Virtual Home Region Multi-Hash Location Management Service (VIMLOC) for Large-Scale Wireless Mesh Networks

CHAPTER · JANUARY 2011

DOI: 10.5772/13139 · Source: InTech

Josep Mangues-Bafalluy
Centre Tecnològic de Telecomu…

Manuel Requena-Esteso
CTTC Catalan Telecommunicati…

10

Virtual Home Region Multi-hash Location Management Service (VIMLOC) for Large-Scale Wireless Mesh Networks[1]

J. Mangues-Bafalluy, M. Requena-Esteso, J. Núñez-Martínez and A. Krendzel
Centre Tecnològic de Telecomunicacions de Catalunya (CTTC)
Av. Carl Friedrich Gauss, 7 – 08860 Castelldefels – Barcelona
Spain

1. Introduction

Wireless mesh networks (WMNs) have recently received much attention not only from the research community, but also from municipalities or non-tech-savvy user communities willing to build their own all-wireless network. One of the factors that has helped in making WMNs become popular is the widespread availability of low-cost wireless equipment, and particularly, IEEE 802.11 WLAN equipment. However, making these WMNs operationally efficient is a challenging task. In this direction, there has been a lot of work on the research issues highlighted in (Akyildiz & Wang, 2005). Nevertheless, such research topic as mobility management did not receive as much attention as others (e.g., channel assignment or routing).

In general, mobility management is split into two main functions, namely handoff management and location management. The former deals with maintaining the communication of the mobile node (MN) while (re-)attaching to a new attachment point, whilst the latter deals with locating the MN in the network when a new communication needs to be established.

Related to mobility, and at an architectural level, a common belief in the research community is that, unlike in an IP context, node identifiers and addresses (i.e., the current location in the network of those nodes) should not be integrated into a single identifier. The main purpose of this is to enable designing efficient mobility management schemes, and as part of them, efficient location management schemes (location services). This is particularly challenging in large-scale WMNs, due to the state information that must be stored in the nodes and the associated control overhead sent through the network. Related to this, position-based (geographic) routing algorithms are expected to improve scalability of large

---

[1] Based on "VIMLOC location management in wireless meshes: Experimental performance evaluation and comparison", by Mangues-Bafalluy et al., which appeared in Proc. ICC-2010, South Africa. © [2010] IEEE; "VIMLOC: Virtual Home Region Multi-Hash Location Service in Wireless Mesh Networks", by Krendzel et al., which appeared in Proc. Wireless Days-2008, United Arab Emirates. © [2008] IEEE; "Wireless Mesh Networking Framework for Fast Development and Testing of Network-level Protocols", by Requena-Esteso et al., which appeared in Proc. of the ICT-Mobile Summit-2009, Spain © [2009].

WMNs. In fact, by exploiting position information of nodes in the network both state information and control overhead can be substantially reduced when compared to more traditional flooding-based approaches.

Two building blocks are required for deploying an operational position-based routing scheme, namely a location management service and a position-based routing/forwarding algorithm (Mauve et al., 2001), (Camp, 2006). The location management service/scheme is needed to map between the identifier of a node (node_ID) and its current position in the network (i.e., location address (LA)) so that an underlying position-based routing/forwarding algorithm could take forwarding decisions based on the location information included in the packet header. A location management scheme is transparent/orthogonal from the viewpoint of the main underlying packet forwarding strategies, such as greedy forwarding (Camp, 2006), GPSR (Karp & Kung, 2000), restricted directional flooding (e.g. LAR (Ko & Vaidya, 2000)), etc.

In this chapter, we focus on a scalable distributed location management (DLM) scheme for large WMNs. Scalability is determined by the efficiency of a scheme in terms of overhead introduced in the network and state volume in the nodes to achieve two main goals: 1) a certain level of robustness, understood as the ability to make the location of a given node accessible even in the presence of impairments in the network, and 2) as accurate as possible location information, i.e., as up-to-date as possible.

Although a large number of location management schemes/services are available for mobile ad hoc networks (MANETs), up to our knowledge, there has not been a DLM scheme specifically designed for WMNs taking advantage of the availability of a highly static and non-power-constrained network backbone. Besides, location management schemes, even for MANETs, have only been simulated and there is no previous experimental evaluation over a real testbed implementation.

This chapter presents, up to our knowledge, the first DLM scheme, called *Vir*tual Home Region *M*ulti-Hash *Loc*ation Service (VIMLOC), specifically designed to provide high robustness and accuracy in large-scale WMNs.

It also presents an experimental performance evaluation of VIMLOC under various network load conditions. Furthermore, it presents what is, up to our knowledge, the first experimental performance comparison over a WMN testbed of three different location management schemes, namely proactive, reactive, and VIMLOC. The interest of proactive and reactive schemes resides in that they represent the two main philosophies of operation in location management (Camp, 2006), and for this reason, they are taken as reference for the comparison with VIMLOC. All three schemes have been implemented in the Click modular router framework (Kohler et al., 1999). An extensive measurement campaign has been carried out to determine the efficiency, robustness, and accuracy each of these schemes.

This chapter is structured as follows. First, the most representative location services found in the literature for WMNs and MANETs are analyzed to define which ideas better match the requirements of large-scale WMNs. Second, these ideas are adapted to design a new robust and accurate DLM location service (VIMLOC) for WMNs, by introducing the new functional entities, components, and procedures. Third, the operation of VIMLOC in combination with a geographic routing scheme is explained. Then, the main building blocks of the implementation of VIMLOC using the Click modular router framework as well as the testbed developed to test the DLM scheme are described. After that, the experimental evaluation of VIMLOC is presented and discussed and its performance is compared over a WMN testbed with two different flooding-based philosophies, namely reactive and proactive.

2. Related work

Up to our knowledge, no location management scheme specially designed to take into account the requirements of a large-scale WMN (scalability, robustness, accuracy, benefits of stable backbone, etc.) can be found in the literature.

The traditional region-based location management scheme used in typical cellular networks and its improvement, called cluster-based location management scheme, have been theoretically analyzed in (Hu et al., 2007), (Hu et al., 2009) in the context of a mesh network based on the WiMAX technology. However, their idea of WMN is not exactly the same as the one we are considering in this chapter. The WiMAX-based mesh network consists of a base station, subscriber stations that act as client-side terminals through which mobile users can access the network, and mobile terminals. It is assumed that packets are forwarded to/from the base station, which serves as a gateway between the external network and the WiMAX mesh network and subscriber stations act as relays of the root base station, hence forming a tree. Therefore, this WMN is not really a fully distributed mesh network. Thus, these management schemes have no direct application to our scenarios.

In general, previous work on distributed location schemes/services may be found mostly for MANETs. As a basis for the development of a location service scheme for WMNs, some features of location services developed earlier for MANETs have to be revisited when taking into account the specificity of WMNs. For this reason, the main location schemes used in MANETs are analyzed below from the viewpoint of its possible applicability to WMNs.

In accordance with Mauve's classification (Mauve et al., 2001), existing location services for MANETs can be defined depending on what nodes actively participate in the location process, i.e., what nodes are servers storing location information. This can be either *all* nodes in the networks or *some* specific nodes. Besides, each server can store location information about positions of *all* nodes in the network or positions of *some* specific nodes.

On the other hand, in accordance with Camp's classification (Camp, 2006), location services can be divided into three types: proactive location database schemes, proactive location dissemination schemes, and reactive location schemes. In *proactive* location schemes nodes exchange location information periodically. Correspondingly, in a *reactive* location scheme location information requested when needed. In a proactive location *dissemination* scheme *all* nodes have location databases for *all* other nodes in the network. Therefore, a node can find in its local location table information about the position of any destination node of the network. On the other hand, in a proactive location *database* scheme, typically *all* nodes in the network maintain location databases for *some* other nodes. Thus, when a node needs position information about a destination node, it first requests the location database servers storing the destination node location.

The DREAM location service (Basagni et al., 1998) is an *all-for-all* proactive location dissemination scheme. From the viewpoint of large scale WMNs, it is not reasonable that each node is considered a server database for all other nodes given the state information required. Besides, it uses flooding to spread location information throughout the network. In other words, the number of one-hop transmissions of a location update procedure is very high and scales with $O(n)$ (Mauve et al., 2001). As a consequence, DREAM has low scalability and does not seem to be appropriate for large-scale WMNs.

The Reactive Location Service (RLS) (Kaseman et al, 2002) is classified as an *all-for-some* reactive location scheme. This scheme also uses flooding, but in its request procedure. Thus, the number of one-hop transmissions of a lookup procedure is very high (Kies, 2003), (Kies et al, 2004). Therefore, this scheme has low scalability as well, and thus, it does not seem to be efficient enough for a large-scale WMN.

Other location services are proactive location database schemes. They do not require flooding since specific nodes in the network serve as location databases for other specific nodes in the network (Camp, 2006).

The Row/Column location service (Stojmenovic, 1999) is a proactive location database scheme that uses the *all-for-some* approach. Spatial orientation in a certain direction (north/south, east/west) for location update and location request procedures is used in the scheme. However, an intersection between the north/south and east/west directions does not always occur, and as a result, the location reply may often contain out-of-date location information. Some improvements (Camp, 2006) to solve this problem lead to high implementation complexity of the mechanism.

The Hierarchical location service (Kies, 2003), (Kies et al., 2004) is another *all-for-some* proactive location database scheme that is characterized by very high implementation complexity, since it deals with several hierarchical levels. Besides, the approach followed to define the appropriate number of levels in the hierarchy is not specified in (Kies, 2003), (Kies et al., 2004). The main idea of the scheme is to select geographical regions (responsible cells) that contain a location server. However, the scheme is not quite robust, since there is just one location server in each of the defined geographic regions, which may lead to loss of location databases if the server fails (Kies et al., 2004).

The Uniform Quorum System (UQS) location service (Haas & Liang, 1999) is a proactive location database scheme that uses a non-position-based routing protocol for the virtual backbone consisting of a fixed number of nodes (a quorum). Location updates are sent to a subset (a write quorum) of available nodes and location requests are referred to a potentially different subset of nodes (a read forum) (Mauve et al., 2001). This feature increases implementation complexity and limits scalability of the service. Besides, the management of the virtual backbone is not described. The services can be configured as *all-for-all*, *all-for-some*, or *some-for-some* depending on how the size of the backbone and the quorum is selected (Mauve et al., 2001). However, it is mostly configured as a *some-for-some* approach.

