A Middleware Support for Location-Based Service Discovery and Invocation in Disconnected MANETs

Nicolas Le Sommer, Salma Ben Sassi, Frédéric Guidec, Yves Mahéo

A Middleware Support for Location-Based Service Discovery and Invocation in Disconnected MANETs

Nicolas Le Sommer — Salma Ben Sassi
Frédéric Guidec — Yves Mahéo

Laboratoire Valoria
Université de Bretagne Sud, France
{Nicolas.Le-Sommer, Salma.Ben-Sassi, Frederic.Guidec, Yves.Maheo}@univ-ubs.fr

ABSTRACT. *Disconnected MANETs show a changing topology and a fragmentation in distinct communication islands. In such networks, service discovery and invocation rely preferably on protocols that can support connectivity disruptions such as opportunistic protocols based on the store, carry, and forward principle. In this paper we present a middleware platform for geolocated services in disconnected MANETs. We detail in particular the location methods we used, the facilities for service discovery, selection, and invocation, and the undelying opportunistic communication protocol. Lastly, simulation results show the advantage of geolocation for service provision in such networks.*

RÉSUMÉ. *Les MANETs discontinus présentent une topologie changeante et fréquemment fragmentée en plusieurs îlots de communication distincts. Dans ces réseaux, la découverte et l'invocation de services sont préférablement assurées grâce à des protocoles de communication tolérant les ruptures de connexion tels que les protocoles de communication opportuniste reposant sur le principe du « store, carry and forward ». Dans cet article, nous présentons une plate-forme intergicielle de services géolocalisés dans les MANET discontinus. Nous détaillons plus particulièrement les méthodes de localisation employées, les fonctionnalités de découverte, de sélection et d'invocation des services, ainsi que le protocole de communication opportuniste sous-jacent. Enfin, des résultats de simulation montrent l'intérêt de la géolocalisation pour la prestation de services dans ce type de réseaux.*

KEYWORDS: *Service provision; Opportunistic networking; Disconnected MANETs*
MOTS-CLÉS : *Fourniture de service ; Communications opportunistes ; MANETs discontinus*

1. Introduction

The number and diversity of mobile devices equipped with wireless interfaces ad hoc communications have increased significantly during the last decade. Typical examples are smartphones, personal digital assistants and laptops, which are used every day by millions of users. Thanks to Wi-Fi or Bluetooth interfaces, such devices can spontaneously form a mobile ad hoc network (MANET), which makes it possible to share data or services without relying on any kind of pre-existing networking infrastructure.

The prospect of using mobile ad hoc networks to provide nomadic people with new kinds of services is appealing, especially for small companies or local authorities who cannot –or do not wish to– deliver these services through cellular phone networks. Yet, mobile ad hoc networks are usually highly dynamic, and often only partially connected due to the mobility and volatility of devices, of the short radio range of their wireless interfaces, and of the sparse and irregular distribution of these devices in their environment. The frequent and unpredictable changes observed in the topology of such networks yield new constraints regarding the way services must be designed and implemented in these networks, and regarding the middleware platforms supporting these services. These platforms must notably feature capabilities that are not required from traditional platforms, such as context-awareness (and especially location-awareness), an aptitude to cooperate in order to forward messages opportunistically, and a form of adaptive and autonomous behavior in order to select and invoke the best service providers at the most favorable time.

