Wireless Networks 10, 423–438, 2004

Spontaneous Group Management in Mobile Ad Hoc Networks

LAURA GALLUCCIO *, GIACOMO MORABITO and SERGIO PALAZZO
Dipartimento Ingegneria Informatica e delle Telecomunicazioni, University of Catania, V.le A. Doria 6, 95125 Catania, Italy

Abstract. This paper deals with the problem of Location Management in mobile ad hoc networks where users are organized in groups. In the following this type of systems are referred to as Mobile Ad hoc Networks for Group Operations (MANGO). This paper proposes a framework for location management which exploits the trend of mobile users to spontaneously form groups in MANGOs. The management procedures required to support such spontaneous groups, which are by nature dynamic, are introduced as well. The proposed spontaneous group management is based on a hierarchical location database architecture and the concept of Group Leader, which is a terminal responsible for the location update of a group of terminals. Objective of the proposed framework is minimizing the burden on location databases and, at the same time, the signaling issued by terminals. In this paper, distributed operations required to support the whole framework are properly introduced and described. Simulation experiments have been run in order to assess the proposed scheme. Performance results show that the introduced methodology allows reduced signaling and location updating.

Keywords: ad hoc networks, group mobility, mobility management, location updating

1. Introduction

In the recent years several scenarios, where users or terminals are organized in groups and have strict communication requirements, have emerged. Some examples include military and disaster recovery operations as well as Vehicular Area Networks (VANs) and Personal Area Networks (PANs).

Mobile Ad Hoc Networks are characterized by ease and speed of deployment together with a complete independence of infrastructure. So, ad hoc networking is the natural design choice for supporting communications in the above dynamic scenarios. In the following we call *Mobile Ad hoc Networks for Group Operations* (MANGO) the networks supporting communications in such scenarios.

In MANGOs two types of communication elements can be logically distinguished: Terminals, which are the portable communication devices held by users, and Nodes, which forward packets and allow connectivity between Terminals. For example, in military and disaster recovery operations as well as in civil applications, Terminals are the portable communication devices carried by the soldiers, disaster recovery operators or users, whereas Nodes are the communication elements installed in the trucks or ambulances. Note that Terminals as well as Nodes are mobile elements.

In this paper we focus on *Location Management* which is a key component of any communication system involving mobile users [2,20]. In fact, terminal location is required for efficient and rapid delivery of information. In MANGOs, terminal location is a challenging task due to the following problems:

- *Problem 1: Wired infrastructure is absent.* The lack of an infrastructure means not having any fixed database in which storing terminal location information. Therefore, this location information must be maintained in the mobile Nodes characterized by lower processing capability than terrestrial Home Location Registers.
- *Problem 2: The probability of nodes' failures is high.* Mobile nodes are subject to frequent failures which could compromise the reliability of the location management. The unavailability of a given database would lead to the unreachability of a certain set of terminals and consequently to their isolation.
- *Problem 3: Users are organized in groups.* Several terminals may change their location at the same time. This causes a burst of terminal location update messages, which may not be processed in the due time by the databases containing the location management and may result in momentary network congestion.

The first two issues are common to all ad hoc networking environments. Solutions for Location Management problem have been proposed in [4,11,13–15,18]. However, none of them deals with problem 3.

In this paper we address the latter problem introducing a a framework for location management which exploits the tendency of mobile users to spontaneously form groups in MANGOs. Such groups are dynamic by natures and thus, appropriate management procedures are required. The proposed spontaneous group management is based on a hierarchical architecture and on the concept of *Group Leader* (GL), which is responsible for location updates of its group. It is intuitive how the use of a group leader can reduce signaling and database updates in scenarios characterized by group movements such as MANGOs. However, groups of users in MANGO are dynamic, i.e., users may change their group. Therefore, we have to introduce an efficient scheme for group management, which is not a straightforward issue because terminals have to recognize that they are leaving a group or joining a new one in an energy efficient way. In this paper, we intro-

* Corresponding author.
E-mail: laura.galluccio@diit.unict.it

duce a comprehensive framework which integrates management of location databases in a hierarchical architecture and dynamic group management in MANGO scenarios where no wired infrastructure is available. We also assess the performance of the proposed solution by simulation and estimate the improvements achieved by using our scheme.

The rest of the paper is organized as follows. The related work is described in section 2. In section 3 we show the architecture of MANGO systems. The targets of Location Management along with the issues specific of MANGOs are given in section 4. The spontaneous group management architecture supporting the location management is introduced in section 5, while the relevant group operations are described in section 6. Issues specific of Nodes' and Group's mobility management are introduced in section 7. The performance of the proposed architecture and protocols is assessed through simulation in section 8. Finally, in section 9 some conclusions are drawn and in appendix all the acronyms used throughout the paper are summarized.

2. Related work

Traditional Location Management algorithms [2] are not suitable for mobile ad hoc networks. This is due to the specific challenging features, such as lack of infrastructure and frequent nodes' failures, that ad hoc networks pose. Accordingly, several solutions specific for mobile ad hoc networks have been proposed in the recent past.

In [13] and [14] two schemes for the location management are proposed. Both of them are based on the concept of *quorum* that is a set of nodes containing location information for all the nodes in the system. This information is stored in a database distributed among all the nodes in the quorum. Several quorum coexist in the same network. In each quorum certain nodes are chosen to compose a *virtual backbone*. The choice is dynamically made looking at the current nodes stability, traffic and mobility patterns. As long as at least one quorum survives updated location information for all the nodes in the system is maintained updated. In [15], the same authors propose how to implement the virtual backbone supporting the operations required in [13,14].

In [4] each node maintains a database that stores position information about all other nodes of the network. Each entry in the database includes a node identifier, the direction and distance to the node, as well as a time value representing the age of the location information. Each node regularly floods control packets to update its position information maintained by other nodes. In order to avoid the obvious scalability problems raising from such a scheme, movement locality is exploited by using different spatial information resolution: more accurate location information is maintained in the neighborhood, less accurate information is maintained farther away.

In [18], the Grid Location Service which is part of the GRID project is introduced. The area of the ad hoc network is divided into a hierarchy of squares. n-order squares contain exactly four $(n-1)$-order squares forming the so called *quadtree*. Each node stores a table of all other nodes within the local first-order square. This table is built with periodic position broadcasts scoped to the area of the first-order square.

In [11] the concept of virtual *Homezone* is used. The Homezone of a node represents where its position information is stored. The position of the Homezone for a node can be derived applying a well-known hash function to the node identifier. All nodes within a circle of radius r and center in the node, maintain position information for the node. Accordingly, when needed, the location information of a node is retrieved in the nodes near its Homezone.

Other solutions requiring the distribution of responsibility among the location databases have demonstrated to be also very promising. To this end, hierarchical-architecture-based schemes for location management have been proposed in [3]. In [3], the proposed hierarchical solution is aimed at supporting a novel routing protocol for ad hoc networks. This architecture assumes the existence of a wired infrastructure which connects the fixed Mobile Switching Centers hosting the databases containing the location information of the nodes in their area.

All the above solutions do not solve the problem of bursts of database updates occurring when users move in groups such as in MANGOs.

In such scenarios nodes can be divided into groups of nodes with movement affinity. A node, the *Group Leader*, can be selected in each group to be the responsible for the location updates of all the nodes in the group. Recently, other efforts have been devoted in this direction, e.g., [16]. However, the algorithm proposed in [16] lacks the specification of how groups can be dynamically managed in a robust way and without wasting energy, which is not straightforward. Also, no leadership rotation is considered, which is an important problem because the Group Leader consumes more energy than the other nodes of the group.

3. System description

Mobile Ad hoc Networks for Group Operations (MANGOs) can be rapidly deployed and, for this reason, are suitable for supporting communications in military and disaster recovery operations but also in civilian environments. For the sake of illustration, in the following, we will refer to these types of operation environments. As shown in figure 1, in MANGO two types of communication elements can be distinguished:

1. **Terminals** are the portable communication devices (simply indicated by T) utilized by users and identified by a unique ID. They are characterized by:
   - *Rapid mobility.* Users move rapidly and unpredictably within the operation areas.
   - *Limited battery capacity.* Terminals are supplied by batteries. The higher the battery capacity, the heavier the device. Since the weight of these devices must be kept low in order o facilitate user movements, batteries have limited capacity.

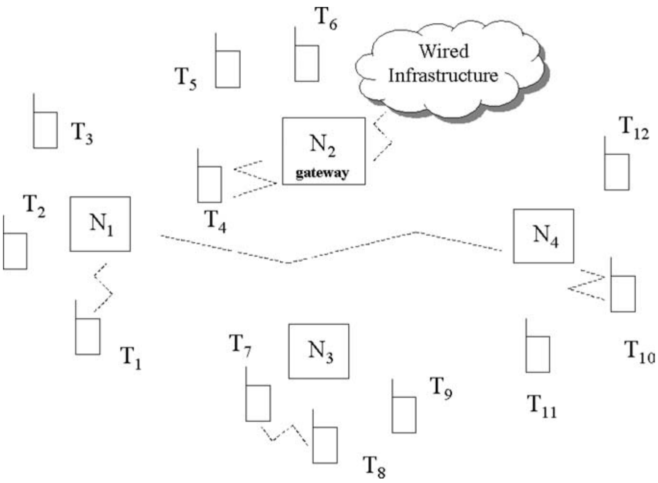


Figure 1. System architecture.

- *Short transmission range.* In order to increase battery life, transmission power is kept low, which results in shorter transmission range.

Observe that users or operators in the considered scenarios can hold simultaneously several different types of user terminals [28–30].

