In the *Proc. of the* 2^{nd} *International Mobile Computing Conference (IMCC 96)*, March 25-27, 1996, Hsinchu, Taiwan, ROC

An Efficient Multicast Protocol for PCS Networks[1]

Vanitha Aravamudhan Karunaharan Ratnam Sampath Rangarajan

ABSTRACT In a Personal Communication Services(PCS) Network, mobile hosts communicate with other mobile hosts through base stations on a wired (static) network. The mobile hosts connect to different base stations through wireless links and the base stations to which mobile hosts are connected change depending on the current location of the mobile hosts. In this environment, the problem of efficiently delivering a multicast message from one mobile host to a group of other mobile hosts becomes challenging. In this paper, we present a multicast protocol that delivers multicast messages from a mobile host to a group of other mobile hosts *without* flooding the wired network. The multicast protocol is built on top of a user location strategy that should follow one of three models of user location described in the paper. The basic multicast protocol proposed guarantees exactly-once message delivery to all mobile hosts in the multicast group and also ensures that multicast messages are delivered in FIFO order from the point of view of the base station that originates the multicast message (referred to as BS-FIFO). More importantly, an extension of the basic protocol is provided that, unlike earlier work, delivers multicast messages in FIFO order from the point of view of the mobile host that initiates the multicast message (referred to as MH-FIFO). The modifications to be made to the multicast protocol to accommodate each of the three models of user location is also described.

1 Introduction

In distributed systems, efficient mechanisms are required that enable a process to multicast a message to a group of other processes [5][13]. In a Personal Communication System (PCS), this problem of multicasting a message from a mobile host to other mobile hosts (henceforth referred to as mobiles) becomes more difficult because of the constant movements of the mobiles. A mobile that initiates a multicast message to a group of other mobiles, sends the message through the wireless link to the base station that is monitoring this initiator mobile. This base station is now responsible for transmitting the message across the wired network and delivering it to the mobiles in the multicast group. In [1], a multicast protocol is presented in which it is assumed that no location information is maintained for the mobiles. Hence, it was required that a multicast message be sent to every base station in the network as the mobiles that belong to the multicast group could be anywhere on the network. Thus, this protocol requires the network to be flooded with the multicast message. The protocol in [1] guarantees that multicast messages that are originated by a base station, will be delivered in FIFO

[1]Authors are with the Department of Electrical and Computer Engineering, Dana Research Building, Northeastern University, Boston, MA 02115, U.S.A.

order to a particular mobile. We refer to this as the BS-FIFO delivery requirement. Further, it is guaranteed that a multicast message is delivered to a recipient mobile exactly once. A related protocol presented in [2] does not require flooding the network but requires information about a "host-view" of each multicast group. Thus, it requires a separate host-view membership algorithm to be integrated together with the multicast protocol.

The maintenance of a Location Directory (LD), be it centralized or distributed, seems to be highly essential to the design of any distributed protocol in a mobile environment [3][4][7][10][17]. A location directory is nothing but a directory which provides information to locate the base station that is currently monitoring a particular mobile. If location directories are replicated at all the base stations, a base station originating a multicast message could send the multicast message to only those base stations that are monitoring one or more of the mobiles in the multicast group, as per the current information available in the local location directory. If the location directory is maintained at a central site, the originating base station can consult this central site to determine the list of base stations to whom the multicast message should be sent. Thus, the network need not be flooded as long as user location is maintained in some fashion.

Based on the assumption that some user location strategy will always be integrated into a PCS network, we present a multicast protocol that efficiently delivers multicast messages. This multicast protocol should be built on top of a user location strategy that has one of the following structures:

1. The location directory is fully replicated where all base stations maintain location information about all mobiles. This information gets constantly updated when the location of a mobile changes [15].

2. Each mobile's location information is maintained only at its *home* network. This model is followed in the basic IS-41 strategy [8] through the use of Home Location Registers (HLR). When a mobile moves, the new location information is updated only at its HLR.

3. In addition to the location information about a mobile being maintained at its HLR, this information may also be cached by a base station that makes frequent connections to this mobile. The decision as to whether information about a mobile should be cached at a base station could be made based on its called frequency from this base station, and also its mobility [9].

In our description of the multicast protocol, we assume a user location strategy based on full replication as in [15] for the following reason. A fully replicated location management protocol should provide a "worst case" scenario in terms of multicast message overhead. In a fully replicated location management scheme, no base station can be guaranteed to have the latest information about any mobile. This means, old location information may lead to a multicast message being forwarded one or more times. In both a central location management scheme as well as one that employs caching, one can always rely on the HLR to give the most recent location information about a mobile. This means, the number of times a multicast message is forwarded will be no more than in a scheme that uses full replication of location information. Thus, the total message overhead for

delivering a multicast message to a mobile for a protocol implementation that uses any other location management scheme will be upper bounded by an implementation that uses fully replicated location directories (plus an additional message that is incurred in contacting the HLR, if required). We provide simulation results for the multicast message overhead for a protocol implemented on top of a fully replicated location management scheme. We also describe implementation changes that are required in order for the protocol to work on top of a user location strategy that uses a central HLR and also a user location strategy that uses caching.

2 Motivation

Although flooding is not required in a multicast protocol that uses a user location strategy, the following situations need consideration. A base station that originates the multicast message will first use the available user location strategy to find the location of the mobiles in the multicast group. One or more of this location information may be old and hence the base station may send this message to base station(s) that are not currently monitoring any of the mobiles in the multicast group. When these base stations receive this message, they have to forward the message to the appropriate base stations(s). Even if the location information used by the originating base station at the time of sending the multicast message is current, by the time the message reaches the destination base station(s), the mobile(s) may have moved out of those base station(s). Again, the message has to be forwarded to the appropriate (new) base station(s). To avoid congestion at the new base station, forwarding of the message should be done only by one base station. The proposed multicast protocol, when it uses a fully replicated location directory, ensures that the a message is forwarded to a new base station by exactly one other base station.

Since the mobiles are constantly moving, two multicast messages originating from a base station, sent to the same mobile, can get delivered to the mobile out of order. The proposed protocol ensures that messages originating from a base station destined to a mobile get delivered to that mobile in the order in which they were originated at the base station. We refer to this property as the BS-FIFO property. In most cases, base stations originate multicast messages on behalf of the mobiles in their cell that wish to send multicast messages to a group of other mobiles. In this case, maintaining FIFO delivery at the destinations for the messages that are sent by a particular mobile may be more important. That is, multiple messages sent by a mobile should be delivered to a destination mobile in the order in which they were sent. We refer to this as the MH-FIFO property. The mobiles, due to their mobility, may send two multicast messages through two different base stations where these messages have one or more common recipients. So, an extension to the basic protocol is presented that, unlike earlier work [1][2], ensures the MH-FIFO property.