Two other proactive location database services have been proposed to eliminate drawbacks of the UQS (Mauve et al., 2001). These are the Grid Location Service (GLS) (Li et al., 2000), (Grid project, 2003) and the Virtual Home Region (VHR) location service (Blazevic et al., 2001), (Wu, 2005), sometimes called the Homezone location service.

They are similar to each other in the sense that each node selects a subset of all available nodes as location servers, i.e. the all-for-some approach is used (Mauve et al., 2001). These services are similar as well from the viewpoint of communication complexity (the average number of one-hop transmissions to make a location update/look up and time complexity (the average time to perform a location update/look up) (Mauve et al., 2001).

However, the main drawback of the GLS is that location update/request procedures require that a chain of nodes based on node_IDs is found and traversed to reach the location server for a given node (Kies et al., 2004). Traversing the chain of arbitrary nodes may lead to significant update and request failures if the corresponding nodes in the chain cannot be reached (Kies et al., 2004). Furthermore, controlling node failures is quite difficult (Kies et al., 2004). Besides, if nodes are uniformly distributed throughout the network, the number of entries about positions of other nodes in the location database of a node (the state volume) increases logarithmically with the number of nodes, while in the VHR the state volume is constant (Mauve et al., 2001). Furthermore, the implementation complexity of GLS is higher than that of the previous schemes, except the UQS (Mauve et al., 2001).

As for the VHR, the position of the geographic (home) region that contains the location servers storing the location information of a certain node is found by applying a hash function to the node_ID. The main disadvantage of the service is the *single* home region (Mauve et al., 2001). As a consequence, if a node is far from its home region, update packets have to travel a long way to reach the home region. If an update packet is lost along this path, the location information stored in the home region for this node may become outdated. Moreover, since in MANETs all nodes can potentially move, it may be usual to have empty home regions, especially if node density is low.

Other schemes like GrLS and FSLS (Derhab & Badache, 2008), (Cheng et al., 2007), and some other similar schemes, are variations of previous location schemes developed to solve specific problems. However, some of the improvements are attained by introducing additional implementation complexity.

In conclusion, all the location schemes described above have some shortcomings when applied to large-scale WMNs. This is mainly because they were designed and tested with MANETs in mind, i.e., all nodes were supposed to have more or less the same characteristics, be mobile, and given their power constraints, they just mounted one radio, and thus, when applied to WMNs, they would not fully exploit the advantages of WMNs. Moreover, all these proposals give performance evaluation via simulation and/or asymptotical quantitative models. Thus, up to our knowledge, there has not been any experimental evaluation or comparison of such schemes neither for ad hoc nor for mesh networks.

The above analysis motivates our work on a DLM scheme for large-scale WMNs, called VIMLOC, which is described in the following section.

3. Overview of location management schemes: VIMLOC vs. legacy schemes

This section introduces the rationale and the main design principles behind our location management scheme (VIMLOC). It also explains the entities and procedures involved in its operation. Furthermore, we also briefly explain the operation of legacy proactive and reactive schemes, as in other sections of this chapter we are quantitatively comparing the performance of VIMLOC with that of such schemes.

3.1 VIMLOC

3.1.1 Motivation

As mentioned in the previous section, none of the location services developed earlier for MANETs can satisfy the requirements to large-scale WMNs. However, by analyzing such services thoroughly, it was found that some features of the VHR location service may be considered as the basis for the development of a location service scheme for WMNs. There are some reasons for this. First, this location service is scalable, i.e., the average number of one-hop transmission required to look up or update the position of a node scales with $O(n^{1/2})$ (Mauve et al., 2001). Second, the service has low implementation complexity compared to, for instance, the UQS or GLS (Mauve et al., 2001). Third, with appropriate modifications, it can take advantage of a mesh network backbone consisting of stable mesh routers that can help to avoid the problem of empty home regions. Fourth, the limitations of having a single home region can be avoided by increasing the number of home regions storing information for each node. Further additions described in the following subsections may as well help to improve the reliability and accuracy of the location service for WMNs.

In these subsections, the detailed description of a location management scheme called Virtual Home Region Multi-Hash Location Service (VIMLOC) is presented. It is based on the VHR concept, but it contains some distinguishing features conceived to increase its robustness and accuracy in the large-scale mesh networking environment for which it is designed.

3.1.2 Main ideas

VIMLOC is a *proactive* location database scheme. Conceived as a *some*-for-*some* approach, it is designed by taking into account the specific characteristics of the WMN architecture, namely the WMN backbone. In particular, and as opposed to the VHR scheme, not *all* nodes are considered as location servers. Since the mesh backbone consists of wireless mesh routers (WMRs) that are more stable (in terms of movement and power constraints) than mobile nodes (MNs), just these WMRs are considered as *some* nodes storing location information in a distributed way.

MNs do not act as location servers and just cache location information related to their flows. Furthermore, location databases do not contain the locations of all nodes in the network, but just of *some* selected ones in the network with their node_ID-to-location mapping. This globally saves location information state, thus, improving scalability.

VIMLOC is mainly conceived to increase robustness and accuracy. These are critical requirements for the successful delivery of packets in a position-based routing environment. Additionally, mechanisms to control the overhead generated by VIMLOC are also considered.

In particular, the distinguishing features of the VIMLOC scheme follow:

- Multiple hash functions to increase robustness, i.e., one node has more than one virtual home region called *HomeGeoCluster* (HGC). This also allows load balancing of location servers
- Visited geographic region called *VisitedGeoCluster* (VGC) around a given node, in addition to its HGCs, for accuracy. It supports fine-grained mobility by diverting packets to the appropriate location as they approach the destination, i.e., arriving packets will follow the trail of the node
- "Lazy location updates" of HGCs to reduce update overhead throughout the network towards multiple HGCs. The VGC is updated more often than HGCs, thus localizing part of the overhead in a small region around the node and also attaining higher location accuracy
- Soft-state entries in location databases to avoid maintenance of stale entries. That is, the timer of the entries in the location table of those WMRs that are not anymore in the VGC allows removing stale entries.

Overall, this results in a scalable mechanism, as the average number of one-hop transmissions required to look up or update the position of a node scales with $O(n^{1/2})$.

Given that VIMLOC was designed to operate in a position-based routing environment, it is assumed that there is a coordinate space that allows assigning addresses to node identifiers, e.g., through a GPS system or by means of virtual coordinate spaces. It is also assumed that each node knows its location (i.e., address). Furthermore, the multiple hash functions used in the network are pre-defined and well-known by all nodes in network. For instance, they could be transferred to them by neighbors when joining the network.

3.1.3 VIMLOC functional entities and components

Each node n, $n = 1...N$ (where N is the number of nodes in the network, i.e., both WMRs and MNs) has a permanent node_ID. The current position of a node is defined by a temporary location address (LA).

The *i*-th *HomeGeoCluster* (HGC_i) of node *n* is the subset of WMRs inside the geographic region whose central location is obtained by applying the *i*-th hash function to the node_ID of node *n*, for *i*=1...*k*, where *k* is the total number of the hash functions used in the network.
The *VisitedGeoCluster* (VGC) of node *n* is the subset of WMRs forming the geographic/physical neighborhood (cluster) around node *n*.
Thus, each node has its own GeoClusters, in particular, some HGCs and one VGC, as it is shown in Fig. 1, in which *k*=2.
All WMRs inside a cluster (HGC or VGC) of node *n* have an entry in their location database for node *n*. In particular, the *location database* of an arbitrary WMR *r* contains an entry for a node *m* if $r \in$ VGC(*m*) or $r \in HGC_i(m)$, for any *i*=1…*k*. WMRs have location tables that are used to answer location queries. Location tables store soft state information to avoid maintenance of stale entries. MNs do not have location tables and just have caches that are used when they send packets to keep location information about nodes involved in their flows. The fields stored in each entry of the location table are: node_ID of node *n*, geographical position of the node (LA), timestamp, type of entry (cached or updated by location update protocol), flags to indicate if this entry corresponds either to the HGC or the VGC of the node_ID.

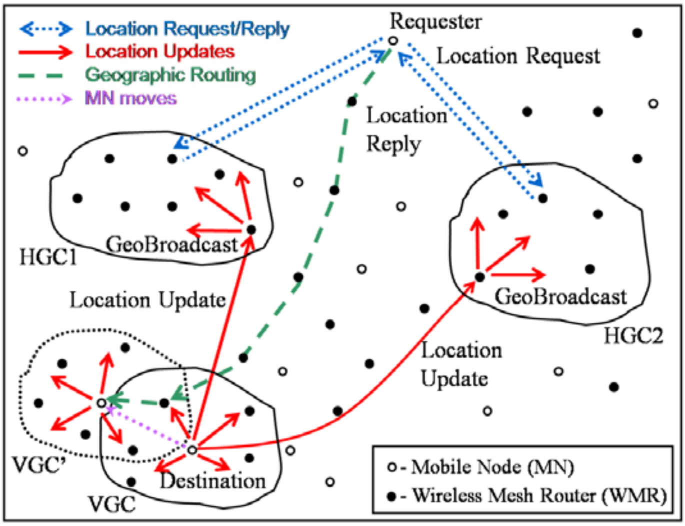


Fig. 1. Example of the operation of VIMLOC when using two hash functions

3.1.4 VIMLOC procedures

3.1.4.1 Location server selection

Location server selection is the procedure by which location servers for node *n* are selected. It is supposed that all WMRs inside a GeoCluster of node *n* are servers that store its location information. In particular, all WMRs inside the VGC and HGC_i (*i*=1…*k)* are considered as database servers. Note that it does not just correspond to the region around node *n*, but also to those around each of the positions obtained by applying the hash functions to the node_ID of node *n*. The size of GeoClusters is defined in accordance with reliability requirements of the network so that each GeoCluster maintains an approximately constant number of location servers (WMRs). It is assumed that routing will allow reaching the location servers inside a cluster.

3.1.4.2 Location server update

Node *n* initiates updates of its location servers in the following cases: 1) the network is turned on, 2) a node joins the mesh, 3) a node moves, 4) a node does not move, but soft-state refreshing is needed. Location updates are initiated by a moving node depending on the chosen scheme. In fact, there is a number of schemes that can be used for initiation of location updates, for instance, distance-based, state-based, and timer-based, among others (Wong & Leung, 2000).

