Journal of Network and Computer Applications 35 (2012) 145–155

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

High level MANET protocol: Enhancing the communication support for mobile collaborative work

Juan Rodríguez-Covili, Sergio F. Ochoa*, José A. Pino

Department of Computer Science, Universidad de Chile, Santiago, Chile

ARTICLE INFO

Article history:
Received 16 September 2010
Received in revised form
7 January 2011
Accepted 24 February 2011
Available online 9 March 2011

Keywords:
MANET
Routing protocol
Mobile collaborative work
Nomadic work
HLMP

ABSTRACT

Mobile collaborative activities involve on-demand interactions among nomad users. Unavailability of communication support in the physical scenario where users are located cannot be a limitation to carry out such collaboration instances. Mobile workers can take advantage of the communication capability embedded in their mobile devices in order to create communication channels between them. The Mobile Ad-hoc Networks (MANET) are infrastructures that can be used to support the nomad users' activities. However, these networks have a short communication threshold; therefore, they need to include a routing protocol as part of its infrastructure to allow mobile workers to collaborate when they are not physically close. This paper presents an application level routing protocol that was designed to support nomad workers performing mobile collaborative activities. The protocol, named High Level MANET Protocol (HLMP), provides several automatic services that are required by mobile collaborative systems. Some of these services are the automatic MANET formation, peer detection and messages routing. HLMP has been implemented in a mobile communication infrastructure and used in several mobile groupware systems.

1. Introduction

Several infrastructure components are typically involved in wireless communication services that allow interactions among mobile devices. Examples of these infrastructure components are Wi-Fi or Bluetooth access points, cell phone antennas, radio signal boosters and amplifiers. Those components currently provide enough signal coverage and stability, which allow mobile collaborative work in the physical area where they are located. However, there are scenarios where this infrastructure is not available, its usage is expensive or simply the mobile users are not able to depend on such infrastructure to carry out on-demand collaboration. Some of these scenarios are disaster relief, police security operations, touristic activities and mobile work in rural areas. Mobile collaborative work typically occurs at some physical workplace, where there is no infrastructure-based wireless communication support (Brugnoli et al., 2005).

Since the independent wireless signal range of mobile devices allows deployment of a Mobile Ad-Hoc Network (MANET) almost anywhere (Corson and Macker, 1999), it is possible to use this type of communication to support collaboration among mobile users located in scenarios where other communication systems have limitations or are unavailable. A MANET is an autonomous and mobile peer-to-peer mesh, which is able to support communication, coordination and collaboration activities performed by mobile user groups. This network can be formed by various types of autonomous mobile devices (e.g. cell phones, laptops or microcomputers installed in vehicles). These devices are usually equipped with wireless network signal transmitters and receptors, mainly Wi-Fi or Bluetooth, which allow them to communicate without making use of fixed infrastructure elements. Previous studies have shown the usefulness of MANETs in scenarios of mobile collaborative work, e.g. catastrophe assistance or coordination in common emergencies (Neyem et al., 2008), construction sites inspections (Ochoa et al., 2011), healthcare (Morán et al., 2007), m-business (Tarasewich, 2003) and mobile learning (Valdivia et al., 2009).

MANETs can be modeled by a graph $G=(V, E)$, where V is the set of nodes representing the mobile devices and E is a set of arcs; each arc models the communicational range intersection between two devices (Perkins, 1998). Figure 1 shows how a set of devices and their respective groups of direct communication connections are represented in this model. However, native ad-hoc wireless networks do not allow communication between devices that are outside the respective wireless signal range. Therefore, each node has to use some routing mechanism to transmit messages to remote devices, which are not adjacent neighbors. Routing is a

* Corresponding author. Tel.: +56 2 9784879.
E-mail addresses: jrodrigu@dcc.uchile.cl (J. Rodríguez-Covili), sochoa@dcc.uchile.cl (S.F. Ochoa), jpino@dcc.uchile.cl (J.A. Pino).

1084-8045/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jnca.2011.02.006

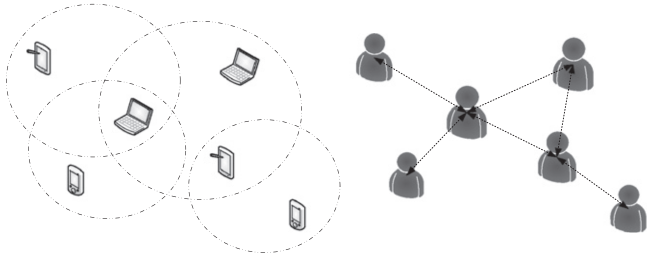


Fig. 1. MANET model.

mandatory requirement in mobile collaboration (Herskovic et al., accepted for publication). If the communication infrastructure supporting the mobile work does not support routing, the nomad users will be able to collaborate with other users that are located within just one hop of distance (the maximum distance for one hop is about 15–20 m in built areas).

This messages routing is possible when intermediate nodes retransmit data packets which are not necessarily of their own interest. Moreover, the routing protocol used to support this behavior has to take into account the dynamics of the graph definition (i.e., the mesh). This mesh can change in an unpredictable way at anytime, because of several reasons; e.g. the users' mobility while carrying the devices, the change of the work context where they are located, or temporal variation and environmental interference on the wireless signal.

This paper presents the High Level MANET Protocol (HLMP), an application level routing protocol that provides a set of communication functionalities required to carry out on-demand mobile collaboration. HLMP is able to automatically assemble and keep a MANET structure, using operating system implementation routines and data transfer protocols, such as UDP or TCP. Since HLMP was designed particularly to support mobile collaboration, the protocol structure eases the implementation of several awareness mechanisms that support users' interactions. This protocol was implemented through a communication infrastructure named HLMP API (Rodríguez-Covili et al., in press). The empirical results obtained using such infrastructure indicates that HLMP has a performance similar to some of the most well-known routing protocol for MANETs (e.g. OLSR). However, HLMP is more robust and reacts faster to changes in the network topology than the previous protocols (Rodríguez-Covili et al., in press).

Next section presents the related work. Section 3 describes the High Level MANET Protocol. Section 4 shows the implementation results, and also presents a set of mobile collaborative applications that used HLMP to support communication among nomad users. Finally, Section 5 presents the conclusions and future work.

2. Related work

Although several studies and initiatives have been reported concerning routing protocol in MANETs (Johnson et al., 2007; Neumann et al., 2008), this is still an open issue (Kiess and Mauve, 2007; Messeguer et al., 2009). The complexity of the ad-hoc communication scenario and the need to deal with low level details (i.e. accomplishment at IP layer) makes communication services hard to implement, adapt or reuse, particularly, if these protocols have to consider the use of different kinds of mobile devices and operating systems.

