# Towards an Opportunistic and Location-Aware Service Provision in Disconnected Mobile Ad Hoc Networks

Salma Ben Sassi, Nicolas Le Sommer

## ▶ To cite this version:

HAL Id: hal-00498324

https://hal.archives-ouvertes.fr/hal-00498324

Submitted on 7 Jul 2010

# Towards an Opportunistic and Location-Aware Service Provision in Disconnected Mobile Ad Hoc Networks

Salma Ben Sassi and Nicolas Le Sommer

VALORIA Laboratory
Université Européenne de Bretagne, France
`{Salma.Ben-Sassi, Nicolas.Le-Sommer}@univ-ubs.fr`

**Summary.** Opportunistic networking has recently appeared as a promising method to support communication in disconnected mobile ad hoc networks. This new communication model relies on the "store, carry and forward" principle, and exploits ad hoc communication and device mobility in order to achieve a network-wide message dissemination. It allows nomadic people to communicate together without resorting to infrastructure-based networks and to have access to services offered by infostations even if they are not in the area covered by these devices. Nevertheless, to be efficient this model requires to guide the message propagation using contextual information, and especially location information.

In this paper, we present the middleware solution we have defined in order to support the provision of location-aware application services in disconnected mobile ad hoc networks using opportunistic communications.

## 1 Introduction

With the significant progress achieved since many years in the domains of hardware computing and wireless communication, smartphones, personal digital assistants and netbooks have become cheaper and more powerful, and are now widely spread. However, the main use of these devices lies so far in accessing the Internet via wireless access point (i.e., hotspots) in order to send and receive email or to surf the Web. Yet, all these devices are equipped with WiFi interfaces capable of ad hoc communication, and could be used by nomadic people to communicate together without necessarily resorting to an infrastructure-based network, and to access application services offered by fixed devices embedded in their physical environment, or even by mobile devices themselves.

Opportunistic networking [15, 11, 12, 2] has recently appeared as a relevant mean to support communication in mobile ad hoc networks (MANETs), which are in realistic conditions intrinsically disconnected due to the sparse and irregular distribution of the mobile devices and the fixed infostations that composed them. Like delay-tolerant networking [16, 17, 5], the opportunistic networking exploits ad hoc communication and device mobility to exchange data allowing hosts to communicate with each other even if a route connecting them does not exist all the time. Indeed, in opportunistic networks, messages are not simply routed in the network. While travelling from host to host in the network, they can be stored temporarily on certain hosts and be forwarded later when the circumstances are favourable. By thus exploiting ad hoc communication and

device mobility, messages can be propagated network-wide. However, application services are often relevant only in specific areas. Consequently, the messages inherent in the discovery, the advertisement and the invocation phases should be restricted to these areas instead of being propagated network-wide thus granting a sensible reduction of the network load.

In this paper, we present a flexible and extensible framework designed to support the development of location-aware application services, as well as the provision of these services using opportunistic communications. This framework defines several representations of the location concept, and several location determination methods. It also implements a location-aware and opportunistic service provision model. With this model, service providers are able to exhibit their location constraints, requirements and properties and to include this piece of information in their service advertisements. Likewise, service clients are aware of their real-time location, and able to perform service discovery, service selection and service invocation based on this location. For instance, a client application can discover and locate all service providers available in its neighbourhood and invoke the nearest one. This service application model is supported by a location-aware communication layer, where the opportunistic dissemination of application messages can be restricted to given geographical areas (i.e., the areas specified as relevant for the service discovery and invocation by the applications).

The remainder of this paper is structured as follows. Section 2 presents a framework for mobile host location and environment modelling. Section 3 describes how application services can use location information in the service discovery and invocation processes as well as in the message dissemination process. Section 4 presents the results we obtained by running our middleware platform on a mobile ad hoc network simulator. Section 5 compares our proposal with works targeting the same objectives as ours, and Section 6 provides a summary of our contribution and concludes by giving some perspectives.