2. **Nodes** are communication devices (simply indicated by N) usually hosted on trucks, tanks or laptops. They provide a backbone for communications between user terminals. They also may allow terminals to communicate with entities outside the MANGO scope. Usually they are less numerous than terminals. For example, in figure 1 nodes N_1 and N_4 provide the communication path between terminals T_1 and T_{10}, whereas N_2 is the gateway which connects the ad hoc network with the infrastructured network: it allows terminal T_4 to communicate with a remote server. Let us specify that in figure 1 the dotted lines represent wireless links.
Nodes are characterized by:

- *Slow mobility.* Nodes move more rapidly than terminals within the operation area because, as we said before, we assume they are mounted on trucks or tanks. However usually, they spend most of their time being steady, whereas terminals tend to move much more.
- *High battery capacity.* Since nodes are usually mounted on trucks, the battery can be re-charged using the power generated by truck's alternator. As a result, there are not stringent battery capacity constrains.
- *Long transmission range.* Given that there are not stringent battery capacity limitations, the transmission power can be high which results in longer transmission range.

A node can be obviously a source and/or a sink of information. If this is the case, we model it as a terminal and a node coinciding.

It is worth noting that, both terminals and nodes, are usually prone to failures due to their nature and environment hostility.

One of the main features of ad hoc networks is multihopping. As a consequence localization and thus packet delivery in the considered system can be done in three ways depending on the distance between source and destination. In the following two examples we will explain better this concept.

- *Case 1: The destination is within the source transmission range*. In figure 1, terminal T_7 wants to communicate with terminal T_8 which is within T_7's transmission range. If this is the case, then the communication occurs directly between the source and the destination as shown in figure 1.
- *Case 2: The destination is outside the source transmission range and no nodes are present in the considered area*. In case no nodes are present within the given area, multihopping will be used to let the information travel throughout the network exploiting neighbors to forward the packet. This behavior is repeated until either the receiver is reached, or an available node is met.
- *Case 3: The destination is outside the source transmission range*. In figure 1, terminal T_1 needs to contact terminal T_{10} which is outside T_1's transmission range. Observe that N_1 and N_4 are the nearest nodes to T_1 and T_{10}, respectively. So, if a certain terminal is surrounded by a given number of nodes, it is straightforward that, except in case of particular situations, e.g., geographical constraints, the closest node (with respect to the terminal) is the one which guarantees the stronger signal power. In the following we will call this node the *responsible node* of the terminal. Observe that each terminal has one responsible node, whereas each node can be responsible for several terminals.

Let us say that N_1 is the responsible node of T_1 and N_4 is the responsible node of T_{10}. The two nodes N_1 and N_4 are appointed to communicate each other in order to localize T_{10} so allowing communication between T_1 and T_{10}. The choice of the routing protocol which can be used to let N_1 and N_4 communicate is out of the scope of the paper. The target of this paper is defining an efficient framework for locating users and only then, routing packets by finding the most efficient path, so that, possibly, QoS requirements can be also satisfied. This means that we want to locate users before routing a message. No a priori restrictions are done about the routing solution which could be employed. However, it is clear that some routing protocols fit better than others within the proposed framework as they could well interact with the location management protocol described in this paper. In particular, some location-aware protocols seem to be especially appropriate for the following reasons:

- **LAR** [17]. It exploits location information and, by defining an Expected Zone and a Request Zone, limits the scope of route request flood.

- **LANMAR** [9,27]. It assumes that users are divided in groups with a unique landmark identifier.
- **Terminodes Routing** [5]. It uses a combination of two addresses in order to localize destination. This allows to have, at first, an approximate localization of the terminal by employing anchored geodesic routing, and then, a more precise estimation by using an accurate final address.
- **GRID** [19]. It maintains location information of a node at certain other nodes of the network, but the density of nodes storing this information decreases as the distance from the considered node increases.

4. Terminal location management issues in MANGOs

The Terminal Location Management is aimed at allowing system to determine the position of the users and thus deliver a call to any terminal of the MANGO. With this objective, it is necessary being able to identify the responsible node of any terminal. Terminal Location Management should be:

- *Scalable*. Complexity of location procedure should be low even when the number of terminals in MANGO is very large.
- *Reliable and rapid*. Users must be located correctly within a short time interval.
- *Energy efficient*. Terminals have limited battery capacity. The number of signaling messages generated by terminals should be kept as low as possible.

The simplest solution of the location management problem might be using a fixed HLR like in cellular networks. This can give great advantages, since such a HLR leads to less probability of failures and allows to disregard scalability problems. However, in case database is far from terminals, there could be problems due to the long update messages propagation delay which may result in information inconsistencies and, thus, unreliability. A different solution is distributing location information among several fixed databases connected with each other through a wired infrastructure. In this case, information can be placed closer to terminals, therefore the long propagation delay problem can be solved. However, it still requires the use of a wired infrastructure, which in most cases is not available in the MANGO area.

If the HLR-like databases are hosted by mobile nodes, existence of the wired infrastructure within the MANGO is not required. However, this solution still presents many problems principally due to the use of distributed databases and to possible nodes failures which lead to unreliability, low scalability and lack of information consistency. In addition to the above problems, terminals may often change their responsible node, and, as a consequence, it may be required to generate frequent location update messages which consume battery energy. Obviously, when the number of users increases, updates to be executed in the databases increase as well. If the number of updates is too high, the location databases may not be able to execute them enough rapidly, which may result in inconsistencies and unreliability as well.

The characteristics of user movements in the group scenario addressed, make this problem crucial. In fact, we may think to the case of squads, battalions or numerous users standing at the bus stop, where bursts of updates may be sent to the location databases almost simultaneously.

Another simple solution for location service is not maintaining any location information at all. The system searches for the receiver through flooding the network with `Connection_Setup` messages containing the receiver identifier. The receiver sends back a `Connection_Establish` message to the source and the end-to-end path is established. This solution can be considered reliable in the sense that the receiver terminal can be located as long as it is reachable, and energy efficient because signaling for terminal location purposes is solely generated by nodes (which do not have energy constraints). However, it is worth noting that using this solution implies flooding the network with signaling, which is not desirable by itself, as it can cause serious network congestion.

The solution we will propose in this paper represents, on the opposite side, an efficient and functional solution to the above discussed problems. We will use a hierarchical approach which allows to restrict the users' search and need for updates in spite of the potential high terminals' mobility. In fact, in spite of the latter and thanks to movement locality, which characterizes scenarios under consideration, the hierarchical architecture can be positively exploited. High level tables can be, consequently, rarely updated, thus allowing to maintain updates local.

In addition to this, we will use the Group Leader concept to reduce signaling overhead. Moreover, since the Group Leader spends more energy than other terminals, leadership role rotation will be considered to achieve fairness.

Reliability is guaranteed by maintaining some redundant information to avoid possible lack of location information due to nodes' failures.

Moreover, thanks to the locality of calls, it is more likely that users will contact terminals which belong to their area, so needing to involve, in localization, only the first level of hierarchy. This, obviously, guarantees a very fast delivery of calls.

5. Spontaneous group management

5.1. Background concepts and solution's basic features

The framework for location management proposed in this paper exploits the tendency of mobile users to spontaneously form groups. Such groups are dynamic by natures and thus, appropriate management procedures are required. The proposed spontaneous group management is based on the hierarchical architecture, described in section 5.1.3, which uses the concepts of *Location Area* and *Group of Users* as introduced in sections 5.1.1 and 5.1.2. Details of the algorithms for the proposed spontaneous group management are provided in sections 6 and 7.

Observe that, differently from [3], we do not rely on any wired infrastructure because this is usually not available in the battlefield or in a meeting room and it may fail in case of disaster recovery. In addition to this, we use the concept of *Group of Users* which was not considered in [3].

5.1.1. Location area and geographical area

In our model, the operation area is divided into non-overlapping areas called *Location Areas* (LA's) according to a grid-like scheme also used in [7,18]. Objective of the location management is keeping updated the information needed to know the LA of the responsible node of any user.

The LAs are divided in disjointed sets of contiguous LAs to form the *Geographical Areas* (GAs).

To this aim, users must inform the system when they change the responsible node and the LA of the new responsible node is different from the previous one.

5.1.2. Groups of users

In our architecture a *Group of Users* is a set of terminals which update their location simultaneously by means of signaling generated by one single terminal representative of the group which is called *Group Leader*.

Each terminal keeps updated a variable which specifies its current Group ID. The latter is assigned to the group at the time of its formation by the nearest node, i.e., the responsible node. Each node can assign a set of Group IDs, in such a way that they are unique and consistent.

Example. As an example of possible solution, considering a node N_i and defining Φ_i as its set of Group IDs, it will be given by:

$$\Phi_i = \{[(i-1)N^{(\mathrm{ID})}] + 1, \ldots, iN^{(\mathrm{ID})}\}. \tag{1}$$

where $N^{(\mathrm{ID})}$ is the number of Group IDs managed by each node.

We denote the set of Group IDs which are currently unused as Ψ_i; obviously the following relationship exists: $\Psi_i \subseteq \Phi_i$. Upon request of a Group ID by a terminal (see section 6.5), the node will choose an available value from the set of unused Group IDs, Ψ_i.

Each terminal must belong to a group of users; an isolated terminal is a group by itself. Each group has a *Group Leader* (GL) which is responsible of location update for the group. The distance between the Group Leader and the other terminals of the group should not exceed a given threshold d_{Thr}. In our scenario we assume that $d_{\mathrm{Thr}} = 100$ mt because users or operators in the scenarios under consideration typically remain within that range of distance one from the other.

When the GL recognizes that its responsible node has changed, it executes the Group Location Update procedure as explained in section 7.1.

Obviously, the Group Leader consumes more energy than the other terminals of the group. As a consequence, the leadership must rotate between the terminals of the group as described in section 6.1.

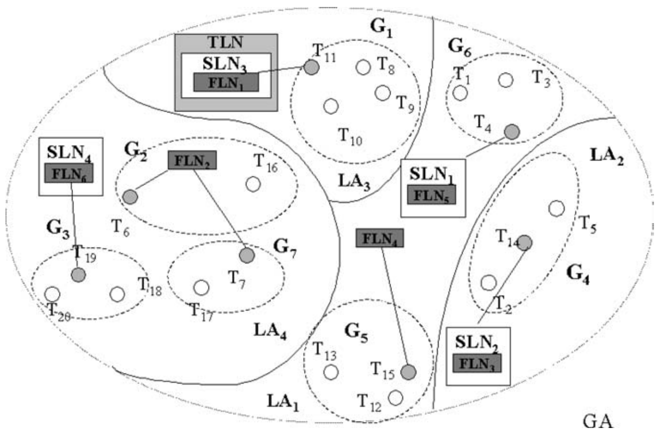


Figure 2. Hierarchical architecture.

