

# Context-Awareness for Impromptu Collaboration in MANETs

Dario Bottazzi, Antonio Corradi, Rebecca Montanari  
*Dipartimento di Elettronica, Informatica e Sistemistica*  
*University of Bologna*  
*Viale Risorgimento 2, 40136 Bologna, Italy*  
*{dbottazzi, acorradi, rmontanari}@deis.unibo.it*

## Abstract

*The growing diffusion of wireless-enabled portable devices and the recent advances in Mobile Ad-hoc NETWORKS (MANETs) open a new scenario where users can benefit from anywhere/anytime impromptu collaboration. However, the development of collaborative services in MANET environments raises new challenges. In particular, unpredictable users/devices mobility, frequent disconnection and reconnection of devices and continuous changes in network topology call for novel middleware solutions to handle properly the communication between transiently collaborating partners. The paper proposes and describes AGAPE, a context-aware group communication middleware, that permits to select collaborating partners, to schedule messages and to tailor their presentation on the basis of group members context, e.g., depending on member's location, attributes, and device properties. In addition, the paper presents an emergency response application prototype to show and evaluate AGAPE functioning.*

## Introduction

The increasing diffusion of portable devices with both fixed and wireless connectivity, the widespread availability of network accessibility in living, working and amusement environments, and the emergence of Mobile Ad-hoc NETWORKS (MANETs) create novel opportunities for users to establish impromptu collaboration. In the new MANET scenario, users sharing common interests and/or goals and located on the same network locality expect to form groups on demand and to join/leave locally available groups of their interest.

However, the design, development, and deployment of collaborative services in MANET environments

raises complex group management issues. In particular, the unique characteristics of MANET environments require novel solutions to handle properly group communication and to rethink traditional group communication systems. Traditional systems are designed to maintain global and synchronized views of all interacting users and to deliver messages to all members with properties of reliability, atomicity and synchronicity [1]. The maintenance of global views and the delivery of synchronized, reliable and totally ordered multicast messages rely on the availability of network environments with high bandwidth capacity and with stable properties. However, the unreliable and asynchronous nature of MANET environments undermine these assumptions. The topology of the network cannot be statically determined making it difficult a priori assumptions on the availability and status of the collaborating entities. The members of a group appear and disappear in an unpredictable manner and often change their position in the network. Disconnection and network partitioning are common events that cause transient communication phases among continuously varying and previously unknown partners. In addition, group members operate from heterogeneous access devices with different properties in terms of computing and memory capabilities, screen size, and network apparatus.

Novel group communication solutions are required that rely on different assumptions and follow different design guidelines. In particular, we claim that the development of MANET collaborative services may benefit from context-aware message-oriented communication solutions: the selection of message recipients should depend on the applicable context and its dynamic evaluation rather than simply depending on recipient names. Different definitions of context have been recently proposed [2], [3]. In the following, we use context as the collection of any information useful to characterize the runtime situation of a communicating entity during its service session, e.g., its location and its profile including for instance, its

---

This investigation is supported by MIUR within the framework of the FIRB Project "WEB-MINDS" and by CNR within the framework of the Strategic Project "IS-MANET".

desired collaboration preferences and its properties. The location and the reciprocal position of the different interoperating parties is a key parameter to take into account to favour interoperation among co-located members, thus reducing the problems deriving in MANETs from the impossibility to rely on stable network connections and to achieve acceptable error rates in message delivery through long-length routing paths. In addition, the visibility of user/device profiles are important to permit to adapt message delivery, presentation and scheduling according to device capabilities.

The paper presents a context-aware group communication service within the AGAPE (Allocation and Group Aware Pervasive Environment) framework that provides support for the design and the deployment of collaborative applications in MANET environments [4]. As key features, the AGAPE communication support allows co-located group members to communicate via context-based point-to-point and multipoint communication patterns. In particular, according to the selected communication pattern, one (or more) group member is dynamically selected for the collaboration on the basis of its attributes and characteristics. In addition, the AGAPE group communication support permits to schedule message order and to tailor message format on the basis of the communicating entities context, such as their profile attributes.

The rest of the paper is organized as follows. Section 2 and Section 3 describe the AGAPE communication model and architecture. Section 4 provides some relevant implementation details, and Section 5 shows the applicability of the AGAPE communication solution in the context of an emergency response application scenario. Section 6 compares AGAPE with related works and, finally, concluding remarks follow.

## 2. The AGAPE Communication Model

AGAPE supports the design, development and deployment of context-aware collaborative applications in MANET environments. Collaboration in AGAPE is based on the metaphor of group of entities: only entities that are member of the same group can interoperate. Each group is characterized by a group unique identifier (GID) and by a group profile that specifies interests, preferences, activities and goals that should be commonly agreed by all group members. In particular, each group member has a personal identifier (PID) which is (statistically) unique within a group. The set of members that compose a group is not a-priori determined, but dynamically

changes due to user mobility, device disconnection/reconnection, unannounced losses of network connectivity and network partitions.