## 2 Location framework

### 2.1 Location requirements for opportunistic service provision

The provision of application services using opportunistic communication introduces new issues that have not been identified so far with the single-hop synchronous communication model that is traditionally used for service provision. Indeed, with opportunistic communication service messages are likely to be propagated network-wide. Hence, application services can potentially be discovered and invoked in the whole network by mobile clients, even if these services must only be accessed in specific areas. Moreover, a network-wide dissemination of messages increases the global network load and reduces its scalability. A message propagation control using an expiration time and a number of hops is not sufficient for service provision, which requires a location-based propagation control. In order to underline both the issues inherent in the service provision using opportunistic communication and the needs of having location-aware services and middleware platforms, let us consider a campus populated with several service providers offering wireless access to printing services for students (see Figure 1).
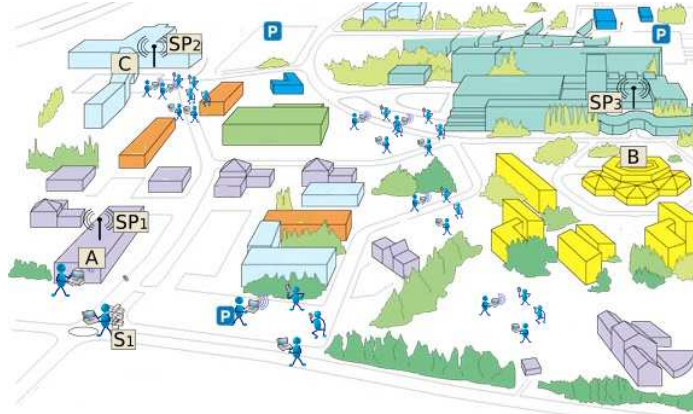
**Fig. 1.** Example of a disconnected mobile ad hoc network

For example, in order to propagate advertisement messages for printing services in the whole campus, service providers $SP_1$, $SP_2$ and $SP_3$ can decide to send their messages with a high number of hops. These messages will be widely disseminated in the campus by the students' devices, but also outside the campus. Indeed, without location constraints these service advertisement messages will be stored, carried and forwarded until their number of hops is equals to zero. For example, student $S_1$ who is going from building $A$ to building $B$ can propagate the service advertisement it received from provider $SP_1$ in $A$, in a location area close to building $B$. These services could thus be discovered –and potentially invoked– outside the campus by service clients, even if that should be prohibited. Furthermore, service clients having no information about their own location and about the location of the providers can achieve bad provider selections and invocations. For instance, after having received an advertisement from providers $SP_1$ and $SP_3$, student $S_1$ (or more precisely the middleware installed on the device used by $S_1$ after an action performed by the latter) can decide to select and invoke the service offered by $SP_1$, whereas he is now closer to $SP_3$ than $SP_1$, since no information allows to distinguish the services offered by $SP_1$ and $SP_3$.

Consequently, it could be convenient to have service providers able to give their location and to define in which location area their service advertisements must be forwarded, as well as to have location-aware service clients. This way, $SP_1$, $SP_2$ and $SP_3$ could specify that their service advertisements must only be propagated in the campus, and $S_1$ could compare its own location with the location of the providers, and thus select the nearest provider without ambiguity. Moreover, it is suitable that service providers can specify, in addition to the location area in which they can be discovered, the location area in which they can be invoked. Indeed, these two location areas can be different. For example in the scenario of Figure 1, providers $SP_1$, $SP_2$ and $SP_3$ should be discovered by the students in the whole campus, but should only be invoked by their respective surrounding students (i.e., by the students who are in the building where the providers are deployed).

Similarly, the service discovery requests and the service invocation requests sent by service clients must not be disseminated in the whole network, but only in specific location areas. For that, clients should be able to specify their own location and the location areas where their service messages can be propagated. Thus, only the providers located in the location area specified by the client can receive these requests and can return a response if they provide the required service. For example, if student $S_1$ has not received an advertisement from service provider $SP_2$ when reaching building $C$, student $S_1$ could initiate a service discovery by sending in the network a service discovery request with a geographic-restricted location area and its own location, and it should receive in return only an advertisement from $SP_2$. On the contrary, if he wants to discover all the services available in the campus, he should specify that its request must be propagated in the whole campus.

The key issues thus lie in the definition of the location concept and location determination methods, and in the design of service-oriented middleware platforms able to efficiently exploit location information in service discovery, service selection and service invocation, as well as in message routing. In the remainder of this section we present a location framework, and in the next section we present how this framework is used in a service management middleware layer and in middleware layer supporting opportunistic communication.