In most cases the assignment of a user to a squad or battalion is definitive. Nevertheless, for the sake of generality, we consider that users can join or leave a group and create a new group. Also, we should consider that different groups may merge into a single group. To this purpose, the operations described in section 6 are defined.

5.1.3. Hierarchical architecture

The logical structure proposed is organized in a hierarchical tree consisting of three levels. Note that three levels are sufficient to cover the following cases:

1. Calls between terminals with the same responsible node (first level).
2. Calls between terminals within the same location area (second level).
3. Calls between terminals within different location areas (third level).

The use of additional levels would result in increased complexity without any significant performance improvements.

The hierarchical architecture is shown in figure 2 and consists of the following elements:

1. **First Level Node (FLN).** At the first level of our architecture there is a node functionality, called First Level Node (FLN), for each node. FLNs know the identifiers of all the users they are responsible for. Accordingly, each FLN maintains updated a

   - **Terminal ID → Group ID Table** (T–G Table)

   that contains its user terminal identifiers and the respective Group IDs. As an example, in table 1 we show the Terminal ID → Group ID Table for *FLN_2* which is the responsible node for groups G_7 and G_2.

2. **Second Level Node (SLN).** At the second level of our hierarchical architecture an additional functionality called Second Level Node (SLN) is defined for each Location Area. The SLN is located in one of the nodes within the Location Area. SLN maintains location information for all the terminals currently in the Location Area by keeping updated two different tables:

Table 1
Terminal ID → Group ID Table for FLN_2.

Terminal ID	Group ID
T_6	G_2
T_7	G_7
T_{16}	G_2
T_{17}	G_7

Table 2
Group ID → Node ID Table for SLN_4.

Group ID	Node
G_2	FLN_2
G_3	FLN_6
G_7	FLN_2

Table 3
Terminal ID → Location Area ID Table for TLN.

Terminal ID	Location Area ID
T_1	LA_1
T_2	LA_2
T_3	LA_1
T_4	LA_1
T_5	LA_2
T_6	LA_4
..	...
..	...

- **Terminal ID → Group ID Table** (T–G Table).
- **Group ID → Node Table** (G–N Table).

The T–G Table of a SLN contains the identifiers of all the terminals in its Location Area and their Group IDs. Therefore, it consists of the union of the T–G Tables stored in the First Level Nodes belonging to the Location Area. The G–N Table, instead, contains the responsible node's ID for each group in the Location Area. As an example, in table 2 we show the G–N Table for SLN_4 in the case illustrated in figure 2.

3. **Third Level Node (TLN).** At the third level of the hierarchical architecture, we group all the Location Areas forming a given Geographical Area. Each Geographical Area is served by a Third Level Node functionality (TLN). Each TLN contains a table in which all the terminals, whose responsible nodes are inside the Geographical Area, are registered together with their responsible node's current Location Area. We call the above table by

   - **Terminal ID → Location Area ID Table** (T–LA Table).

   As an example, table 3 is the T–LA Table for TLN.

The use of such tree-based hierarchical structure is straightforward and it is obvious how Call Locality can be exploited to improve communication performance. It is well known that hierarchical architecture helps exploiting Movement Locality [2] to decrease the number of updates per database. Moreover, the organization of terminals in groups allows a further decrease of database update at the second level of the architecture which is usually the most stressed. In fact, when a group of users changes its responsible node but not the Location Area, we only need to update the value in the G–N Table of the SLN regardless of the number of users in the group. This helps solving the problem of bursts of update requests in the databases highlighted in section 4.

Nodes may be prone to failures, therefore recovery from nodes' failures is a critical issue. In our architecture:

1. **If a FLN fails**, the SLN initiates a paging of all the groups whose responsible node was the failed FLN. The SLN can retrieve this information from its G–N Table.

2. **If a SLN fails**, then a new SLN must be selected in the Location Area. Note that the information in the SLN cannot be completely reconstructed using the information in the FLNs and the TLN. As a consequence, a backup copy of the information in the SLN must be kept updated. This further highlights the importance of minimizing database updates at the second level of the architecture. We achieved this target using the concept of Group of Users.

3. **If a TLN fails**, in order to recover its lost information the SLNs in its Geographical Area can be used. In case the TLN resides in the same node of a SLN, the backup copy of the latter can be used to recover the lost information supporting the correct behavior of the system. It is worth noting that, including a FLN, SLN and TLN in the same node would result in too much burden in terms of energy consumption, storage and computation capabilities. So eventually it would be more efficient choosing different databases for the two upper-layer functionalities.[1]

Nevertheless, even if the three functionalities are taken by the same node, the correct behavior of the system is guaranteed. In fact, should this node fail, the existing backup copy of the SLN would guarantee the lost FLN's and SLN's information recovery. The TLN's information could be recovered from all the SLNs of that GA.

Once the hierarchical architecture has been defined, the procedures to be executed to localize any terminal of the system are obvious and, therefore, not reported in this paper.

6. Group operations

As discussed above, grouping represents an efficient way to organize a certain set of users which spontaneously move together. In order to reduce the unnecessary signaling associated with these groups, it is necessary to elect a representative of all the group's users. Also, given the dynamic nature of groups (i.e., terminals can join or leave a group at any time), numerous operations are necessary to guarantee the correct behavior of the system. These operations are:

- **Group Leadership Rotation.** This operation is used to select the new representative of the group, i.e., the Group

[1] Observe that since they are placed in the same communication device, usually when a TLN (or a SLN) fails a failure of a FLN is involved as well.

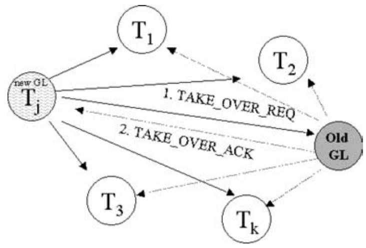


Figure 3. Group Leadership Rotation procedure.

Leader, letting the leadership rotate among all the members of the group in order to preserve the fairness of the system.

- **Group Joining.** It is initiated by a node which wants to join a group.
- **Group Querying.** This procedure is initiated by group terminals in order to verify which terminals are currently in their proximity.
- **Group Merging.** This operation can be invoked by Group Leaders when they detect another Group Leader in the closest proximity.
- **Group Formation.** It is initiated by a terminal which wants to be a group on its own and so asks for the assignment of a new group ID.

Observe that no explicit operation is defined for a terminal *leaving* a group. As it will be clear in the following, this is indeed done implicitly.

In the following sections each of these operations will be described in detail. We will see that they require the estimation of the mutual distance between terminals or terminals and nodes. Such estimation can be executed by evaluating the signal power attenuation in the wireless channel [10,26]. This results in distance values which may have a significant uncertainty. However, we stress out that the proposed operations are robust to errors in the distance estimation.

6.1. Group leadership rotation

Suppose that at time τ_0 terminal T_j belonging to the group G_i of n_i users is going to assume the leadership of G_i. Terminal T_j will inform all other terminals of the group by broadcasting a `Take_Over_Req` message containing its identifier T_j and its Group ID G_i. The current Group Leader measures the power level of the `Take_Over_Req` message and evaluates the distance from the source, d. If the distance d is lower than d_{Thr}, i.e., $d \leqslant d_{\mathrm{Thr}}$, then the Group Leader responds with a `Take_Over_ACK` message containing T_j ID, the Group ID G_i and the number of users currently in the group n_i. Accordingly, terminal T_j becomes the new Group Leader of G_i. This case is illustrated in figure 3.

If the Group Leader does not respond to the `Take_Over_Req` message, then terminal T_j will invoke a Group Querying procedure which is explained in section 6.2. When T_j sends a `Take_Over_Req` sets a timer τ_B. If τ_B expires before terminal T_j receives a `Take_Over_ACK`, a new Group Querying is performed. After hearing the `Take_Over_ACK` message, other terminals of the group generate a random variable exponentially distributed with a certain average value depending on the terminal type, the battery status and the time elapsed since the last time they had group leadership.

Example. Let us assume that at time τ_1 terminal T_l of group G_i hears the `Take_Over_ACK` message which confirms that T_j is the new Group Leader. Terminal T_l generates a random time value, x_l, exponentially distributed with average value equal to $1/(\lambda_l t_l)$; where:

- λ_l: is a dimensional constant value which may depend on the type of terminal and the battery status. The dimension of λ_l is $\sec^{-2}$.
- t_l: is the time interval elapsed since the last time T_l has been a Group Leader.

At time $\tau_1 + x_l$, terminal T_l will attempt to assume the leadership of the group.

6.2. Group querying

A terminal T broadcasts a `Group_Probing` message containing its own terminal ID. Group Leaders which receive the above message measure the received power in order to evaluate the distance d from terminal T and then generate a `Group_Advertising` message containing the Group ID, the distance d and the number, n, of terminals currently in the group. Four different cases are possible:

1. *No* `Group_Advertising` *message is received by* T. This is the evidence that there are no groups in the proximity of terminal T; therefore, T initiates a Group Formation, which is explained in section 6.5.
2. *Terminal does not receive the* `Group_Advertising` *message transmitted by its current Group Leader GL but receives* `Group_Advertising` *messages from other GLs.* This may occur because of three reasons: (i) the user left the old group; (ii) the GL left the group; (iii) the GL failed and thus it is not still available. In all the above three cases, the user must join another group if possible, otherwise must form a new group. Accordingly, if the distance information, d, contained in at least one of the received `Group_Advertising` messages is lower than d_{Thr}, the terminal initiates a Group Joining procedure with the GL which transmitted the `Group_Advertising` message with the lowest d value. Otherwise, the user initiates a Group Formation procedure.
3. *The terminal receives the* `Group_Advertising` *message from its GL but distance d is higher than* d_{Thr}. This may occur for two reasons: (i) the user is leaving the group, (ii) the GL is leaving the group. In both cases, the user must join another group if possible, otherwise must form a new group. Accordingly, if the distance, d, contained in at least one of the received `Group_Advertis-`

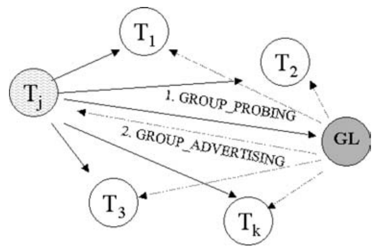


Figure 4. Group Querying procedure when the `Group_Advertising` message is received from the Group Leader.

