *b* in its MFT, i.e., MTI | IP_*b*. Route II may be quite different from route I (Case 1) or partly share the route II (Case 2). Clearly, there is one branching node for the Case 2 and no one for the Case 1. To explore which of these two cases happening, the source *S*, will send an eBNM with a list of the destinations of IP_*a* and IP_*b*, shown in Fig 1c. This process is called explicit searching of branching nodes. An intermediate multicast-enabled router (IMR, i.e., one of routers along route I and II, from the source to destinations), say router N_1, will receive this eBNM. Router N_1 will process this message and be aware of whether or not it is a branching node for the destinations of IP_*a* and IP_*b*.

If N_1 is not a BNR, it will just pass eBNM to a next IMR and no any processing will be done, meaning that route II is partly sharing with route I. Otherwise, if a branching node is found, say N_2, will send a rBNM to its previous BNR, where at the moment, it is the multicast source *S*. Meanwhile, an entry is created in its MFT, i.e., MTI | IP_*a* & IP_*b*. Upon receiving this message, the previous BNR will update its MFT, i.e., MTI | IP_N_2 instead of MTI | IP_*a* & IP_*b*. This means that a DBT for *a* and *b* is now set up.

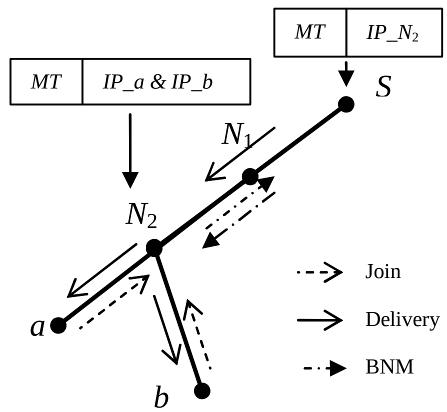


Fig 1. Initial process of multicasting in MoSReM.

The second example is about the joining of a new member of *G*, say *x*, which is joining at an IMR between two BNR, as figured in Fig 2.

BNR N_1 will send an eBNM with the destination of IP_N_3 and IP_*x* towards its next BNR (i.e.N_3). One of IMR (say N_2) will check out that it is a new BNR. And then an MFT will be created at N_2 and a rBNM will be sent back to the BNR N_1. Upon receiving this message, BNR N_1 will update its MFT, i.e. , using MTI | IP_c & IP_ N_2 instead of MTI | IP_c & IP_ N_3 .

Now, turn to consider an example of the process of leaving of members in MoSReM. Let us use the Fig1 one more. Assume that *x* decide to leave the multicast group. Node *x* will send a leave-signaling (LS) message to N_2. Upon receiving this message, the entry for *c* in the N_2's MFT will be removed and the delivery of packets to *c* is then terminated. Meanwhile, N_2 will be ware of itself no longer being a BNR in this example and N_3 (instead of itself) being the new BNR. N_2 will forward the LS message to N_1 to inform of itself (N_2) no longer being a BNR and N_3 (instead of itself) being the new BNR. Upon receiving LS message, N_1 will update its MFT, i.e., using MTI | IP_c & IP_N_3 instead of MTI | IP_c & IP_ N_2. Note that, no any affect is incurred, including MFTs at other BNRs and behavior of BNRs delivering packets in the rest of the existing DBT.