Many research publications have created standard terminology, problem definitions and solutions for several network topology issues related to MANETs (Holland and Vaidya, 1999; Ni et al., 1999; Wongsaardsakul and Kanchanasut, 2007). Typically, these solutions have been implemented through two families of protocols: *proactive* and *reactive*. Proactive protocols implement and keep a routing table, which is used to determine the best path to deliver a message. Reactive protocols do not pre-establish a path; such path is dynamically defined (with each hop) while the messages travel towards the destination. Regardless of the protocol type, each solution must address the problem of delivering unicast and multicast messages. Concerning unicast routing protocols, two of the most promising proposals are DYMO (Chakeres and Perkins, in preparation) and OLSR (Clausen and Jacquet, 2003).

On the one hand, Dynamic MANET On-Demand (DYMO) is a reactive unicast protocol that creates routes on-demand, by sending request and response control packets. Therefore, no global topology information is available. On the other hand, OLSR is a proactive unicast protocol; therefore, it maintains a routing table with information to deliver the messages. An interesting feature included in OLSR is the TC messages. These messages have topology information, which is exchanged by means of a controlled flooding.

Concerning MANET multicast protocols, some of the most well known are: ALMA (Krishnamurthy and Faloutsos, 2006) and PAST-DM (Gui and Mohapatra, 2003). Application Layer Multicast Algorithm (ALMA) creates a tree of logical links between the group members. The protocol aim is to reduce the cost of each link in the tree, by reconfiguring the tree under mobility and congestion situations. Progressively Adaptive Subtree in Dynamic Mesh (PAST-DM) is an overlay multicast protocol implementing a dynamic virtual mesh. The mesh is dynamically maintained through the exchange of link state packets.

Most of these routing protocols are not focused on supporting mobile collaborative work, for that reason they do not manage users (just IP addresses). They do not provide information to implement awareness mechanisms either. The protocol to be presented in the next section was designed to support mobile collaboration activities; therefore, it provides particular services that ease interactions among nomad users.

3. High level MANET protocol

MANET based applications have to be aware of the routing protocol functionality and also about the network topology. These systems need information about the multiple hops that set up the communication schema in order to provide a suitable service.

The procedures to manage the messages transmission and routing can be kept in either the network or application layers (Krishnamurthy and Faloutsos, 2006). Most routing protocols implement those procedures at the network layer; however, if the protocol is going to support mobile collaboration activities it is convenient to implement them at the application layer. The rationale for this decision is as follows: (a) the protocol provides flexibility to groupware specific requirements; (b) it becomes simple to develop, test and deploy; (c) it is possible to make communication decisions based on the network topology; and (d) information transmission can be done using just simple structures.

The main purpose followed in the design of HLMP was to establish a set of automated high-level procedures, able to create, keep and use a MANET, which includes routing capabilities. The design of these procedures should consider some key requirements to support mobile collaborative work; e.g. the protocol must be fully distributed and able to run on a wide range of computing devices (from a cellular phone to a laptop) (Herskovic et al., accepted for publication). Mobile devices using the protocol should be able to participate in a network and collaborate on-demand with other devices, by sending messages to any other node inside the mesh.

The key concept stems on the periodical delivery of a datagram known as "I'm Alive" message. This packet contains information about the sender node and its arcs set (i.e. the neighbors). Composing these packages it is possible to identify the network current topology; i.e. the MANET graph. This information is kept and updated into every node's memory. Since the protocol must act based on fully distributed services, each node must decide, based on a pre-established heuristic, how to route each received packet. Counting on information about the network topology in each participating node, makes feasible the task of finding the optimal paths to deliver the messages.

Provided that MANETs can change their behavior in short time periods, HLMP recalculates the path for a message in every intermediary node of the route. This operation is done based on the current knowledge of the MANET graph and it is not based on data analysis or statistical information.

HLMP delegates the low level functionalities to the operating system; however, the protocol establishes the high level procedures to manage the messaging. It also decides the kind of transport protocol that is suitable to use in order to provide communication functionalities between two neighboring nodes. Next sections describe the main features and components of this routing protocol.

3.1. Application layer MANET routing

Several peer-to-peer systems have adopted the idea of moving the routing procedures to the application layer in order to overcome the common limitations of the transportation layer (Garcia et al., 2009). This strategy affects important issues concerning efficiency (in terms of the latency-stretch) and not-generic services characteristics that are available for existing applications. However, this kind of design structure offers considerable advantages that justify a tradeoff. Figure 2 compares the HLMP approach with the commonly used routing protocols approach. In this design, the routing procedures are moved over the transportation layer to provide information about the network topology and routing decision-key elements to the mobile collaborative application. Next we present various reasons that justify the decision to allocate the routing processes at the application layer.

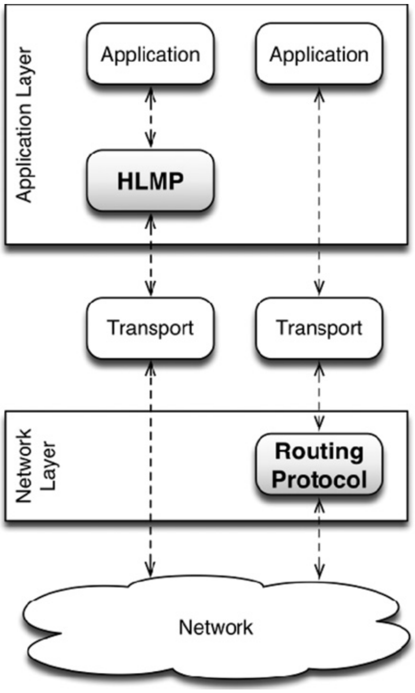


Fig. 2. Routing protocols at network layer vs. application layer.

(1) *Flexibility to groupware specific requirements*: Complex requirements for groupware applications may need specific behavior for routing procedures, message control or network traffic. An application layer routing mechanism must offer a set of parameters or adaptable software modules that provide flexibility to the mobile applications.

(2) *Simplicity for development, test and deployment*: Building software that is going to run at the application layer is easy to develop and test, due to the many tools and frameworks available for that purpose. Most pre-built software components are platform-independent, which simplifies the source code compilation. Moreover, the reuse and deployment of groupware services based on MANET systems become an "out of the box" application with a simple deployment functionality. Simplicity is provided to the final user who does not have to install/configure OS elements.

(3) *Topology aware communication decisions*: The routing protocol can provide information to the overlay application in order to make communication decisions when this component is at application layers. An evaluation of the network topology can determine the kind of transportation protocol to use, the messages size, and waiting times. The peer-to-peer MANET environment can be reflected also to the groupware users through awareness mechanisms, in order to provide network information. Applications running on unstable network scenarios can take advantage of this information and offer better collaboration possibilities.