The procedure of updates (or refreshing) of location servers inside the VGC is different from that of location servers inside HGCs. Inside the VGC, node *n* broadcasts updates to location servers inside the neighborhood. For HGCs, node *n* sends geobroadcast updates (Seada & Helmy, 2006) throughout the network to the positions obtained by means of the hash functions of node *n* (as explained above). That is, first, geographic routing is used so that the location update message reaches any server inside an HGC, then, the message is geobroadcasted inside the HGC to update the location information of all location servers of node *n* inside this cluster. Note that for HGCs, a location update message is sent by node *n* to all HGCs in parallel to increase system robustness, although it leads to additional overhead. Furthermore, to control the overall location update overhead of VIMLOC, a “lazy” location update procedure is applied. That is, when a MN moves, WMRs inside the VGC of the MN are updated more frequently than WMRs in HGCs. In other words, different thresholds are used in the selected scheme for triggering updates in the VGC and HGCs, e.g., different distance values in case the distance-based scheme is chosen. When a node does not move, the refreshing procedure of all HGCs and VGC is periodically carried out. In summary, the VGC of a MN is refreshed more often than its HGCs or the VGC of a WMR (as it is static).

3.1.4.3 Location request

The location request procedure is as follows. Firstly, a source node looks up the destination node_ID in the local table by checking its cache and by checking if the current node acts as location server of the destination. If an entry is not found, the source node calculates all hash functions and selects the closest one. This approach is applied to decrease the location request overhead. Location requests are sent to the best (e.g., the closest one) HGC using geoanycast (Seada & Helmy, 2006). Other options, like sending the request to all HGCs at the same time, would have other overhead-robustness trade-offs. Geographic routing is used until a location request message reaches any server (WMR) inside the HGC. The first server inside the HGC, receiving the geoanycasted request replies.

Note that the above-described VIMLOC protocol can run in parallel to any position-based routing algorithm, such as greedy forwarding (Camp, 2006) or restricted directional flooding (Ko & Vaidya, 2000). The location scheme may as well be used in conjunction with hierarchical routing approaches, i.e., those that combine both position routing for wide area routing and non-position-based algorithms for local area routing (e.g., Terminodes (Blazevic et al., 2001), Ballistic geographical routing (Rousseau et al., 2008)). However, the location scheme should be slightly modified in this case, as illustrated in (Rousseau et al., 2008).

The next subsection provides a detailed description of VIMLOC when combined with a geographic routing protocol.

3.1.5 Operation of VIMLOC in combination with geographic routing

In this subsection, the operation of VIMLOC is explained with the help of Fig. 1. When a MN first joins the network, it is loosely attached (i.e., ad-hoc mode is used) to all WMRs from which it receives beacons and selects the best one at any time instant, e.g., by choosing the least loaded. From then on, the MN periodically sends its location to its HGCs by geobroadcasting location update packets (solid lines in Fig. 1), as explained in section 3.1.4.2. When a MN moves, its VGC moves together with it (VGC' in Fig. 1), i.e., WMRs inside the neighborhood of the node change. In this way, the VGC enables packet diverting to compensate the potentially outdated information received from distant servers (or servers not updated recently). In this way, fine-grained mobility is supported. Besides, the timer of the entries in the location table of those WMRs that are not anymore in the VGC of the MN allows removing stale entries.

When a source node (Requester in Fig. 1) gets a packet from an application that must be sent to the destination node_ID, it uses VIMLOC to obtain the LA of the destination. The packet is in the buffer of the source/requester node while the node is obtaining the corresponding LA of the destination. By applying all the hash functions to the destination ID, the source node (requester) obtains the central location of all the HGCs of the destination node. Among them, it may choose the closest one or it may select a subset (or all) of them, mainly depending on the overhead-response time trade-off. In particular, in Fig. 1, the request is simultaneously sent to both HGCs (dotted lines).

The location request is geoanycasted, that is, once any of the WMRs inside the HGC receives the request, it sends the reply back to the source node, and it is not further forwarded inside the HGC. After receiving the position reply, the source node puts the location information of the destination node into the packet header and sends the data packet through intermediate WMRs to the destination node using the underlying geographic routing protocol (the dashed line in Fig. 1).

When an intermediate WMR receives a packet, it first checks whether the destination LA is its own or the address of a MN attached to the WMR. If this is the case, the packet is delivered. Otherwise, the WMR checks whether the destination node_ID is among its location table entries (only entries with flags corresponding to VGC are checked) to appropriately divert the packet, if needed. In other words, it is checked whether the packet has reached a location server inside the VGC of the destination node. In this case, the destination LA field in the packet header is overwritten with the value obtained from the entry in the location table corresponding to the destination node_ID. Then, geographic forwarding eventually delivers the packet to the correct destination inside the VGC, even if the information initially used by the source node was a bit outdated. On the other hand, if there is no entry for the destination node_ID in the location table (e.g., because the packet did not reach the VGC), the packet is forwarded based on its current LA. Note that the same procedure is applied to location replies in case the source/requester node is also moving, i.e., the LA of the source node in a header of a reply packet may be updated by intermediate WMRs while the packet approaches the node.

After both communicating nodes establish a communication, location tracking (Blazevic et al., 2004) is used. Therefore, data packets periodically piggyback the current locations of communicating nodes. If there are no data to send, nodes send location control packets with their location information.

3.2 Proactive and reactive location management schemes

For the comparison with VIMLOC, one proactive and one reactive location management are also considered, as they represent the two main philosophies of operation in location management (Camp, 2006). The proactive scheme under consideration may be classified as a proactive location dissemination scheme (Camp, 2006), e.g., DREAM (Basagni at al., 1998). As all nodes have location databases that store information about all other nodes in the network, it is an *all*-for-*all* approach (Mauve at al., 2001). Moreover, all nodes periodically flood the network so that all WMRs update the LA of that node. Therefore, there are no location requests sent through the network, as they are answered by looking up the local location table.

As for the reactive scheme, e.g., RLS (Kaseman at al., 2002), before a node sends a packet towards a certain destination node_ID, a location request asking for the LA of the destination node is sent to all nodes by flooding the network. The WMR owning this location information (i.e., the WMR to which the destination node is attached) sends a reply back to the requester with its node_ID-to-LA mapping. This approach may be classified as an *all*-for-*some* approach, that is, every node in the network maintains location information on some other nodes in the network (Mauve at al., 2001). In our case, it is assumed that each node only maintains its own location information. And thus, there is no location update procedure in the scheme.

The Click environment is used (Kohler et al., 1999) for the implementation of these three protocols (VIMLOC, proactive, and reactive schemes). The wireless mesh networking framework, including the testbed and some implementation issues of the protocols, is presented in the next section.

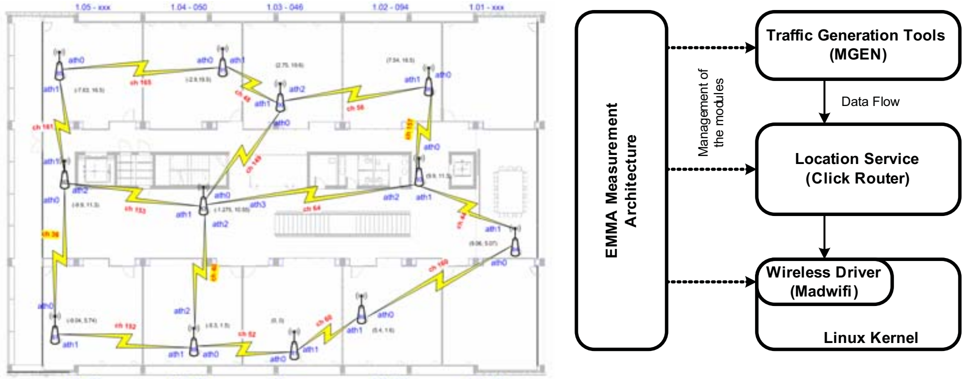
4. Wireless mesh networking framework

This section describes the main implementation choices and the testbed over which all the results presented in this chapter have been obtained, as well as the automated measurement framework that was developed to gather them. It also presents the main parameters that characterize the scenarios under evaluation.

4.1 Wireless mesh networking testbed

An indoors wireless mesh networking testbed was built to evaluate the VIMLOC distributed location management scheme in conjunction with greedy forwarding and to compare it with simple proactive and reactive schemes. The experimental setup includes a 12-node multi-radio backbone WMN, as shown in Fig. 2(a), over an approximate area of 1200 square meters. All nodes run Click 1.6.0 over a Linux kernel 2.6.24. Backbone nodes (WMRs) are built based on a mini-ITX board (Pentium M 1.6 GHz) and mount up to four CM9 wireless cards (802.11abg) with Madwifi driver v0.9.4. One of these cards may be used for offering access to MNs. Notice that antennas are omnidirectional and a link is established between two nodes if they have cards assigned to the same channel. In this way, the topology of the testbed can be easily modified by modifying channel assignment. For simplicity, channels are assigned in the network so that all the links are in different channels in order to minimize contention and interferences. External interference with other wireless networks usually configured in 2.4 GHz band is avoided by configuring the wireless cards to 5 GHz band (i.e., 802.11a mode).

Experiment automation benefits from the capabilities of the EXTREME Testbed® (EXTREME, 2010). The autoconfiguration software provides automation to scenario configuration tasks. It is composed of custom made code and as well as code from various open source software projects.



(a) (b)

Fig. 2. (a) Wireless Mesh Networking testbed scheme. Plan of the building showing the positions of backbone nodes, links, and channels configured in each link between nodes, and (b) Software framework in all nodes of the testbed

4.2 Software framework

Fig. 2b presents the software framework in all nodes of the testbed for the evaluation of the location management schemes. The software architecture for location management is based on the Click modular router (Kohler et al., 1999), which is modular and easy to extend. Click was designed to implement networking protocols for flexible and configurable routers. A Click router is composed by generic and simple packet processing elements and a configuration file that defines the interconnection of the processing elements and how the packets flow through the router. We exploit its capabilities by using its elements, but we have also developed new elements for location management. Moreover, developing simple elements allows reusing these elements to implement different location management protocols only by changing the Click configuration file of the Click router.