The AGAPE group model recognizes two entity roles: the Managed Entity (ME) and the Locality Manager Entity (LME) role. MEs are group members that exploit the AGAPE group management support services to collaborate. LMEs are group members that can not only collaborate, but also support group management operations on behalf of MEs: they promote the run-time creation of a new group, allow entities to join a group and maintain an updated list of co-located group members (context-dependent views).

The notion of locality is central to AGAPE and permits to effectively organize and manage groups: each LME supports group management operations for the group members in its locality. The locality of one LME is a logical grouping abstraction defined as the set of all AGAPE entities whose devices are connected to the LME device by a route path of a maximum length of  $h$  hops. The LME represents the center of one locality identified by LME geographical coordinates. The value  $h$  expresses the maximum radius of a locality measured in network hops. The value of  $h$  is chosen on the basis of the application scenario.

AGAPE group members collaborate via a context-aware group communication support. As a key feature AGAPE entities select message recipients on the basis of recipient context information instead of names. In MANET environments where it is difficult to guarantee name identity uniqueness a group member name may be un-informative or insufficiently trustworthy [5]. But even if a priori knowledge about interoperating partner names is available and is meaningful, the exploitation of names for communicating with a group member may require complex and heavy weight location tracking mechanisms that are inappropriate for MANET environments. The AGAPE group communication support provides group members with the following two basic communication patterns:

- **context-based any-cast** communication pattern that enables unreliable and asynchronous point-to-point communication between two interoperating group members. When one group member has to communicate, one and only one co-located member entity that matches a specified profile is selected;
- **context-based multi-cast** communication pattern that enables unreliable and asynchronous point-to-multipoint communication among various interoperating group members. In particular, the pattern permits to deliver the same message to all the co-located entities matching a desired profile.

It is worth stressing that these communication patterns can serve as a basis either to obtain more specific patterns, such as traditional point-to-point and point-to-multipoint group communication patterns, or to build more complex patterns, e.g., tuple spaces, with reliability, atomicity and synchronicity properties. For instance, the context-based any-cast pattern may implement traditional point-to-point communication if the profile information used to locate the communicating member reduces to its unique name and/or address, whereas the context-based multi-cast pattern may implement traditional point-to-multipoint semantics for group communication if all locally available members are assumed to have the same default profile.

### 3. The AGAPE Context-Aware Communication Support

AGAPE services are organized in two logical layers, as Figure 1 shows, built on top of the Java Virtual Machine. We detail in the following the key services for supporting ad-hoc group communication.

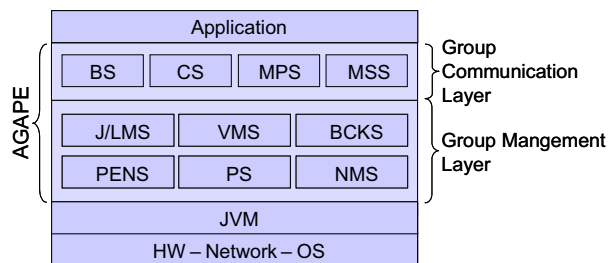


Figure 1. The AGAPE architecture.

#### 3.1 Group Management Layer

The group management layer provides required services to create/dissolve and manage groups. In particular, the *Network Manager Service* (NMS) allows all AGAPE entity's devices to send and receive UDP packets to/from the Mobile Ad-Hoc network. NMS supports both point-to-point and multipoint communication. In particular, the multipoint communication relies on the GOSSIP3 protocol and limits message dissemination within a defined number of hops [6]. The *Proximity Service* (PS) permits AGAPE members—both MEs and LMEs—to advertise their on-line availability in the locality at regular times. The *Proximity Enabled Naming Service* (PENS) randomly generates (statistically) unique group identifiers (GIDs) and ME/LME personal identifiers (PIDs) by exploiting a naming approach similar to the one proposed for P2P environments [7].

In addition, PENS senses the on-line advertisement packets of MEs/LMEs and builds a table which associates each available AGAPE entity with an entry containing its GID/PID, its role (ME or LME) and its IP address. The *BaCKup Service* (BCKS) allows LMEs to decide whether to distribute or not context-dependent views. BCKS helps to reduce un-necessary view propagation, for instance, when multiple LMEs belonging to the same group and defining the same locality attempt to disseminate all the same view to co-located group members. The *Join/Leave Manager Service* (J/LMS) allows AGAPE entities to join/leave the group and to re-qualify themselves when their profile information change. In particular, during the joining phase, all AGAPE entities provide J/LMS with their user/device profiles. If the entity is allowed into the group, J/LMS returns to the new entity the group profile and its GID/PID.