(4) *API structures*: Developers can use a standardized API that implements a high level routing protocol and provides the communication services using simple objects and structures. When the MANET is controlled as a middleware component, it is easy to specify software modules interactions without breaking the strict layering of the system architecture.

3.2. The three wireless signal layers

The behavior of the wireless communication signal has been reported in various studies (Duchamp and Reynolds, 1992; Eckhardt and Steenkiste, 1996; Messeguer et al., 2009). Based on such studies we know that several factors affect such signal. Some of the most common factors are the following ones: the datagram size, the kind of network interface, the processing power of the device, the wireless threshold, the protocol implementation in the operative system, and also the environmental factors such as walls, waves interference, noise by heat, weather conditions, distance dispersion, Doppler effect, etc.

These studies also show there is uncertainty about what kind of protocol is most suitable under particular conditions. In order to reduce this uncertainty, the authors have empirically found that the wireless signal emitted by a mobile device can be modeled as three main layers, as shown in Fig. 3.

The WLan layer is defined as the distance the device can use to create an active wireless ad-hoc network with another neighboring device. This communication threshold is usually longer than the distance necessary to create TCP or UDP efficient procedures.

The UDP layer is defined as the distance the device can use to send UDP multicast messages with a reasonable packet loss rate. This layer enables the peers' detection control system, which is described in Section 3.4. The UDP layer is usually longer than the distance necessary to create fast TCP connections. Finally, the TCP layer is the distance needed to create TCP links, in order to connect two devices with a reliable link.

HLMP uses this model to separate the processes and functionalities it provides. The WLan layer is used to perform connection procedures and establish network IP addresses identification. The UDP layer is used to perform the peer detection mechanism and the MANET graph creation (e.g. the network topology). Finally, the TCP layer is used to establish direct paths between the nodes in order to send and route reliable messages.

3.3. Connection procedure

Whenever a new device wants to access an HLMP MANET, it has to perform a connection procedure. Figure 4 shows the three macro-components of this process: WLan Ad-Hoc connection, IP address self-configuration, and TCP and UDP services start.

(1) *WLAN ad-hoc connection*: This is the first step of the connection procedure. The node must delegate the configuration of a wireless network profile to the operating system when trying to access the MANET, using as Service Set Identifier (SSID) a common word selected by the groupware application. A WLan is created when another device is detected and it delivers the same profile. This profile has to be transmitted also with the Independent Basic Service Set modality (defined by the IEEE 802.11 standard), which allows direct links between devices (ad-hoc behavior) without using any kind of access points (IEEE Computer Society, 2007). Most operating systems use an XML profile to represent this information.

(2) *IP address self-configuration*: IP address self-configuration is a desirable requirement in a MANET, because the mobile collaboration processes are performed on-demand, and usually new mobile devices need to enter or leave the network. Therefore, a unique network address must be automatically settled for each one of these nodes. This kind of distribution is usually performed with DHCP, but such mechanism requires a central server to provide an automated configuration. Using centralized components is not recommended for mobile collaborative solutions because they limit the nomad users' interaction capability (Neyem et al., 2008).

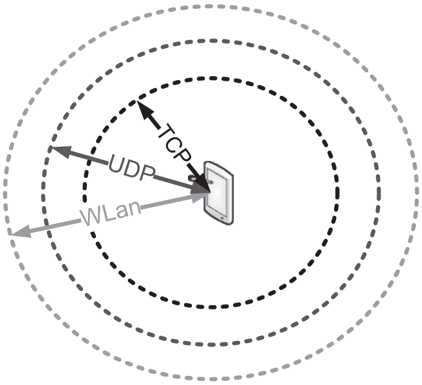


Fig. 3. Wireless signal ranges.

A plausible approximation of this process consists of choosing a tentative network address. HLMP randomly selects the IP address and a fixed sub-net mask which defines the number of possible nodes inside the MANET. After the IP configuration, the devices have to perform a Duplicate Address Detection process (DAD) in two stages: a strong DAD and a weak DAD (Vaidya, 2002).

Strong DAD is delegated to the operating system. Such process can detect IP address duplications during the WLan conformation. Consequently, the strong DAD can only detect devices addresses belonging to the closest devices set.

Weak DAD is managed by HLMP and consists of a verification process. Such process is periodically executed when receiving any kind of message. It checks the sender's original IP address and compares it with its own IP address in order to detect duplicate addresses. If an IP duplication is detected, then the device has to go back on the configuration procedure and perform a new random selection of the IP address.

Since this IP configuration process is automatable, it facilitates the mobile users' connection/re-connection (Bernados et al., 2007). This service is mandatory in mobile collaborative applications because most nomad users do not want to perform a manual IP configuration process each time they have to reconnect to the network (Perkins and Belding-Royer, 2003). This automatic IP configuration service and MANET formation influence directly the usability of a mobile solution.

(3) *TCP and UDP services start*: Finally, the node starts the corresponding services in order to initiate the communication mechanisms. These are TCP services that run at the previous configured IP address, and allow sending and receiving packets under an agreed port. An UDP service, subscribed to an agreed multicast group address, is also started using a second agreed port.

3.4. Message structure

A Network Message (or HLMP datagram) is composed of a header and a body. The header is a four bytes field that indicates the size of the inner data. The body, named Communication Message (Fig. 5(a)), contains the message itself. A Communication Message consists of an organized packet of bytes containing data related to a high level message.

The information contained in the header depends on the required mechanisms to send and route a message. HLMP defines five main delivery mechanisms; each one with a unique identifying code named Meta Type. These mechanisms are the following ones:

- *Multicast*: The protocol performs just an attempt to send the message to all nodes in the network, using the UDP multicast groups channels.
- *Safe Multicast*: An attempt is made to send the message to all nodes in the network, using the TCP bridges.
- *Unicast*: The message is sent to only one node in the network using the TCP channels.
- *Fast Unicast*: The message is sent just to one node in the network using the UDP layers.

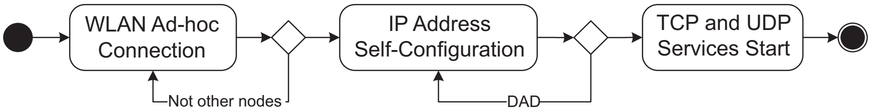
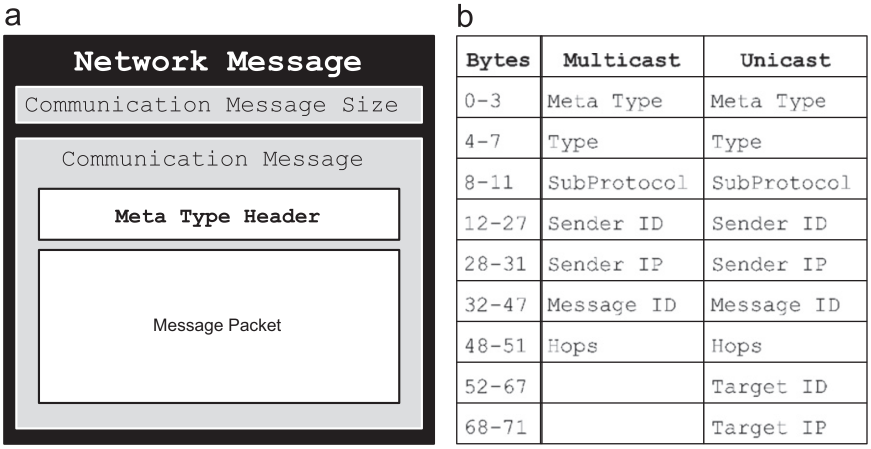


Fig. 4. Network connection procedure.