The main building blocks of the Click stack for the VIMLOC implementation are highlighted in Fig. 3.

In the upper-left part, there is the new User interface. The User interface is in charge of the communication between the user applications (including traffic generation tools) and the Click router. That is, Click generates a new virtual interface that can be used by any legacy user application instead of the real wireless interfaces. All such virtual interfaces in all the nodes of the network are assigned IP addresses that belong to the same subnetwork, and the operating system (OS) sends regular IP packets with the user data to this *fake local* subnetwork. In this way, the OS routing tables and forwarding mechanisms can be bypassed and those implemented in Click can be used instead. However, what the legacy application treats as an IP address is in fact treated as a node_ID by the Click stack. So, the IP packet is encapsulated into what we may call a geographic packet. Its header contains the source and destination IDs (i.e., the IP addresses assigned to the virtual interfaces), source and destination LAs (provided by the location scheme), packet type, and additional information of the location protocol.

The lower part of Fig. 3 shows wireless interfaces that are in charge of the communication between the wireless devices and the Click router. Packets might be sent using unicast MAC

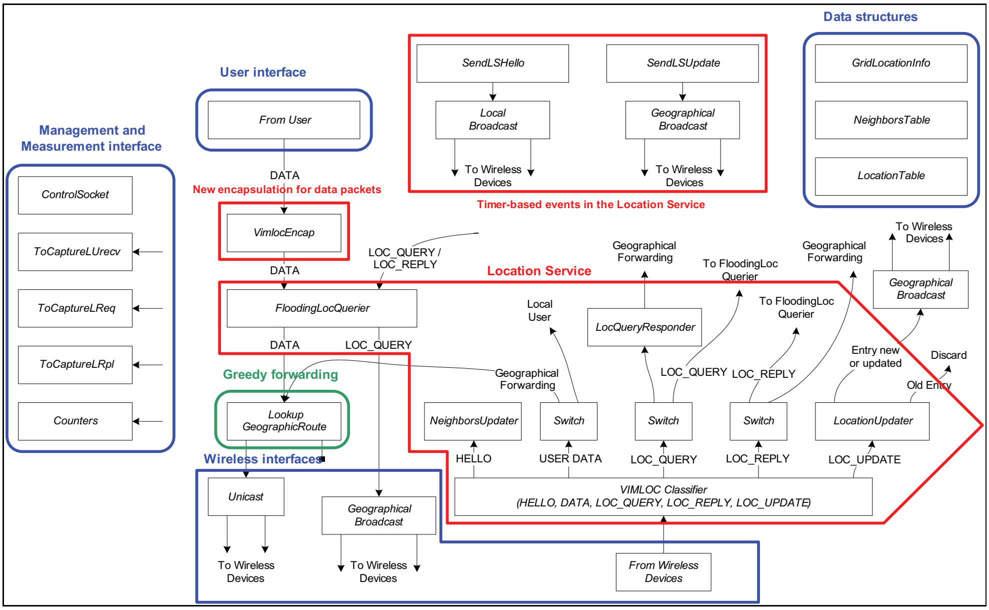


Fig. 3. Software building blocks, including a simplified diagram of the Click stack for the VIMLOC protocol

addresses obtained by means of a geographic forwarding strategy, or in broadcast mode for some packets (e.g., location server updates of HGCs for VIMLOC). Eventually, the geographic packet is encapsulated into an Ethernet packet before delivering it to the wireless card driver.

The core of the VIMLOC (location service engine) and geographic routing implementation is presented in the central part of Fig. 3. In the current implementation, greedy forwarding is used as geographic routing protocol, i.e., a node forwards the packet to its neighboring node that makes the most forward progress (Camp, 2006) in terms of distance towards the destination. It comes mainly from the one implemented by the Grid project (Grid project, 2003) with some adaptations for the location management service and the testbed.

Firstly, there are some important data objects in the central part: the *LocationInfo* element keeps the LA of the WMR, and the *NeighborsTable* and the *LocationTable* elements gather the information of the location database.

Secondly, there are the processing elements. (1) The *FloodingLocQuerier* element is in charge of receiving geographic data packets, and it also looks for the LA of the destination WMR. If there is no information in the location database for this WMR, it starts the location query request procedure by sending location requests to the appropriate location servers in the HGC or the VGC according to the location server selection procedures. It also receives the location replies for the packets waiting in the queue to be transmitted. Once the LA is resolved, it sends the packet to the following processing element. (2) The *LookupGeographicRoute* implements greedy forwarding, it receives a packet for a destination LA, and it looks up in the location database of the neighboring WMR that is closer to the destination LA. (3) The *VIMLOC Classifier* dispatches the received packets to the suitable element that processes it. Some of them will be used locally to populate the location database (e.g., location update packets), and others will be forwarded geographically or/and processed locally depending on whether they already reached or not the destination region towards which the packet is sent. (4) The *NeighborsUpdater* element populates the *NeighborsTable* with information in the incoming packets of the neighboring WMRs in the VGC. (5) The *LocQueryResponder* element answers the location queries if the node processing the packet belongs to an HGC or the VGC of the destination node. (6) The *LocationUpdater* element populates the *LocationTable* with the mapping between node_IDs and their LAs.

In parallel, VIMLOC has some timer-based events. Mainly, it sends location updates to the VGC in local broadcast mode and to the HGC in geobroadcast mode through the *SendLSHello* and *SendLSUpdate* elements, respectively.

This is only a high-level picture of the implementation details. There are other important elements to check if the packet reached the HGC or VGC; to send geobroadcast packets; to send geoanycast packets; and to calculate the regions of the HGC and VGC (i.e., the hash functions). There is also the Management and Measurement Interface that is in charge of the communication between the Click router and the management and measurement tools. It generates the counter files and packet capture files (see Section 4.6) according to the management messages received from the measurement tools.

Simple proactive and reactive location protocols have also been implemented in the testbed to compare their performance with that of the VIMLOC protocol.

The same Click environment for the implementation of these two protocols was used to obtain comparable results. Notice also that Click allows modifying the nodes so that they implement one or the other scheme by simply changing the configuration. In this way, one avoids having to develop the whole processing path from scratch. Furthermore, and due to the modularity and flexibility of the Click Modular Router, the Click stack for each of these two protocols is a simplified version of that of VIMLOC. For instance, in our implementation of the simple proactive dissemination scheme, the concept of VGC and HGC is missing. It just floods location updates periodically to every node in the network. As for location replies, they are answered by just querying the local database, i.e., there is *no location request procedure* for the services. This translates into a simplified Click protocol stack where all the processing elements related with the location queries and the location replies are missing.

Similarly, in the simple reactive scheme, there is *no periodical location update procedure* and WMRs flood a location query to the entire network in case a node initiates a communication and needs the location for another node. As a result, nodes do not have the location database to maintain location update entries. Only the node with that destination node_ID answers with a location reply. Thus, the generation and processing of the location updates is missing from the Click protocol stack.

Click configuration files are different for each node in the testbed. To ease the deployment and management of location schemes, a mechanism to generate the Click configuration file for each node in the testbed has been developed. Fig. 4a illustrates how it works. In particular, a generic template configuration file and different variable files are used for each node. A generation tool uses the generic template as a model and generates all the Click configuration files according to the values in the variable files. In Fig. 4b, an example of this variable file is shown. It contains the specific values for each node.

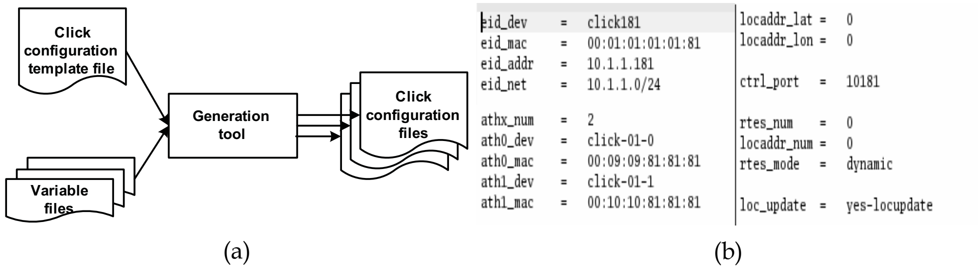


Fig. 4. (a) Operation of dynamic Click configuration files, and (b) Example of variable file

4.3 Parameters defining the test environment

All average values presented in this paper are based on 30 replications of the experiment. The duration of each replication is 120s. Background traffic does not generate any control traffic for location management. For each link, one bi-directional constant bit rate UDP flow is generated. That is, the destination is the neighboring node through that link. Packet rate is varied to study the operation under different traffic loads in the network. Thus, low load corresponds to no background traffic at all. High load corresponds to a packet rate of 850 pkts/s. This value was chosen to reach network saturation conditions. Finally, medium load corresponds to 550 pkts/s. A packet size of 1000 bytes was chosen. The duration of background flows is the same as that of the replication.

Besides, given the interest on evaluating distributed location management, reference flows were also generated, i.e., those causing location requests. A new data flow is started every 10s by each node towards a random destination. However, as the focus of this chapter is on assessing the control overhead, data traffic is just used to trigger the location requests and is not actually sent through the network. Considering the duration of 120s and the size of the network (12 nodes), 144 requests are generated in the network in each replication. The number of retransmissions in each link is configured to three and the link rate is fixed to 54 Mbps. The transmission power is fixed to 50mW.