The *View Manager Service* (VMS) permits LMEs to create and disseminate group views to AGAPE group members at regular times. Each group member receives a view (context-dependent view) that contains the list of only group members located within the scope of their responsible LME locality. This list associates each group member reference (obtained from PENS) with user, device/group profile information (obtained from J/LMS). When group members connect or disconnect from the network or when they change access device and/or group profile, AGAPE reports the view changes to the members into the locality. VMS coordinates with PENS to obtain the notification of arrival, departure and disconnection of a group member entity. In particular, these event notification causes VMS to update the group view accordingly, by inserting/removing the new/old member.

#### 3.2 Communication Layer

The communication layer provides various primitives for message-oriented communication and for run-time adaptation of message presentation and scheduling. In particular, the *Communication Service* (CS) supports asynchronous unreliable message-oriented communication and implements both context-based any-cast and multi-cast patterns. To start a communication, a group member has to provide several information: a *Searching Profile* (SP) that specifies its collaboration preferences and the desired communication pattern (either context-based any-cast or context-based multi-cast). In the case of any-cast, the member is also required to specify a *Designation Criteria* that is exploited to determine the proper recipient at message delivery time. This criteria is used

only if several members match with the target profile for collaboration: in particular, random, first/last fit designation policies are available (see Section 4). According to provided information, CS delivers the messages by coordinating with the Binding Service.

The **Binding Service** (BS) is in charge of managing bindings between communicating parties. BS supports two different binding strategies: Early-Binding and Late-Binding. The Early-Binding strategy determines the set of members matching the characteristics specified in the searching profile associated to a binding request, i.e. the Target Member Set (TMS), by keeping into account the availability of members at the time the binding is created. In particular, the TMS is identified at the event of binding request depending on the Searching Profile and is kept for the whole communication session. The Early-Binding strategy assumes short-lasting communication sessions. The Late Binding strategy determines TMS dynamically each time a communication message is to be sent. In the latter case, TMS varies dynamically depending on the availability of co-located group member recipients.

The **Message Scheduler Service** (MSS) assigns dynamically a priority to exchanged messages depending on application-specific scheduling preferences. In particular, the application associates each user/device/group profile with a priority level and MSS builds a Priority Table registering these priorities. When a message is received, MSS retrieves from VMS the profile data associated to the message sender, controls whether the profile data have an entry in the Priority Table and extracts from Priority Table entries the corresponding priority levels. If no entry is found, the message is assigned a default priority level. MSS can also discard messages depending on application-specific preferences, e.g., messages coming from undesired members, and on current operating condition, e.g., limits in available memory.

The **Message Presentation Manager Service** (MPMS) permits to plug-in and to select different filters to adapt the content of exchanged messages. In particular, MPMS extracts messages from MSS and uses user/device profile information to choose the most suitable tailoring filter among available ones. Message content adaptation can involve very different operations, ranging from data filtering to data transcoding and down scaling. For instance, down scaling operations convert images from GIF to JPG format when delivering images to a resource-limited mobile device.

#### 4. Implementation Details

We have developed two different AGAPE middleware prototypes. One release is for supporting LME management operations and includes all AGAPE services. This release fits portable devices with rich computational resources (large amount of memory, file system, powerful cpu, and long-life batteries). The other AGAPE release supports ME operations and includes only a sub-set of AGAPE services: the J/LMS and VMS client side, along with the CS, BS, MSS, PS, PENS and NMS. Let us note that MPMS is not installed in the release for a resource-constrained device being too high the computational load required for supporting message tailoring. This section details the service interaction for group communication and focuses on the functioning of BS and CS, that are the key communication services.

When a group member requires to establish a communication the CS coordinates with the BS to build a binding between the two interoperating entities. In particular, BS retrieves from the client-side VMS component the currently available context-dependent views and filters them according to the preferences specified in the communication request, i.e., in the Searching Profile. Then, BS builds the list of all eligible members for message delivery (the TMS), associates the group member with a Binding Table (BT) that records all its active communication sessions and returns an handler that the group member exploits to send messages over the established communication channel (see Figure 2A). Each handler uniquely identifies an entry in the BT. Each entry in the BT represents a specific binding and includes several fields: the SP, the TMS, the IP address of all members into the TMS, the desired communication pattern, the binding strategy and, in the case of the context-based any-cast communication pattern, the specified designation criteria.

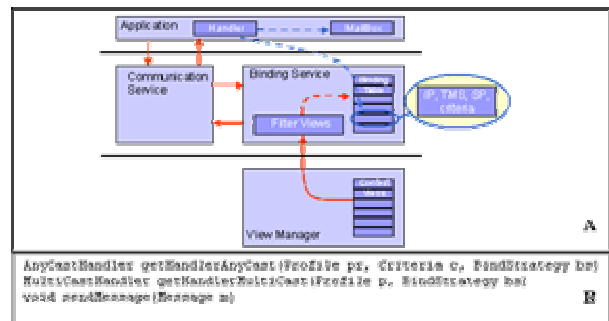


Figure 2. Creation of handlers and BTs (Figure 2A), and AGAPE communication primitives (Figure 2B).