Bytes	Multicast	Unicast
0-3	Meta Type	Meta Type
4-7	Type	Type
8-11	SubProtocol	SubProtocol
12-27	Sender ID	Sender ID
28-31	Sender IP	Sender IP
32-47	Message ID	Message ID
48-51	Hops	Hops
52-67		Target ID
68-71		Target IP

Fig. 5. Network message structure and meta type header.

- *Safe Unicast*: The message is sent to one node in the network using the TCP channels, but the receiver must confirm the delivery of the packet.

Figure 5(b) shows the header definition for multicast and unicast kind of messages. The information contained in the Meta Type header specification corresponds to:

- *Meta Type*: The Meta Type code of the message.
- *Type*: The type of message contained in the body field. The value assigned to this field is used to access specific services required by groupware applications.
- *SubProtocol*: The code of an optional HLMP sub-protocol responsible for attending the message. The codes are established in the same way than the message type codes.
- *Sender ID*: The identification code of the sender, which corresponds to a high level code for unique identification of the user at the application layer.
- *Sender IP*: The sender's IP address.
- *Message ID*: The globally unique 128-bit message identification code. This code is generated by the protocol at the moment of sending it. The delivery process uses an algorithm that guarantees an extremely low collision probability.
- *Hops*: The number of hosts in which the message has been received and routed.
- *Target ID*: The identification code of the message addressee.
- *Target IP*: The IP address of the message addressee.

Finally, the body of a Communication Message consists of data packets composing the message that wants to be delivered by the groupware services.

3.5. Multicast transmission process

This process establishes the functionality required to send and route Multicast and Safe Multicast (Meta Type) messages. The devices do not require any kind of information about the network topology to carry out this procedure. However, a Message ID List is necessary to allow the temporary storage of received messages' identification numbers. The list must be composed of a FIFO queue and hash table. This structure allows detection of copied messages that have been received from two or more different paths. Therefore, it is possible to avoid most message duplication problems. The reasons used to define the size of this list are discussed in Section 3.5.6.

```
Send Multicast Message M to Everyone:
add M to MessageIdList;
send M to multicast group;
end;
```

Algorithm 1. Procedure to send a multicast message.

```
Receive Multicast Message M:
if MessageIdList  does not contains M {
   add M to MessageIdList;
   send M to multicast group;
   process M as a received message;
}
end;
```

Algorithm 2. Procedure to receive a multicast message.

3.5.1. Algorithms

The Multicast process is described using the Algorithms 1 and 2 below. Essentially, it consists of a message transmission to all possible nodes that are in the UDP multicast group of the sender device. When a message of this kind is received, it must be retransmitted again to all possible nodes, like flooding the network with the packet. In order to send Safe Multicast messages, a difference is made when performing the flooding. Safe Multicast uses the TCP neighborhood surrounding the node, instead of its UDP multicast group.

In order to keep control over the flooding, the Message IDs of sent and received packets are saved into the Message IDs List to check and avoid possible collisions.

3.5.2. Example

Figure 6 shows an example of the transmission process of one Multicast message into a simple MANET. The sequence of steps is the following: (a) node A wants to send a Multicast message M, it sends the message to its multicast group, which corresponds only to nodes C and B; (b) nodes C and B receive and processes the message, resending it to their multicast group; (c) nodes A, B and C detect the duplication of message M, therefore it is dropped when received; nodes D and E receive and process the message, resending it to their respective multicast group; (d) only node F receives, processes and resends the message to its multicast group, but then, node E will detect the copy of M, dropping it away. Finally, message M has been flooded on the network and all users have received it and processed it just once.

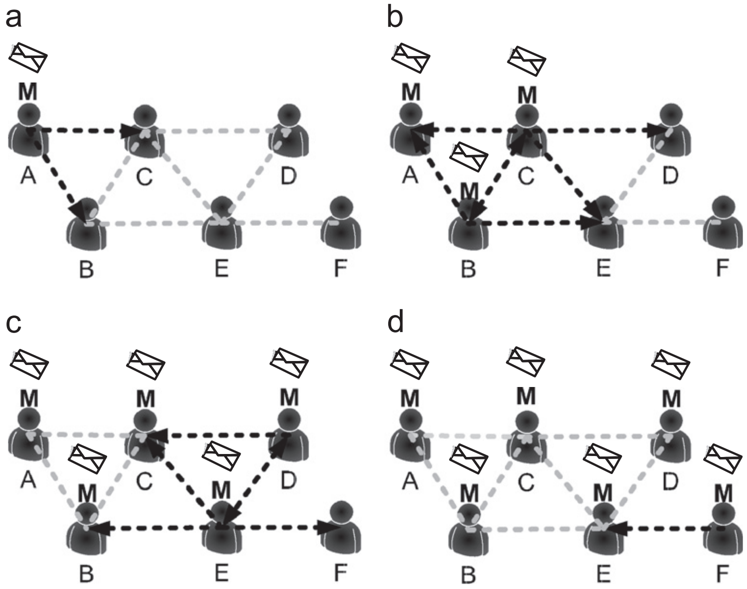


Fig. 6. Example of a multicast message transmission.

```
Process Received I'm Alive Message M:
if graph does not contains sender of M {
   add sender of M to graph;
}
else {
   update sender of M;
}
if neighborhood does not contains sender of M {
   if hops of M is equal to 1 {
      add sender of M to neighborhood;
   }
}
end;
```

Algorithm 3. Processing an I'am alive incoming message.

3.5.3. *Peer detection mechanism*

The nodes must send and receive networking information in order to perform a peer detection activity. This process allows the generation of the MANET graph into each node's memory. The method is based on the transmission of a Multicast message named "I'm Alive", which contains the set of arcs of the sender, corresponding to the neighborhood created by direct TCP active connections between two nodes.

When connected to the MANET, nodes have to constantly send their "I'm Alive" message every one second. New nodes will send empty messages, and the old ones will send their corresponding neighborhood. Nodes have to process the received "I'm Alive" messages using Algorithm 3 below.