Due to the mobility of the mobiles, a situation can arise where a mobile receives a message through a base station and then moves into the cell of another base station which has a copy of the message. In such a case, the protocol may need to ensure that the message is not delivered again. That is, it is not only required that a message be delivered *at-least once* to a mobile, but it may be also be required that the message be delivered *at-most once* [1][2]. To satisfy the at-most once requirement,

the protocol should ensure that two or more base stations do not deliver the same message to a mobile. The at-least once requirement can be satisfied by ensuring that if a base station receives a copy of a multicast message for a mobile in its cell, it will deliver the message to this mobile. The proposed protocol ensures both these conditions, thereby ensuring *exactly once* delivery of the message.

In summary, the above considerations have resulted in the following characteristics for the protocol presented in this paper.

1. The network is not flooded with the multicast message and the protocol, if built on top of a fully replicated location directory, will ensure that the multicast message is not sent to a base station more than once.

2. The proposed multicast protocol satisfies the BS-FIFO delivery requirement.

3. An extension to the protocol is presented that can be included if MH-FIFO requirement needs to be satisfied.

4. The protocol satisfies the exactly-once delivery semantics through data structures maintained at the base stations. It does not require a mobile to carry information about the multicast messages that it has received so far.

The paper is organized as follows. Section 3 describes the system model which defines the terminology used. Section 4 provides a detailed description of the protocol including correctness arguments. Examples that illustrate the execution of the protocol are presented in Section 5. Section 6 presents conclusions and future work. A specification of the protocol in the form of a finite-state machine that provides implementation level details, is presented in the Appendix.

3 System Model and Assumptions

The PCS network reference model that we assume is shown in Figure 1. The Service Control Point (SCP) maintains the service specific information (service logic). Home Location Register (HLR) maintains the profiles of all the mobiles (MH) registered with the home network and a mobile registers with only one home network. The information maintained in the HLR includes a pointer to the Visitor Location Register (VLR) where the mobile has currently roamed into. A set of base stations (BS), which provide the wireless connection to the mobiles within its coverage area (cell), are connected to the Mobile Switching Center (MSC). Given the i.d. of a mobile, all MSCs can uniquely determine its HLR.

Every mobile is registered with a home network, the HLR of which maintains the mobile's current physical location. If the mobile has roamed in to another region, then it has to register with the VLR that covers the new region. As part of registration and authentication, the VLR contacts the mobile's HLR, and the HLR updates the mobile's profile to reflect its new location. In the rest of the paper, we will refer only to HLRs and base stations. Whenever a HLR needs the location information of a

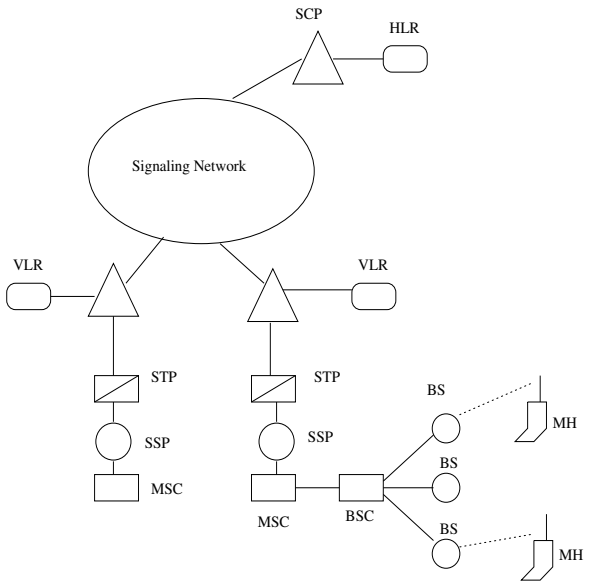

Figure 1: Network model

mobile, it is understood that the HLR will contact the appropriate VLR, which in turn will contact the appropriate MSC to get that information.

Our protocol is based on the following assumptions:

1. The wireless channel has arbitrary but bounded delay.

2. No message chases a mobile forever without being delivered to the mobile.

3. There are no base station failures.

4. Messages are reliably delivered over the wired network.

In this paper, any message that is sent to any subset of mobiles will be considered as a multicast. The protocol will not be any different whether a multicast message is sent to a multicast group that is formed dynamically, or if it is sent to an already existing multicast group.

4 Multicast Protocol Description

We first provide a description of the data structures that are maintained at each of the base stations in order to execute the multicast protocol. Then, we provide details of the protocol that ensures BS-FIFO delivery of messages and show how the data structures are updated. After that, we present correctness arguments to show that the protocol ensures the desired properties. An extension to this protocol that ensures MH-FIFO delivery of messages is presented in a later section.

4.1 Data Structures

The following data structures are maintained at each of the base stations :

- M_{id} : This is a counter that indicates the number of multicast messages that have been originated so far from the base station. Whenever a new message is originated, this counter is incremented and the value is sent along with the message. Thus, the i.d. of the originating base station, together with M_{id}, can uniquely identify a multicast message sent on the wired network.

- $SEQ[1 \ldots M]$: This is a $1 \times M$ array maintained at a base station, where M is the total number of mobiles in the network. The entry $SEQ[j]$ at BS_i indicates the total number of multicast messages sent so far to the jth mobile MH_j from BS_i. Thus, while M_{id} provides a total count of the multicast messages originated at a base station, the $SEQ[1...M]$ array provides a count of the number of messages sent so far to each of the mobiles from that base station.

- $MH_{rcvd}[1 \ldots N]$: Each mobile has an associated MH_{rcvd} array (of N elements, where N is the total number of base stations on the wired network) which at any point in time, is maintained at the base station which is monitoring this mobile at that time. As the mobiles move out of the cell of a base station, this array gets transferred to the new base station responsible for that mobile. Thus, in steady state, the MH_{rcvd} array of a mobile is located at only one base station. The $MH_{rcvd}[i]$ entry for mobile MH_j provides the number (i.d.) of the next multicast message that MH_j can receive from BS_i. This array is used along with the SEQ array to maintain BS-FIFO and *exactly-once* delivery as shown later.

- C matrix : This is a matrix of size $N \times N$. The entry C_{ij} in the matrix indicates the cost of forwarding a message from BS_i to BS_j under normal traffic conditions. This cost matrix is replicated at every base station and is used to decide which base station should forward a multicast message to another base station when forwarding is required. Thus, this matrix is used to minimize the cost overhead incurred in forwarding a message and also to ensure that a base station gets a message forwarded to it only once. One of the objectives is to make sure that in the multicast protocol, exactly once forwarding takes place. The idea of the cost matrix is just a mechanism to improve the efficiency of forwarding by also making sure that among the base stations that are capable of forwarding a message, the one that incurs the least cost overhead does so. Obviously maintaining a matrix like this could limit scalability. So, to make the protocol more scalable we can use other mechanisms that choose one base station to forward a message. One such mechanism is described later in the paper.

- $local_{list}$: This is a list of all the mobiles that the base station is currently monitoring. Thus, this list changes dynamically.

4.2 Details of the Protocol

The multicast protocol is described assuming a user location strategy that uses fully replicated location directories (LD) at each of the base stations. Modifications that are required to suit the other location

strategies are provided in a later section.