As Figure 2B shows, application developers can benefit from two different primitives to obtain a binding: the *getHandlerAnyCast* primitive that enables any-cast communication, and the *getHandlerMultiCast* primitive that implements multicast communication. In both cases the application must specify the profile of desired partners for collaboration along with the selected binding strategy (either early binding or late binding) and is returned with an handler. In addition, in the case of any-cast the profile designation criteria must be specified. The designation criteria is selected within a list of available designation criteria. In the current AGAPE prototype the list includes random designation, and first/last fit designation criteria. In the case of random designation criteria a target member is randomly selected among all the members referred by the TMS associated to the binding, i.e., among all members that match the desired profile for collaboration. In the case of first (last) fit designation criteria the first (last) available member into the TMS is selected. However, we are extending current AGAPE designation capabilities to allow application designers to exploit more complex designation criteria, e.g., a criteria that allows to choose message recipients on the basis of attributes that maximise (or minimise) an application-dependent cost function.

Once obtained a binding, the application exploits the *sendMessage* primitive to send messages to the selected collaborating partners. Note that, while in the case of any-cast communication the *sendMessage* primitive delivers the message to one and only one partner that is selected according to the member profile, the binding strategy, and the designation criteria, in the case of multi-cast communication the message is delivered to all actually available collaborating partners that are in the TMS of the Binding Table entry referred by the handler.

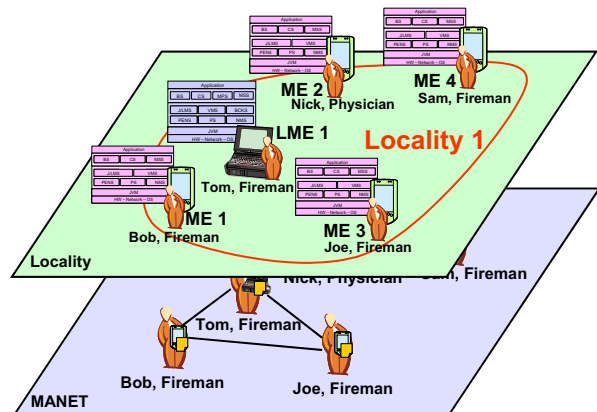
At run-time bindings may need to be updated to reflect the current availability of addressed members. The update varies depending on the binding strategy chosen. In the case of the Early Binding strategy the TMS associated to the binding is never updated once it has been setup. In the case of Late Binding the TMS is updated according to the actual availability of possibly collaborating partners. To update a binding, the Binding Service checks whether changes into the context-dependent view are related to members referred by any binding into the binding table. In particular, let us consider the case of any-cast binding strategy. When the actually selected message recipient becomes unavailable, a substitute must be identified. In particular, the new recipient is chosen by exploiting the TMS associated to the binding. It is worth noticing

that, according to the specified designation criteria, only one member is selected among the available ones.

## 5. Case Study

For sake of illustrating the AGAPE communication support let us consider a simplified emergency response application scenario. The application puts together emergency response operators, such as firemen and physicians, cooperating within the same area to provide prompt assistance to people in need of help. The application provides emergency response operators with the visibility of their co-located colleagues and allows them to communicate and share instructions or images, such as maps and building plans, via messages exchanges. Emergency response operators typically must make important decisions quickly by relying on incomplete information and the visibility of co-located colleagues can facilitate emergency response operators decision making. In fact, emergency response operators can promptly ask neighbours for all details (necessary) to have a clear picture of the intervention scenario. Let us note that the possibility to exchange written messages is of paramount importance for disseminating instructions and warning in this scenario, because rescue operation often occur in noisy environments where it is difficult to communicate via audio channels [8].

Different users have different roles and competences and, as a consequence, they are characterized by different profiles. These profiles, along with the used access terminal characteristics, have been modelled as CC/PP profiles [9]. For description simplicity let us suppose that the profiles include the only operator's name and skills (e.g. fireman, physician, and so on).



**Figure 3.** Deployment settings of the emergency response application prototype.

In our scenario users can interoperate in a MANET. In our prototype, in particular, as depicted from Figure 3, the MANET is dynamically constituted by exploiting

802.11b-enabled wearable devices operating as laptops that permit user hand-free interoperation and iPAQ PDAs provided to the different emergency response operators [10]. All wearable devices run Linux, J2SE 1.4 and the AGAPE release for resource rich devices while all PDAs run Linux Familiar, Personal Java and the AGAPE release for resource constrained devices. We have implemented the MANET infrastructure by installing the AODV routing protocol on each mobile device [11]. Moreover, due to the lack of standard addressing schema for MANETs, we have statically configured device IP addresses. These deployment setting choices do not undermine the generality of results. We have designed the AGAPE infrastructure without any assumption of the availability of a specific routing protocol or addressing schema.

In the following we detail two main AGAPE group management aspects: group creation/joining and communication.