When a sender node is added to the MANET graph, the node information and all its arcs defined in the "I'm Alive" message are kept. When a sender node is added to the neighborhood, a TCP handshaking must be performed in order to establish a constant reliable connection. If a TCP connection is not possible, then the sender node has to wait a time interval, usually of 10 s, after trying to connect again. This time interval is configurable, and it establishes how fast the system reacts to neighborhood changes.

If a TCP connection is dropped later, then the link is just removed from the neighborhood arcs set. The MANET graph obtained using this mechanism in each node is then used to send Unicast messages.

This automatic peers' detection mechanism eases the implementation of some key awareness services, such as users' connection (e.g. online/offline) and status (e.g. available/busy). Since mobile collaborative work typically involves on-demand users interactions, these awareness mechanisms ease their implementation.

3.5.4. *Nodes signal quality*

Nodes signal quality is measured in order to detect old information within a MANET graph (i.e. a portion of the graph that has not been updated recently), and also nodes that have passed to an offline status or left the network.

When a node is added to the MANET graph, a quality flag is set to value 25. The flag of all users in the graph is reduced by 1 every one second. If the flag of a node reaches a zero value, then it is assumed the node has left the network and it is deleted from the graph. If the information of a node is updated when performing Algorithm 3, then the quality flag is increased by 5.

Signal quality is then divided into three main sets. If a node has a flag value between 1 and 10, then the node has a Critical quality. If a node has a flag value between 11 and 20, then the node has a Low quality. Finally, nodes with a flag value between 21 and 25, have a Normal quality communication link. This quality value represents how updated is the information on a specific node of the MANET graph. It is helpful information to determine the Unicast messages procedure to be used to deliver messages.

A high value of the signal quality also represents the neighbor node is physically close to me. This information can be used to implement awareness of users' proximity. Such awareness service is also useful as promoter of on-demand collaboration instances among nomad users.

3.5.5. *Nodes traffic state*

Node traffic state is a local measure of each node. A flag value is set into each device and it counts how many messages are being received per second. The traffic state is also divided into three main sets in order to manage this information propagation and make communication decisions. The value of this variable is calculated depending on the device capacity, processing power, memory and estimated message size.

A minimum set of configuration values were found, after several testing with PDAs/smartphones and messages with a body content of 200 kbytes as maximum weight. The analysis of tests results indicated that traffic between 0 and 10 corresponds to nodes with a Normal traffic state. A flag value between 11 and 20 indicates the node is Overloaded. Finally, a flag value over 20 indicates the node has a Critical state in terms of traffic.

The calculated state is then attached to every "I'm Alive" message sent by the node. This measure represents how much processing delay can have a message when passing through that node. It is helpful information to determine the Unicast messages routing paths.

3.5.6. *Message ID list size*

The Message ID List works as a FIFO queue that helps detect duplicated messages. In order to identify an appropriate size of this list, we have observed the following probabilistic event:

- A message M is received, and its identification number is stored into the Message ID List.
- Then, other messages are received and their identification numbers are also stored into the Message ID List. This causes a shift of the M identification towards the final positions of the list. If there is message overflow, then message M's ID is removed from such list.
- Finally, an unexpected copy of M is received. A search in the list shows message M's ID does not exist. The system will include M's ID in the list, and therefore the node will process the message again.

This situation is what we define as the *message duplication problem*. It generates an erroneous processing of a message that has already been processed, due to an unsuitable size of the Message ID List.

In order to determine a maximum size for the list, we have created a simple process to estimate the current number of messages in the network. Using such information it is possible to establish a size for the ID list, which helps us to avoid the message duplication problem.

A possible solution can be to oversize the list; however, it will jeopardize the performance of the protocol when it is used in computing devices with scarce hardware resources (e.g. a cellular phone).

Figure 7 depicts the model used to estimate the current number of messages in the MANET. Periodically every network node counts how many messages it receives from the MANET during 1 second, and also how many nodes are connected to the network. Let us suppose that a particular node counted an average of n messages during the last three observations, and also an average of z users were connected. If we assume that the local numbers recorded by a node are in some way representative of the MANET status, then the total number of packets in the whole network can be estimated in $z{*}n$ messages.

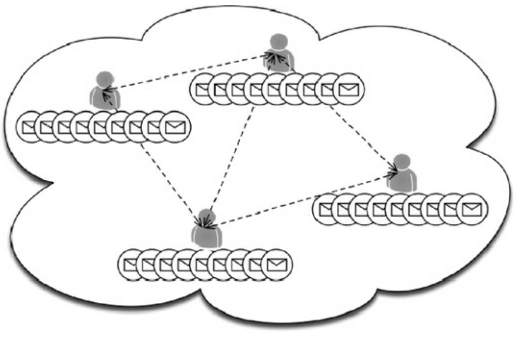


Fig. 7. Simple MANET model.

After a node has received and stored a message M's ID, and a copy of M (named N) is somewhere in the network, there is a probability P_a of receiving N as follows:

$$P_a = \frac{1}{z{*}n} \tag{1}$$

The probability P_b of not receiving N is as follows:

$$P_b = \frac{z{*}n-1}{z{*}n} \tag{2}$$

If several messages different from N are received afterwards, M's ID is shifted to the final positions of the Message ID List. If we call L the size of the Message ID List, then the number of events before M's ID get out of the Message ID List is also L. Eq. (3) shows the probability P_c for that to happen, i.e., the probability of occurrence of the *message duplication problem*

$$P_c = \left(\frac{z{*}n-1}{z{*}n}\right)^L \tag{3}$$

L can be obtained from Eq. (3). Its value can be estimated by setting P_c as low as possible, in order to reduce the message duplication problem occurrence. Eq. (4) shows how to calculate the limit of the Message ID List in real time in terms of P_c, and the average number of messages and users in the network. The size of the ID list changes periodically according to the network traffic

$$L = \frac{\ln(P_c)}{\ln(z{*}n-1)-\ln(z{*}n)} \tag{4}$$

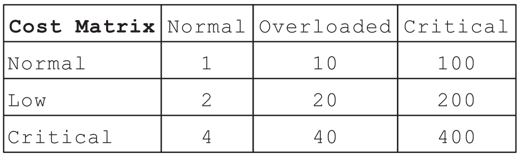
3.6. *Unicast transmission process*

This process establishes the functionality required to send and route Unicast, Fast Unicast and Safe Unicast (i.e. Meta Type) messages with the goal to be delivered to only one node within the MANET. The devices use the knowledge obtained by the peer-detection mechanism about the MANET and the quality and traffic state values in order to generate a path cost matrix (Fig. 8). Each node keeps its own matrix, which is used to assign cost weights to the paths on the MANET graph. The matrix rows indicate the traffic state of a node, and columns indicate the signal quality value. When performing optimal paths calculation, the resulting combination value is set to all paths surrounding the node.