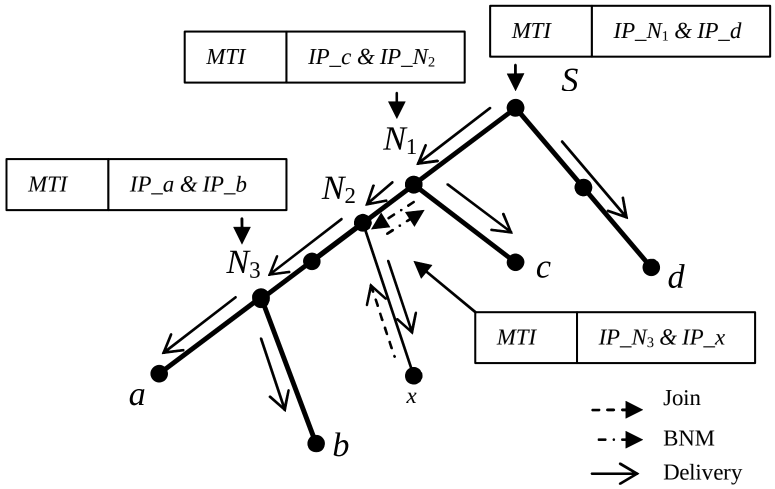


Fig2. A new member is joining in MoSReM

3.2 . Signalling messages in MoSReM

A. Branching node messages

In MoSReM, enquiring branching messages (eBNM) should look like [src= IP_branching node | group address = (S,G) | dest= IP_new router, IP_next branching node| previous branching node], where a group or MTI is identified by (S, G). And, similarly but simply, replying branching messages (rBNM) look like [src= IP_ new branching node | group address =(S, G) | dest = previous branching node].

In MoSReM, sending BNMs could be initiated at the source or any one of BNRs to search for new BNRs. However, BNRs are usually responsible for searching for new BNRs as the joining of new members, unless a new member joins the multicast session via an IMRs which is between the source and its BNRs. This means that the multicast tree is built gradually and locally and no remarkable joining and leaving latency is incurred. This is quite different from the way in SEM, where the multicast tree is built in advance based on all of addresses of destinations involved in the multicast group and needs to be rebuilt whatever any new joining and leaving happens.

B. Join / Leave Signaling (JS/LS) messages

In MoSReM, IGMPv.3 is used to initiate the procedure of multicast. Any end user or receiver which intends to join / leave a multicast group sends a join / leave message (JoinM/LeavM) to the local multicast router (LMR).

On behalf of their attached multicast receivers, LMRs send registration messages to the multicast source. These messages are: registration request messages (RqMs) and registration reply messages (RpMs). In MoSReM, RqMs and RqMs use a similar format as IGMP messages. However, there are two changes made. First, the Max Resp Time field in IGMP messages is substituted by a flag filed in MoSReM registration messages, where at the moment, only one bit in the flag field, called 'S' bit, is defined. Secondly, another address field is added to payload of the message. This field is filled with the address of a receiver.

C. Roaming Control Message

In MoSreM, in order to support receivers mobility, two controlling message are introduced, i.e., Roaming-in controlling message (Rm_IN) and Roaming-out controlling message (Rm_OUT). These two messages are used to inform LMRs and BNRs of update of multicast destination tables (MDTs) and MFTs, respectively, or in some scenario, to trigger an update of DBT.

In MoSReM, we assume that there are Mobile IP-based mobility protocols (micro-, macro-, or both) to implement local and wide-area movement, such as Columbia Cellular IP, HAWAII, and Hierarchical IP (HIP) [14]. Whenever detecting its movement (from one LMR covered area or domain to another LMR covered area or domain), a mobile receiver, say *x*, will send an Rm_OUT to its LMR, depending on LMR's status (i.e. the state of MDT), the LMR update its MDT and will then decide if it need to forward Rm_OUT towards its up-level BNR: 'Yes, Forward' if the *x*'s MDT is empty, 'Not Forward' otherwise. If the later happens, an update process of DBT is then triggered. There are two ways to implement this DBT update as follows.

First, use the same process as a receiver leaves a multicast group. In this way, Rm_OUT messages will terminate at LMRs, i.e., upon receiving a Rm_OUT, the LMR will send a leave message RqM towards the LMR's up-level BNR to trigger a DBT update process. For this, a new bit R in RqM is needed to set, i.e., 'R=1' means this is a RqM regarding roaming and it will not be forwarded to the source, and 'R=0' means that it is the RqM regarding leaving. One of advantages of using this way is: the control message traffic due to receivers' roaming is tremendously reduced because roaming messages never traverse within the multicast tree. Another advantage is that the Rm_OUT messages may use the similar format as JoinM/LeavM with minor modification. The disadvantage of this way is that a modification with respect to join/leave and registration messages is needed for the compatibility between MoSReM and its original version SReM.