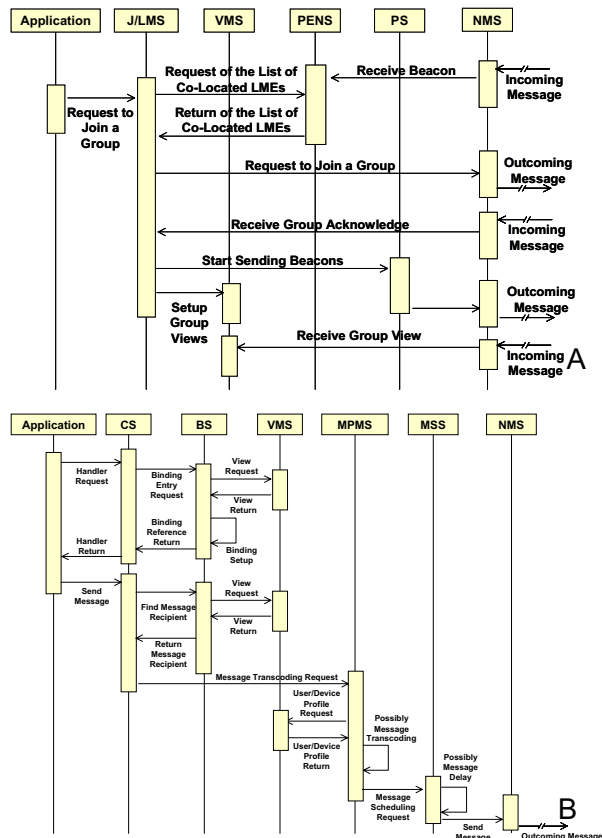


Figure 4. AGAPE service interaction diagrams for group joining (Figure 4A), and group communication (Figure 4B).

**Group Management.** When a fire brigade receives a request for intervention, it drives to the fire scene. While approaching the fire scene, the brigade captain

promotes the dynamic formation of the emergency response group, on the field. The application client module of the captain device allows him to specify the group profile—in our example “Emergency Response”—along with the user profile attributes—for example “Tom, Fireman”. His lap-top acts as an LME and exploits the locally installed PENS to generate the GID/PID, the VMS to initialise context-dependent views and the PS to advertise the on-line availability of the new group. Since no other co-located LMEs are available, the VMS installed on the captain device is the one LME that propagates context-dependent views that initially include only the captain GID/PID and her profile. All devices connected to the MANET that are placed within the captain locality acquire the visibility of the captain LME.

The other emergency response operators, such as firemen and physicians specify the required group profile, i.e., “Emergency Response”, the user profile, e.g., “Bob, Fireman”, and attempt to join the group promoted by the captain’s device. The J/LMS service installed on the firemen PDAs discovers the LME by coordinating with the locally available PENS instance and allow firemen to join the group (see Figure 4A). The J/LMS instance installed on the captain device allows the new members to enter the group and coordinates with locally available PENS to obtain a PID for the new members. In addition, the J/LMS installed on the captain device coordinates with locally available VMS to update the context-dependent views with the profile data of new members. Similar operation occur for the joining of the other emergency response operators, e.g., physicians.

**Group Communication.** Let us now consider how AGAPE handles communication for the any-cast communication pattern. Similar considerations apply for the multi-cast communication pattern. In our application prototype the any-cast communication pattern is used by a fireman to alert one physician—regardless to her identity—to ask for her intervention: similar operations occur for multi-cast communication pattern. As depicted in Figure 4B, to communicate the alert message the application on the fireman’s lap-top must obtain a communication handler. To this aim, the application requires an handler of the specified type (any-cast in our example) through the *getHandlerAnyCast* primitive. The CS requires a binding to the BS that creates a new entry into the Binding Table. Then, because the early binding strategy has been selected, BS filters the context-dependent view obtained from the VMS to identify all members whose profile matches the provided Searching Profile, i.e. all co-located physicians. According to the specified designation strategy, one of

them is selected for collaboration. The BS therefore returns a reference to the requested binding into the binding table to the CS that can return the communication handler to the application. The application exploits the obtained communication handler to deliver the message to the selected physician via the *sendMessage* primitive. Note that in this example, the Early Binding strategy is adopted because it is necessary to provide firemen with the possibility to refer the same physician for the delivery of several consecutive messages. If the fireman has to interoperate with a neighbour colleague, e.g. to get data such as the temperature of the environment, a Late Binding approach could be more suitable. In fact, in that case, any co-located colleague with a device equipped with a thermometer could provide the information.

As Figure 4B depicts, all messages exchanges require both scheduling and format adaptation. Let us focus on the interactions involving a physician to describe AGAPE message scheduling and format adaptation support. One physician may be involved in multiple emergency scenarios, e.g., when more than one co-located firemen have asked for help. The physician can be the primary responsible caregiver only in one emergency case at a time, whereas in several others she can only provide advices. It is obvious that the physician assigns a higher priority to her care-related messages. Other messages may be delayed or even discarded if the physician is already occupied. In particular, the application can associate each user/device/group profile with a priority level. According to the specified preferences, when a message is sent (or received), MSS retrieves from VMS the profile data associated to the message recipient (or sender), and assigns a priority level to the message. The priority level rules the transmission order of sent messages and the presentation order of received messages. When the physician completes her tasks, she can change message priorities to reflect new preferences and needs.