## 2.2 Location modelling

In many related works, the concept of location is often reduced to GPS coordinates –which are defined by a longitude and a latitude expressed in decimal degrees with respect to the World Geodetic System 1986 (WGS84), and an elevation above the WGS84 ellipsoid that is expressed in meters. Yet, in a pervasive environment similar to that considered in Figure 1, obtaining GPS coordinates is not always feasible, and this expression of location is not suitable in many cases. For instance, service providers $SP_1$, $SP_2$ and $SP_3$ deployed within buildings –where GPS signal cannot be received– should not define their location and their discovery and invocation location areas using GPS coordinates, but instead with symbolic names that can be expressed as a textual address information divided into fields (e.g. street, postal code, city, etc.). The term *location* thus should refer to the generic concept of a place that can be identified either by a symbolic name, a position expressed in a specific coordinates system, or an area. Our location framework, supports these different representations of the location concept, and defines methods making it possible 1) to estimate the distance between two locations when specifying intermediary locations or not, 2) to determinate if a location is included in another, and 3) to return the method used to obtain (or to estimate) this location. Four kinds of locations are defined in our framework: 1) waypoints, 2) landmarks, 3) location areas identified by a symbolic name, and 4) proximity-oriented location areas.

Waypoints are locations with no particular significance. They are only described by their position in a specific coordinate system. In contrast, Landmarks are locations characterised by both a position and a symbolic name defined as textual address information divided into fields (e.g. street, postal code, city, etc.). They are typically used to identify a place on a map.

Like landmarks, the location areas are identified by a symbolic name defined as textual address information and/or by coordinates and a geometric shape. They can be typically used by providers to specify where they are to define service discovery and invocation areas, as well as by the clients to define their own location, the area in which their service discovery requests can be propagated, and the area of the provider they try to invoke if necessary. A location area can be defined as an assembly of pre-existing areas. A primitive location area is mainly defined by a symbolic name and/or by coordinates and a geometric shape. A composite location area is defined so as to include other location areas, and to define how these areas are composed together using binding operators such as the union, the intersection, and difference (the default operator is the union). Architectural elements of physical environments, such as buildings, can be defined with location areas. They can be identified by an address and a position, and be described by the floors that compose it, which themselves can be structured as sets of several rooms. Thanks to such a composite modelling, a location information can be easily refined or extended by our framework when needed, thus offering several levels of precision to locate service clients and service providers.

Finally the proximity-based areas are characterised by a shape and geographic co-ordinates. They are mainly used by mobile clients in order to discover what services are in their neighbourhood, and by service providers to define in which area they can be accessed knowing their own position.

### 2.3 Location determination method

A location can be determined using several methods and technologies. The technologies used in location determination have been classified in [7] according to properties such as media, type of information, method, scale, cost, accuracy and point of computation. The media refers to the underlying technique (e.g., acoustic, video, electromagnetic). The type of information can be a physical position or some symbolic information; both can be absolute or relative. Symbolic location is defined to be a position relative to some known entity whose location may or may not be precisely known. In [7], three generic location determination methods are identified: proximity, triangulation and scene analysis or pattern recognition. These methods are also classified according to the point of computation, which can be server-based or client-based.

Our framework was designed with a similar classification in mind. All the objects modelling a location thus define a method returning an object that specifies the location type, the computation method and the technology used to obtain the location, and the point of computation. A location type can be either absolute, relative or symbolic. Landmarks and way points are absolute locations that are expressed with a position in a specific coordinate system (the default system is the WGS84), and which can be obtained using a GPS typically. Relative location is defined as a position estimation with respect to another location information. Proximity-oriented locations are examples of relative locations. Such location can be obtained using a proximity or triangulation computation methods on the basis of wireless signals for instance. Finally, symbolic locations are location areas identified by a symbolic name. In our current implementation only two kinds of technology are considered in the location determination: the GPS and the wireless communication interfaces. Wireless technologies are