The same Message ID List used in the Multicast message transmission process is utilized to avoid the message duplication problem. A Unicast acknowledge message (i.e. an Ack) is required in this type of transmission. Such message transports the ID of a received message in order to confirm its reception.

3.6.1. *Algorithms*

The Safe Unicast process to send a message is described using Algorithm 4. The sender node selects the best neighbor (i.e. the first node in the optimal path) when trying to send a message, and it uses a TCP connection to do it. Instead of saving the path into the message, the protocol recalculates the route in every node. The sender node holds the message for a while (until a timeout), in order to resend it if an acknowledgement is not received. When the path finding algorithm does not return any route, it means there are no suitable ways to reach the destination



Cost Matrix	Normal	Overloaded	Critical
Normal	1	10	100
Low	2	20	200
Critical	4	40	400

Fig. 8. Path cost matrix.

```
Send Safe Unicast Message M to Node N:
while Ack of M has not been received {
   Path <- minimum path to N;
   if there exist a Path {
      send M to first node in Path;
      wait a Time interval;
   }
   else {
      process M as a failed message;
      end;
   }
}
end;
```

Algorithm 4. Send a safe unicast message.

```
Receive Safe Unicast Message M:
if target of M is not myself {
   Path <- minimum path to target of M;
   if there exist a Path {
      send M to first node in Path;
   }
   else {
      process M as a failed message;
   }
}
else {
   if MessageIdList does not contains M {
      add M to MessageIdList;
      send Ack of M to source of M;
      process M as a received message;
   }
   else {
      send Ack of M to source of M;
   }
}
end;
```

Algorithm 5. Receiving a safe unicast message.

host. In that case the message is processed as a failed message using Algorithm 6.

Unicast messages are delivered using a similar process, but their delivery is not confirmed. Therefore, no acknowledgement messages or retransmissions are required. Fast Unicast applies the same strategy, but it uses UDP channels to send the packets.

Algorithm 5 shows the procedure followed by a node when a Safe Multicast message has been received. On the one hand, if the node detects the message target node is not itself, then it has to find a path towards the destination through the best neighbor to route the message. On the other hand, if the receiver node is the message destination, then it has to use the Message ID List to keep track of the reception, because the original node could be sending copies of the message due to times delays, messages lost for disconnection or other causes. Then, the receiver node has to send the Ack message to the sender, indicating message reception.

If any node detects that a copy of a Safe Unicast message has been received, then it has to send back the Ack message again, because it is unknown which situation generated the message duplication; i.e. a delay/loss of the original message, or a delay/loss of the Ack.

Algorithm 6 describes the procedure for processing a message labeled as a "failed message" by Algorithms 4 or 5. While performing this operation, the node has to check if the destination node still exists in the network. If there is not a TCP path to such node, then it is possible to wait a time interval in order to clarify the status of the destination node; e.g. the node is re-connected to the network or it finally left the MANET. After this time, the message is sent again using the corresponding algorithm. If the destination node is not detected in the network, then the protocol assumes the node is disconnected, and therefore the message is dropped.

```
Process Failed Safe Unicast Message M:
if target of M is in UserList {
   wait a Time interval;
   send M again;
}
else {
   if source of M is myself {
      warning failed delivery of M;
   }
}
end;
```

Algorithm 6. Process failed safe unicast message.

The retry procedure can be controlled using a maximum number of attempts. If the same message is processed as failed too many times, then it is assumed there is a low probability to find a path to the destination. In that case the message is also dropped.

3.6.2. *Example*

Figure 9 shows an example of the transmission process for one Safe Unicast message through a simple MANET. The transmission involves disconnection events. Let us suppose node A wants to send a Safe Unicast message M to node D; therefore, it calculates the optimal path and sends the message to node C (its best neighbor). Node C receives the message and the topology of the network changes; therefore node C recalculates the path and it sends the message to node E. Node E receives the message and routes it to node D. Node D receives and processes the message. Then, it computes the path to send the corresponding Ack. Such message is sent to A through node C.

4. Implementation results

HLMP has been implemented in a mobile communication infrastructure named HMLP API (Rodríguez-Covili et al., in press). That infrastructure exposes an application programming interface (API) that allows accessing the set of automatic services described in the protocol; for example MANET formation, peers detection, IP assignment, management of users connections/disconnections, and routing services using several delivery strategies. The infrastructure also implements several awareness mechanisms that support the mobile collaboration process, such as users' availability, physical distance between users and teamwork composition and physical distribution.

HLMP API has been compared with OLSRd, a well-known implementation of the OLSR routing protocol (Clausen et al., 2009). The obtained results show that both routing protocols have a similar performance; however, HLMP is more stable and react faster to changes in the MANET topology (Rodríguez-Covili et al., in press).

HLMP API has also been used as communication support for several mobile collaborative applications, such as Construction Inspector (COIN), which supports the work of inspectors during construction inspection activities (Ochoa et al., 2011; Rodríguez-Covili et al., 2011); MobileMap that supports the work of firefighters during emergencies (Monares et al., 2011; Rodríguez-Covili et al., 2011); and MeetU that assists physicians and nurses during hospital work (Morán et al., 2010; Rodríguez-Covili et al., 2011). All these applications have been used in real scenarios. The feedback received from the mobile workers indicates that HLMP not only is able to support mobile collaborative activities, but it also affords an appropriate performance.

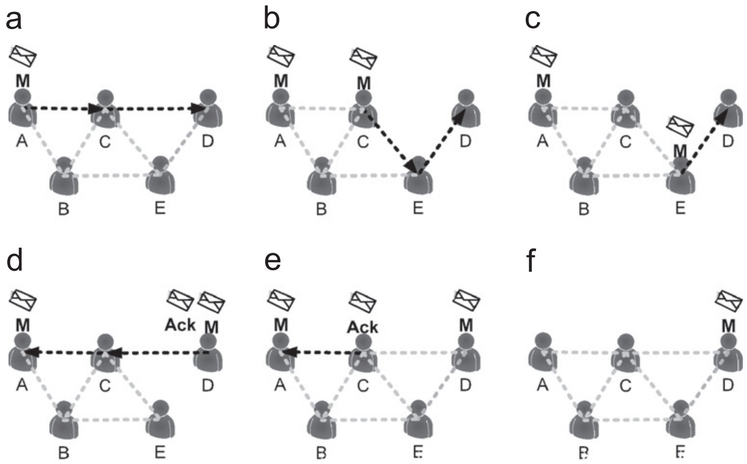


Fig. 9. Safe unicast message transmission process example.