Message format adaptation occur when both messages are received and sent to accommodate different device properties. For example, a physician requires to interoperate in an hand free fashion while assisting a patient, MPMS installed on the physician device extracts incoming messages and, for example can transforms them from text-to-speech based format (e.g. by exploiting the java speech api). Message format adaptation also takes place in message sending, e.g., when a physician sends a message to another group member that operates via PDA, the message format must be tailored to fit the limited resource capabilities of the PDA. MPMS on the physician

device coordinates with VM to retrieve the device profile and transforms message formats accordingly before sending.

## 5.1 Experimental Results Evaluation

During the deployment of the emergency response prototype, we have conducted a number of measurements to evaluate the overhead and the amount of resources required by AGAPE to support group management and communication. Because view management and beacon dissemination are the main performance impact factors in AGAPE, we here specifically focus on their introduced overhead. In particular, we have evaluated the network overhead and memory occupation that are critical parameters for resource-limited devices.

**Network Overhead.** We have measured AGAPE services overhead in term of used bandwidth. Measurements have been taken on a MANET constituted by one lap-top acting as an LME and a variable number of PDAs acting as MEs. Measurements have been taken in two different operating conditions: in the worst operating case to obtain an upper-bound for the AGAPE services overhead and in ordinary operating conditions.

**Worst Case.** The worst case deployment setting is constituted by a dense MANET where the devices of all members in the locality are in reciprocal visibility, i.e. all members are placed within 1 hop distance. We consider this situation “the worst case” because all members re-transmit all received views and beacons, thus introducing a high network traffic.

Figure 5A shows the overhead determined by beacon dissemination. The results are obtained by considering an average beacon message size of 21 bytes, a number of ME devices allocated within the transmission radius of the LME (NME) that varies from 1 to 19 PDAs and a variable transmission time interval between two consecutive beacons ( $T_b$ ) (5, 10, 15, 20 and 25 seconds).

Figure 5A shows that the used bandwidth grows quadratically with the number of MEs in the locality. In addition, the used bandwidth obviously decreases for growing intervals of time between two consecutive beacons.

We have also measured the overhead due by view dissemination. In our prototype, group view entry size is on the average of 512 bytes. Figure 5B shows the overhead of view dissemination with a number of MEs devices allocated within the transmission radius of LME device that ranges from 1 to 19 and with a variable transmission interval time between two

consecutive group views dissemination ( $T_v$ ) (20, 40, 60 and 80 and 100 seconds).

The used bandwidth grows quadratically with the number of MEs devices in the locality. In addition, the used bandwidth decreases for growing values of  $T_v$ .

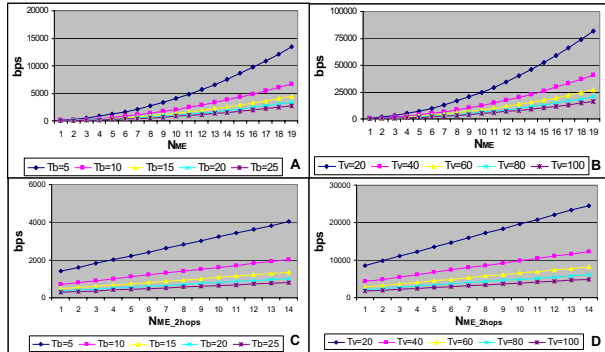


Figure 5. Beacon and view dissemination overhead.

**Ordinary Operating Conditions.** In our application scenario, emergency rescue operators typically collaborate via a multi-hop network. We have deployed a MANET network constituted by one laptop (acting as LME), 5 PDAs (acting as MEs) that are allocated at 1 hop distance from the lap-top, and a variable number of PDAs (acting as MEs) that are placed at 2 hops distance from the central lap-top.

We have investigated the bandwidth used for beacon dissemination. In particular, we have measured the overhead within the area covered by the transmission radius of the LME device because of its group management responsibilities. In our measurements we have varied the number of MEs devices allocated at a 2 hops distance from the LME device ( $N_{ME\_2hops}$ ) from 1 to 14. In addition, we have varied the interval of time  $T_b$  between the transmission of two consecutive beacons by testing values of 5, 10, 15, 20 and 25 seconds.

Figure 5C shows that the used bandwidth increases linearly with the number of MEs allocated at 2-hops from the LME device ( $N_{ME\_2hops}$ ). In addition the used bandwidth decreases when we increase the interval of time between two consecutive beacons ( $T_b$ ).