`ing` messages is lower than d_{Thr}, then the user initiates a Group Joining procedure with the Group Leader which transmitted the `Group_Advertising` message with the lowest d value. Otherwise, the user initiates a Group Formation procedure.

4. *Terminal receives the* `Group_Advertising` *message from its GL and the distance d is lower than* d_{Thr}. In this case (figure 4), the user is still within its group and no actions are required.

Observe that Group Querying involves signaling by the terminal initiating the procedure and the GLs in the proximity, which results in energy consumption. For this reason, in our Location Management framework, Group Querying is initiated only when strictly required, i.e., when a terminal needs to verify its belonging to a group.

Nodes continuously transmit a beacon signal containing information about the node identity and its current LA. For a terminal, receiving a beacon signal from a node with a power level higher than the beacon signal from its responsible node plus a proper hysteresis threshold, δ_{HT}, is the evidence that the responsible node of the entire group is changing. If the terminal still belongs to the old group, it is likely that within a short time period, the GL will invoke the Group Location Update. Therefore, the terminal waits for a time period equal to τ_{Safety} and then, if no Group Location Update is initiated by the GL, invokes the Group Querying. Note that Group Querying is immediately triggered, i.e., before τ_{Safety} expires, by a terminal in case it wants to establish a connection.

In addition, it must be noted that Group Querying is also triggered by a terminal at the time of its turning on.

6.3. Group merging

Suppose the Group Leader GL_l of group G_l composed of n_l terminals receives a `Group_Advertising` message from the Group Leader GL_m of group G_m composed of n_m terminals. GL_l measures the power level of the received message and evaluates the distance d from GL_m. Then, if $d \leqslant d_{\mathrm{Thr}}$ and

- $n_l \leqslant n_m$, then GL_l invokes the Group Merging.
- $n_l > n_m$, then GL_l transmits a `Group_Advertising` message. Observe that GL_m will hear this message and will initiate a Group Merging of G_m and G_l.

Suppose that GL_1 wants its group to merge into group G_2. The procedure is initiated by GL_1 broadcasting a `Group_Merging_Req` message which contains the old Group ID, G_1, the number, n_1 of terminals in the group and the new Group ID, G_2. This message will arrive to the responsible node which will accomplish location updating as explained in the following, and will broadcast a `Group_Merging_Conf` message which contains G_1, n_1, and G_2. Upon receiving this message all terminals of G_1 change group, i.e., they set their Group ID equal to G_2; meanwhile GL_2 updates the value of the number of users in the group, n_2.

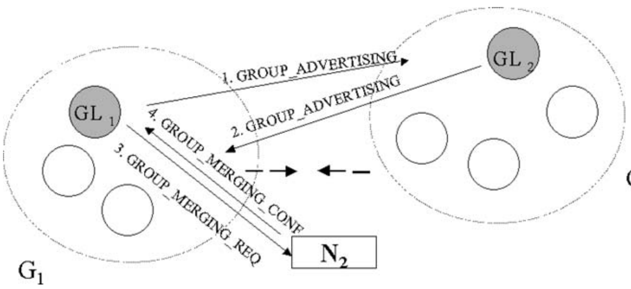


Figure 5. Group Merging procedure when the two groups have the same responsible node.

When a Group Merging occurs, the old Group ID, G_1, becomes available. Let N_h be the node which released the Group ID G_1, i.e., $G_1 \in \Phi_h$; node N_h must be informed that group G_1 does not exist anymore so that it can update Ψ_h as follows: $\Psi_h := \Psi_h \cup \{G_1\}$.

As far as location updating is concerned, we must distinguish two cases:

1. *The two groups have the same responsible node*. In this case (figure 5) only the T–G Tables in the FLN of the responsible node and the respective SLN must be updated. In particular the Group ID of the terminals of G_1 must be set equal to G_2.
2. *The two groups have different responsible nodes*. Suppose that the responsible nodes of G_1 and G_2 are N_1 and N_2, respectively. We can consider this case as the composition of two events:
   - Group G_1 changes the responsible node, i.e., it passes from N_1 to N_2 as explained in section 7.1.
   - Group G_1 merges into G_2, both under the same responsible node. This case has been dealt previously in this section.

6.4. Group joining

Suppose that terminal T belonging to group G_1, whose previous responsible node was N_1 located in the Location Area LA_1 wants to join group G_2 whose responsible node is N_2 located in the Location Area LA_2. A user can initiate the Group Joining procedure by simply broadcasting a `Group_Joining` message containing the values T, G_1, N_1, LA_1, G_2, N_2 and LA_2.

This message will be received by N_2, i.e., the responsible node of G_2, which will process it as described in the follow-

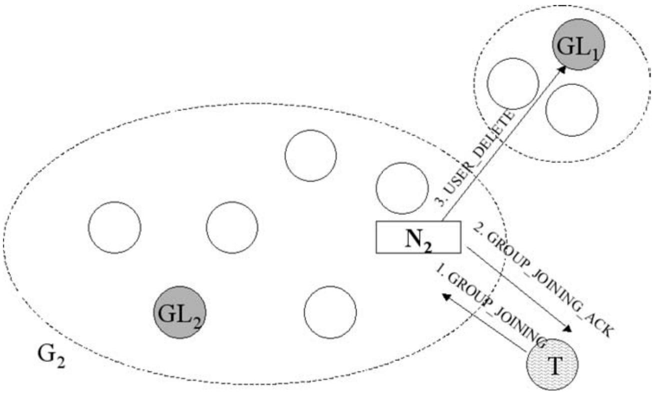


Figure 6. Group Joining procedure when a terminal is joining a new group where the responsible node is the same of its own previous group.

ing and then will transmit a `Group_Joining_ACK` message containing the identifiers T and G_2. Upon receiving this acknowledgement, terminal T will set its group ID, G, equal to G_2; whereas G_2's Group Leader increases the value of the number of terminals, n_2, by one. Note that also in the new group, terminal T maintains the value of the time instance when it will try to assume group leadership.

Furthermore, the Group Leader of the old group, G_1, will be informed by its responsible node that T left the group by means of a `User_Delete` message and accordingly decreases the number of users, n_1, by one.

As far as location updating is concerned, four different cases can be distinguished:

1. *The responsible nodes of the old group G_1 and of the new one G_2 are the same*. Upon receiving the `Group_Joining` message, the FLN within the responsible node modifies the Group ID value of terminal T in its T–G Table. Also the T–G Table in the SLN must be properly modified. The FLN triggers this operation by sending to its SLN a `SL_Group_Update` message with T, G_1, N_1, LA_1, G_2, N_2 and LA_2. This case is illustrated in figure 6.

2. *The responsible nodes of G_1 and G_2 are different but within the same LA*. Upon receiving the `Group_Joining` message, the FLN in N_2 inserts an entry for T in its T–G Table. Then, it transmits to the SLN a `SL_Group_Update` message with T, G_1, N_1, LA_1, G_2, N_2 and LA_2 as parameters. The SLN updates its T–G Table accordingly, and sends a `FL_User_Delete` message containing the terminal identifier, T, as parameter to the FLN in the old responsible node N_1. Accordingly, the FLN in N_1 deletes the entry about T in its T–G Table.

3. *The responsible nodes of G_1 and G_2 are in different LAs but within the same GA*. Initially, the operations are the same as in the previous case. Then, upon receiving the `SL_Group_Update` message, the SLN detects that T is not in its T–G Table and creates an appropriate entry. Furthermore, the new SLN sends a `TL_Group_Update` message to the TLN in order to inform it that the LA of the terminal T is changed. The parameter sent to TLN are T, G_1, N_1, LA_1, G_2, N_2 and LA_2. After that, TLN informs the SLN of the old LA that the entry of terminal T should be deleted by the T–G Table. This is achieved, through sending to the SLN of LA_1 a `SL_User_Delete` message with the following parameters: T, G_1, N_1, LA_1, G_2, N_2, and LA_2. The entry for terminal T should be deleted by the FLN of the old responsible node as well. To this aim, the SLN of the old LA sends a `FL_User_Delete` message as in the previous case and the procedure continues accordingly.

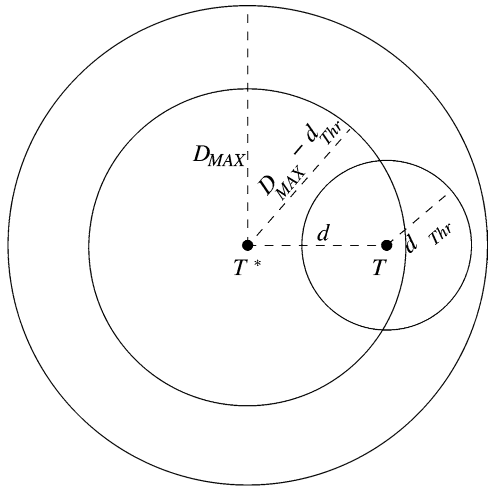


Figure 7. An example where Group Joining is triggered.