characterised by their communication range, their type (e.g., 802.11, 802.15), and other information such as the signal level or the noise level. Moreover, four kinds of computation methods have been identified: the direct method, the pattern-matching-based method, the proximity-based method, and the triangulation method. In the current implementation of our framework only the last method is not implemented yet. The direct method is a simple method that is used to compute the coordinates obtained from a GPS receiver. The proximity-based method is used for wireless technologies and implements a linear attenuation model. The pattern-matching-based method relies on a hierarchical pattern matching mechanism on textual address information divided into fields (e.g. street, postal code, city, etc.). The location estimation can be computed either locally or remotely by another device in the network. Indeed, if a device cannot estimate its location itself, it can ask for a remote host (or a set of remote hosts) to compute this location according to the messages they receive from the mobile device and of the characteristics of the wireless technology.

### 2.4 Maps projection and environment modelling

The environment modelling and maps projection are key features for the location-aware service provision. Indeed, it is often necessary to build location databases, and to deploy them on mobile devices in order to associate address information with GPS coordinates for instance, or to indicate the position of people (or service providers) on maps. Coordinate projection and conversion functions have thus been implemented in our framework. Thanks to the landmarks and location areas, our framework provides general, yet easy to use, means for environment modelling. Obviously, our framework does not perform a 3D representation of the environment like CityGML [6], which is dedicated to 3D urban representation and that covers a very large field of places. Mobile devices do not need to be equipped with a location database natively, they can build themselves their own database (and therefore their own perception of the environment) by exploiting the location information included in service messages roaming in the network.

Figure 2 shows an application that has been developed using our framework, and that runs on a PDA. This application makes it possible, for instance for students, to model the campus, to see where they are, and to locate the application services available in the campus, such as the printing services.

## 3 Middleware support for location-aware application services

In this section, we first present how service messages are structured, and how they are handled by the service management and the communication middleware layers. Then we present how service clients and providers use the framework detailed in the previous section in order to express their location requirements, constraints and properties, and how location information is currently exploited in message routing.

### 3.1 Location information and service message structure

All the messages exchanged opportunistically by the devices are structured in two parts: a set of headers and a content (see Figure 3). Some of message headers are compulsory,

**Fig. 2.** A service location application developed using our framework and running on an IPAQ
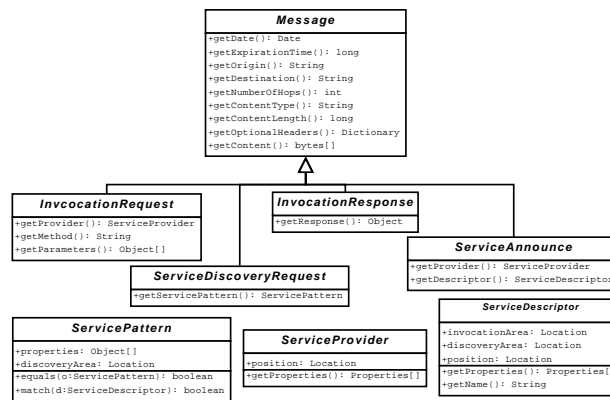


**Fig. 3.** UML methods of message structure.

such as *origin* and *destination* for routing purposes, *number of hops* to limit the propagation of messages in term of hops, and *expiration time* and *date* to define the temporal validity of messages. Other headers are optional, and are generally some complementary information specified by the application services so as to help in service selection and message routing. The content of messages depends on the type of messages. For example, the content of a service discovery request is a service pattern, whereas the content of a service advertisement is a service descriptor (see Figure 3). The service location properties and constraints mentioned in the previous section (e.g., service discovery location area, service invocation location area, and service provider location) are specified within the content of such messages. Some of these properties and constraints are also defined as optional headers by the service management middleware layer in order to specify how these messages should be processed by the opportunistic communication middleware layer.