Whenever a base station wants to originate a multicast message to a group of mobiles, it executes the multicast protocol. Initially, the M_{id} and the $SEQ[1 \ldots M]$ array at a base station will have zero entries indicating that no multicast message has been originated from this base station. When a base station (referred to as (BS_{orig})) originates a multicast message to a group of mobiles (indicated by (MH_{list})), it increments its M_{id} counter and also the entries in the SEQ array corresponding to each of the mobiles in the MH_{list}. So for example, if mobile MH_i is in the MH_{list}, then $SEQ[i]$ is incremented. BS_{orig} then creates a tuple $< (MH_{id}, SEQ[MH_{id}] >$ for each of the mobiles in MH_{list} and then creates a *tuple list* $< (MH_{id1}, SEQ[MH_{id1}]), (MH_{id2}, SEQ[MH_{id2}]), \cdots >$ which contains all the tuples. Then, BS_{orig} looks up the local location directory and creates a list of base stations (referred to as BS_{list}) that are currently monitoring one or more of the mobiles in MH_{list} (the location information in the local location directory may not be current, hence forwarding may be required at the base stations in BS_{list}). BS_{orig} now sends the multicast message in the following format to all the base stations in BS_{list}:

$$< M_{id}, BS_{orig}, BS_{list}, < (MH_{id1}, SEQ[MH_{id1}]), (MH_{id2}, SEQ[MH_{id2}]), \cdots >, message >$$

Here, $message$ refers to the contents of the multicast message, the rest forming part of the header for the message. The combination of M_{id} and BS_{orig} uniquely identify a multicast message.

When a base station receives a multicast message, the following algorithm is executed for each tuple in the tuple list to check if a local mobile is *eligible* to receive a message. A mobile is eligible to receive a multicast message from a originating base station BS_{orig}, if it has received all the previous messages from BS_{orig} destined for it. The eligible mobiles are included in the list $local_dest_list$, which is initially empty.

local_dest_list $= \{\}$
while $(tuple\ list \neq empty)$
do {
  If $((MH_{id} \in local_list)$
  && $(MH_{rcvd}[BS_{orig}] == SEQ[MH_{id}]))$
    add MH_{id} *to the local_dest_list*
  else go to the next entry
}

The base station sends the multicast message to each of the mobiles in the $local_dest_list$ and then creates an empty ack_list, which is the list of all the mobiles that have received the multicast message. So, the following sequence of operations take place.

while (local_dest_list $\neq$ *empty)*
do (for each mobile MH_i in local_dest_list) {
  if (ack) received from MH_i{

add MH_i *to ack_list*
remove MH_i *from local_dest_list*
remove MH_i *and* $SEQ[MH_i]$ *from the message*
Increment $MH_{rcvd}[BS_{orig}]$ *for* MH_i
}
elseif time out on (ack) for MH_i
remove MH_i *from local_dest_list*
}

The $MH_{rcvd}[BS_{orig}]$ array entry for a mobile is incremented to ensure that the same message does not get delivered again to the mobile if it enters the cell of another base station that also has a copy of this message. When the $local_dest_list$ is empty, the ack_list is sent to BS_{orig} and the message is stored in the buffer, along with an empty BS_{fwd} list. This is the list of base stations to which this message has been forwarded to by this base station. When a mobile in the MH_{list} that was not delivered a multicast message, enters the cell of a base station that does not have a copy of this message, one of the base stations in BS_{list} (or BS_{orig}) will forward the message and include the new base station in its BS_{fwd} list for that message. As shown in the next section, a message will be forwarded to the new base station by only one other base station. That means, the BS_{fwd} list for the same message at different base stations will be different and non-overlapping.

When a mobile registers with a base station, it is expected to provide this new base station with the i.d. of the base station that was its previous monitor. The new base station adds this mobile to its *local_list* and then requests the previous monitor for the MH_{rcvd} array for this mobile. Using the MH_{rcvd} array, it checks its message buffer for any pending message for that mobile. If the mobile is eligible to receive any of the pending messages, the algorithm explained above for the creation of $local_dest_list$ and ack_list is performed for this mobile.

At the originator base station BS_{orig}, every time an ack_list arrives from the various base stations, the corresponding mobiles in the list are deleted from MH_{list}. Once MH_{list} becomes empty, a *delete* multicast message is sent (with BS_{orig} and M_{id} information included in this message) to all the base stations in the BS_{list}. At each base station, the BS_{orig} and M_{id} information uniquely identifies a multicast message. This multicast message is deleted from the message buffer at the base station, and the $delete$ message is forwarded to all the base stations in the BS_{fwd} list of the multicast message. Thus, the forwarding of delete messages also takes place efficiently.

4.2.1 Forwarding of Messages

When a mobile that is in the MH_{list} of a multicast message, moves into the cell of a base station (let us refer to it as BS_{new}) without receiving the message at its previous base station, this message should be forwarded to BS_{new}. This forwarding should be done by only one of the base stations to avoid congestion of messages at BS_{new}. This is done as follows.

In a user location strategy, where fully replicated location directories are used [15], updates to the location directory are propagated to all the base stations. When a location directory update about a mobile arrives at a base station, say BS_i, then BS_i checks the message buffer to see if there is any pending multicast messages for this mobile. If such a message exists, then BS_i checks the BS_{list} for that message. If BS_i is either the originator of the message or an element of BS_{list}, and BS_{new} is neither an element of BS_{list} nor the originator of the message, the message has to be forwarded to BS_{new}. To avoid forwarding of the message to BS_{new} by all the base stations in the BS_{list} of the message, the cost matrix C is used. This matrix is stored at all the base stations and is the same everywhere. BS_i traverses the cost matrix to see which of the base stations in BS_{list} and BS_{orig} has the least cost in transmitting the message to BS_{new}. Given that C is the same at all the base stations, if the cost matrix is traversed in the same order at all the base stations, only one base station that has the least cost overhead will forward the multicast message to BS_{new}. This cost matrix is not an absolute requirement for the functioning of the protocol. Other mechanisms can be designed to guarantee that only one base station forwards a message. For example, a base station in BS_{list} can use a mapping function as follows. A base station in the BS_{list} can compute the value of *id of* (BS_{new}) $mod|BS_{list}|$ where $|BS_{list}|$ denotes the size of BS_{list} and *id of* (BS_{new}) refers to the an identification number for the base station. If this values evaluates to the base station's position in BS_{list}, it will forward the message to BS_{new}. Otherwise, it will not forward the message. This way, exactly one base station will forward the message to BS_{new}. Other such mapping functions can be designed. This would make the protocol very scalable but the trade off is this forwarding may not be the "most optimal" in the sense of cost overhead. In the rest of the paper our description assumes the availability of a cost matrix.

The base station that forwards the message will add BS_{new} to the BS_{fwd} list for that message (if BS_{new} is already not present in the list). If BS_i is neither in BS_{list} not is the originator of a message, it cannot consider forwarding the message because the message to BS_i itself was forwarded by some other base station, and this information may not be known to all the other base stations in BS_{list}, which could then result in multiple stations forwarding the same message to BS_{new}. Thus, only the base stations in BS_{list} can forward a message, and base stations in the BS_{fwd} for the message are not allowed to forward the message further.