4. *The responsible nodes of G_1 and G_2 are in different Geographical Areas*. Initially, the operations are the same as in the previous case. Then, upon receiving the `TL_Group_Update` message, the TLN of G_2 detects that T was not in its Geographical Area and creates an appropriate entry in its T–LA Table. Furthermore the TLN of G_2 sends a `TL_Group_Delete` message to all the other TLNs with parameters T, G_1, N_1, and LA_1. The TLN which previously had T ID, removes the entry for T from its T–LA Table and sends a `SL_User_Delete` message to the SLN of LA_1. After that, the procedure continues as in the previous case.

A terminal can initiate a Group Joining procedure on the following of a Group Querying, as we said in section 6.2. Also, a Group Joining can be invoked by a terminal on the following of a Group Querying procedure triggered by another terminal. As an example, suppose that at a certain time, terminal T belonging to group G_1 hears the `Group_Probing` and/or `Group_Advertising` messages caused by a Group Querying procedure initiated by some other terminals. If T receives the `Group_Advertising` message broadcast by its Group Leader, GL_1, and estimates that the current distance from GL_1 is higher than d_{Thr}, then T checks the distance from the nearest Group Leader, GL: if the latter is less than d_{Thr}, T invokes a Group Joining with GL_2, otherwise it initiates a Group Querying.

Group Joining can be initiated in another case as well. Suppose that terminal T, whose current Group Leader is GL, hears a `Group_Probing` message generated by another terminal T^*, distant d. If $d < D_{\mathrm{MAX}} - d_{\mathrm{Thr}}$ and GL is less distant from T than d_{Thr}, then GL is supposed to hear the `Group_Probing` message generated by T^* and answer with a `Group_Advertising` message to T^* as shown in figure 7. If the Group Leader of T does not answer with a `Group_Advertising` message to T^*, then the distance d' between GL and T^* is higher than D_{MAX}. It follows that, if $d < D_{\mathrm{MAX}} - d_{\mathrm{Thr}}$, then the distance, d'', between T and GL is higher than d_{Thr}. In fact, according to the triangular property,

$$d' \leqslant d + d''. \tag{2}$$

From equation (2) we can derive:

$$d'' \geqslant d' - d \geqslant D_{\mathrm{MAX}} - (D_{\mathrm{MAX}} - d_{\mathrm{Thr}}) = d_{\mathrm{Thr}} \tag{3}$$

which demonstrates that the distance between T and its group leader GL is higher than d_{Thr}. Accordingly, if the distance between terminal T and the nearest Group Leader, GL_{Near}, is less than d_{Thr}, then T invokes a Group Joining procedure with GL_{Near}; otherwise terminal T initiates a Group Formation.

6.5. Group formation

Any terminal, T, can initiate a *Group Formation* procedure by broadcasting an `ID_Assignment_Req` message which contains the terminal ID. Each node N_i, receiving the `ID_Assignment_Req` message, measures the signal power level in order to estimate its distance, d_i, from terminal T, and selects a group ID, G_i, out of the unused group IDs; node N_i transmits a `Group_ID_Prop` message containing the terminal ID, the node ID, the proposed Group ID, G_i, and the estimated distance value, d_i. Terminal T will consequently receive the `Group_ID_Prop` message from several nodes. Let N_{i^*} be the nearest node, i.e.,

$$d_i > d_{i^*} \quad \forall i\colon i \neq i^*. \tag{4}$$

The terminal broadcasts a `Group_Formation` message containing its own ID, the ID, G_1, of the group which it was previously belonging to, the ID, N_1, of the node where G_1 was registered, the nearest node ID N_{i^*} and the proposed Group ID G_{i^*}. Upon receiving this message, node N_{i^*} updates the set of unused Group IDs as $\Psi_i := \Psi_i \setminus \{G_{i^*}\}$ and processes it as explained in the following. In addition, N_{i^*} transmits a `Group_Formation_ACK` message containing the terminal ID and the Group ID G_{i^*}. Node N_1 that heard the `Group_Formation` message, generates a `User_Delete` message informing the Group Leader of G_1, which accordingly updates the value of the number of users in the group n_1. Upon receiving the `Group_Formation_ACK` message, terminal T updates its group identifier, $G = G_{i^*}$ and sets the number of users in the group equal to one, i.e., $n_{i^*} = 1$. Obviously, terminal T is the initial Group Leader of group G_{i^*}.

Group Formation phase is concluded by the terminal T broadcasting a `Group_Advertising` message. This results in the activation of a Group Merging procedure.

As far as location updating is concerned, the first task to be accomplished is the creation of the group entry for G_{i^*} in the G–N Table of the SLN of N_{i^*}'s Location Area. To this purpose, the FLN in N_{i^*} sends an `SL_Group_Formation` containing the new Group ID, G_{i^*}, and the node identifier, N_{i^*} to SLN. Once the group entry has been introduced in the G–N Table, the other operations required to update the location information are the same as in the Group Joining of terminal T into group G_{i^*}. In case a certain group of users detaches from its original group and decides to create a new group, the first terminal which realizes group detachment, invokes Group Formation. Other terminals, which successively become conscious of the situation, simply invoke Group Joining procedure.

7. Mobility management

In MANGO scenarios, we must consider both group and node mobility whose management will be the subject of sections 7.1 and 7.2, respectively.

7.1. Group mobility management

While moving, a group can change the responsible node. If this is the case, then a Group Location Update procedure is required. The operations of the Group Location Update depend on the relative position of the old and new responsible nodes. We must distinguish three cases:

1. *Change responsible node but not LA.* The nodes continuously broadcast a beacon signal containing their ID and LA. Each GL calculates its distance from the nodes from which it receives the signal and selects its responsible node. Suppose that at a certain time, a group leader, GL_h, whose responsible node was N_i, detects that the closest node is now N_j. In order to change the responsible node, GL_h transmits to N_j a `Change_RN` message, with its Group ID G_h, N_i and N_j as parameters. This message is processed by the FLN in N_j which sends an `Update_RN_SLN` message to the SLN with the same parameters. The identifiers of terminals belonging to G_h are stored in the T–G Table of the SLN, which:
   - sends the FLN in N_j a `Add_Group_Users` message containing the Group ID and the identifiers of the terminals belonging to the group (accordingly, FLN in N_j updates its T–G Table);
   - sends a `Delete_Group_Users` message containing the Group ID to the FLN in N_i which deletes from its T–G Table the entries for the terminals of group G_h;
   - updates its G–N Table.
2. *Change responsible node and LA but not GA.* In addition to the procedure given for the previous case, we need to move the entries for group G_h from the SLN of the old LA to the SLN of the new LA. Furthermore, we need to update the T–LA Table in the TLN related to the GA. The

new SLN accomplishes this by sending the TLN of the GA an `Update_LA_TLN` message containing the identifiers of the terminals of group G_h and the new LA identifier.

3. *Change responsible node, LA and GA*. Differently from the previous case, two different TLNs are involved in the mobility procedure. Therefore, the information about the terminals of group G_h must be moved from the old TLN to the new TLN. To do this, the new TLN sends an `Add_Users_TLN` message to all the other TLNs. This message contains the identifier of the group. Upon receiving this message, the TLNs interrogate the SLNs and when the answers are received, the old TLN deletes the entries for all the users belonging to the group G_h in the LA.

7.2. Node mobility management

While moving, a node, N_i, can

1. *Change LA but not GA*. If this is the case, then the node informs the terminals of the groups it is responsible for, that the LA is changed by broadcasting a `LA_Change` message. Databases of the SLNs and TLN are updated using the procedures introduced in item 2 of section 7.1 for each group under N_i.
2. *Change LA and GA*. Terminals of the groups under N_i are informed as explained in the previous case. Databases of the SLNs and TLNs involved in the process are updated using the procedures introduced in item 3 of section 7.1 for each group under N_i.

Observe that due to node mobility, a group can change its responsible node. Obviously, this case can be handled as explained in section 7.1.

8. Performance evaluation

8.1. Simulation scenario

In our simulation we consider the scenario depicted in figure 8. MANGO system extends over a square area with side equal to 6 km divided in squared Location Areas with side equal to 1 km. For the sake of simplicity we assume that Geographical Areas are squared as well, and their side is equal to 2 km. Also, we assume that M_N and M_T are the number of nodes and terminals in the system.

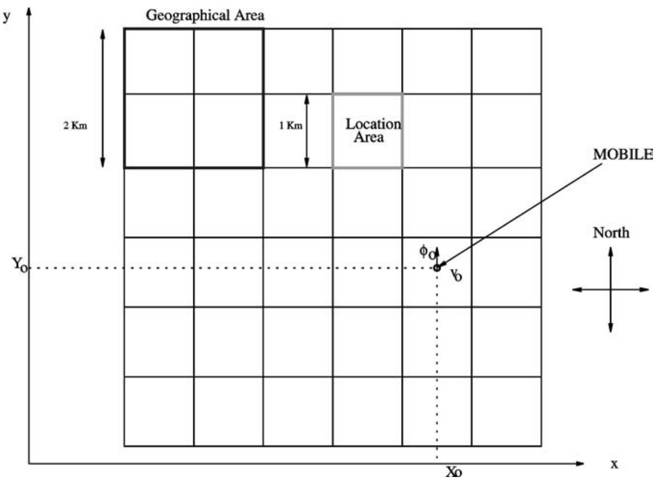


Figure 8. Simulation scenario.

In MANGO systems, users, and therefore their terminals, are organized in groups. In our simulations, we assume groups merge if they are distant from each other less than $d_{\mathrm{Thr}} = 100$ mt for more than one minute. A user leaves the current group every 30 minutes and becomes a group by himself.

As regards mobility characterization, it is worth noting that, recently, there has been much research effort to characterize mobility in ad hoc networks. However, the outcome of all this effort is rather contradictory because mobility strictly depends on the application scenario and the topology of the area [1]. Therefore, in our simulation a simple model was used for both nodes and groups. The mobility model used for nodes and groups is the same. However, in order to consider that usually nodes are placed in trucks, tanks or laptops whereas groups are composed of users walking, we use different parameters in the node and group mobility models.