We have also measured the amount of bandwidth required by group view propagation. In particular, we have measured the overhead within the area covered by the transmission radius of the LME device. We have varied the number of MEs allocated at a 2 hops distance from the LME device ( $N_{ME\_2hops}$ ) from 1 to 14 and we have varied the interval of time  $T_v$  between the transmission of two consecutive group view by testing values of 20, 40, 60, 80 and 100 seconds.

Figure 5D shows that the used bandwidth linearly increases with the number of MEs allocated at 2-hops

from the LME device ( $N_{ME\_2hops}$ ). In addition the used bandwidth decreases with increasing values of  $T_v$ , i.e. the interval of time between two consecutive group views.

**Memory Requirements.** We have measured the memory occupied to support view, beacon and binding management in order to evaluate AGAPE suitability for resource constrained devices.

With regard to beacon management, the main factor that impacts on memory occupation is the table that associates any available AGAPE entity with an entry containing its GID/PID, its role (ME or LME) and its IP address. Each table entry is 21 bytes in size.

Context-dependent views can also require a high memory capacity. The memory occupation is determined by the list that stores the views. In particular, each entry in the list has an average size of 520 bytes.

Finally, we have measured the memory occupation required to support binding management. In particular, the binding table handled by the BS determines most of the binding management memory usage. Each entry in the binding table is associated to a communication session and has a variable size depending on several factors: the Searching Profile (in our prototype, on the average, it has a size of almost 512B), the number of addressed entities into the TMS and, for context-based any-cast communication pattern, the specified designation criteria (in our, prototype it has a size of 1B). Let us note that the TMS has a variable length depending on the number of addressed group members and that each entry into the TMS has a size of 12B.

Table 1 shows the memory usage for beacon and view management. In particular, we have measured the memory required by one ME placed in a locality constituted by a variable number of group member entities. Similar considerations apply to LMEs.

Group Members	5	10	15	20
Beacon Management	105 B	210 B	315 B	420 B
View Management	2600 B	5200 B	7800 B	10400 B

Table 1. Memory occupation for beacon and view management.

The measurements show that the memory required both for beacon and view management linearly grows with the number of members into the locality.

Table 2 details the memory used by an AGAPE member for handling an increasing number of communication sessions with a varying number of addressed members per communication session.



		Open Sessions			
		2	4	6	8
Addressed Members (TMS)	2	1072 B	2148 B	3222 B	4296 B
	3	1098 B	2196 B	3294 B	4392 B
	4	1122 B	2244 B	3366 B	4488 B
	5	1146 B	2292 B	3438 B	4584 B

**Table 2.** Memory occupation for binding management.

The memory usage for the binding management linearly grows with the number of open connections and also linearly grows with the number of addressed group members.

Let us note that, as the above measurements show, AGAPE requires a limited amount of memory to properly work and, consequently, it is suitable for resource-constrained devices.

## 6. Related Works

Few recent research efforts have been directed toward the development of innovative support solutions for collaborative applications in ad-hoc network environments. As it has been highlighted above, two research mainstream solutions can be identified [12], [13], [14], [15].

On the one hand, Profile-Based collaborative systems exploit the visibility of contextual information, such as the availability and the characteristics of neighbour entities, to support group communication. Profiles are the means that convey contextual information among the various interoperating entities. Proem is a significant example of Profile-Based system [12]. In Proem entities are assumed to belong to the same group as long as they are co-located and implement the same application-level protocols, e.g., a file sharing protocol. The visibility of available neighbours along with their profiles is propagated up to the application level and is used to dynamically determine message recipients at execution time. Neighbours entities broadcast profiles at regular times.

AGAPE extends the profile-based collaboration model proposed in [12]. In particular, AGAPE programmers express the desired contextual information that interoperating entities should exhibit without the need to explicitly handle context information. The AGAPE middleware provides the needed services to manage context gathering and to handle the binding and the communication among interoperating entities. In addition, AGAPE clearly decouples group management from group

communication features by providing different sets of primitives and support services to arrange/dissolve groups in MANETs and to support communication.

On the other hand, Global Virtual Data Space (GVDS)-based collaborative systems promote a context-transparent group communication model [13], [14], [15], [16]. GVDS systems constitute a single space as the union of all the partition of the GVDS allocated on connected entities. According to the operation defined on the data space, each entity can transparently access and modify the available section of the data space. In GVDS-based systems, the data space model affects the programming model: in particular, it the GVDS to determine the set of legal operations on the space and, as a consequence, to impose the communication model.

However, the highly dynamic nature of mobile ad-hoc environments makes it necessary to provide applications with the ability to tailor their behaviour by directly handling contextual information. To address this need some GVDS-based systems include reactive primitive with the aim to achieve reasonable tradeoff between context-awareness and allocation-transparency. For example, Lime provides support to reactive statements that permit to change application behavior in response to event occurrence triggered by GVDS access operations.