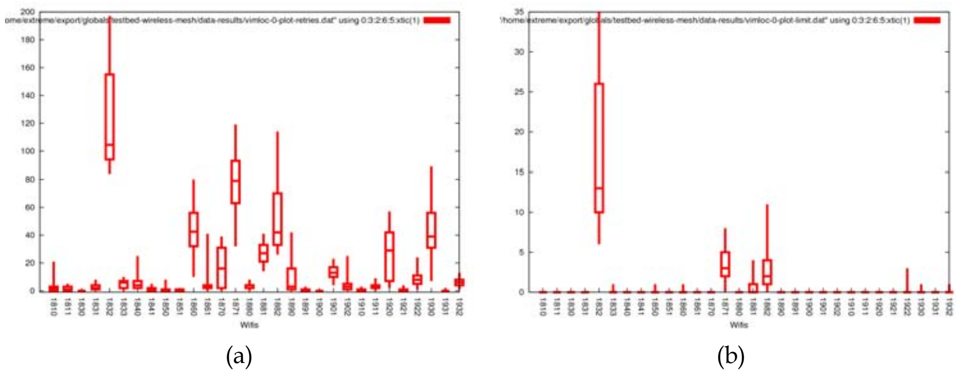
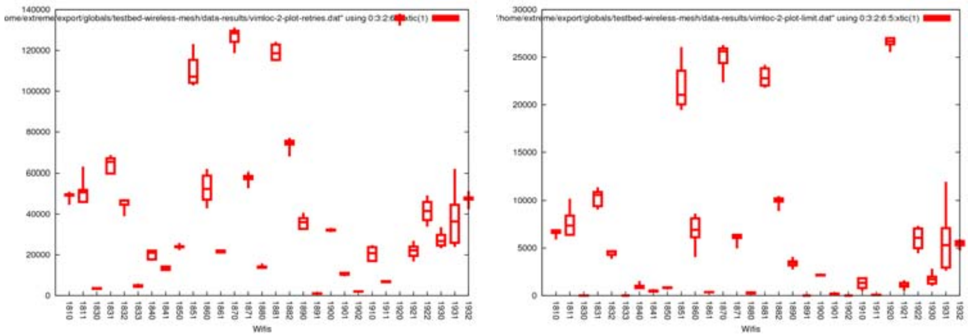


Fig. 5. Box plot of (a) number of retransmissions and (b) number of times retransmissions were exhausted for all the wireless cards of the testbed under low load conditions



(a) (b)

Fig. 6. Box plot of (a) number of retransmissions and (b) number of times retransmissions were exhausted for all wireless cards of the testbed under high load conditions

Note that some additional work to evaluate the link quality of the testbed has been carried out. This work is used for representing the link quality status when performing an experiment, and is used to understand the outcome of the experiment results. Fig. 5a shows the box plot of the number of retransmissions and Fig. 5b the box plot of the number of times retransmissions were exhausted in each wireless card of the testbed under low load conditions. Hence, in the X-axis, 1831 means node 183 and wireless card number 1. In the Y-axis, the number of total layer 2 retransmissions is plotted. The main goal is to check the status of the network during the time while any developed network protocol is evaluated. For instance, it helps detecting anomalous operation of a certain link, meaning that the antenna fell or new furniture was added, thus modifying the path propagation conditions. These figures also help to differentiate losses in channels and other causes (e.g. buffer overflow at the nodes).

The same graphs were obtained under various conditions showing the same trend, excepting for the substantial increase in the Y-axis values when load is increased (see Fig. 6). In fact, when load is increased the total number of losses in the network substantially grows because more packets traverse each wireless link, hence generating more contention, buffer overflows, and collisions. As observed in Fig. 6, the scenario with high load represents really adverse conditions (saturated network) for the operation of the network, which should never be reached if it is appropriately managed. In any case, we also evaluate our location service in this scenario to test its robustness when many losses occur in the network.

4.4 Measurement framework

The Extreme MeasureMent Architecture (EMMA) (Portoles et al., 2006) is used to define and control the traffic characteristics (e.g., source-destination pairs, packet rates, packet sizes) as well as important events during experiment runs (e.g., new flow, end of flow). EMMA provides the basis for handling the required number of experiment replications to obtain statistically significant results. All traffic is generated using the MGEN v4.2b6 tool.

Additionally, some enhancements were developed to gather, parse, and present the results. More specifically, the following functionalities were added:

- Random Flow Generation: EMMA was expanded to randomly generate source-destination pairs for reference traffic. To send reference traffic in a random manner from each node in the network, the implementation was expanded with the choice of random destination node, hence generating random source-destination pairs. The set of potential destination nodes are all the nodes in the network except for the source node.

- Interaction between the Click framework and EMMA: Counters were introduced at key points in the Click stack to count and/or dump packets containing relevant information to calculate the parameters of interest (see Section 4.6). The Click and the EMMA frameworks were synchronized through the use of Click handlers. Two Click handlers are launched from the EMMA code indicating the beginning and the end of each experiment replication. After the reception of these events, the Click framework updates the information in its counter files so that all the packets taken into account are within the duration window of the experiment replication. This was done for all nodes and for the three location management schemes under evaluation.
- Post-processing parsers: A series of post-processing scripts download all the experiment data from the local storage of each node to a central server through an out-of-band control network. Data stored can be managed by post-processing parsers either to obtain results related to the whole set of repetitions of the experiment or to obtain results separately from each repetition in the experiment. Their purpose is to generate statistical values of the assessed parameters.
- Plot automation functionalities: Due to the increase in the amount of tests and parameters, plot generation was automated in EMMA. Graph results are stored in a fixed location in the central server. Furthermore, another observed requirement was to generate the same graphs in different formats. This would imply repeating the execution of all the post-processing parsers. With high amounts of data, this task can be highly time-consuming. To avoid this, an input parameter was added to EMMA so that it can be used only as a plot generation facility.

4.5 Practical implementation issues of the location schemes

The following options are initially chosen to ease the implementation of VIMLOC. Each node has two HGCs and one VGC to provide the diverting capability. Thus, two different hash functions provide two geographical positions that correspond to the centers of the HGCs. Each HGC is composed of two WMRs and the VGC is composed of the neighbors of a certain node. For the location update procedure, the timer-based scheme is applied. The update interval of WMRs inside HGCs is twice that of the VGC (i.e., 10s and 5s, respectively). There is no retransmission in the location update procedure. In the location request procedure, a requester sends location queries to two HGCs in parallel. And, in the current setup, each node sent one request every 10s. Entries in location tables expire after 22s. Periodic Hello packets for discovering the coordinates of neighbors are sent every 10s by each node.

4.6 Gathering of measurement results

For the gathering of measurement results, counters for sent and forwarded packets, as well as packet capture tools to obtain information on received packets, have been implemented for the three above protocols. They were implemented by means of Click and EMMA and deployed in backbone nodes (WMRs).

As an example, the diagram presented in Fig. 7 illustrates the approach to gather results for the procedure of the location update of the VIMLOC scheme (VIMLOC-LU) in a WMR.

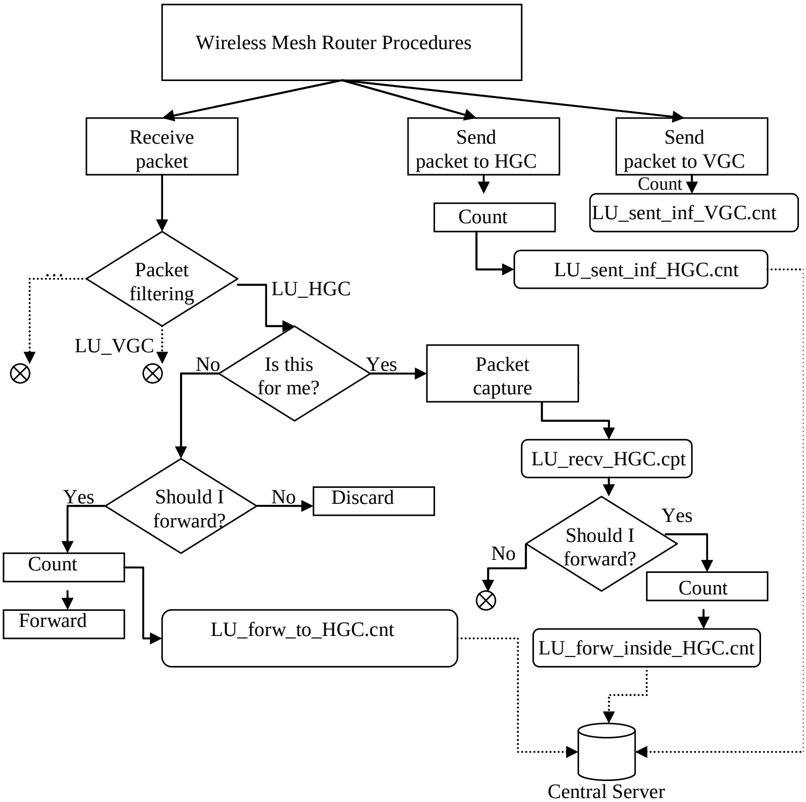


Fig. 7. Approach to gather measurement results (the VIMLOC-LU procedure)

In particular, for the VIMLOC-LU procedure, a counter for packets sent towards the HGCs is installed in each interface of a WMR (LU_sent_inf_HGC.cnt), i.e., a packet sent to two different HGCs in parallel is counted as two packets since two interfaces are used, although packets have the same sequence number. A similar counter is implemented for packets sent towards the VGC (LU_sent_inf_VGC.cnt). There are also two more counters to make the distinction between packets forwarded to HGCs and packets geobroadcasted inside HGCs (LU_forw_to_HGC.cnt and LU_forw_inside_HGC.cnt).

Note that all information about sent/forwarded packets gathered by these counters in each WMR is eventually transferred to the central server. An example of the structure of a counter file stored in the central server that merges counter files coming from various nodes is shown in Table 1.

Node_ID (1..12)	Number of sent/forwarded packets
5	15
8	32
…	….

Table 1. Structure of a merged counter file stored in the central server

Besides, packet capture tools are used to collect information on the behavior of WMRs when acting as location server for one or more nodes. Location updates received when the WMR belongs to an HGC are logged in the LU_recv_HGC.cpt file and in the LU_recv_VGC.cpt file when it belongs to a VGC. An example of the structure of a capture file saved in the central server (after merging files coming from different WMRs) is shown in Table 2.

Node_ID	Source_ID	Seq #	TTL
8	7	1	7
8	2	1	4
8…	7	2	7
9	3	1	5
9…	7	1	6
...	...	...	...

Table 2. Structure of a capture file

Thus, a capture file contains information about Node_ID (the ID of the WMR that sent its capture file to the server), Source_ID (the node for which the WMR acts as location database), sequence number (of the received location update message), and TTL.

The approach to gather results for the location request-reply procedures of VIMLOC (VIMLOC-LR), the LU procedure of the proactive scheme, and the LR procedure for the reactive scheme is similar to the above-described one for VIMLOC-LU procedure.