The position of a mobile is described by a pair (x, y) according to a Cartesian reference system, as shown in figure 8. At any time, t_0, a mobile is characterized by a position (x_0, y_0), a direction ϕ_0 and a velocity v_0. In order to simplify the model, we consider four possible values for ϕ_0: towards North, East, South, West. Obviously, if the direction and the velocity do not change, at time t_1, with $t_1 \geqslant t_0$, the position of the terminal will be $(x_1, y_1) = (x_0, y_0 + v_0(t_1 - t_0))$ if the direction ϕ_0 is North. For other values of the direction, ϕ_0, position can be easily estimated likewise.

A mobile changes direction according to an exponential random process. We denote

- λ_{TR}: the average rate by which a mobile turns right.
- λ_{TL}: the average rate by which a mobile turns left.
- λ_{TB}: the average rate by which a mobile turns back.

Note that if we denote $\lambda_{(\phi',\phi'')}$ the rate of a mobile element changing the direction from ϕ' to ϕ'', then:

$$\lambda_{(\mathrm{North},\mathrm{West})} = \lambda_{(\mathrm{West},\mathrm{South})} = \lambda_{(\mathrm{South},\mathrm{East})} = \lambda_{(\mathrm{East},\mathrm{North})} = \lambda_{\mathrm{TR}},$$
$$\lambda_{(\mathrm{North},\mathrm{East})} = \lambda_{(\mathrm{East},\mathrm{South})} = \lambda_{(\mathrm{South},\mathrm{West})} = \lambda_{(\mathrm{West},\mathrm{North})} = \lambda_{\mathrm{TL}},$$
$$\lambda_{(\mathrm{North},\mathrm{South})} = \lambda_{(\mathrm{West},\mathrm{East})} = \lambda_{(\mathrm{South},\mathrm{North})} = \lambda_{(\mathrm{East},\mathrm{West})} = \lambda_{\mathrm{TB}}.$$

At any time the speed of a mobile can increase or decrease by δ_v. We assume that both, speed increase and decrease occurrences, happen according to an exponential random process. We define λ_{SI} and λ_{SD} the average rates of speed increase and decrease occurrences, respectively. Obviously, mobile speed cannot be lower than zero and we also impose a maximum value for mobile speed, v_{MAX}.

In table 4 we give the parameters of the mobility models used in our simulations.

As far as call establishment is concerned, in order to take call locality into account, we define a parameter, α, representing a measure of call locality. Let T_x be a terminal initiating

Table 4
Mobility parameters used in our simulations.

	Group	Node
λ_{TR} (sec^{-1})	8.33e−2	1.33e−3
λ_{TL} (sec^{-1})	8.33e−2	1.33e−2
λ_{TB} (sec^{-1})	5e−3	8.33e−4
λ_{SI} (sec^{-1})	1.33e−2	5e−4
λ_{DS} (sec^{-1})	1.33e−2	5e−4
v_{MAX} (m/sec)	1.33	8.33
δ_v (m/sec)	1.66e−1	1.66e−1

a new call and let $d(T_x, T_y)$ represent the distance between terminals T_x and T_y. The destination of the initiating call is chosen according to the following probability law:

$$\Pr\{\text{Destination is } T_y\} = \frac{d(T_x, T_y)^{-\alpha}}{\sum_{z=0,\, z\neq x}^{M_T} d(T_x, T_z)^{-\alpha}}. \qquad (5)$$

Observe that:

- if $\alpha = 0$ the probability law in equation (5) is independent of the distance, i.e., there is no call locality;
- if $\alpha > 0$ probability of calls towards neighboring terminals is higher than towards far terminals.

Obviously, the following relationship holds:

$$\sum_{y=0,\, y\neq x}^{M_T} \Pr\{\text{Destination is } T_y\} = 1. \qquad (6)$$

Moreover, we note that the higher α, the more probable are the calls towards neighbor terminals.

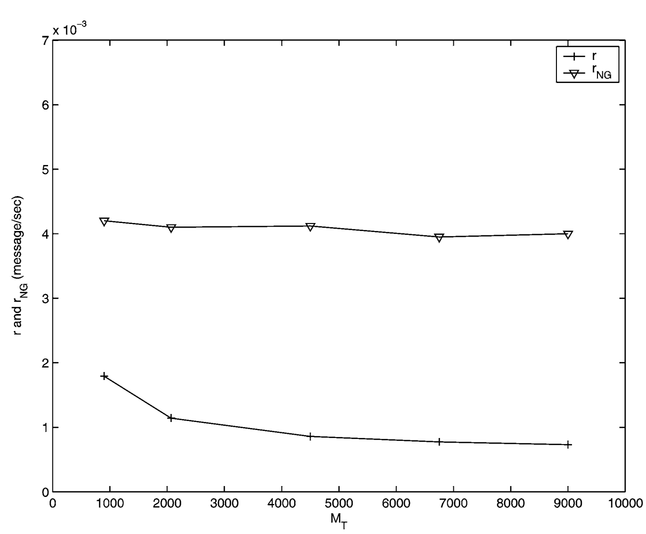
8.2. Simulation results

We evaluated the performance of the proposed scheme in terms of signaling messages issued by terminals and location information updates in the FLN, SLN and TLN tables.

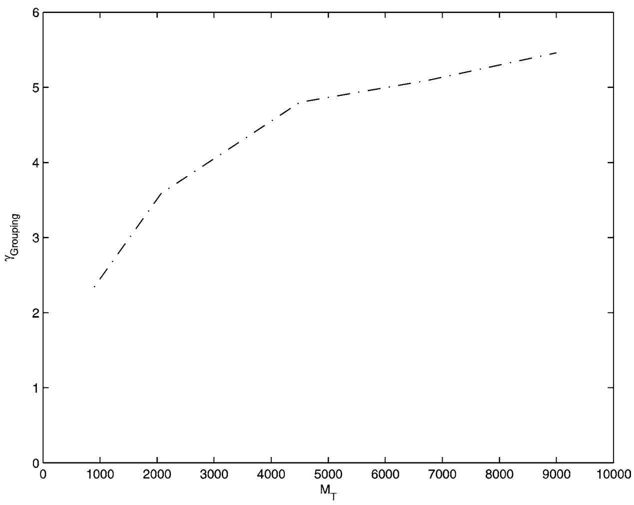
In figure 9(a) we show the transmission rate, r, of messages issued by each terminal, for location management purposes vs. the number of terminals, M_T. The number of nodes was set equal to 50, i.e., $M_N = 50$. In this plot the rate r decreases as the number of terminals, M_T, increases. This is because when there are more terminals, groups are likely more numerous and therefore, percentage of time spent by terminals as Group Leaders, is lower. In order to highlight the advantages of using the concept of group, in figure 9(a) we show the transmission rate, r_{NG}, obtained when the concept of group is not used. Using the concept of group, less messages must be generated which results in battery energy savings. This is further highlighted in figure 9(b) where we show the function $\gamma_{Grouping}$ defined as $\gamma_{Grouping} = r_{NG}/r$, which is always higher than 1.

In figure 10(a), we show the number of updates occurring per second in the T–G Tables of FLNs and SLNs, in the G–N Tables of the SLNs and in the T–LA Tables of TLNs. In the same figure, we show the total number of updates per second as well. In figure 10(b), instead, the number of updates required by each FLN, SLN and TLN are depicted. Obviously, in both plots the effort required in terms of information updates increases as the number of terminals, M_T, increases. However, the effort required by SLN nodes to keep terminal location information updated is low and almost equal to the effort required by FLN nodes.

Figure 11(a), represents the increase of r and r_{NG} as the number of nodes increases. However, r is always lower than r_{NG} and the gain function $\gamma_{Grouping}$ shown in figure 11(b) increases with M_N. Instead, figure 12(a) shows the number of updates occurring per second in the T–G Tables of FLNs and SLNs, in the G–N Tables of the SLNs and in the T–LA Tables of TLNs. In the same plot, the total number of updates per second is shown as well. Moreover, in figure 12(b), we show the number of updates required by each FLN, SLN and TLN are depicted. Figures 11 and 12 were obtained setting $M_T = 5000$ and varying the number of nodes, M_N.

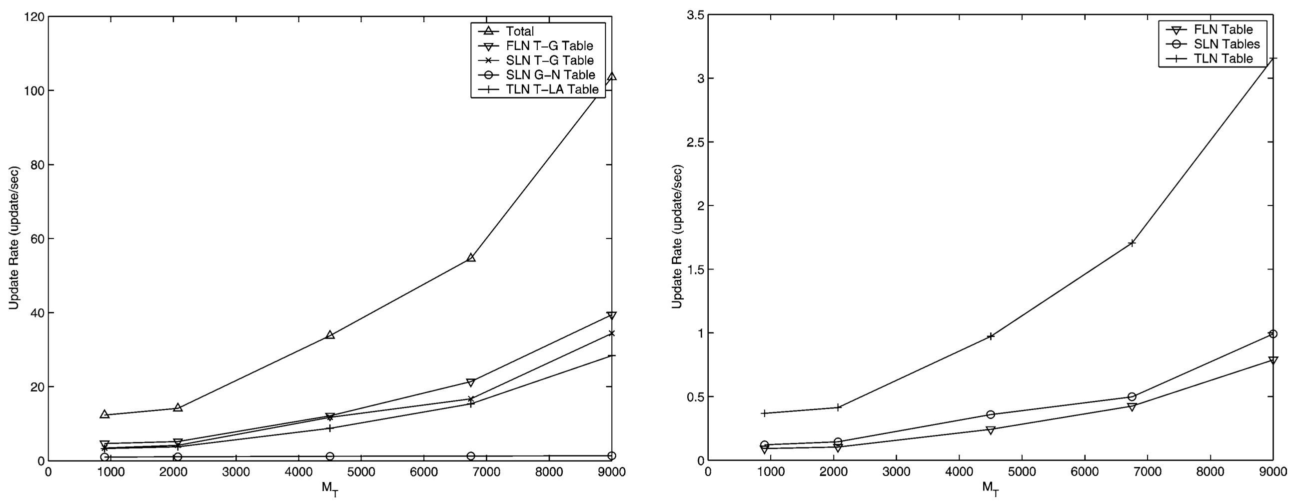


(a)