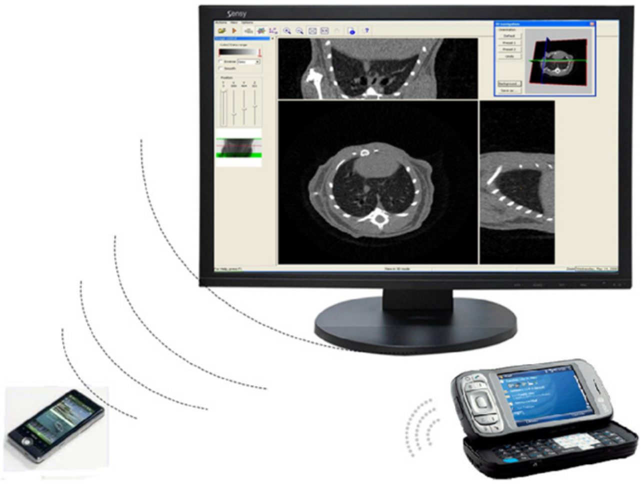


Fig. 10. MobileConnector work scenario.

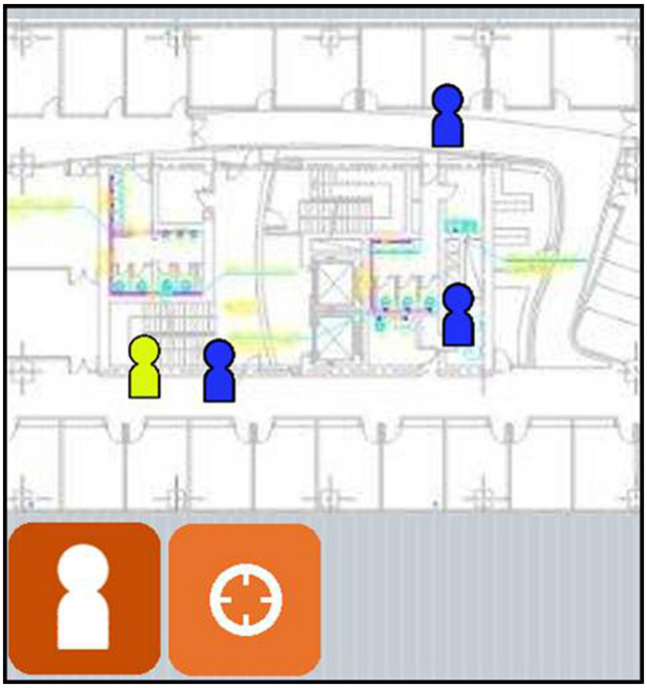


Fig. 11. Main user interface of UserLocator.

This performance contributes to the usability of the mobile applications implemented over it. Next section describes a couple of new collaborative applications that have recently been implemented over HLMP API.

4.1. MobileConnector

MobileConnector is an application integrating several devices into a MANET, and allows deploying the screen of a particular device on the other devices' screen. Figure 10 shows the integration of a smartphone, a pocketPC and a LCD. This application was designed to support the mobile work done by medical personnel at a hospital. Particularly, it eases information exchange, informal interactions and ad-hoc meetings among mobile workers.

Let us suppose that Fig. 10 depicts an ad-hoc meeting between two physicians who must the diagnosis of a patient. The physician using the smartphone deploys relevant information on the LCD. A nettop connected to the LCD allows such device be part of the MANET. Eventually the other physician can retrieve a copy of such information from the LCD or the smartphone in order to perform an in-depth analysis later. Alternatively, that physician can also display information that he/she has in the pocketPC on the LCD, and that could be useful for the diagnosing process.

All communication services that allow this application to work properly are provided by the HLMP API. The services are automatic MANET formation, peers detection and IP configuration, file transfer and messaging. Since the application involves co-localized users, the routing capabilities and the management of disconnection are available in the system, but they are not particularly useful in this work scenario.

4.2. UserLocator

Locating users in indoor workplaces not only is a challenge, but also it is a key awareness mechanism to promote on-demand collaboration (Herskovic et al., accepted for publication). UserLocator is a mobile collaborative application that deploys, on a map of a physical workplace, the information about the location of a list of users that are part of a MANET (Fig. 11) (Vera et al., 2010). This is a generic application that can be used to locate people in various built scenarios, such as shopping malls, office buildings or hospitals.

The system uses a prediction model based on the signal strength that mobile users receive from the access points deployed in the workplace. In addition, mobile users share the location information among them in order to reduce the location error range. As a consequence of it, each user can see the location and the movements of his/her partners on a map with an average error of 3–4 m.

Similar to the previous case, this system uses HLMP API to form the MANET, detect users, share information, etc. However, in this case the shared information needs to be routed through the network because the users are not co-located. Moreover, the services to manage automatic users' connection/disconnections are highly required because of the typical interference produced by built scenarios. Since mobile collaboration involves on-demand interactions, locating potential collaborators can be helpful to promote such instances.

In most collaboration scenarios the mobile applications require to address particular quality requirements, such as security or privacy. The current version of HLMP does not consider these issues, therefore the components that provide these services must be implemented in an ad-hoc way into each application. Rodríguez-Covili et al., (2011) proposed a reference architecture for mobile collaborative applications, which is composed by three layers: communication, coordination and collaboration. The components addressing these transversal issues should be part of the coordination layer. Thus such components will keep an abstraction level that allows developers to reuse them.

5. Conclusions and future work

Several protocols have been proposed to route messages on MANETs in the last few years. Some of them are general and others are specific for certain application domains. However, these protocols have not been designed to support mobile collaborative work. Therefore, they do not provide services that are usually required to support this type of activities.

This paper describes the HLMP routing protocol, which offers a significant communication base to mobile groupware applications that do not have fixed infrastructure dependence. HLMP is able to manage, using automatic mechanisms, several services such as users' connections and disconnection, IP assignment (dealing with IP collisions), messages routing and peers detection. The protocol eases the implementation of various users' awareness mechanisms based on that information. Examples of awareness information needed for collaborative work are the distance between two users and the MANET composition.

This protocol has been implemented in a mobile communication infrastructure named HLMP API (Rodríguez-Covili et al., in press). The capabilities of this infrastructure have been compared with OLSRd (Clausen et al., 2009). The obtained results are highly encouraging. HLMP API has also been used as communication support of several mobile collaborative applications; two of them were briefly introduced in this article.

The variety of communication mechanisms and message delivery strategies supported by this protocol provides flexibility to mobile groupware developers. Moreover, the high level procedures implemented in HLMP API allow reusing this protocol for various types of devices and operating systems.

The next steps in the HLMP evolution consider defining and implementing reusable coordination mechanisms on top of the current communication platform in order to offer a more comprehensive set of services to mobile groupware developers. Furthermore, the authors are currently performing a low level tuning process to the HLMP API in order to improve its throughput in scenarios with any type of users' mobility.

Next version of this protocol will address extra quality requirements, such as security and privacy, which are required in most collaboration scenarios. Thus, the solutions implemented to deal with these issues will be easy to reuse by applications using HLMP as communication support.