The second way is straightforward: the LMR just forwards the Rm_OUT towards its up-level BNR. Upon receiving the Rm_OUT, the up-level BNR updates its MFT and decide if it is still a BNR or not. If yes, this up-level BNR will terminate forwarding the Rm_OUT and there is no more impact on remaining DBT. Otherwise, the Rm_OUT will be forwarded to the next up-level BNR, say Y, which start a DBT update process just like leave process in SReM. Note that forwarding of the Rm_OUT message will be terminated at Y. There are also pros and cons of this implementation. The advantage is that it is fully compatible with SReM and there is no need to modify the join/leave and registration messages. The disadvantage is that it requires a fully new message format to be used, might quite different from JoinM/LeavM or RqM. Using new format messages result in more complexity in roaming process, esp., at the time when leaving and roaming process coincide. In this case, LMRs must be able to decide the priority: to deal with leaving process first and then roaming process.

When entering into a new domain (a multicast area covered by a new LMR), a mobile receiver will send an Rm_IN message. The Rm_IN message will result in MDT update and may trigger a DBT update. Upon receiving the Rm_IN, the new LMR will update its MDT or set up a MDT (if no entry yet). Depending on the status of the new LMR's MDT, a DBT update may be triggered: DBT updating is needed if the newly incoming mobile receiver is the first member of multicast group at the new LMR, no DBT updating otherwise. There are also two ways to carry on this DBT update.

The first way is that the new LMR sends a modified RqM message towards its up-level BNR on behalf of the newly incoming mobile receiver. However, this modified RqM message will only trigger a partial registration request process, i.e., resulting in a DBT update (due the join of the new LMR to existing DBT) but not being forward toward the source (for registration of new member of multicast group). This modification of RqM message avoids the overwhelming traffic occurred by the roaming of mobile receivers.

The second way seems simple: the new LMR just forwards the Rm_IN message towards up-level BNR on behalf of the newly incoming mobile receiver. Upon receiving the Rm_IN message, the DBT update may then be launched. The disadvantage of this way is that there are two types of control messages traversing the routers in the DBT, i.e., RqM message (response to the joining /leaving of members of multicast group) and Rm_IN message (due to th e movement of receivers), resulting in complex processing for routers to deal with. However, this way keeps the backwards compatibility with SReM. Because no modification is made to the original set of control messages (RqM messages).

D. Roles of Signaling Messages in MoSReM

D.1. Joining/leaving process

In MoSReM, the two classes of multicast session signaling messages, i.e., JoinM/Leav and RqM/RpM, play their independent roles and cooperate (in turn) to support multicasting process. Below is an example of how they take part in joining process in MoSReM. The leaving process is similar to joining process.

First, let us consider the joining process, shown in Figure 3a. A new member of a multicast group, say *x*, sends a JoinM to its LMR. Upon receiving this message, *x*'s LMR will then send a RqM on behalf of *x*.

Depending on whether or not *x* is the first member of a multicast group attached to its LMR, *x*'s LMR will choose to send a RqM^1 or RqM^0, where the upper subscript '1'and '0' corresponds the states of 'S' bit in RqM. The LMR will send a RqM^1 if *x* is the first member (of a multicast group attached to its LMR), otherwise it will send RqM^0. It is straightforward for a LMR to find out which case is happening. Let us consider the first case, shown in Figure 3a.