In mobile ad hoc networks, the synchronous communication model used traditionally to provide services is not suited. This model implies that two devices can communicate only if they are active simultaneously in the network, and if an end-to-end route exists between them. This route can be determined either proactively of reactively using dynamic routing protocols such as OLSR or AODV. On the other hand, with opportunistic communication protocols, which rely on ad hoc communications and which exploit the mobility of devices to achieve transmissions, mobile devices are not expected to acquire any information about the network topology and to maintain end-to-end routes between nodes in order to propagate messages network-wide. Messages are processed asynchronously and can be stored on some devices in order to be re-emitted later, thus making it possible for a message to reach its destination even if this one was not present in the network when the message was originally sent by the sender. Each device is liable to serve as a relay, and when processing a message it is responsible for choosing which of its neighbors, if any, this message should be forwarded to. To achieve this goal, opportunistic communication protocols usually implement heuristics in order to select the best device(s) for delivering a message, as well as heuristics that are meant to control the message propagation spatially and temporally. These heuristics are intended to reduce the global network load and to avoid an epidemic dissemination of messages in the network. In this paper we notably show that heuristics based on location criteria can prove to be efficient for opportunistic service discovery and invocation. Indeed, most of the services offered to nomadic users are relevant only in specific geographic areas. These services should thus define the areas where they can be discovered and invoked rather than in the whole network. Middleware platforms devoted to supporting such services should thus be able to exploit these location properties in order to forward messages only in the concerned areas, and to select autonomously the best service providers based on this location information.

In this paper we present the architecture of the middleware platform we designed in order to address the issues pertaining to service discovery and invocation in disconnected MANETs. This platform is mainly composed of two layers: a service management layer and an opportunistic communication layer. Cross-layering allows these two layers to interact and to share common information, such as the location properties associated with services. Several representations of the concept of location are supported by the platform, as well as several location methods. They are exploited by the service management layer to select service providers based on their location, and by the communication layer to determine the areas where messages should be forwarded.

The remainder of this paper is organized as follows. Section 2 presents a scenario that highlights the problems that must be solved, and justifies the solutions we propose. The general architecture of our platform is described in Section 3. In Section 4 we present the service management layer, focusing on the definition and the exploitation of location properties in service discovery, selection, and invocation. Section 5 shows the results we obtained while running our service platform on a mobile ad hoc network simulator. Related work is discussed in Section 6. Section 7, which concludes this paper, provides a summary of our contributions and proposes directions for future work.

2. Scenario and motivation

In this section we present a scenario that motivates our work, and that shows the challenges that must be addressed. In this scenario we consider a campus where infostations are deployed in order to offer services to students, as shown in Figure 1. We assume that the students are equipped with mobile devices such as smartphones, personal digital assistants, or laptops. We further assume that these devices can all communicate in ad hoc mode using IEEE 802.11 (Wi-Fi) wireless interfaces.

In the remainder of this section we focus on a specific set of services, namely printing services. Such services should be discoverable campus-wide, and be invokable only within the buildings where the infostations are located. In the following we discuss the advantage of exploiting location information for communication and service provision.

2.1. *Opportunistic communication with and without location*

Opportunistic communication protocols exploit the mobility of devices and their ability to store messages temporarily and re-emit them later in order to perform the message forwarding in a disconnected MANET. These protocols rely on the so-called "store, carry, and forward" principle. In order to control the propagation of messages in the network, each message usually includes a lifetime and a maximal number of hops. Yet such parameters can prove to be insufficient. For example, in the scenario we consider (Fig. 1), service providers SP_1, SP_2 and SP_3 should all assign a long lifetime and a high number of hops to the

Figure 1: Example of a disconnected MANET.

service advertisement messages they broadcast, so these messages can be disseminates in the whole campus. By doing so these advertisement messages will probably be propagated outside of the campus, which is not desirable. Hence, although student S_1 is not in the campus anymore, his/her device will keep relaying to other devices an advertisement message it has obtained from provider SP_1 (until this message has reached the end of its lifetime), and these devices will in turn contribute to relay this message even farther from SP_1.

If the middleware platform installed on student S_1 's mobile device was aware of its own location and could process the location information exhibited by services, it could refrain from relaying advertisement messages outside of the discovery area defined by a service provider. Messages would thus be circumscribed geographically. Furthermore, middleware platforms taking location and movement properties into account could select the best device(s) for transporting messages to their destination(s). The exploitation of such properties in message dissemination contributes to reduce the global network load, and thus to improves the scalability of the whole system.