Acknowledgements

This work was partially supported by Fondecyt (Chile), Grant no. 1080352, and LACCIR Grant nos. R0308LAC004 and R1210LAC002, and Proyecto Enlace VID 2010 (University of Chile), Grant ENL 10/10.

References

Bernados C, Calderon M, Moustafa H. Survey of IP address auto-configuration mechanisms for MANETs. IETF Internet Draft, October 2007.

Brugnoli MC, Davide F, Slagter. R. The future of mobility and of mobile services. In: Cunningham P, Cunningham M, editors. Innovation and the knowledge economy: issues, applications, case studies. Amsterdam: IOS Press; 2005. p. 1043–55.

Chakeres I, Perkins C. Dynamic MANET On-demand (DYMO) Routing. IETF Internet-Draft, March 8, in preparation.

Clausen T, Dearlove C, Jacquet P. The Optimized Link State Routing Protocol version 2. IETF Draft, September 25, in press.

Clausen T, Jacquet P. Optimized Link State Routing Protocol (OLSR). IETF RFC 3626, October 2003.

Corson S, Macker J. Mobile Ad hoc Networking (MANET): routing protocol performance issues and evaluation considerations. IETF, RFC 2501, January 1999.

Duchamp D, Reynolds NF. Measured performance of a wireless LAN. In: Proceedings of the 17th IEEE conference on local computer networks 1992:494–9. September.

Eckhardt D, Steenkiste P. Measurement and analysis of the error characteristics of an in-building wireless network. In: Proceedings of ACM SIGCOMM'96 1996:243–54.

Garcia P, Gracia R, Espelt M, Paris G, Arrufat M, Messeguer R. Topology-Aware Group Communication Middleware for MANETs. In: Proceedings of the 4th international ICST conference on COMmunication System softWAre and middlewaRE (COMSWARE'09). Dublin, Ireland: ACM Press; 2009. June 15–19.

Gui C, Mohapatra P. Efficient overlay multicast for mobile ad hoc networks. In: Proceedings of the WCNC'03, New Orleans, USA, March 2003.

Herskovic V, Ochoa SF, Pino JA, Neyem A. The iceberg effect: behind the user interface of mobile collaborative systems. Journal of Universal Computer Science, accepted for publication.

Holland G, Vaidya N. Analysis of TCP performance over mobile ad hoc networks. In: Proceedings of the 5th ACM/IEEE international conference on mobile computing and networking 1999:219–30.

IEEE Computer Society. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Standard for information technology, telecommunication and information exchange between systems, LAN & MAN: specific requirements, Part 11, IEEE 802.11, 2007.

Johnson D, Hu Y, Maltz D. The Dynamic Source Routing Protocol (DSR). IETF RFC 4728, February 2007.

Kiess W, Mauve M. A survey on real-world implementations of mobile ad-hoc networks. Ad Hoc Networks 2007;5(3):324–39.

Krishnamurthy G, Faloutsos M. Application versus network layer multicasting in ad hoc networks: the ALMA routing protocol. Elsevier Ad Hoc Networks Journal 2006;4(2):283–300.

Messeguer R, Medina E, Ochoa SF, Pino JA, Neyem A, Navarro L, et al.. Building real-world ad-hoc networks to support mobile collaborative applications: lessons learned. In: Proceedings of the 15th international workshop on groupware (CRIWG). Lecture notes in computer science, vol. 5784. Douro, Portugal, 2009. p. 1–16.

Monares A, Ochoa SF, Pino JA, Herskovic V, Rodriguez-Covili JF, Neyem. A. Mobile computing in urban emergency situations: improving the support to firefighters in the field. Expert Systems with Applications 2011;38(2):1255–67.

Morán AL, Rodríguez-Covili JF, Mejía D, Favela J, Ochoa SF. Supporting informal interaction in a hospital through impromptu social networking. In: Proceedings of the 16th CRIWG conference on collaboration and technology, Maastricht, The Netherlands, September 20–23, 2010. Lecture notes in computer science, vol. 6257. p. 305–20.

Morán EB, Tentori M, Gonzalez V, Favela J, Martínez-Garcia. A. Mobility in hospital work: towards a pervasive computing hospital environment. International Journal of Electronic Healthcare 2007;3:72–89.

Neumann A, Aichele C, Lindner M, Wunderlich S. Better Approach To Mobile Ad-hoc Networking (B.A.T.M.A.N.). IETF, Internet-Draft, April 2008.

Neyem A, Ochoa S, Pino J. Integrating service-oriented mobile units to support collaboration in ad-hoc scenarios. Journal of Universal Computer Science 2008;14(1):88–122.

Ni S, Tseng Y, Chen Y, Sheu J. The broadcast storm problem in a mobile ad hoc network. In: Proceedings of the 5th ACM/IEEE international conference on mobile computing and networking, 1999; 151–62.

Ochoa SF, Bravo G, Pino JA, Rodriguez-Covili. J. Coordinating loosely-coupled work in construction inspection activities. Group Decision and Negotiation 2011;20(1):39–56.

Perkins C. Mobile ad hoc networking terminology. IETF Internet Draft, November 1998.

Perkins C, Belding-Royer E. Ad hoc On-Demand Distance Vector (AODV) Routing. IETF RFC 3561, July 2003.

Rodríguez-Covili JF, Ochoa SF, Pino JA, Messeguer R, Medina E, Royo D. A communication infrastructure to ease the development of mobile collaborative applications. Journal of Network and Computer Applications, in press. doi:http://dx.doi.org/10.1016/j.jnca.2010.12.014.

Rodríguez-Covili JF, Ochoa SF, Pino JA, Herskovic V, Favela J, Mejía D, et al. Towards a reference architecture for the design of mobile shared workspaces. Future Generation Computer Systems 2011;27(1):109–18.

Tarasewich P. Designing mobile commerce applications. Communications of the ACM 2003;46(12):57–60.

Vaidya NH. Weak duplicate address detection in mobile ad hoc networks. In: Proceedings of the 3rd ACM international symposium on mobile ad hoc networking & computing, 2002. p. 206–16.

Valdivia R, Nussbaum M, Ochoa SF. Modeling a collaborative answer negotiation activity using IMS-based learning design. IEEE Transactions on Education 2009;52(3):375–384.

Vera R, Ochoa SF, Aldunate R. EDIPS: an easy to deploy indoor positioning system to support loosely-coupled mobile work. Personal and ubiquitous computing, Valencia, Spain, September 2010. p. 79–86. doi:10.1007/s00779-010-0357-x.

Wongsaardsakul T, Kanchanasut K. A structured mesh overlay network for P2P applications on mobile ad hoc networks. Lecture Notes in Computer Science 2007;4882:67–72.