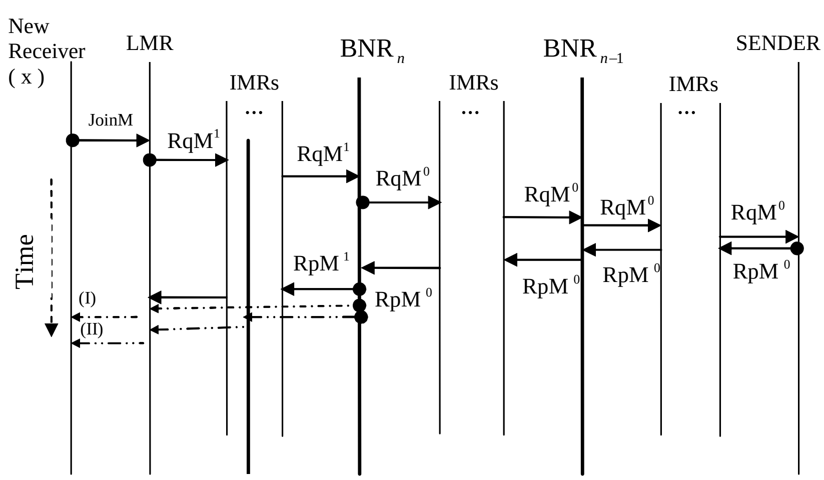
Upon receiving RqM^1, each of IMRs just passes and forwards it towards the source. As soon as a BNR (called the First BNR) receives RqM^1, it will fulfil two tasks (operations). The first is to create an entry for *x* in its MFT. This means that the multicasting process for *x* will be started at the First BNR. On other hand, the First BNR will change the 'S' bit in RqM^1, i.e., from the state '1' to the state '0'. The amended message is then an RqM^0. RqM^0 is used to inform following BNRs of no any processing needed, being done by checking the 'S' bit. From now onwards, instead of RqM^1, RqM^0 will be forwarded towards the multicast source. As receiving RqM^0, both following IMRs and BNRs just pass and forward it towards the source, no any processing is needed for them. Due to this fact, i.e., that both IMRs and BNRs don't need to do any process, it will take a very short time (called Reg_time_up) for RqM^0 from the First BNR to reach the multicast source.

Upon receiving RqM^0, the multicast source will send back RpM to confirm that the joining of new member is accepted. Corresponding to RqM messages, there also is a pair of RpMs in MoSReM, i.e., $RpM^0$and RpM^1. RpM^0 is a reply message sent by the multicast source while RpM^1 is one sent by the First BNR. RpM^1 is generated by the First BNR only as a RpM^0 is received. Except for the First BNR, no any processing is needed for all of the IMRs and other BNRs as receiving RpMs (RpM^0or RpM^1). Clearly, it will also take a very short time (called Reg_time_down) for RpM^0 from the source to the First BNR, for both IMRs and BNRs between them don't need to do any processing. After processing by the First BNR, a RpM^1 will be then produced and it will finally reach the LMR which the new member is attached to, informing of LMR being authorized to deliver multicast packets for the new member. This means the registration processing is finished.

As soon as the First BNR receives the RpM^1, the process of delivering multicast packets for the new member, may start. Note that this process is started from the First BNR. However, depending on whether or not a new BNR is found by IMRs (exactly the IMRs between LMR and the First BNR), the multicast packets will be forwarded via a New BNR (case (I)), or not (case (II)).

Now, turn to the second case, i.e., there exist already other members of a multicast group attached to the *x*'s LMR. In this case, *x*'s LMR will directly send a RqM^0 other than RqM^1, directly informing the multicast source of that *x* is joining. Therefore, there is no processing needed for all of following IMRs, BNRs and the multicast source. This case is shown in Figure3b.

Note that in MoSReM, the multicast source is always in charge of the controlling process of the delivery of multicast packets (i.e. the registration process), but it doesn't take part in the delivering of multicast packets. In fact, the delivering of multicast packets is fulfilled through BNRs, exactly a 'local' BNR for a new member. Although the part of registration process initiated from BNRs is always towards and through the multicast source, this process, which will take a short time (i.e., the sum of Reg_time_up and Reg_time_down), is fast and is expected to be finished almost at the time when BNRs are in place to deliver multicast packets. This leads to the efficiency and less delay in delivering of multicast packets. Therefore, MoSReM is effective and of less latency.