Forwarding may also be required in a slightly different situation. If the originator base station sends a multicast message to a certain BS_{list}, and (say) there is a LD update at one of these base stations for one of the mobiles in the MH_{list} before the multicast message reaches that base station. So to ensure at least once delivery in this case, every base station BS_i, as soon as it gets the multicast message, should compare the BS_{list} with the list of base stations that monitor the MH_{list} according to its location directory and if there is a base station that does not have a copy of the message and if BS_i is the one that should forward the message to this base station, then it should forward it.

4.2.2 State Information

Most of the time the only information maintained in the base stations is the multicast message along with the BS_{list} and the MH_{list}. Whenever the initiator receives the *acklist*, a search is made in its

list of messages and the relevant mobiles are removed from the MH_{list}. Apart from this, whenever a base station sends a message to a mobile and gets a reply, an *acklist* is prepared to send to the initiator base station. As all operations are asynchronous handshakes and the base stations need not maintain any states, it can just do a look up as and when a message arrives. Thus, the amount of information maintained in a base station does not increase with increase in number of base stations or mobile hosts thus making the data structure maintenance very scalable.

A specification of the protocol in the form of a finite-state machine diagram which gives implementation level details, is provided in the Appendix.

4.3 Correctness of the Protocol

In this section, we show that multicast messages originating at a base station get delivered to mobiles in the multicast group exactly-once and in BS-FIFO order. Similar to [1], we make the assumption that a mobile makes only a finite number of moves without receiving any "deliverable" messages. This eliminates the possibility that a mobile does not stay long enough in any cell to get any message delivered at all. Also, in order to prove the BS-FIFO property, we assume that the wireless channel from a base station to a mobile is FIFO.

4.3.1 BS-FIFO Delivery

The protocol ensures that two multicast messages that originate from a base station destined to a particular mobile are delivered in the order in which they are originated. This property is maintained using the SEQ array at the originator base station and also the MH_{rcvd} array corresponding to the mobile at the monitoring base station of the mobile. Consider the following:

Let BS_i be a base station that originates a multicast message. Initially, all the entries of the SEQ array at BS_i will be zero. The entry $SEQ[a]$ corresponding to mobile MH_a, indicates the number of multicast messages that have been sent from BS_i to MH_a so far. Similarly, the ith entry of the MH_{rcvd} array for MH_a will contain the number (i.d.) of the next multicast message destined to MH_a that it is expecting from BS_i. That means, $MH_{rcvd}[i]$ of MH_a will start with a 1 indicating that it is waiting for its first message from BS_i. Let us now assume that BS_i initiates the first multicast message destined to MH_a and MH_b. Then, the following updates are made in the SEQ array at BS_i:

$$SEQ[a] = 1$$

$$SEQ[b] = 1$$

If both these mobiles are in the cell of base station (say) BS_j, the multicast message will be sent to BS_j as:

$$< M_{id}, BS_i, < BS_j >, < (MH_a, 1), (MH_b, 1) >, message >$$

At base station BS_j, since the ith entry of the MH_{rcvd} array for both mobiles MH_a and MH_b are 1, and since these match the sequence numbers sent along with the multicast message, the message is delivered to MH_a and MH_b. Then, $MH_{rcvd}[i]$ corresponding to both MH_a and MH_b are incremented to 2. Note that Base station BS_j needs to have the MH_{rcvd} array for MH_a and MH_b before it can deliver messages to these mobiles.

Now suppose that MH_a moves from the cell of BS_j to the cell of BS_k without receiving the first multicast message. Assume that BS_i now originates a second multicast message to MH_a and sends it to BS_k (assuming it gets the current location information for MH_a; if not, it will send it to BS_j and forwarding will take place). The SEQ array at BS_i is updated again as follows:

$$SEQ[a] = 2$$

The multicast message is sent as follows:

$$< M_{id}, BS_i, < BS_k >, < (MH_a, 2) >, message >$$

Since MH_a has not received the first multicast message, the $MH_{rcvd}[i]$ entry at BS_k corresponding to MH_a will be equal to 1 (again, BS_k needs to get the MH_{rcvd} array for MH_a before it can deliver messages to this mobile). When BS_k receives the second multicast message where the sequence number corresponding to MH_a is 2, this message is not delivered to MH_a as it is not equal to the $MH_{rcvd}[i]$ entry corresponding to MH_a. So the message is held up until the first message gets forwarded to BS_k, is delivered to MH_a and the $MH_{rcvd}[i]$ entry is updated to 2 making it eligible to receive the second message from BS_i. Thus, messages do not get delivered to mobiles in random order but are strictly delivered in the order in which they are originated at a base station.

4.3.2 Exactly Once Delivery

The protocol ensures *exactly-once* delivery of a message to a mobile even though the mobile may move through different base stations while the message delivery protocol is in execution. The MH_{rcvd} array is used to accomplish exactly-once delivery. Consider an extension to the scenario used in Section 4.3.1. Now, assume that mobile MH_b moves from the cell of BS_j into the cell of BS_i which is the originator of the first message and hence has a copy of it. The MH_{rcvd} array for MH_b is transferred from BS_j to BS_i after BS_i makes a request for it. For the exactly-once property to hold, the first multicast message should not be delivered to MH_b by BS_i, as it has already been delivered to MH_b by BS_j. This condition will hold because the $MH_{rcvd}[i]$ entry for MH_b is now 2, and is greater than the sequence number corresponding to MH_b in the first message (which is 1). So, BS_i will not deliver the first message again to MH_b. Thus, every message gets delivered *at most once* irrespective of the mobility of the mobiles. We assume that message delay in the wired network is low compared to the mobility of the mobiles so that no message will infinitely chase a mobile without being delivered to it. Hence, the message will be delivered *at-least once* to a mobile. The combination of *at-most once* and *at-least* once delivery results in *exactly-once delivery* delivery. It should be noted that *at least once* delivery is guaranteed assuming that a mobile does not disappear completely, in which case no

delivery can be guaranteed. If this were to happen the resources can be freed by the originator base station by having a time out mechanism.

A crucial situation arises when a mobile that has received a multicast message at a base station (refer to this as BS_{old}), sends an acknowledgment and then moves into the cell of another base station (refer to this as BS_{new}). If the request from BS_{new} for the MH_{rcvd} array corresponding to this mobile is received at BS_{old} before this acknowledgment is received, BS_{old} needs to wait for this acknowledgment to be received. It will then update the MH_{rcvd} array, and only then transfer the array to BS_{new}. The updated MH_{rcvd} array makes it known to BS_{new} that the mobile has already received that message.

4.4 Extension to the Protocol to Ensure MH-FIFO

The protocol described above ensures that the messages that originate from a base station, destined to a particular mobile, are delivered to the mobile in the order in which they are originated. Thus BS-FIFO ordering is ensured. We will now consider an extension to the basic protocol that ensures MH-FIFO delivery ordering for the messages i.e. messages initiated by a mobile, destined to a set of other mobile, reach them in the order in which they are sent.