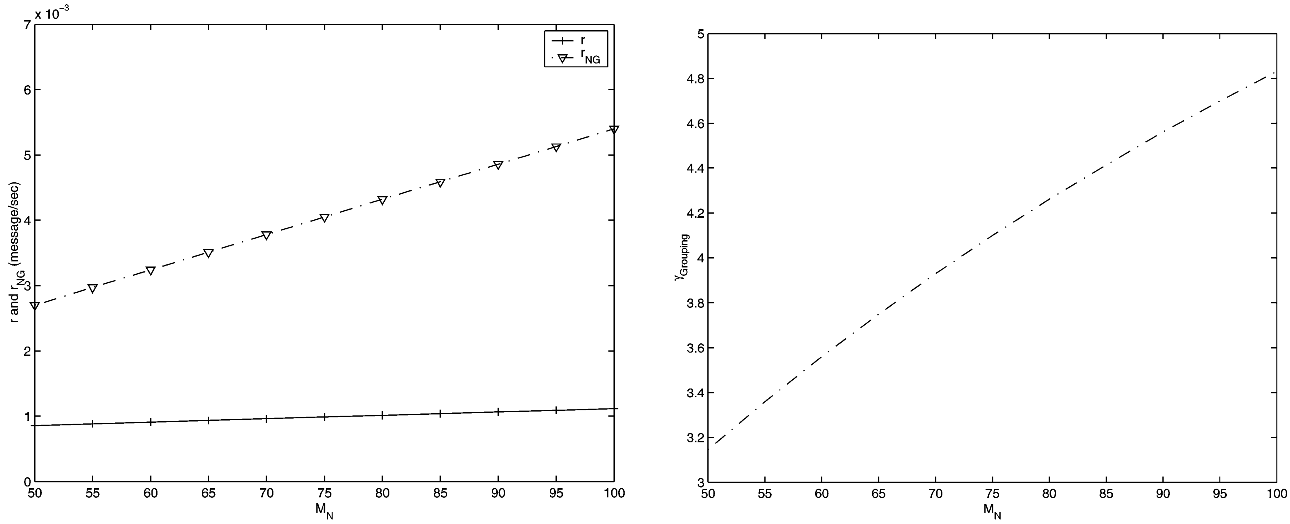
(b)

Figure 9. Mobility management signaling vs. the number of terminals, M_T.



(a) (b)

Figure 10. Location table update rates vs. the number of terminals, M_{T}.



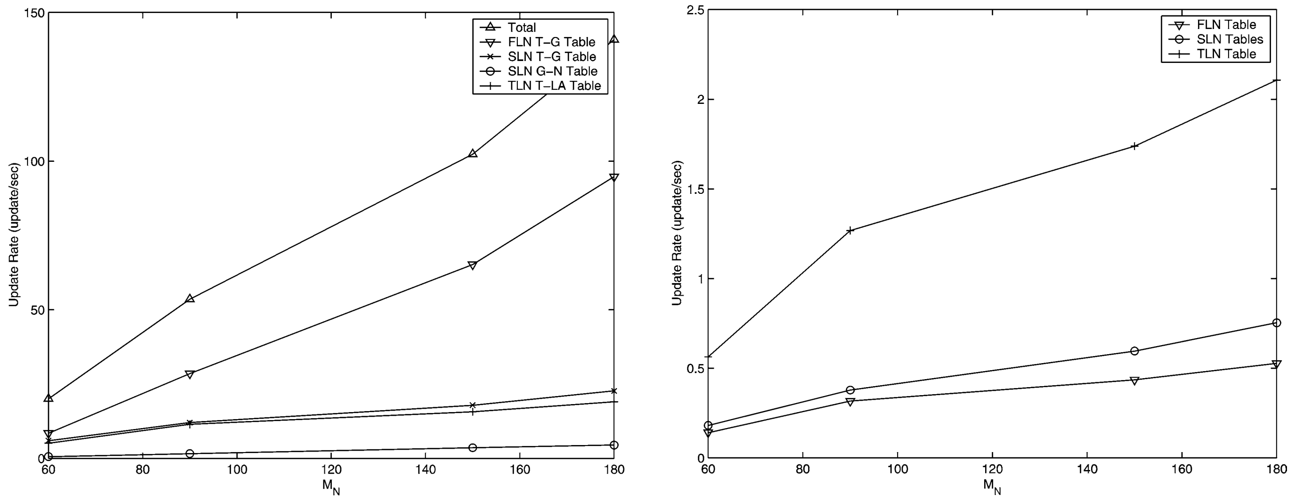
(a) (b)

Figure 11. Mobility management signaling vs. the number of nodes, M_{N}.

Suppose that a node in the communication path of an ongoing connection fails. We distinguish three cases:

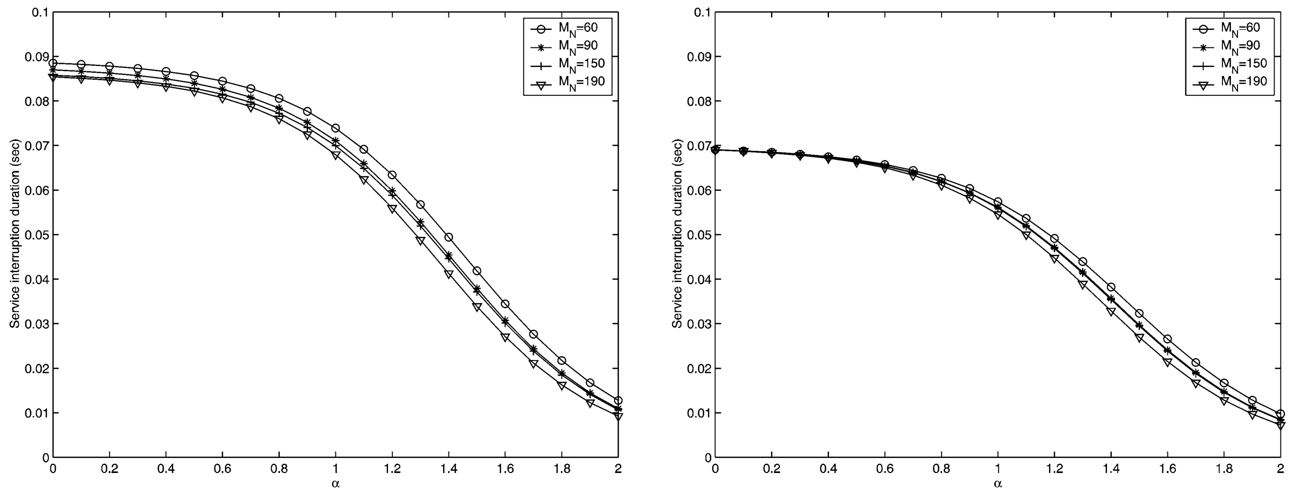
1. *The failed node is the responsible node of the source terminal T_x*. In this case, the service interruption lasts for the time required by the new responsible node of T_x to recover the information about the responsible node of the destination T_y. The average duration of the service interruption is given in figure 13(b) vs. the parameter α. As it can be seen, the duration of the service interruption decreases when α increases. This is because, when α is high, most of the connections are established between neighboring terminals and therefore, the new responsible node can obtain the information about the responsible node of the destination T_y very rapidly. Observe that even if $\alpha = 0$, the service interruption duration is very short.
2. *The failed node is between the responsible node of T_x and the responsible node of T_y*. If this is the case, the service interruption is caused by the failure of a link involved in the end-to-end communication. In this case, re-routing is executed according to the specific routing protocol in use. Performance measures can be found in [5,6,9,17,19,27].
3. *The failed node is the responsible node of the destination terminal T_y*. In this case, the service interruption lasts for the time required by the new responsible node of T_y to update the routing tables at the hierarchical levels involved in the end-to-end communication. The average duration of the service interruption is given in figure 13(a) vs. the parameter α. Observe that also in this case the service interruption duration decreases when α increases.

Note that the service interruption duration is higher in case 1 than in case 3. The reason for this is that, in case 1,



(a) (b)

Figure 12. Location table update rates vs. the number of nodes, M_{N}.



(a) (b)

Figure 13. Service interruption duration vs. the parameter α.

the re-establishment of the communication may require the partial reconstruction of the routing tables.

9. Conclusions

Terminal location is a key component of Mobile Ad hoc Networks for Group Operations (MANGO) systems because it is required for efficient and rapid delivery of information.

In this paper we introduced a framework for terminal location based on a three level hierarchical architecture and on the concept of *Group Leader* (GL), which is responsible of location updates for its group.

Our solution for Terminal Location is completely independent of the existence of a wired infrastructure and exploits the group mobility, as well as the movement and call locality characterizing MANGO scenarios.

In the paper distributed operations required to support the whole framework have been described in details.

The performance of the proposed solution has been evaluated by simulation. Numerical results show that the use of Group Leader concept gives dramatic reduction in the number of signaling messages transmitted by each terminal for location updates. This results in energy saving and therefore, longer duration of batteries.

Acknowledgements

This work was partially supported by MIUR under the VICOM project.