AGAPE is located at a lower abstraction level than GVDS-based systems. In particular, AGAPE does not aim to solve the coordination problems arising among interoperating entities. Notwithstanding, AGAPE, according to application-specific requirements, facilitates the design and the development of GVDS. In particular, the AGAPE middleware may ease the merging of available GVDS partitions by exploiting the group management support: GVDS partitions allocated on co-located members of the same group may merge. In addition, AGAPE communication patterns may provide effective support to implement access primitives on remotely available GVDS partitions. In fact, the AGAPE group communication support may permit to require a co-located member both to return data from and to write data to its GVDS partition.

## 7. Conclusions and Ongoing Work

Group communication in MANETs raises novel challenges that require the design and development of advanced group membership and communication support solutions. AGAPE is a novel group membership and communication framework for the design, development and support of collaborative

services in MANET environments. AGAPE enables impromptu collaboration among previously unknown users by adopting the metaphor of group. In particular, the group communication support exploits the visibility of the location of group members along with their characteristics to select and refer the collaboration partners for message delivery. Group members can interoperate by exploiting various context-aware communication patterns that permit to implement both point-to-point and point-to-multipoint communication channels. In addition, AGAPE permits to tailor message order and presentation according both to the actually applicable context and the application requirements.

We are currently validating our middleware with a wide variety of collaborative scenarios. The first experiences in the use of AGAPE have demonstrated its suitability in the implementation of collaborative services in Mobile Ad-Hoc environments. These early results are stimulating further research along different guidelines to improve the current prototype. In particular, we are currently working on the integration of AGAPE with a session manager service to enable stateful communication among collaborating group members even in typically highly dynamic settings. We are also investigating the security concerns raised by group management and starting to integrate initial security support services in AGAPE.

## References

- [1] D. Powell (ed.), "Group Communication", Special Issue on Group Communications, Communications of the ACM, ACM Press, Vol. 49, No.4, April 1996.
- [2] A.K. Dey, et. al., "Towards a Better Understanding of Context and Context-Awareness", in Proc. of the 19th Conference on Human Factors in Computing Systems (CHI 2000), ACM Press, Netherlands, April 2000.
- [3] T. Rodden, et. al., "Exploiting Context in HCI Design for Mobile Systems", in Proc. of the 1st Workshop on Human Computer Interaction with Mobile Devices, ACM Press, Scotland, May 1998.
- [4] D. Bottazzi, et. al., "AGAPE: a Location-Aware Group Membership Middleware for Pervasive Computing Environments", in Proc. of the 8th International

- Symposium on Computers and Communications (ISCC 2003), IEEE Press, Turkey, July 2003.
- [5] M.J. Fisher, "Impossibility of Distributed Consensus with One Faulty Process", in Journal of the ACM (JACM), ACM Press, Vol. 32, No. 2, April 1985.
- [6] Z.J. Haas, et. al., "Gossip-based Ad-Hoc Routing", in Proc. of the 21st Annual Joint Conference of the IEEE Computers and Communication Societies (Infocom 2002), IEEE Press, USA, June 2002.
- [7] A. Rowstron, et. al., "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems", in Proc. of the 4th International Conference on Distributed Systems Platforms (Middleware) (IFIP/ACM), ACM Press, Germany, November 2001.
- [8] X. Jiang, at. al., "Ubiquitous Computing for Firefighters: Field Studies and Prototypes of Large Displays for Incident Command", in Proc. of the 23rd Conference on Human Factors in Computing Systems (CHI 2004), ACM Press, Austria, April 2004.
- [9] World Wide Web Consortium, Composite Capability/Preference Profiles, [www.w3.org/Mobile/CCPP](http://www.w3.org/Mobile/CCPP).
- [10] J. Zieniewicz, et. al., "The Evolution of Army Wearable Computers", in IEEE Pervasive Computing, IEEE Press, Vol. 1, No. 4, October-December 2002.
- [11] C.E. Perkins, et. al., "Ad hoc On-Demand Distance Vector Routing.", in Proc. of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 1999), IEEE Press, USA, February 1999.
- [12] G. Kortuem, et. al., "When Peer-to-Peer comes Face-to-Face: Collaborative Peer-to-Peer Computing in Mobile Ad-hoc Networks", in Proc. of the 1st International Conference on Peer-to-Peer Computing (P2P2001), IEEE Press, Sweden, August 2001.
- [13] G. Picco, et. al., "Lime: Linda meets Mobility", in Proc. of the 21st International Conference on Software Engineering (ICSE1999), ACM Press, USA, May 1999.
- [14] C. Mascolo, et. al., "XMIDDLE: A Data-Sharing Middleware for Mobile Computing", in Personal and Wireless Communications Journal, Kluwer, Vol. 21, No.1, April 2002.
- [15] G. Cugola, et. al., "PeerWare: Core middleware support for Peer to Peer and Mobile Systems", Technical report, Politecnico di Milano, 2001.
- [16] G.P. Picco, et. al., "On global virtual data structures", in D. Marinescu and C. Lee, editors, Process Coordination and Ubiquitous Computing, CRC Press, 2002.