To ensure BS-FIFO ordering, the $MH_{rcvd}[1 \ldots N]$ array was used. The MH_{rcvd} array corresponding to a mobile indicated the number (i.d.) of the next message that this mobile can receive from each of the base stations. To ensure MH-FIFO ordering, we will use a MH_{rcvd} array that will be size $1 \times M$, where M is the total number of mobiles in the network. The $MH_{rcvd}[i]$ entry for a mobile will then indicate the number (i.d.) of the next message that this mobile can receive from mobile MH_i. Further, analogous to the SEQ array maintained at each base station to ensure BS-FIFO, an array $MH_{sent}[1 \ldots M]$ will be maintained at a base station for each of the mobiles that is currently being monitored by that base station. $MH_{sent}[i]$ for a mobile MH_a will indicate the number of multicast messages that MH_a has sent to MH_i. In the multicast message, the entries from the MH_{sent} array will be sent along with the M_{id}. A mobile (say) MH_i is "eligible" to receive a message from MH_a only if $MH_{rcvd}[a]$ of MH_i is equal to the $MH_{sent}[i]$ of MH_a sent along with the message. Thus MH_{sent} and MH_{rcvd} arrays are used to ensure MH-FIFO and *exactly once* delivery in the same way SEQ and MH_{rcvd} arrays are used to ensure BS-FIFO and *exactly once* delivery. Both MH_{sent} and MH_{rcvd} arrays get transferred from one base station to another when a mobile changes base stations. BS-FIFO and MH-FIFO delivery are neither achievable at the same time nor is it necessary to achieve both at the same time. MH-FIFO is a stronger requirement, and ensuring MH-FIFO makes the BS-FIFO semantics unnecessary as messages are always sent from a mobile to a set of other mobiles and the base stations merely originate messages on behalf of the mobiles.

4.5 Modification of the Protocol to accommodate other User Location Strategies

The protocol has been described for a model in which the location information is fully replicated and updated at all the base stations. In this section, we discuss modifications to the protocol required in order for it to work under the following two models: 1) location information for a mobile is available

only at its home, for example at the HLR, 2) in addition to the information being available at the home, it may also be available in a local cache.

1. Case 1 : Location information is available only at the HLRs of the mobiles

   In this model, when a base station initiates a multicast message, the BS_{list} for the message will contain the list of the HLRs for all the mobiles in the MH_{list}. The message is then sent to the HLRs in this list. The MH_{rcvd} and SEQ array are maintained as before at the base stations. At the HLRs, the MH_{list} of the message is checked and the message is sent to the base stations that are monitoring the mobiles at that time. However, in such a case where the location information for the mobiles is distributed over different HLRs, we cannot guarantee that a message will be forwarded only once to a base station. Two mobiles belonging to two different HLRs could be located in the same base station. In this case, both HLRs will forward the message to this base station. This is an overhead that we incur as a consequence of not replicating location information. Since only one HLR is responsible for a particular mobile, when a location update takes place at a HLR for a mobile in the MH_{list}, only this HLR forwards the message to that base station. So, the issue of congestion at this new base station is not a major concern. The MH_{rcvd} array gets transferred in the same fashion as before. As soon the originating base station of a message gets the ack_list covering all the hosts in the MH_{list}, it sends a $delete$ message to all the HLRs that are in the BS_{list} of the message, which will then forward this message to the corresponding base stations in their BS_{fwd} lists.

2. Case 2 : Location information is cached

   In this model, the location information about certain mobile hosts could be cached locally at the base stations. In this case, the originating base station checks the MH_{list} and in creating the BS_{list} checks its local cache to see if location information is available for any of the mobiles in the MH_{list}. If this information is available in the cache, the message is sent to the corresponding base station and also to the HLR for that mobile. If this information is not available in the cache, the message is sent only to the HLR. Thus, the BS_{list} could contain a combination of HLRs and base stations. When a HLR receives the message, it checks both the MH_{list} and the BS_{list} sent with the multicast message. For the mobile hosts for which this HLR is the home, if the monitoring base station is not in the BS_{list}, the message is forwarded to it. When a base station receives a message, it delivers the message to the mobiles that are local to it, but *no forwarding* is done by a base station.

Thus, in both the above cases, the base stations do not forward multicast messages, as they do not get location updates. Only the HLRs do the forwarding and maintain the BS_{fwd} list for a message. The only difference between the above two cases is that if location information is cached, it is reasonable to use this information and send the message to that location (base station) instead of

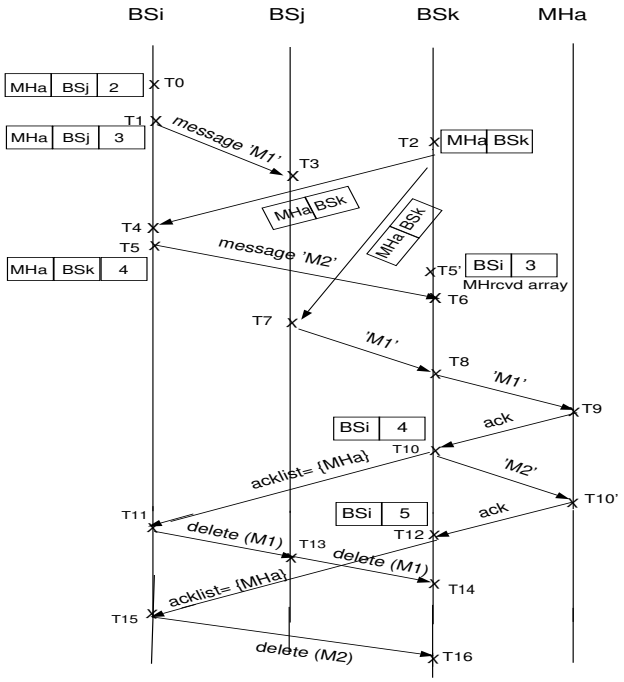

Figure 2: Event-time diagram for the protocol execution example

just sending it to the HLR. If the cached location information is current, this base station can right away deliver the message to the mobile instead of waiting for the message to be forwarded from the HLR. Note that if the cached location information is current, the HLR will not forward the message to the base station and hence the base station will receive this message only once. The *exactly-once* delivery semantics as well as the BS-FIFO delivery semantics are maintained in both the above cases. The same extension discussed earlier can be provided to ensure MH-FIFO.

5 Protocol Execution Example

Let us consider an example that illustrates the execution of the multicast protocol. The example is given based on a user location strategy that uses full replication, such as the one in [15]. The following scenario shows how the protocol ensures BS-FIFO and *exactly once* message delivery.