### 3.2 Location management in the service management middleware layer

With our middleware platform each mobile host can maintain its own perception of the services available in the network thanks to a dedicated service register. This register is responsible for performing service discovery and service selection. The service discovery can be achieved either proactively or reactively. The reactive discovery consists in listening and analysing the unsolicited service advertisements sent by providers in the network. The proactive service discovery consists in sending service discovery requests in the network and in listening and analysing the service advertisements returned in response by providers.

```
//Environment modelling: cShape is the campus's shape
// bShape is the building's shape
// coordinates of the campus and of the building
GeographicCoordinates cCoord, bCoord;
cCoord = new GeographicCoordinates(47,64588,2.74516);
bCoord = new GeographicCoordinates(47.64504,2.74852);
CompositeLocationArea campus; Builing buildingA;
Floor floor0;
campus = new CompositeLocationArea ("UEB Campus", cShape, cCoord);
buildingA= new Building("A,UEB Campus",bShape,bCoord);
campus.add(buildingA);
floor0 = new Floor("Floor0", buildingA);

//Service descriptor
Location invocationArea= buildingA;
Location discoveryArea= campus;
Location position = floor0;
ServiceDescriptor desc=new ServiceDescriptor("printing service","fr.ubs.casa.
    PrintService",position, invocationArea, discoveryArea);

//Service registration
ServiceRegister.register(desc,this);
```

**Fig. 4.** Example of descriptor definition for a location-aware service.

Figure **??** shows how a printing service, comparable to those considered in the scenario presented in Section 2.1, can describe its location properties and constraints in its own service descriptor before registration. This descriptor will be then used by the service management layer in order to build a service advertisement for this service. The first block of instructions shows how to model the place where the service provider is located using a symbolic building (called A) with one floor (floor0), and whose address is "building A, UEB Campus. The second block of instructions defines a service descriptor including the interface exhibited by the service, its non-functional properties, and its locations properties and constraints. The last block of instructions registers the service in the local service register. In short, it describes a printing service with floor0 as invocation access area and the entire campus as discovery access area.

Local service clients can invoke the local service register in order to discover the remote services available in its neighbourhood. For example, to discover all printing services available in a range of 200 meters, the client must create a service pattern describing the service it requires and specifying its location constraint. Figure **??** shows

an example of the definition of such a service pattern using an object of type *Proxymit-yArea*. This service pattern will be included by the service register in a service discovery request if it has no information about a provider satisfying the required service interface and the location constraints expressed by the client. This discovery request will then be sent in the network in order to perform a proactive service discovery. The selection process aims at choosing the "best" provider to invoke among a set of providers, and at returning its reference to the client application. Meaning that we have to ensure that the selected provider is invokable (i.e, that the client is located within the invocation area when invoking) and that the chosen provider is the closest one to the client's current position and thus would be the fastest one to respond. The reference is then used by the client in order to build its service invocation request.

```
// Location of the client
Location hostLoc= LocationProvider.getCurrentLocation();

// Definition of the discovery area
Location discoveryArea=new ProximityArea(hostLoc.getCoordinates(), 200);

ServicePattern pattern = new ServicePattern("fr.ubs.casa.PrintService",
    discoveryArea);

// Service lookup
ServiceProvider refSrv=ServiceRegister.lookup(pattern);
```

**Fig. 5.** Example of service look-up.

In our framework, the asynchronous service invocation is achieved using notably the *InvocationRequest* and *InvocationResponse* messages and the *ServiceInvoker* and *ServiceResponseHandler* objects. To invoke asynchronously a remote service (or a set of remote services), a local client is expected to use the *asyncInvoke()* methods defined by the *ServiceInvoker* object. The first method asyncInvoke() takes as parameters a *InvocationResquest* object and a *ServiceReponseHandler* object. The handler object is used by the client to handle the responses it receives following an event-based programming approach. This handler takes as parameters a timeout –which should be equals to the expiration time specified in the request– and a number specifying how many responses must be handled. This handler is designed so as to receive responses from the network. When the timeout it received as parameter is triggered, the handler is expected to unregister itself from the *ServiceResponseListener*. A *ServiceResponseListener* is used to listen to a service response from the network and to dispatch them to objects of type *ServiceInvoker* and/or *ServiceResponseHandler*. The second method *asyncInvoke()* only takes as parameter an *InvocationRequest*. This method is designed to be running until a response is received from the network. When the timeout specified in the request is triggered, the method is expected to return a null value. Since temporal, spatial and contextual properties are intrinsically service-dependent, client services and service providers are responsible to specifying these properties themselves in the *ServiceRequest* and *ServiceResponse* objects respectively.