(a)

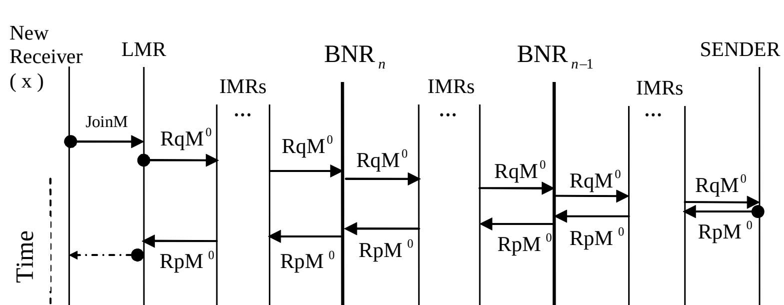


Figure 3 Joining and delivering process in MoSReM

D.2. Roaming process

D.2.1 Registration/De-registration Process

Join/leave registration/de-registration is just as that in SReM. Normally (the scenario which mobile receivers not in roaming), LMRs on behalf of mobile receivers will register or de-register with the multicast source when mobile receivers decide to join or leave a multicast group. In contrast, when a mobile receiver is roaming, further consideration is needed. This is because re-registering with the source for the existing multicast session (at new LMR) should be avoided for mobile receivers. However, a re-registration process between new LMRs and mobile receivers is needed for the incoming receiver to be recognized by its new LMR for multicast service. To tackle this issue, it is needed for the new LMR to be able to obtain information (from the Rm_IN message) about incoming receivers' qualification to receive multicast packets. For this, we may design the Rm_IN in such a way that there is a 'Marked Bit' (say, R bit) or identity included in Rm_IN message for the roaming–in mobile receiver to prove the new LMR (which a mobile receiver is roaming in) that it is legal or authorised (i.e. registered already) to secure the multicast serve.

D.2.2 Roaming Process

When a mobile receiver is in roaming, there are two Messages (Rm_IN, Rm_OUT) to be involved. Briefly, as shown in Fig.5, the roaming procedure includes the process as follows:

(i) Mobile node (MN) sends an Rm_OUT message towards its current LMR (old LMR).

(ii) Old LMR trigger a procedure of update of DBT, if no more MN attach to the old LMR.

(iii) Mobile node (MN) sends an Rm_IN message toward its new LMR.

(iv) New LMR trigger a procedure of update of DBT, if the MN is the first member attached to the new LMR

(v) The process of update of DBT is similar the join/leave operation in SReM, except for no registration/de-registration towards the source.

The following examples provide the detailed regarding roaming process. This first one is the case of that the multicast tree (i.e. DBT tree) is unchanged when a mobile receiver is roaming. The second shows the scenario where the DBT is needed to be change as a result of the mobile receiver's roaming. Let us use the following denotation: *x* for mobile receiver, oLMR for the LMR to which *x* attached before *x* roaming, nLMR for the LMR to which *x* to attach after x roaming. Furthermore, assume that m and k are the number of mobile receivers currently attached to the oLMR and nLMR, respectively.

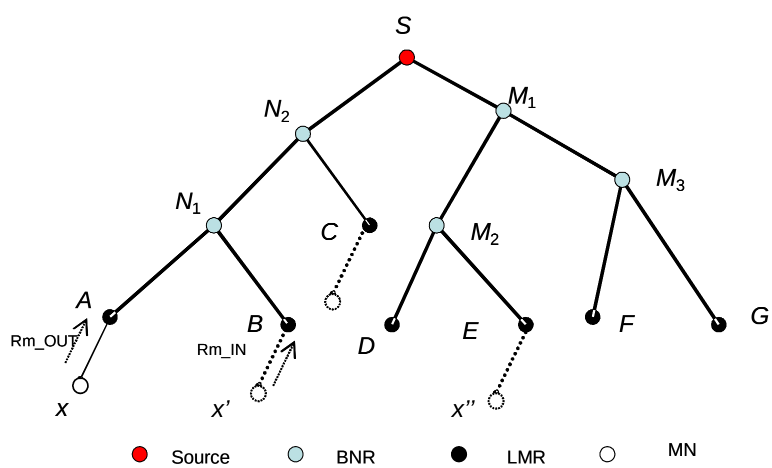


Figure 5. The roaming process in MoSReM

In the first scenario, the mobile receiver *x* must be not the last one attached to the oLMR or the first one attached to the nLMR. For this case, the whole roaming procedure is simple. *x* send a Rm_OUT message towards its oLMR. Upon receiving this message, *x*'s oLMR will update its