Consider the scenario illustrated in Figure 2. To simplify the explanation, let us consider a multicast message which includes just one mobile in its destination list. At time T_0 the location directory information at BS_i for MH_a indicates that BS_j is currently monitoring MH_a, and the SEQ corresponding to MH_a is 2. That is, BS_i has sent two multicast messages to MH_a so far. MH_a at this instant may or may not have received both the messages. At time T_1, BS_i originates another multicast message $M1$ to MH_a. The SEQ corresponding to MH_a at BS_i, $SEQ[a]$, is incremented to 3. The M_{id} at BS_i is also incremented. Then, BS_i sends $M1$ (whose structure is $< M_{id}, BS_i, < BS_j >, < (MH_a, 3) >, message >$) to BS_j. By time T_2, MH_a has received the two multicast messages from BS_i and hence its $MH_{rcvd}[i] = 3$ indicating that the SEQ of the next message

it is expecting from BS_i is 3. At time T_2, MH_a moves into the cell of BS_k. Then, BS_k makes a request to BS_j for the MH_{rcvd} array for MH_a and disseminates information about the new location of MH_a to all the base stations. At time T_3, BS_j receives message $M1$ and tries to deliver the message to MH_a. Since MH_a has moved out of its cell, this will not be successful. At time T_4 BS_i receives the location directory update for MH_a. At time T_5 BS_i originates another multicast message $M2$ to MH_a. The M_{id} is incremented again. The SEQ number corresponding to MH_a at BS_i is incremented to 4. Then, BS_i sends $M2$ (whose structure is $< M_{id}, BS_i, < BS_k >, < (MH_a, 4) >, message >$) to BS_k. At time T_6 $M2$ reaches BS_k which checks the $MH_{rcvd}[i]$ entry for MH_a. Since this entry is 3 and is less than the SEQ entry for $M2$, MH_a is not included in the *local_dest_list* of $M2$. At time T_7 the location directory update for MH_a reaches BS_j which forwards $M1$ to BS_k and adds BS_k to its BS_{fwd} list. At time T_8 $M1$ reaches BS_k. Since the SEQ entry for $M1$ corresponding to MH_a at BS_k is 3 and is equal to the $MH_{rcvd}[i]$ entry for MH_a, this mobile is included in the *local_dest_list* for $M1$ and the message is sent to MH_a. An empty *ack_list* is also created for this message. At time T_9, MH_a receives $M1$. At time T_{10}, the acknowledgment message from MH_a is received at BS_k. BS_k then increments $MH_{rcvd}[i]$ for MH_a to 4, adds MH_a to the *ack_list*, and removes it from the *local_dest_list* and MH_{list} of $M1$. Since this makes the *local_dest_list* empty, the *ack_list* is sent to BS_i.

Since $MH_{rcvd}[i]$ for MH_a is now equal to the SEQ number corresponding to MH_a sent along with $M2$, MH_a is now placed in the *local_dest_list* of $M2$ and is sent to MH_a. At time T_{11} the *ack_list* for $M1$ reaches BS_i and since the message has been delivered to all the mobiles in MH_{list} (which contains only MH_a), BS_i sends a *delete(M1)* message to the base stations in BS_{list} (which contains only BS_j). At time T_{12}, MH_a acknowledges the receipt of $M2$ and $MH_{rcvd}[i]$ for MH_a at BS_k is incremented to 5. BS_k then removes MH_a from the *local_dest_list* and the MH_{list} for $M2$. It then adds MH_a to the *ack_list* and sends it to BS_i. Thus the messages from the same base station are delivered to the mobile in the order in which they were originated. At time t_{13} the *delete(M1)* message reaches BS_j, which checks the BS_{fwd} list for $M1$ and forwards the delete message to BS_k. At time t_{14} the *delete(M1)* message reaches BS_k. Thus the $M1$ message is removed from all the base stations. At time t_{15}, the *ack_list* for $M2$ from BS_k reaches BS_i and it sends the *delete(M2)* message which reaches BS_k at t_{16}.

The *exactly once* delivery can be explained using the above scenario. Suppose at time t_{14}, the mobile MH_a moved into the cell of BS_i which has $M2$, the message will not be delivered to MH_a again because the $MH_{rcvd}[i]$ entry for MH_a is now 5 and not equal to the SEQ number corresponding to MH_a for $M2$. Thus, a message will be delivered to mobiles only once.

6 Simulation Results

The correctness of the multicast protocol was verified through a simulation study. An event driven simulation tool called CSIM [16] was used for this purpose. The protocol was tested with different number of mobiles and base stations. The overhead for the protocol was measured in terms of the number of messages that carry context information that are exchanged in the wired network even when

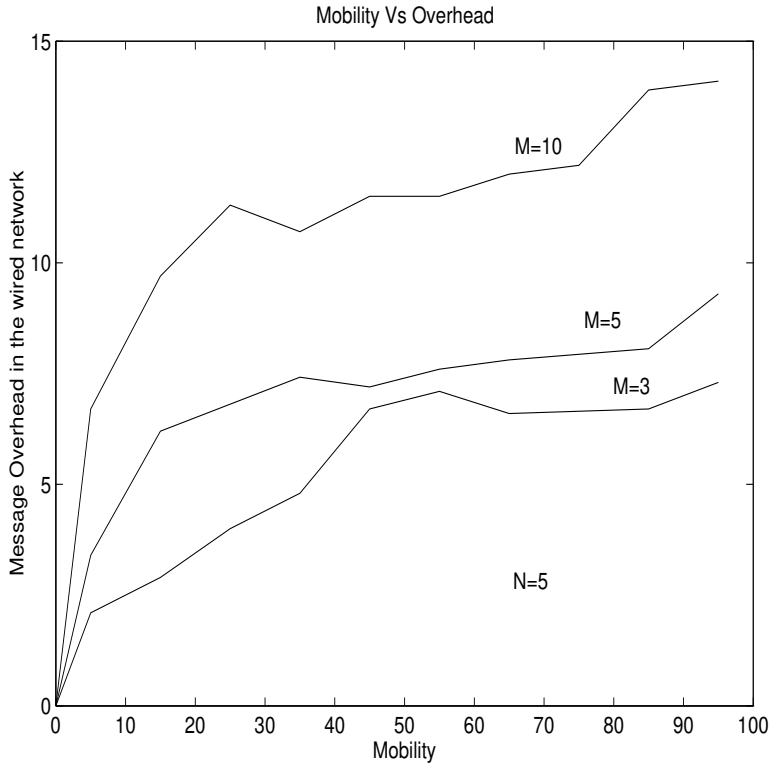

Figure 3: Context information message overhead

there is no multicast message to be transmitted. We further measured the total number of messages exchanged in the wired network during the delivery of a multicast message. These simulations verified the *exactly once delivery* and *exactly once forwarding* semantics of multicast messages. The protocol was simulated on top of a location management protocol that uses full replication of location information [11]. As described earlier, the total message overhead for delivering a multicast message to a mobile (using any location management protocol) will always be upper bounded by that of a fully replicated scheme (plus an additional message that is sent to the HLR, if required).