Appendix. List of Acronyms

MANGO	Mobile Ad hoc Network for Group Operations
T	User Terminal
N	Node
GL	Group Leader

G	Group of Terminals
LA	Location Area
GA	Geographical Area
Group ID	Identifier of a Group of Terminals
ϕ	Set of Identifiers available at a certain node
$N^{(\mathrm{ID})}$	Number of Group IDs managed by each node
ψ	Set of Identifiers available and currently not used at a certain node
d_{Thr}	Threshold distance between the GL and terminals of its group
FLN	First Level Node
SLN	Second Level Node
TLN	Third Level Node
T–G Table	Terminal ID $\rightarrow$ Group ID Table
G–N Table	Group ID $\rightarrow$ Node ID Table
T–LA Table	Terminal ID $\rightarrow$ Location Area ID Table
d	Distance between the GL and the terminal who is requesting to take the leadership
n	Number of users in a group
M_{N}	Number of mobile nodes in the system
M_{T}	Number of mobile terminals in the system
λ_{TR}	Average rate by which a mobile terminal turns right
λ_{TL}	Average rate by which a mobile terminal turns left
λ_{TB}	Average rate by which a mobile terminal turns back
λ_{IS}	Average rate by which a mobile terminal increases its speed
λ_{DS}	Average rate by which a mobile terminal decreases its speed
v_{MAX}	Maximum value of mobile terminal's speed
δ_v	Amount of speed increase or decrease at any time
r	Transmission rate issued when group leader is used
r_{NG}	Transmission rate issued when no group leader is used
$\gamma_{\mathrm{Grouping}}$	r_{NG}/r
α	Call Locality parameter

References

[1] S. Agarwal, R.H. Katz, S. Krishnamurthy and S.K. Dao, Impact of group movement on energy consumption in ad-hoc wireless networks, in: *Proc. of IEEE Infocom'01*, Anchorage, AK (April 2001).

[2] I.F. Akyildiz, J.Y. McNair, J.S.M. Ho, H. Uzunalioglu and W. Wang, Mobility management in next generation wireless systems, IEEE Proceedings 87(8) (1999) 1347–1385.

[3] I.F. Akyildiz, W. Yen and B. Yener, A new hierarchical routing protocol for dynamic multihop wireless networks, in: *Proc. of IEEE Infocom'97*, Kyoto, Japan (April 1997).

[4] S. Basagni et al., A Distance Routing Effect Algorithm for Mobility (DREAM), in: *Proc. of ACM Mobicom'98*, Dallas, TX (October 1998).

[5] L. Blazevic, S. Giordano and J. le Boudec, Self-organized terminode routing, Journal Cluster Computing 5(2) (2002) 205–218.

[6] J. Broch, D.A. Maltz, D.B. Johnson, Y.C. Hu and J. Jetcheva, A performance comparison of multi-hop wireless ad hoc network routing protocols, in: *Proc. of ACM Mobicom'98*, Dallas, TX (October 1998).

[7] B. Chen, K. Jamieson, H. Balakrishnan and R. Morris, Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks, in: *Proc. of ACM Mobicom'01*, Rome, Italy (July 2001).

[8] C. Eynard, M. Lenti, A. Lombardo, O. Marengo and S. Palazzo, A methodology for performance evaluation of data query strategies in Universal Mobile Telecommunication Systems (UMTS), IEEE Journal on Selected Areas in Communications 13(5) (1995) 893–907.

[9] M. Gerla, X. Hong and G. Pei, Landmark routing for large ad hoc wireless networks, in: *Proc. of IEEE GLOBECOM 2000*, San Francisco, CA (November 2000).

[10] J. Gibson, *The Mobile Communications Handbook* (IEEE Press, 1999).

[11] S. Giordano and M. Hamdi, Mobility management: The virtual home region, EPFL Technical Report (October 1999).

[12] S. Giordano, I. Stojmenovic and L. Blazevic, Position based routing algorithms for ad hoc networks: A taxonomy, in: *Ad Hoc Wireless Networking*, eds. X. Cheng, X. Huang and D.Z. Du (Kluwer, 2003).

[13] Z.J. Haas and B. Liang, Ad hoc mobility management with uniform quorum systems, IEEE/ACM Transactions on Networking 7(2) (1999) 228–240.

[14] Z.J. Haas and B. Liang, Ad hoc mobility management with randomized database groups, in: *Proc. of IEEE ICC'99*, Vancouver, Canada (June 1999).

[15] Z.J. Haas and B. Liang, Virtual backbone generation and mainteinance in ad hoc network mobility management, in: *Proc. of Infocom 2000*, Tel-Aviv, Israel (March 2000).

[16] X. Hong and M. Gerla, Dynamic group discovery and routing in ad hoc networks, in: *Proc. of Med-Hoc-Net 2002*, Chia-Laguna, Italy (September 2002).

[17] Y.B. Ko and N.H. Vaidya, Location Aided Routing (LAR) in mobile ad hoc networks, in: *Proc. of ACM Mobicom'98*, Dallas, TX (October 1998).

[18] J. Li, J. Jannotti, D.S.J. De Couto, D.R. Karger and R. Morris, A scalable location service for geographic ad hoc routing, in: *Proc. of ACM Mobicom 2000*, Boston, MA (August 2000).

[19] W.H. Liao, Y.C. Tseng and J.P. Sheu, GRID: A fully location-aware routing protocol for mobile ad hoc Networks, in: *Proc. of IEEE HICSS*, Maui, HI (January 2000).

[20] M. Mauve, J. Widmer and H. Hartenstein, A survey on position-based routing in mobile ad hoc networks, IEEE Network 15(6) (2001) 30–39.

[21] A.B. McDonald, A mobility-based framework for adaptive dynamic cluster-based hybrid routing in wireless ad-hoc networks, Ph.D. Thesis (January 1999).

[22] M.R. Pearlman and Z.J. Haas, Determining the optimal configuration for the Zone Routing protocol, IEEE Journal on Selected Areas in Communications 17(8) (1999) 1395–1414.

[23] G. Pei, M. Gerla, H. Hong and C.C. Chiang, Wireless hierarchical routing protocol with group mobility, in: *Proc. of IEEE WCNC'99*, New Orleans, LA (September 1999).

[24] E.M. Royer, Hierarchical routing in ad hoc mobile networks, Wireless Communications and Mobile Computing 2(5) (2002) 515–532.

[25] E.M. Royer and C.K. Toh, A review of current routing protocols for ad hoc mobile wireless networks, IEEE Personal Communications 6(2) (1999) 46–55.

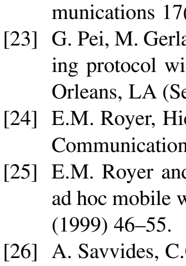
[26] A. Savvides, C.C. Han and M.B. Strivastava, Dynamic fine-grained localization in ad-hoc networks of sensors, in: *Proc. of ACM Mobicom 2001*, Rome, Italy (July 2001).

[27] P.F. Tsuchiya, The Landmark Hierarchy: a new hierarchy for routing in very large networks, Computer Communication Review 18(4) (1988) 35–42.

[28] K. Van Dam et al., From PAN to BAN: Why body area networks, in: *Proc. of WWRF Meeting*, Helsinki, Finland (May 2001).

[29] T.G. Zimmerman, Personal Area Networks (PAN): Near-field intra-body communication, M.S. Thesis, MIT Media Laboratory, Cambridge, MA (September 1995).

[30] T.G. Zimmerman, J.R. Smith, J.A. Paradiso, D. Allport and N. Gershenfeld, Applying electric field sensing to human-computer interfaces, in: *Proc. of CHI'95 Human Factors in Computing Systems*, Denver, CO (May 1995).



Sergio Palazzo received his degree in electrical engineering from the University of Catania in 1977. Until 1981, he was at ITALTEL, Milano, where he was involved in the design of operating systems for electronic exchanges. He then joined CREI, which is the center of the Politecnico di Milano for research on computer networks. Since 1987 he has been at the University of Catania, where is now a Full Professor of Telecommunications Networks. In 1994, he spent the summer at the International Computer Science Institute (ICSI), Berkeley, as a Senior Visitor. He is a recipient of the 2003 Visiting Erskine Fellowship by the University of Canterbury, Christchurch,

New Zealand. Since 1992, he has been serving on the Technical Program Committee of INFOCOM, the IEEE Conference on Computer Communications. He has recently been the General Vice Chair of the ACM MobiCom 2001 Conference. He has also been the General Chair of the 2001 International Tyrrhenian Workshop on Digital Communications, focused on "Evolutionary Trends of the Internet". Since 1997, he is a member of the Editorial Board of the IEEE Wireless Communications Magazine (formerly the IEEE Personal Communications Magazine). Recently, he joined the Editorial Boards of the journals Computer Networks, Ad Hoc Networks, and Wireless Communications and Mobile Computing. He was a Guest Editor of Special Issues in the IEEE Journal of Selected Areas in Communications ("Intelligent Techniques in High-Speed Networks"), in the IEEE Personal Communications Magazine ("Adapting to Network and Client Variability in Wireless Networks"), in the Computer Networks journal ("Broadband Satellite Systems: a Networking Perspective"). He is the recipient of the 2002 Best Editor Award for the Computer Networks journal. He also is a member of the Steering Committee of the European Virtual Center on Wireless Internet (EVC-WIN). His current research interests include mobile systems, wireless and satellite IP networks, intelligent techniques in network control, multimedia traffic modelling, and protocols for the next generation of the Internet.
E-mail: sergio.palazzo@diit.unict.it

Laura Galluccio received her laurea degree in electrical engineering from University of Catania, Catania, Italy in 2001. She is Ph.D. student with the Dipartimento di Ingegneria Informatica e delle Telecomunicazioni, University of Catania and also she currently is a CNIT Researcher in the VICOM project. Her research interests include satellite and wireless networks, ad hoc networks and network performance analysis.
E-mail: laura.galluccio@diit.unict.it

Giacomo Morabito received the laurea degree in electrical engineering and the Ph.D. in electrical, computer and telecommunications engineering from the Istituto di Informatica e Telecomunicazioni, University of Catania, Catania, ITALY, in 1996 and 2000, respectively. From November 1999 to April 2001, he was with the Broadband and Wireless Networking Laboratory of the Georgia Institute of Technology as a Research Engineer. Since April 2001 he is with the Dipartimento di Ingegneria Informatica e delle Telecomunicazioni of the University of Catania where he is currently Assistant Professor. His research interests focus on mobile and satellite networks, Quality of Service and traffic management.
E-mail: giacomo.morabito@diit.unict.it