MDT: MDT (*m*)→MDT(*m*-1), where *m* is the number of receivers attached and $m \geq 1$. No any further action against the existing DBT needed to be taken due to *x*'s roaming out of its oLMR. Similarly, *x* needs to send a Rm_IN message towards its nLMR when x roaming into the area covered by its nLMR. After receiving Rm_IN message, nLMR's MDT is needed to be updated: MDT(*k*)→MDT(*k*+1), where $k \geq 1$. As *x*'s roaming-out procedure above, *x*'s roaming-in does not influence on the existing DBT, i.e., there is no need to update MFTs at all BNRs and MDTs at LMRs (except oLMR and nLMR). As mentioned previously, nLRM is able to be aware of the *x*'s membership belonging to multicast groups. Therefore, from then on, mobile receiver is able to receive multicast packets directly from its nLMR.

The second scenario is a little bit complex because the DBT is to be updated as a result of *x*'s roaming-in and/or roaming-out. Depending on the status of MDTs at oLMR and nLMR, there might be three combination cases to happen, i.e.,

(i) MDT(1) → MDT(0) for oLMR and MDT(1) → MDT(0) for nLMR;
(ii) MDT(1) → MDT(0) for oLMR and MDT(*k*) → MDT(*k*+1) for nLMR;
(iii) MDT(*m*) → MDT(*m*-1) for oLMR and MDT(1) → MDT(0) for nLMR.

All of these cases will trigger a DBT update, including BNRs update and MFTs update at BNRs. Let us take case (i), for example, to explain the roaming procedure of this scenario. When the mobile receiver *x* detects that it is in roaming[1], it will send an Rm_OUT message towards its oLMR. Upon receiving this message and checking its MDT, *x*'s oLMR will be aware of it will no longer be in a multicast branch. As a result, *x*'s LMR will delete its MDT for the multicast group and then send a DBT update message[2]. This DBT update message is then forwarded to the up-level BNR (say N_1 in Fig. 5) of the existing DBT. The up-level BNR (or even and its next up-level BNR, say N_2 in Fig. 5) will be in charge of the DBT update process by the use of BNMs messages as that in SReM.

---

[1] We assume that mobility detection is managed by Mobile IP or other mobility protocols.
[2] As aforementioned, the DBT update message could take a form of modified RqM message or directlt use the Rm_OUT message.

On the other hand, when the mobile receiver *x* roams into a new multicast domain which is covered by an nLMR, it will send a Rm_IN message to inform the nLMR of its roaming-in. Due to *x* is the first member of the multicast group (attached to nLMR), upon receiving the Rm_IN message, the nLMR will set up a MDT for the multicast group and simultaneously send a DBT update message (as mentioned above, it might be a modified RqM or Rm_IN) towards the multicast source. One of BNRs in the existing DBT will receive the DBT update message. This BNRs will then be in charge of the DBT update process by the use of the pair of BNMs messages. Meanwhile, all the MFTs at the BNRs (existing or newly updated) will be updated. After the completion of the DBT update, the mobile receiver x is then able to receive the multicast packet from the nLMR.

D.2.3 Smooth handoff management

We assume that the Mobile IP with the extension of route optimization is supported by IP networks [18], called extended Mobile IP hereafter. In order to use the route optimization extension, the existing Mobile IP mobility agent advertisement extension has been made a minor change, i.e., a flag bit replaces a previously unused reserved bit in the extension to indicate that the foreign agent support smooth handoffs.