The simulation was carried out for different system configurations where the number of base stations and mobile hosts were varied. The number of base statiosn were varied between 5 to 10 and the number of mobile hosts were varied between 3 to 8. The number of base stations that overlap with each other also varied. The message overhead for the protocol was measured as a function of the mobility of the mobiles. The overhead due to the exchange of context specific information is incurred due to the transfer of the MH_{rcvd} array when mobiles switch base stations. This overhead however need not be considered as a "multicast" overhead because all the future PCS applications like teleporting, mobile database transactions and file accesses would require such *context sensitive* information to be passed on between base stations as the mobiles switch cells. The MH_{rcvd} array can be treated as one such context sensitive information. The simulation results illustrated in Figure 3 shows this overhead for a configuration with 5 mobiles. The cell size of a base station was fixed and hence this overhead is a function only of the number of mobiles and their mobility, and should

not change if the number of base stations change. It is seen that this overhead increases with mobility but the increase is less at higher values of mobility. This is to be expected as an increase in mobility leads to a higher rate of switching between base stations, which leads to a higher rate of transfer of context sensitive information.

Figure 4 illustrates the total number of messages exchanged on the wired network for the successful delivery of a multicast message (from the time a multicast message is initiated until the time all the resources are released at all the base stations that have a copy of the multicast message). This overhead is measured per multicast message and hence it does not depend on the number of base stations and the number of mobiles, but depends on the size of the multicast group (referred to as *destination hosts* in the figure). This number exhibits a very different behavior compared to the behavior of the messages that carry context information. The total number of messages increases with an increase in mobility up to a certain value and then starts to decrease with a further increase in mobility. This behavior can be explained as follows. For higher values of mobility, there will be an increase in the number of context information messages exchanged as a mobile constantly switches between the same two base stations. But in this case, multicast messages are not forwarded as such forwarding occurs only when a mobile enters a new base station that has not been previously visited since the multicast message was originated. Thus, the number of forwarded messages does not increase monotonically with mobility. Moreover, given that there are M base stations and N mobile hosts, an upper bound on the total number of messages in the wired network (before a multicast message is successfully delivered to all the mobiles in the multicast group) will be $2(M-1)+N$. But our simulation results have shown that the probability of reaching this upper bound is extremely small. The lower bound on the total number of messages is 0 and this situation occurs when all the destination mobile hosts are found in the cell of the originator base station.

The movement of the mobiles was simulated to be quite random. In reality, the movement of the mobiles will usually be more predictable and hence the number of messages can be expected to be less than that observed in the simulations. As we mentioned earlier, if the location information is not fully replicated, there is a smaller chance of an unnecessary forwarding. Thus the performance of this protocol if built on top of centralized location management protocol (where location information is maintained only at the HLRs) should be better. If the location information is cached, the message is forwarded to a base station according to both the cached information as well as that found in the HLRs. Thus, the performance of the protocol can be at its very best the same as that of an implementation that uses a central location directory. In the worst case, when the cache information is stale, the performance will still be not worse than an implementation that uses fully replicated location information.

7 Conclusions

We have presented an efficient multicast protocol for PCS networks that uses user location strategies to avoid sending a multicast message to all the base stations on the network. The protocol satisfies the exactly-once delivery requirement and also ensures that the messages are delivered in BS-FIFO

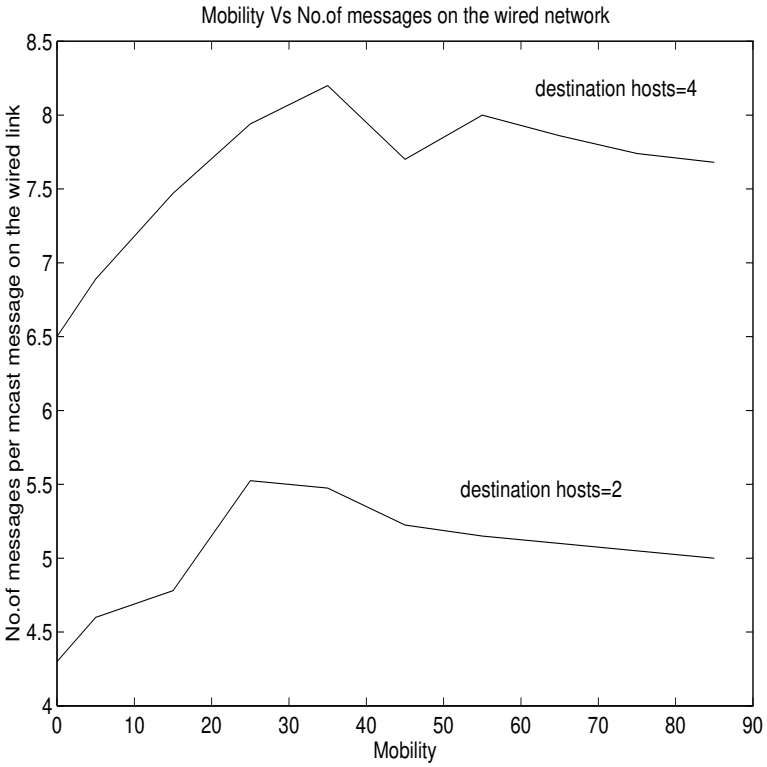

Figure 4: Message overhead per multicast message

order. Extensions to the basic protocol were described that ensures delivery of messages in MH-FIFO order. Future work includes the implementation of the protocol on a mobility test-bed.

8 Bibliography

[1] A. Acharya and B.R. Badrinath, "Delivering multicast messages in networks with mobile hosts", *Proc. 13th International Conference on Distributed Computing Systems*, May 1993.

[2] A. Acharya and B.R. Badrinath, "A framework for delivering multicast messages in networks with mobile hosts", manuscript.

[3] B.R. Badrinath, T. Imielinski and A. Virmani, "Locating strategies for personal communication networks", *Proc. of the IEEE Globecom Workshop on networking of Personal Communication*, December 1993.

[4] P. Bhagwat and C. Perkins, "A mobile networking system based on Internet Protocol (IP)", in *Usenix Symposium on Mobile and Location-Independent Computing*, August 1993.

[5] S.E. Deering and D.R. Cheriton, "Multicast routing in datagram internetworks and extended LANs", *ACM Trans. Computer Systems*, May 1990.

[6] J. Ioannidis, D. Duchamp and G.Q. Maguire, "IP-based protocols for mobile internetworking, *Proceedings of ACM SIGCOMM*, pp. 235–245, 1991.

[7] J. Ioannidis et. al, "The design and implementation of a mobile internetworking", *Proc. of Winter USENIX*, January 1993.

[8] "Cellular Radio-telecommunications Intersystem Operations" EIA/TIA IS-41 Rev. A, December 1990.

[9] R. Jain, Y-B. Lin, C. Lo and S. Mohan, "A Caching Strategy to Reduce Network Impacts of PCS", *IEEE Journal on Selected Areas in Communication*, pp. 1434–1444, October 1994.

[10] P. Krishna et. al, "Location Management in Distributed Mobile Environments", *Proc. of the Third International Conference on Parallel and Distributed Information Systems*, September 1994.

[11] K.C. Lee, "A testbed for mobile environment simulation", Master's Thesis, Department of Electrical and Computer Engineering, Northeastern University, in progress.

[12] C. Perkins, "Providing continuous network access to mobile hosts using TCP/IP", *Joint European Networking Conference*, May 1993.

[13] B. Rajagopalan, "Reliability and scaling issues in multicast communication", in *Proc. ACM SIGCOMM*, pp. 188-197, 1992.