In particular, for the VIMLOC-LR procedure, in the implementation scenario, a source node (requester) sends a location request with the same sequence number in parallel to both HGCs consisting of two WMRs. As a result, a counter for sent location request packets (LR_send_inf.cnt) is installed in each interface of a WMR. Besides, for the calculation of performance parameters, it is needed to obtain a counter just for those sent location request packets that have different sequence number (LR_sent.cnt). There is also a counter for forwarded request packets towards an HGC (LR_forw.cnt).

When a location request packet reaches the first server inside HGC (the replier) and a location reply packet is sent to the requester, the replier changes packet type and source ID in the packet fields though TTL does not take its initial value. To count sent and forwarded location replies towards the requester, two more counters are implemented (LRpl_forw.cnt and LRpl_sent.cnt). Besides, a capture file is also generated (LRpl_recv.cpt) for received location replies. The LRpl_recv.cpt file just keeps information about the first reply packet that reached the requester, i.e., the second reply packet (with the same sequence number) obtained during the same LR procedure (from the second HGC) is discarded and is not dumped to the file.

For the proactive scheme, having just the LU procedure, three counters are needed, namely, LU_sent.cnt, LU_sent_inf.cnt, LU_forw_int.cnt, and a capture file for received location update packets (LU_recv.cpt).

Correspondingly, for the reactive scheme, having just the LR procedure, three counters for sent/forwarded location requests (LR_sent.cnt, LR_sent_inf.cnt, LR_forw_inf.cnt) and two counters for sent/forwarded location replies (LRpl_sent.cnt and LRpl_forw.cnt) were deployed, and one capture file for received location reply packets was generated (LRpl_recv.cpt).

Note that the structure of counter and capture files is similar to the one presented in Tables 1 and 2, respectively.

5. Parameters assessed

Based on the gathered measurement results, the post-processing scripts calculate performance parameters to compare the three location management schemes, namely VIMLOC, the proactive scheme and the reactive scheme. In particular, the following performance parameters have been defined for this purpose.

5.1 Success Rate (SR)

For the *LU procedure,* the SR is the fraction of LU packets (out of the total number of updates) successfully delivered to the nodes acting as location servers for the originator of the LU, that is, all WMRs for the proactive scheme, and WMRs inside HGCs and VGCs for VIMLOC.
For VIMLOC, the SR for LUs is calculated based on the captured files and counters for the chosen implementation options (Section 4.6) as follows:

$$SR_{LU\_VIMLOC} = \frac{S_{LU\_recv\_HGC} + S_{LU\_recv\_VGC}}{S_{LU\_sent\_\inf\_HGC} + S_{LU\_sent\_\inf\_VGC} + S_{LU\_forw\_inside\_HGC}} \cdot 100\% , \quad (1)$$

where $S_{LU\_recv\_HGC}$ is the number of LU packets received by WMRs inside HGCs (the number of entries in the table of the LU_recv_HGC.cpt file), $S_{LU\_recv\_VGC}$ is the number of LU packets received by WMRs inside VGC (the number of entries in the table of the LU_recv_VGC.cpt file), $S_{LU\_sent\_inf\_HGC}$ is the total number of LU packets (including packets with the same sequence number) sent through all node interfaces to HGCs (i.e., the sum of the number of sent packets in all entries of the table in LU_sent_inf_HGC.cnt), $S_{LU\_sent\_inf\_VGC}$ is the total number of LU packets sent through node interfaces to VGCs (the sum of the number of sent packets in all entries of the table in LU_sent_inf_VGC.cnt), $S_{LU\_forw\_inside\_HGC}$ is the total packets of forwarded LU packets inside HGCs (the sum of the number of forwarded LU packets in all entries of the table in LU_forw_inside_HGC.cnt).
For the proactive scheme, the SR for LUs is defined as:

$$SR_{LU\_PRO} = \frac{S_{LU\_recv}}{(N-1)S_{LU\_sent}} \cdot 100\% , \quad (2)$$

where S_{LU_recv} is the number of LU packets received by all WMRs (the number of entries in the LU_recv.cpt table), S_{LU_sent} is the total number of LU packets with different sequence number sent by WMRs (the sum of the number of sent packets in all entries of the table in the LU_sent.cnt), and N is the number of nodes in the network (12 in the current testbed implementation).
For the *LR procedure,* the SR is defined as the fraction of requests whose reply is successfully delivered to the requesting node.
For VIMLOC, SR for the LR procedure is calculated as

$$SR_{LR\_VIMLOC} = \frac{S_{LRpl\_recv}}{S_{LR\_sent}} \cdot 100\% , \quad (3)$$

where S_{LRpl_recv} is the total number of received location reply packets (the number of entries in the file LRpl_recv.cpt), S_{LR_sent} is the total number of sent location request packets that

have different sequence number (the sum of the number of sent packets in all entries of the table in LR_send.cnt).
For the reactive scheme, the SR for LR procedure is calculated by the same formula.

5.2 Communication Complexity (CC)

The CC is the average number of one-hop transmissions required 1) to update the position of a node (it applies to VIMLOC protocol and the proactive scheme), or 2) to look up the position of a node (the LR procedure), which applies to VIMLOC and the reactive scheme. Correspondingly, for VIMLOC, the CC for the LU procedure can be calculated as:

$$CC_{LU\_VIMLOC} = S_{LU\_sent\_\mathrm{inf}\_HGC} + S_{LU\_sent\_\mathrm{inf}\_VGC} + S_{LU\_forw\_to\_HGC} + S_{LU\_forw\_inside\_HGC}, \quad (4)$$

where $S_{LU\_forw\_to\_HGC}$ is the total number of forwarded packets towards HGCs (the sum of the number of forwarded packets in all entries of the table in LU_forw_to_HGC.cnt) and the rest of parameters was already defined above.
For the proactive scheme, the CC for the LU procedure is

$$CC_{LU\_PRO} = S_{LU\_sent\_\mathrm{inf}} + S_{LU\_forw\_\mathrm{inf}}, \quad (5)$$

$S_{LU\_sent\_inf}$ is the total number of sent LU packets (the sum of the number of sent packets throughout in all entries of the table in LU_sent_inf.cnt), $S_{LU\_forw\_inf}$ is the total number of forwarded LU packets (the sum of the number of sent packets in all entries of the table in LU_forw_inf.cnt).
For VIMLOC, the CC for the LR procedure is defined as:

$$CC_{LR\_VIMLOC} = S_{LR\_sent\_\mathrm{inf}} + S_{LR\_forw} + S_{LRpl\_sent} + S_{LRpl\_forw}, \quad (6)$$

where $S_{LR\_sent\_inf}$ is the total number of sent location requests (the sum of the number of sent packets in all entries of the table in LR_sent_inf.cnt), S_{LR_forw} is the total number of forwarded LR packets (the sum of the number of sent packets in all entries of the table in LR_forw.cnt), S_{LRpl_sent} is the total number of sent location replies (the sum of the number of sent packets in all entries of the table in LRpl_sent.cnt), S_{LRpl_forw} is the total number of forwarded LRpl packets (the sum of the number of sent packets in all entries of the table in LRpl_forw.cnt).
For the reactive scheme, the CC for the LR procedure is calculated by means of the same formula.

5.3 Overall Overhead (OO)

The OO is the total amount of bytes sent to the network for a certain procedure, and this is calculated as the addition of the number of packets of each type (i.e., update, request, or reply), multiplied by their respective size (P_{size}).
Thus, for VIMLOC and the proactive scheme, the OO for the LU procedure is defined as

$$OO_{LU} = CC_{LU} \cdot P_{size}, \quad (7)$$

and for VIMLOC and the reactive scheme, the OO for the LR procedure is defined as

$$OO_{LR} = (S_{LRpl\_sent\_\mathrm{inf}} + S_{LRpl\_forw\_\mathrm{inf}})P_{LR\_size} + (S_{LRpl\_sent} + S_{LRpl\_forw})P_{LRpl\_size}. \quad (8)$$

5.4 Efficiency Factor (EF)

The EF is defined for both the LU and the LR procedures as the ratio between the number of "useful" one-hop transmissions and the number of overall one-hop transmissions. The number of *useful one-hop transmissions* is equal to the number of hops of the most efficient path followed to deliver a packet to the appropriate location server node. The number of overall one-hop transmissions means the total number of hops used in the LU/LR procedure, i.e., this is the CC. With this parameter, we try to capture the how inefficient in terms of wasted transmissions (e.g., by flooding) each of the schemes.
The expression for calculation of the EF is the same for both the LU and LR procedures and can be defined as follow:

$$EF_{LU/LR} = \frac{UHT_{LU/LR}}{CC_{LU/LR}}, \tag{9}$$

where $UHT_{LU/LR}$ is the number of useful one-hop transmissions for LU or LR procedures. It is determined from the corresponding tables of capture files, where the number of "useful" hops traversed by successfully received location update/reply packets may be counted by means of the initial and final values of TTL carried in packets. In the case of VIMLOC, this parameter is the sum of the number of useful one-hop transmissions for packets received by nodes belonging to VGCs and HGCs.

5.5 State Volume (SV)

The SV is measured as the average number of entries in the location database of a node. And it is defined just for the proactive scheme (from the LU_recv.cpt file) and VIMLOC (from the LU_recv_HGC.cpt and the LU_recv_VGC.cpt files), since the reactive scheme does not contain a location database.

5.6 Successful Communication Complexity (SCC)

The SCC is the average number of one-hop transmissions required to have a successful reception of either 1) one location update packet for the proactive scheme and the VIMLOC-LU procedure, or 2) one location reply packet for the reactive scheme and VIMLOC-LR procedure.

6. Results and discussions

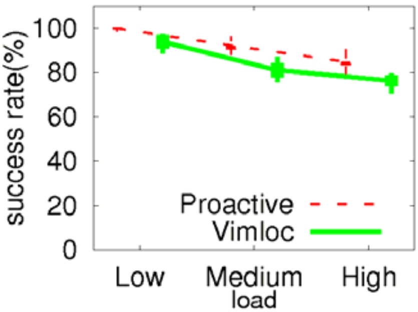
This section presents the results of the experimental evaluation of the parameters defined above. Boxplots are used to present the minimum, 25 percentile, median, 75 percentile, and maximum. Additionally, the curve for each scheme represents the average values for the various background loads under test. Note that VIMLOC was designed for medium/large scale environments. However, it is being tested in a small scale testbed and the results presented are expected to improve with respect to flooding-based approaches as the size of the network increases.