### 3.3 Exploitation of location information while routing

As underlined in the scenario presented in Section 2.1, with opportunistic communication, service messages may be forwarded outside the area where they are relevant. In order to cope with this issue and relieve the network from irrelevant messages, we have introduced, in addition to the compulsory headers presented in Figure 3, an optional header, called *restriction header,* to restrain messages geographically. The opportunistic communication middleware layer is designed to compare the current location of the mobile host with the value of this header before relaying a message. Thus, mobile hosts can forward a message if and only if they are within the area described in the *restriction header* of this message. This header is specified by the service management middleware layer on the basis of location information specified by application services in *service descriptor* and *service pattern* and it usually represents the *access area* or the *discovery area* of the targeted providers.

## 4 Evaluations

In order to evaluate our model, we have made a series of simulations using the Madhoc simulator[1], a metropolitan ad hoc network simulator that features the components required for both realistic and large-scale simulations, as well as the tools essential to an effective monitoring of the simulated applications. This simulator, which is written in Java, allow us to run our middleware platform on it.

In this section, we focus on a particular experiment whose objective was to measure the ability to satisfy the client service invocation efficiently using location information. For that, we compared our location-aware service discovery, selection and invocation model with a classical model that does not exploit the location properties exhibited by service providers and mobile devices. Moreover in the first case, we use an opportunistic communication protocol based on an epidemic model that implements both a hop-based and location-based message propagation control, and in the second case we use an opportunistic communication protocol relying on an epidemic model implementing only a hop-based message propagation control (messages are limited to 4-hops in our simulation). In the location-aware model, clients must await to be into the *invocation area* of a provider offering the service they require in order to start invoking this service, while in the other model, their service messages are forwarded by the other mobile hosts freely and thus client applications do not require to meet any conditions before invoking randomly one of the providers discovered (regardless of their positions or *access areas*).

The simulation environment we consider is an open area of about $1km^2$ containing 4 buildings and populated with a 100 devices equipped with Wi-Fi interfaces: 10 fixed infostations located in the buildings and 90 mobile hosts that evolve from a building to an other following a random way point mobility model and taking a 5 to 10 minutes time pause in each building they reach. Each mobile host moves with a speed varying between 0.5 and 2 m/s.

---

[1] http://agamemnon.uni.lu/~lhogie/madhoc/

70 mobile hosts are chosen periodically to act as relay nodes. Relay nodes are expected to forward immediately all service messages they receive while the other hosts are expected to forward their messages with a periodicity of 20 seconds. Each message has an expiration time of 4 minutes (when a message has expired, it is removed from the cache of messages and cannot be forwarded). Among the mobile hosts, only 60 hosts run client applications for the services offered by the infostations. When they have discovered a provider, the clients try to invoke them periodically (every 5 minutes) with a different request.

|  | Random invocation | Location-based invocation |
|---|---|---|
| Average waiting time before invocation (s) | 1,08s | 73,40s |
| Average delay for successful invocations (s) | 109,6s | 20,1s |
| Standard Deviation for the delay(s) | 134,0s | 51,9s |
| Average invocation success ratio (%) | 32,60% | 66,25% |

**Table 1.** Simulation results for the service discovery and invocation process

|  | Restricted dissemination | Unrestricted dissemination |
|---|---|---|
| Global network load (messges number) | 3537988 | 1948290 |
| Invocation traffic load (messges number) | 2345738 | 743502 |
| Inter-areas traffic rate | 40,11% | 2,31% |
| Intra-area traffic rate | 59,89% | 97,68% |

**Table 2.** Simulation results for the network load

In Tables 2 and 1 we summarise the simulation results we have obtained. The Table 2 shows that the network load has indeed been globally reduced by half and that the message traffic is controlled and restricted within the predefined discovery and invocation areas. Furthermore, the location-awareness has an impact on the service selection and service invocation since the success invocation rate increases from 32,60% for a random invocation to 66,25% for a location-based invocation (see Table 1). The average delay for a successful invocation has also been considerably reduced attaining an average value of 20s for location-aware applications. However, Table 1 also shows that the location-aware invocation model we propose has induced an extra waiting time of 72s due to the fact that a client application is not allowed to invoke a provider until it reaches its invocation area. A possible solution for this side effect would be to initiate the invocation before attaining the invocation area.