Based on the extended Mobile IP, there are many proposals to deal with fast and reliable multicasting, such as pre-active hand-off [16] and fast hand-off [17]. All of the proposals uses the mechanism of predicting the movement of mobile receivers and anticipating new points of attachment [15], where assume foreign agents can detect or anticipate the movement or direction of mobile receivers by taking advantage of link layer and radio specific information. For example, in pre-active handoff scheme, after detecting that a mobile receiver is about to perform a handoff to a different location, the mobile receiver's serving foreign agent sends a binding update request to the "new" foreign agent prior to handoff. This proactive binding update contains the mobile receiver's home address, security related information, as well as the serving gateway foreign agent's address.

In contrast, fast handoff proposal assumes that the serving foreign agent anticipates the movement of mobile hosts by sending multiple copies of the traffic to potential neighbour foreign agents. "Bicasting" is used to support data forwarding to the previous and new foreign agents while the mobile receiver is moving between the old and new access points. Fast handoff predicts the movement of mobile receivers through coupling with layer-2 functionality, which is dependent on the type of access technology used. Bicasting uses simultaneous bindings, where the mobile receiver sets the "S" bit in the registration request. Depending on the networking model (i.e., flat or hierarchical model) the receiving agent (home agent, gateway foreign agent or regional foreign agent) will add a new binding for the mobile receiver. Fast handoff completes the Mobile IP handoff prior to establishing layer-2 connectivity or forwarding data.

In MoSReM, we can use the similar mechanism (of the prediction of movement) above to carry out smooth handoff. For this, we assume that all the LMRs are able to and actively participate to perform the detection of movement of mobile receivers. As part of smooth handoff (procedure), the mobile receiver may request that its nLMR[3] attempt to notify its oLMR on its behalf, by including an oLMR notification extension in its registration request message(of Mobile IP) sent to the nLMR. The nLMR then builds a binding update message (of Mobile IP) and transmits it to the mobile receiver’s oLMR as part of registration, requesting an acknowledgement from the oLMR. Upon receiving the binding update message, the oLMR will then send a notification, which includes the mobile receiver’s nLMR, or new care-of address, allowing the oLMR (or its foreign agent) to create a binding cache entry for the mobile receiver to serve as a forwarding pointer to its new location. After this procedure, any multicast packets for the mobile receiver that arrived at its oLMR will then be tunneled by the oLMR to the mobile receiver’s nLMR or new care-of address.

3.3. Delivery of packets

To multicast data packets, a hop-by-hop-changing MoSReM header will be added to each of packets. This is done by branching node routers, including the multicast source or sender. Each of branching node routers will encode the address of one following next branch node in data packets. This leads to distinct headers for packets towards the different next branching nodes.

[3] We assume that LMRs may work with the same functionality as foreign agents in Mobile IP.

At the starting of multicast, the source or root will send a copy of packets with respective MoSReM headers to all of branching nodes in its MFT. Upon receiving this packet, each of branching nodes needs to properly process the MoSReM header. The standard processing for a branching node is as follows: a) Check its MFTs and replicate the packet so that there's one copy of the packet for each of its next branching nodes, b) Modify the MoSReM header with destination in each of the copies substituted by the address of next branching nodes, c) Send the modified copies of the packets on to the next branching node, which is listed in its MFT.

For non-branching nodes, packets are forwarded just like normal unicast. Finally, when arriving at the last-hop multicast router, multicast packets will delivered by the use of standard multicast.

IV. FUTURE WORK

In our future work, we will further explore the details about MoSReM, such as the format of all signaling messages in MoSReM. The performance analysis of MoSReM and comparison with other mobile multicast protocols will be made in our next stage’s work.

We will also investigate the possibility of how to use Session Initiation Protocol (SIP) to implement MoSReM or a new version of MoSReM based on SIP.