6.1 Robustness of VIMLOC

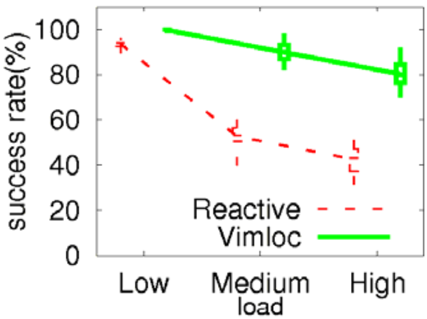
The SR is used in this section to compare the robustness of the three mechanisms. Let us first recall that robustness refers to the ability of the mechanism to carry out an LU or LR procedure even in the presence of impairments in the network. Such impairments may come from the variability of the wireless medium or the potential losses due to the background load introduced in the experiments.

Fig. 8a presents the comparison of the SR for the LU procedure under different background loads in the network. The proactive scheme works perfectly (i.e., SR=100%) for low loads. This is due to the flooding procedure, as there are multiple paths that a flooded update packet may follow to reach each WMR of the network. Thus, even in case one update is lost in one of these paths, there are others that allow reaching each of the WMRs.

The SR for VIMLOC is a bit less than 100% for low loads, because the losses due to fading in the wireless channel make more likely that the update is lost in the unicast part of the geobroadcast procedure. There is just a slight decrease of the SR as load increases for both schemes since background load introduced in the network makes more likely that the packet is lost in all the possible paths that could be used to reach a certain WMR. Anyway, they both show a similar behavior despite their very different operating principles.



(a)



(b)

Fig. 8. Success Rate vs. load. (a) Location update (Proactive and VIMLOC) and (b) Location request (Reactive and VIMLOC)

For the request-reply (LR) procedure (Fig. 8b), the SR for VIMLOC is equal to 100%. It decreases for medium and high loads. The reactive mechanism presents SR values smaller than VIMLOC for low loads, but the SR dramatically decreases when load increases. This is because the reply packet for the reactive scheme is sent back in unicast mode to the requester node. And, as explained above for VIMLOC, the loss of a unicast packet is quite likely, especially for medium and high loads. On the other hand, in the current setup, there are two HGCs from which a reply could be received, and thus, even if unicast, the loss probability of at least one of them is smaller. Therefore, the SR is much better for VIMLOC.

6.2 Trade-off between robustness and overhead

The SR allows quantifying the robustness of the mechanisms. However, the way VIMLOC and the flooding-based protocols achieve their respective SR is different in the sense that the latter do it by introducing a huge amount of overhead, which is inherently inefficient. For this reason, we present some results to quantify the trade-off between robustness and the overhead introduced by each of the mechanisms.

The Y-axis of Fig. 9 presents the overall overhead (OO) for each mechanism for the LU procedure (Fig. 9a) or the LR procedure (Fig. 9b). And the X-axis measures how successful each of these procedures was. In Fig. 9a, it presents the probability of having outdated location information at nodes (POI), and it is calculated as 1-*SR* for location updates. In Fig. 9b, it presents the probability of not getting an answer to a request (PNA) and it is measured as 1-*SR* for the location request-reply procedure. Each point in the figure represents the OO for one replication of the experiment. Recall that 30 replications were ran for each of the three loads considered.

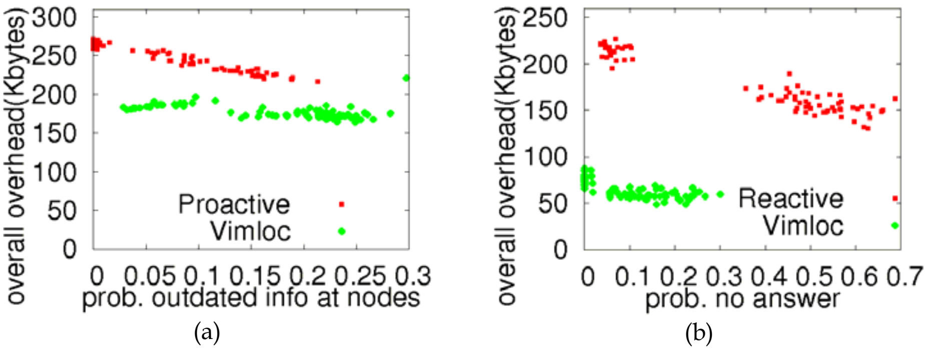


Fig. 9. (a) OO vs. probability of outdated info at nodes (update procedure) and (b) OO vs. probability of no answer (request-reply procedure)

Focusing on Fig. 9a, one observes that VIMLOC generates less overhead than the proactive scheme, but with a bit worse POI values. However, note that the accuracy is twice worse for the proactive scheme, because, with VIMLOC, forwarding can benefit from updates sent to certain key nodes (VGC) that are updated twice as often as in the proactive scheme. As a consequence, they have information which is twice fresher, which is fundamental in highly mobile environments. Achieving the same level of accuracy for the proactive scheme would imply flooding the network twice as often, thus doubling the control overhead and generating more congestion in the network.

With respect to Fig. 9b, one observes that the overhead of the request-reply procedure for the reactive mechanism is much higher than that for VIMLOC, since it follows a flooding-based behavior. Besides, VIMLOC has less probability of not getting an answer to a request than the reactive scheme. This is because the potential loss of unicast packets is compensated by the fact that VIMLOC sends two requests in parallel, thus doubling the chance to receive at least one reply.

6.3 Efficiency of VIMLOC

The state volume (SV) parameter illustrates the efficiency of VIMLOC and the proactive schemes in terms of number node_ID-to-LA mapping entries stored in the location database. The location database of an arbitrary WMR for VIMLOC contains an entry for a node if the WMR belongs to one of the two HGCs of the node or the WMR belongs to the VGC of the node. As it is seen from Fig. 10, the average number of entries in the location table of a WMR for the proactive scheme is equal to 11. That is, each node contains an entry for all other nodes in the network. And it is 7 entries for VIMLOC. Therefore, the state location information stored in a WMR for VIMLOC is about 40% less than for the proactive scheme. But one should notice that, as explained above, VIMLOC has twice better accuracy. If we assign a similar accuracy for VIMLOC and the proactive scheme, the state information stored in a WMR for VIMLOC is almost three times less in accordance with our measurements. Moreover, taking into account the flooding-based nature of the proactive scheme, it is expected that the difference between the state volume values of the two schemes will significantly increase with the size of the network.

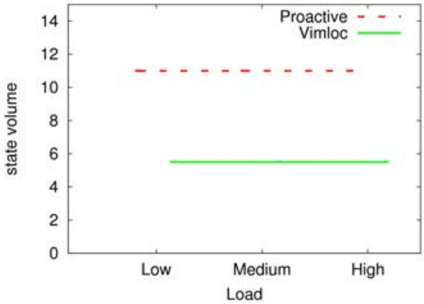


Fig. 10. State Volume vs. load. Location update (Proactive and VIMLOC)

On the other hand, the efficiency factor (EF) illustrates the efficiency of the LU/LR procedure of VIMLOC, the proactive, and the reactive scheme. And it is calculated as the ratio between 1) the number of one hop transmissions really involved in generating a successful location update/request-reply delivery and 2) the total number of one hop transmissions. Out of the total number of one-hop transmissions caused by flooding in the proactive scheme, just around 50 % are used to deliver successful location updates, as one may observe in Fig. 11a. Thus, half of the total one-hop transmissions are not useful in the LU procedure of the proactive scheme. On the other hand, the useful amount of one-hop transmissions for VIMLOC for the LU procedure is close to 100% for low loads and it decreases to 80% for high loads. But even for high loads, VIMLOC is 30% more efficient than the proactive scheme.

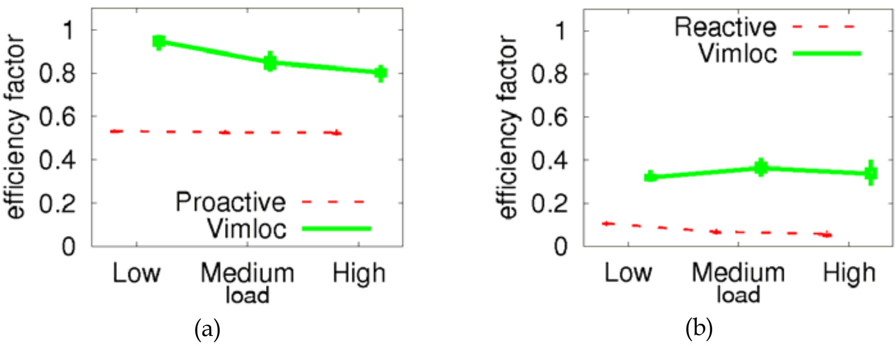


Fig. 11. Efficiency Factor (EF) vs. load. (a) Location update (Proactive and VIMLOC) and (b) Location request (Reactive and VIMLOC)

Fig. 11b shows that more than 90% of one-hop transmissions caused by flooding in the LR procedure of the reactive scheme are useless. In this sense, the VIMLOC EF (around 35%) is more than three times more efficient than the reactive scheme. And these values seem to be constant for all loads tested. The remaining 65%, corresponding to useless transmissions, is the price paid for the high success rate of the VIMLOC-LR procedure (Fig. 8b), since the location request is sent in parallel to two HGCs. In fact, if two successful replies arrive at the requesting node, just the first one received is taken into account for EF calculations. Thus, VIMLOC is much more efficient in terms of the number of useful one-hop transmissions than the proactive and the reactive schemes.