## 5 Related work

Many recent works have focused on problem of the opportunistic message forwarding in mobile ad-hoc networks, and some of them [2, 17, 14, 4] have considered contextual properties in order to define "smart" forwarding policies. However, despite the spreading use of positioning devices, only few location-based message forwarding protocols

have been proposed so far. GeOpps [12] is an example of such protocols. It implements a store, carry and forward mechanism dedicated to vehicular networks, and exploits location information available via navigation systems to select vehicles that are likely to carry a message towards its destination. However such a protocol is not suited for a human-based mobility model. Indeed, in contrast to vehicles, people do not necessarily follow predefined routes while they are moving, making it difficult to determine if or when they will reach a given destination. GPSR [10], Terminode routing [1], and GRA [8] are also location-based routing protocols, and they use only neighbour location information for forwarding data packets. Routing is done in a greedy way by forwarding the packet to the closest neighbour to the physical location of the destination. This local optimal choice is repeated at each intermediate node until the destination is reached or the message's time to live (TTL) expired.

As far as the exploitation of location information at the service level is concerned, few works have been achieved to the best of our knowledge. In [13], Meier et al. present a proximity-based service discovery protocol (PDS) for mobile ad hoc networks. Like our framework, PDS proposes a discovery approach of ad hoc services that exploits the fact that the relevance of such services is often limited to a specific geographical scope. Service providers are thus expected to define the areas (so-called proximities) in which their services are available. In PDS, clients register interest in specific services and are subsequently informed whenever they come into a "proximity" within which these services are available. In PDS, proximity areas are characterised by a given shape and coordinates and are included in discovery requests in order to select relevant service providers. However unlike our framework, PDS does not support several location models, and especially the address-based model that is best suited for indoor places.

Other works have also been done in the domains of the location management and location/environment modelling. These works have led to standards such as the location API for Java (JSR 179) [9], the OpenGIS Location Services (OpenLS) [3] and CityGML [6]. JSR 179 defines a framework to express a location with GPS coordinates expressed in the World Geodetic System 84. It also makes it possible to associate a location with an address-based information thanks to the concept of landmark, and defines a publish/subscribe paradigm that enables subscribers to be notified when a given location is reached. The OpenLS specifications detail the core services and abstract data types that comprise the GeoMobility Server. This server is an open location services platform for Web services defined by the Open Geospatial Consortium[2]. It notably defines functionalities to create routes and driving directions, to support map portrayal and to find points of interest using either a proximity-based model or an address-based model. It relies on a complete XML-based location modelling and querying. Nevertheless in comparison to our framework, this specification does not define a simple and flexible API that could be used by service providers to specify where they are, what is their service access areas, etc.

CityGML is a specification aiming to represent an urban environment in 3D. Different elements such as buildings or city furniture are identified using their geometrical, topological, semantic, and appearance properties. These elements can also be described using address information that conforms to the XML Schema definition of the Exten-

---

[2] http://www.opengeospatial.org/

sible Address Language (xAL) and to the rules for representing address information in CityGML. However, since CityGML is dedicated to 3D representation and thus covers a very large field of places that are unnecessary for our work such as water body or city furniture and defines lots of irrelevant properties such as topology, colour or material type.

## 6 Conclusion and future work

In this paper, we have presented a middleware platform for location-aware services. This platform implements an opportunistic communication protocol in order to support service discovery and invocation in disconnected mobile ad hoc networks. It also defines several kinds of location models and location determination methods. These models and methods are used by application services in order to specify their own location and their discovery and invocation areas, as well as by the middleware itself in order to discover what services are available in a given location area, and to select and invoke service providers according to their location properties.

The simulation results we obtained by running our middleware platform on a mobile ad hoc network simulator show that our middleware offers a more reliable and efficient service provision than a classical random invocation model, while reducing the global load of the network and thus improving the network scalability.

In the future, we plan to improve our middleware platform so as to opportunistically route messages toward targeted geographical locations, and not only limit their propagation to certain geographical locations as in our current implementation.