[14] S. Rangarajan, A.T. Dahbura and E. Ziegler, "A Distributed System-Level Diagnosis Algorithm for Arbitrary Network Topologies", *IEEE Transactions on Computers*, *Special Issue on Fault-Tolerant Computing*, vol. 44, no. 2, pp. 312–334, February 1995.

[15] S. Rangarajan, K. Ratnam and A.T. Dahbura, "A Fault-Tolerant Protocol for Location Directory Maintenance in Mobile Networks", *Proc. of the 25th International Symposium on Fault-Tolerant Computing*, June 1995.

[16] H. Schwetman, *CSIM Reference Manual (Revision 17)*, Mesquite Software Inc., Austin, TX.

[17] S.F. Wu and C. Perkins, "Caching location data in mobile networking", *IEEE Workshop on Advances in Parallel and Distributed Systems*, October 1993.

[18] J.C. Wang, "A Fully Distributed Location Registration Strategy for Universal Personal Communication Systems", *IEEE Journal on Selected Areas in Communication*, pp. 850–860, August 1993.

9 Appendix

In this section, we give a specification of the multicast protocol built on top of a fully replicated location directory. The specification is provided in terms of finite-state machine descriptions of the behavior of a mobile (MH) and of a base station (BS) that execute the protocol. The finite-state machine descriptions are shown in Figure 5. In the figure, ? is used to denote the message that is received, and !, to denote the message that is sent in response. A ?/! is preceded by the name of the base station that the message is received from/sent to. If a message is to be sent to a list of base stations, then the ! is preceded by the list. For example, in $?null/BS_{list}!delete$, *delete* is send to all the base stations in the list BS_{list}. The messages that are received and those that are sent as responses are separated by a /. Events that are triggered locally are denoted without anything preceding the ? or !.

9.1 Messages

The following are the messages that are sent/received between base stations during multicasting:

- *mcast_init* refers to the multicasting of message initiated by a mobile.
- *mcast* refers to the multicasting of a message. A mcast is sent by an originating base station BS_{orig} to a list of base stations BS_{list}.
- *ack_list* is sent to the originator by each base station that has delivered the multicast message successfully to at least one mobile in the MH_{list}. ack_list contains the list of mobiles that each base station has delivered the message to. This is sent in order for BS_{orig} to determine whether all the destination mobiles have received the multicast message.
- *mcast_fwd* is the forwarding of the multicast message, and it is sent by any of the base station in BS_{list} to the base station that need to be forwarded the message.
- *delete* is sent by the originating base station of the multicast message to the base stations in the BS_{list}. But, this same message is then forwarded to the other base stations that received the multicast message by forwarding.

9.2 States of a base station

At any point of time, a base station can be in one of the following eight states.

State 1:

(*Wait*) The base station is currently not processing any multicast messages.

State 2:

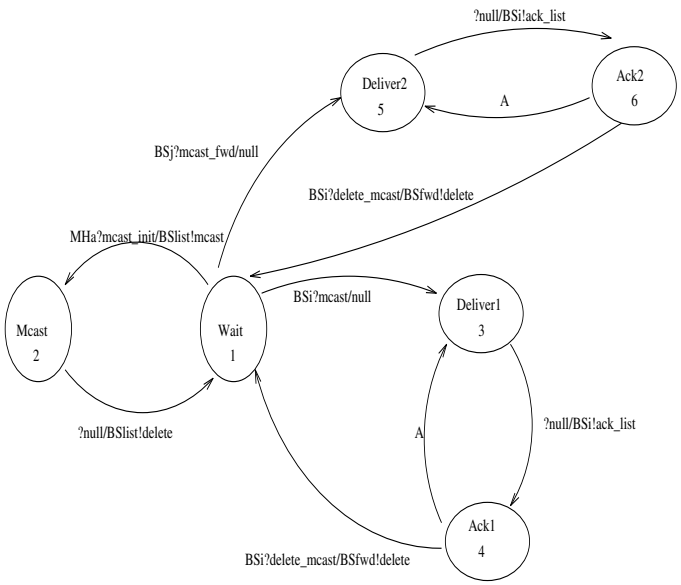

Figure 5: Base station state transitions

(*Mcast*) In this state, a base station has received a request for multicasting of message from one of its local mobile.

State 3:

(*Deliver1*) In this state, the base station has received the multicast message, and it is doing one of the following: a) processing the message, b) delivering the message to its local mobiles, or c) waiting for the acknowledgment from the mobiles.

State 4:

(*Ack1*) In this state, the base station has sent the ack_list to the originating base station, and it is waiting for the delete message or any location updates to arrive on the wired network.

State 5:

(*Deliver2*) This state is taken only by the base stations that receive the message after being forwarded by the initial set of base stations BS_{list}. This state is similar to *Deliver1*.

State 6:

(*Ack2*) This state is similar to *Ack1*, but this state is taken only by the base stations that receive the message after forwarding.

9.3 State transitions of the base stations

The details of the state transition in the finite-state machine (Figure 5) are described below.

1. Transition out of State 1:

   (a) Transition to State 2: This transition is taken only by the initiating base station after receiving a *mcast_init* message from a mobile. The base station creates a BS_{list}, and a tuple list (see Section 4.2), and sends *mcast* to all the base stations in BS_{list}.

   (b) Transition to State 3: This state transition is taken by all base stations in BS_{list} after receiving the *mcast* from the originating station.

   (c) Transition to State 5: This transition is taken only by the base stations that receives the *mcast_fwd* as part of the forwarding.

2. Transition out of State 2:

   Note here that, though an initiating base station may very well be a deliverer of the message to a set of local mobiles, for the reason of simplicity, we have not included that scenario in the finite state machine (FSM) diagram (Figure 5). Several instances of the FSM is assumed to be running at the same time, so that, a initiating base station can be in the Mcast state, and it runs another instance of FSM if it is also a deliverer. This is true for other base stations as well when they handle more than one multicast message.

   (a) Transition to State 1: Once the originating base station receives acknowledgment for the delivery of the message to all the listed mobiles, it sends a *delete* to BS_{list}.

3. Transition out of State 3 (or 5):

   States 3 and 5 are very similar, except the former is taken only by the base stations that are in the BS_{list} and the later is taken only by the base stations that receive the multicast as part of forwarding. Hence, State 5 involves no forwarding.

   (a) Transition to State 4 (6): Once the *local_dest_list* becomes empty, the base station sends the *ack_list* to BS_{orig} and moves to State $4(6)$.

4. Transition out of State 4 (or 6):

   (a) Transition to State 1: When a base station receives *delete*, it cleans all the buffers allocated for the multicast message, and goes back to *Wait* state. However, if the base station is in the BS_{list}, it sends *delete* to all the base stations it had forwarded the message, and then moves to the *Wait* state.

   (b) Transition to State 3 (5): In this state, if a mobile that has not received the message moves into to a base station's cell, that base station goes back to State 3 (5), and adds the new mobile i.dto the empty *local_dest_list*. This transition is denoted by A in the figure.