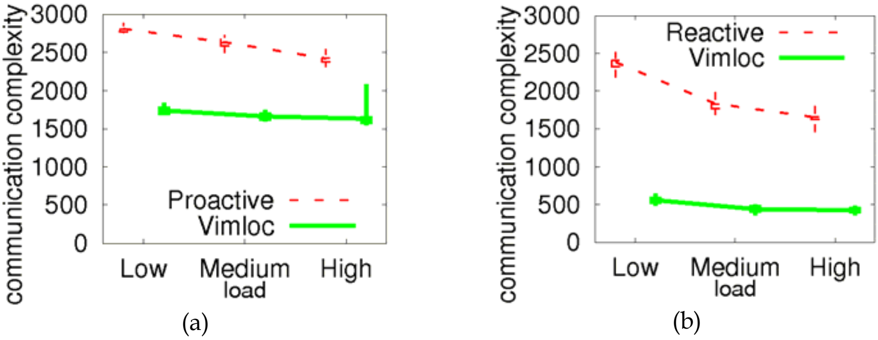


Fig. 12. Communication Complexity (CC) vs. load. (a) Location Update (Proactive and VIMLOC) (b) Location request-reply (Reactive and VIMLOC)

The communication complexity (CC) shows how many one-hop transmissions (on average) it takes to deliver one location update (Fig. 12a) or request-reply (Fig. 12b). As it is shown in Fig. 12a, VIMLOC generates 45% less one-hop transmissions than the proactive scheme for low loads in the LU procedure, whilst having twice higher accuracy. At the same time, the CC for VIMLOC is almost four times smaller than that of reactive schemes for the LR procedure, as illustrated in Fig. 12b. The difference between values slightly decreases with background load due to packet losses in the LU/LR procedure. But in any case, the advantage of VIMLOC is substantial.

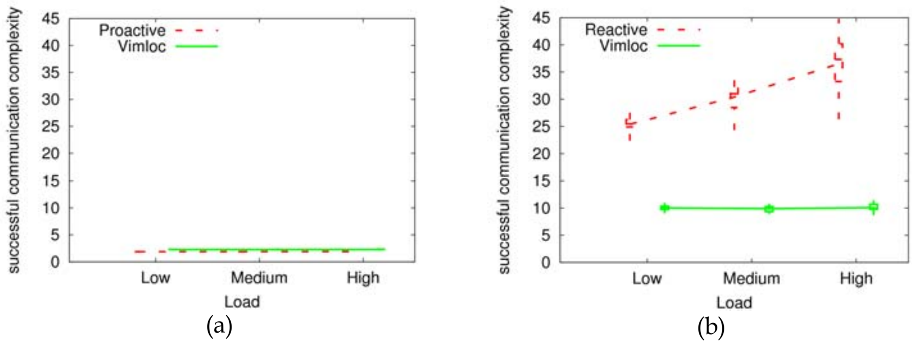


Fig. 13. Successful communication complexity (SCC) vs. load. (a) Location Update (Proactive and VIMLOC) (b) Location request-reply (Reactive and VIMLOC)

Successful communication complexity (SCC) shows how many one-hop transmissions (on average) it takes to deliver one successful location update (Fig. 13a) or request-reply (Fig.

13b). As it is shown in Fig. 13a, curves for VIMLOC and the proactive schemes almost coincide and do not change with load. However, as emphasized above, VIMLOC has twice higher accuracy. If levels of accuracy are the same, VIMLOC shows a reduction of around 30% in SCC, according to our measurements.

SCC for VIMLOC is approximately between one third and one fourth that of the reactive schemes for the loads tested (Fig. 13b). Furthermore, this difference is expected to substantially increase with the size of the network due to the better scaling properties of VIMLOC with respect to the reactive scheme.

7. Conclusion

This chapter presents VIMLOC, which stands for VIrtual home region Multi-hash LOCation management service. VIMLOC is a novel distributed location management scheme that works in conjunction with any position-based routing scheme. Up to our knowledge, this is the first location service specially designed for large-scale WMNs. In this sense, it exploits: 1) the stable and non-power-constrained nature of the backbone of a wireless mesh network, 2) the use of geographic positions, and 3) multiple hash functions. This work also seems to be the first implementation, experimental evaluation, and comparison over self-organized wireless networks (including ad hoc and mesh networks) of various location management schemes. More specifically, a comparison of the performance of VIMLOC with canonical proactive and reactive location mechanisms was carried out in a 12-node testbed. The experimental results show that VIMLOC outperforms these two latter approaches. First, the accuracy of location information is not compromised even though VIMLOC does not flood the network. Second, the state volume stored at each WMR does not grow linearly with the size of the network. Finally, VIMLOC provides robustness across a range of different workloads environments. Besides, it also has much better scaling properties, which mainly comes as a consequence of exploiting geographic information. Overall, this renders VIMLOC a promising solution for location management when wireless mesh networks are used to provide broadband wireless access.

Future work will mainly consist in assessing other parameters, such as reaction time, and to provide mechanisms to automatically tune the parameters already evaluated to the particular network scenario in which VIMLOC is deployed.

8. Acknowledgement

This work was supported in part by the Spanish Ministry of Science and Innovation under grant number TEC2008-06826 (ARTICO), by the Catalan Regional Government under grant 2009SGR-940, and by the European Commission project WIP under contract 27402.

9. References

Akyildiz, I.F. & Wang, X. (2005). A Survey on Wireless Mesh Networks, *IEEE Communications Magazine*, September 2005.

Basagni, S.; Chlamtac, I.; Syrotiuk, V.R. & Woodward, B.A. (1998). A Distance Routing Effect Algorithm for Mobility (DREAM). *Proceedings of the MobiCom´98*, pp. 76-84, Dallas, October, 1998.

Blazevic, L.; Buttyan, L.; Capkun, S.; Giordano, S.; Hubaux, J.-P. & Le Boudec, J.-Y. (2001). Self-organisation in Mobile Ad Hoc Networks: the approach of Terminodes*, IEEE Communications Magazine*, June 2001.

Blazevic, L.; Le Boudec, J-Y. & Giordano S. (2004). A location-based routing method for Mobile Ad-hoc Networks. *IEEE Transactions on mobile computing*, Vol. 3, No. 4, October-December, 2004.

Camp, T. (2006). Location services in Mobile Ad-hoc Networks (2006), *Handbook of algorithms for wireless networking and mobile computing*. 14: 319-341, 2006.

Cheng, H.; Cao, J. & Cheng, H.-H., (2007). GrLS: Group-based location services in Mobile Ad Hoc Network. *Proceedings of IEEE ICC-07*, 24-28 June, 2007.

Grid project (2003). Grid Ad hoc Networking Project. Information available at: http://www.pdos.lcs.mit.edu/grid

Derhab, A. & Badache, N. (2008). Balancing the tradeoffs between scalability and availability in mobile ad hoc networks with a flat hashing-based location service. *Ad Hoc Netw.* 6, 7, September, 2008.

EXTREME, (2010). EXTREME Testbed® of the CTTC. More information available at: http://www.cttc.cat/en/project/EXTREME.jsp

Haas, Z.J. & Liang, B. (1999). Ad-Hoc Mobility management with Uniform Quorum Systems, *IEEE/ACM Trans.* Net., Vol. 7, No.2, April 1999.

Hu, W.; Zou, S. & Cheng, S. (2007). Performance analysis of location management schemes in WiMAX Mesh Network, *Proceedings of Communications and Networking conference (CHINACOM'07)*, China, 22-24 Aug, 2007.

Hu, W.; Zou, S. & Cheng, S. (2009). A Novel Location Management Scheme for Wireless Mesh Networks, *Int. Journal of Distributed Sensor Networks,* Vol. 5, Is. 1, January, 2009.

Kasemann, M.; Fusler, H; Hartenstein, H & Mauve, M. (2002). A reactive location service for mobile ad hoc network. *Technical report TR-02-014,* Department of Science, University of Mannheim, November, 2002.

Karp, B. N. & Kung, H.T (2000). GPRS: Greedy Perimeter Stateless Routing for Wireless Networks, *Proceeding of the MobiCom-2000*, pp. 243-254, Boston, USA, August 2000.

Kies, W. Hierarchical location services for Mobile Ad-hoc Networks (2003). *Master's thesis*. Department of Computer Science, University of Mannheim, Germany, 2003.

Kies W.; Fusler, H.; Widmer, J. & Mauve M. (2004). Hierarchical location services for Mobile Ad-hoc Networks. *Mobile Computing and Communications Review*, Vol. 1, No. 2, 2004.

Ko, Y.B. & Vaidya, N.H. (2000). Location-Aided Routing (LAR) in Mobile Ad Hoc Network*, ACM/Baltzer WINET J.*, vol. 6, no. 2, pp. 307-21, 2000.

Kohler, E. et al. (1999). The Click Modular Router. *Operating Systems Review,* 34 (5), *Proceedings of the 17th Symposium on Operating Systems Principles*, pp. 217-231, December 1999.

Li et al (2005). A Scalable Location Service for Geographic Ad Hoc Routing. *Mechatronics and Automation, IEEE Int. Conf.,* Vol. 2 , pp. 831-836, August, 2005.

Mauve, M.; Widmer, J. & Hartenstein, H. (2001). A survey on position-based routing in mobile ad hoc networks, *IEEE Network,* 15: 30-39, December, 2001.

Portolés-Comeras, M.; Requena-Esteso, M.; Mangues-Bafalluy, J. & Cardenete, M., (2006). EXTREME: Combining the ease of management of multi-user experimental facilities and the flexibility of proof of concept testbeds, *Proc. TRIDENTCOM 2006*, Barcelona (Spain), March 1-3, 2006.

Rousseau, F.; Theoleyre, F.; Duda, A.; Krendzel, A.; Mangues-Bafalluy, J. & Requena-Esteso, M., (2008). Geo-mobility and Location Service in Spontaneous Wireless Mesh Networks, *Proceedings of ICT-Mobile Summit 2008,* Stockholm, Sweden, June 2008.

Saltzer, J. (1993). On the naming and binding of network destinations, *RFC 1498*, M.I.T. Laboratory for Computer Science, August, 1993.

Seada, K. & Helmy, A., (2006). Geographic services for wireless networks. *Handbook of algorithms for wireless networking and mobile computing*. 15: 343-364, 2006.

Stojmenovic, I. (1999). A scalable quorum based location update scheme for routing in ad hoc wireless networks. *Technical Report, TR-99-09,* University of Ottawa, September 1999.

Wong, V. W.-S. & V Leung, C. M., (2000). Location Management for Next-Generation Personal Communications Networks. *IEEE Network*, pp. 18-24, September/October 2000.

Wu, X. (2005). VPDS: Virtual Home Region based Distributed Position Service in Mobile Ad Hoc Networks. *Proceedings of 25-th IEEE Conf. on Distributed Computing Systems* (ICSCS-2005), 2005.