V. CONLUTION

In paper, a new mobile multicast scheme is proposed, named mobile scalable recursive multicast (MoSReM). MoSReM is an extension version of our previously proposed multicast protocol SReM, suitable for applications in mobile IP networks. Similar to SReM, multicast tree in MoSReM is built gradually as the joining / leaving of members of multicast groups and the delivery of multicast is done recursively via branching node routers. This is implemented by the use of a pair of branching node messages (BNMs). Based on extended Mobile IP and introduction of roaming control messages Rm_IN and Rm_OUT, MoSreM supports receivers mobility. MoSReM have two advantages: being scalable and lower joining / leaving latency.

Our future work will include: detail the MoSReM protocol and evaluate the performance of MoSReM. Another investigation will be made in how to combination SIP with our proposed protocol MoSReM.

References

[1] Christophe Jelger and Thomas Noe, Multicast for mobile hosts in IP networks: progress and challenges IEEE Wireless Communications (formerly: IEEE Personal Communications) , Volume: 9 Issue: 5 , 2002 Page(s): 58 -65

[2] Jiunn-Ru Lai and Wanjiun Liao, Mobile multicast with routing optimization for recipient mobility , Consumer Electronics, IEEE Transactions on , On page(s): 199 - 206 Feb. 2001 Volume: 47 Issue: 1.

[3] Ki-Il Kim, Jeoung-Lak Ha , Eun-Hee Hyun and Sang-Ha Kim, New approach for mobile multicast based on SSM, Networks, 2001. Proceedings. Ninth IEEE International Conference on On page(s): 405 – 408, Oct.2001.

[4] C. Perkins, ed., IP mobility support, RFC 2002, IBM (October 1996).

[5] D. Waitzman, C. Partridge and S. Deering, eds., Distance vector multicast routing protocol, RFC 1075, BBN STC and Stanford University (November 1988).

[6] J. Moy, Multicast routing extensions for OSPF, Communications of the ACM 37(8) (1994) 61–66.

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

[8] Yewen Cao and Khalid Al-Begain, ‘‘Scalable Recursive Explicit Multicast (SReM)’’, in Proc. 5th *World Wireless Congress* (*WWC’04*), San Francisco, USA, May 2004, pp.457-462.

[9] H.W. Holbrook and D.R. Cheriton, “IP multicast channels: EXPRESS support for large-scale single-source applications,” in SIGCOMM’99, Cambridge, Massachusetts, Aug. 1999.

[10] I. Stoica, T. Ng, and H. Zhang. REUNITE: A recursive unicast approach to multicast. In Proceedings of IEEE INFOCOM'00, Tel Aviv, Israel, Mar. 2000.

[11] R. Boivie et al., Explicit multicast (Xcast) basic specification,<draft-ooms-xcast-basic-spec-00.txt>, 2000.

[12] M. Shin et al., ‘‘Explicit Multicast (Xcast+) Supporting Initiated Join,’’ <draft-shin-xcast-reciever-join-00.txt>, 2001.

[13] A.Boudani and B.Cousin, ‘‘ Simple explicit multicast,’’ <draft-boudani-simple-xcast-01.txt>, Jan. 2002.

[14] Andrew T. Campbell, et al., ‘‘Comparison of IP Micro-mobility Protocols,’’ IEEE Wireless Comm., pp.2-12, Feb.2002.

[15] Andrew T Campell and Javier G. Castellanos, ‘‘ IP Micro-Mobility Protocols,’’ Mobile Computing and Communications Review, Vol.4, no.4, pp.45-53, Oct. 2004

[16]James Kempf, Pat R. Calhoun, Cbandana Pairla, “Foreign Agent Assisted Hand-off ”, Internet Draft, draft-calhoun-mobileip-proactive-fa-01 .txt, Work in Progress, June 2000.

[17] Gopal Dommety, Local and Indirect Registra- K. E1 Malki, H. Soliman, “Fast Handoffs in Mo-bile IPv4”, Internet Draft, draft-elmalki-mobileip-fast-handoffs-03.txt, Work in Progress, September 2000.

[18] C. E. Perkins, Mobile IP Design Principles and Practices. Massachusetts: Addison-Wesley Longman, 1998