## References

1. Ljubica Blazevic, Jean-Yves Le Boudec, and Silvia Giordano. A Location-Based Routing Method for Mobile Ad Hoc Networks. *IEEE Transactions on Mobile Computing*, 3(4):1–15, October 2004.
2. Chiara Boldrini, Marco Conti, Iacopo Iacopini, and Andrea Passarella. Hibop: a history based routing protocol for opportunistic networks. In Marco Conti, editor, *Proc. IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks WoWMoM 2007*, pages 1–12, 2007.
3. Tom Bychowski, Jonathan Williams, Harry Niedzwiadek, Yaser Bishr, Gaillet Jean-Francois, Neil Crisp, Will Wilbrink, Mike Horhammer, Greg Roy, Serge Margoulies, Gil Fuchs, and Geoffery Hendrey. *OpenGIS Location Services (OpenLS): Core Services*. Open Geospatial Consortium, marwa mabrouk, esri edition, 9 September 2008.
4. Marco Conti, Franca Delmastro, and Andrea Passarella. Context-aware file sharing for opportunistic networks. In *Proc. IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems MASS 2007*, pages 1–3, oct 2007.

5. Kevin Fall. A Delay-Tolerant Network Architecture for Challenged Internets. In *Proceedings of ACM SIGCOMM03*, August 2003.
6. Gerhard Groger, Thomas. H Kolbe, Angela Czerwinski, and Claus Nagel. *OpenGIS City Geography Markup Language (CityGML) Encoding Standard*. Open Geospatial Consortium, August 2008.
7. Jeffrey Hightower and Gaetano Borriello. Location Systems for Ubiquitous Computing. *IEEE Computer*, 34:57–66, August 2001.
8. Rahul Jain, Anuj Puri, and Raja Sengupta. Geographical Routing Using Partial Information for Wireless Ad Hoc Networks. *IEEE Personal Communications*, 8:48–57, 2001.
9. Java Community Process, jsr-179-comments@jcp.org. *JSR 179 Location API for J2ME Specification*, nokia corporation edition, Febrary 2006.
10. Brad Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *6th International Conference on Mobile Computing and Networking (MobiCom00)*, pages 243–254. ACM Press, August 2000.
11. Jérémie Leguay, Anders Lindgren, James Scott, Timur Friedman, and Jon Crowcroft. Opportunistic content distribution in an urban setting. In *CHANTS '06: Proceedings of the 2006 SIGCOMM workshop on Challenged networks*, pages 205–212, Pisa, Italy, 2006. ACM Press.
12. Ilias Leontiadis and Cecilia Mascolo. GeOpps: Geographical Opportunistic Routing for Vehicular Networks. In *IEEE Workshop on Autonomic and Opportunistic Communications (Colocated with WOWMOM07)*, pages 1–6, Helsinki, Finland, jun 2007. IEEE Press.
13. René Meier, Vinny Cahill, Andronikos Nedos, and Siobhán Clarke. Proximity-Based Service Discovery in Mobile Ad Hoc Networks. In *Proceedings of the 5th IFIP INterntational Conference on Distributed Applications and Interoperable Systems (DAIS'05)*, volume 3543 of *LNCS*, Athens, Greece, June 2005. Springer.
14. Mirco Musolesi, Stephen Hailes, and Cecilia Mascolo. Adaptive Routing for Intermittently Connected Mobile Ad Hoc Networks. In *Proceedings of the IEEE 6th International Symposium on a World of Wireless, Mobile, and Multimedia Networks (WoWMoM 2005), Taormina, Italy*. IEEE press, June 2005.
15. Luciana Pelusi, Andrea Passarella, and Marco Conti. Opportunistic Networking: Data Forwarding in Disconnected Mobile Ad Hoc Networks. *IEEE Communications Magazine*, nov 2006.
16. Ritesh Shah and Norman C. Hutchinson. Delivering Messages in Disconnected Mobile Ad-Hoc Networks. In *Proceedings of ADHOC-NOW 2003*, Montreal, October 2003.
17. Guiseppe Sollazzo, Mirco Musolesi, and Cecilia Mascolo. TACO-DTN: A Time-Aware COntent-based dissemination system for Delay Tolerant Networks. In *MobiOpp 07: Proceedings of the 1st international MobiSys workshop on Mobile opportunistic networking*, pages 83–90, New York, NY, USA, jun 2007. ACM Press.