2.2. *Service provision with and without location*

Location information must also be taken into account in service discovery, selection, and invocation. Actually, it is necessary that the platform provides service providers with means to be aware of their own location, so they can specify the geographical areas where they wish to be discovered and invoked. These areas can be either similar or different, as shown with the printing service we consider as an example. Location information must thus be included in the service descriptors broadcast by service providers. For example providers SP_1, SP_2 and SP_3 can all specify that their discovery area covers the whole campus, while their respective invocation area is the building where they are located. Thus this kind of information can be used by in both the service management layer and communication layer of the middleware platform installed on all devices. The middleware platform could compare the location properties included in the service descriptors received from the network with its own location in order to select the nearest service provider, which is presumably the most relevant to provide the required service.

A middleware platform that does not have such capabilities would not be able to differentiate service providers based on their location and could therefore fail to select the best providers, which would in turn impact on subsequent service invocations. For example, after receiving service advertisements from SP_1 and SP_3, student S_1 (or more precisely the middleware platform running on his/her device) can decide to select and invoke the service offered by SP_1, although he/she is far closer to SP_3 and could therefore probably expect a better (i.e. faster) service from this provider.

Similarly, the middleware platform should allow services to specify their own location, as well as the geographical area where they wish to look for another service (or several services) proactively. By including such parameters in service requests, these client services would only get advertisements from service providers located in the desired area.

Finally, by distinguishing a service invocation area from its discovery area, it is possible to anticipate the discovery and selection of service providers, deferring service invocation (or message transmission) until the client is close enough to the provider. The delay inherent to

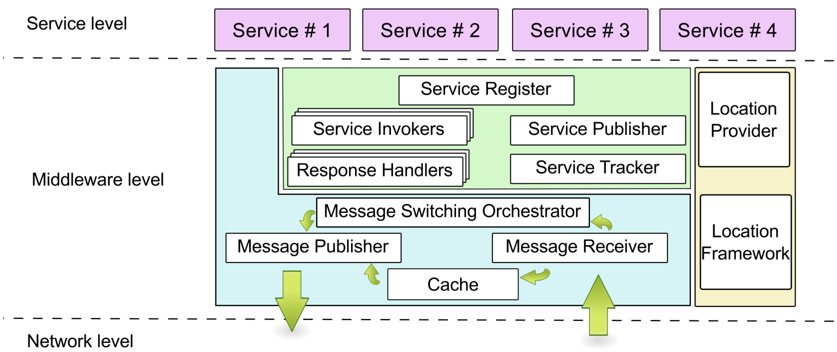

Figure 2: General architecture of the middleware platform.

opportunistic service invocation can thus be reduced, and the resulting network traffic limited to a given geographical area.

3. General architecture of the middleware platform

In this section, we present the general architecture of the service-oriented middleware platform we have designed in order to support the discovery and the invocation of location-aware services in disconnected mobile ad hoc networks. This platform is composed of a service management layer, of an opportunistic communication layer and of a location framework. The general architecture of the platform is depicted in Figure 2.

The different representations of the concept of location and the various location determination methods provided by our framework are used by both the application services deployed on our middleware platform and the layers of this one. In the remainder of this section, we present the main elements of each layer, and explain how they operate. We also motivate the design choices we made in order to allow these layers to share common information, and to act on one another by defining or updating this information.

3.1. *Location modeling and determination*

In many related work, the concept of location is often reduced to GPS coordinates, which are defined by a longitude and a latitude expressed in decimal degrees in the WGS84 geodetic system (World Geodetic System 1984), and by an elevation above the WGS84 reference ellipsoid, expressed in meters. In some environments, it is not always possible to obtain such coordinates, and these ones are not explicit enough in many cases. For instance in indoor places, where the GPS signal cannot be received, service providers should not express their location using GPS coordinates, but instead using a symbolic name. In our point of view, the term of location thus refers to a more general concept of place that can be identified by a symbolic name, a position expressed in a given coordinate system, or a geographic area defined either by a list of coordinates defined in a given coordinate system or by both a geometric shape and a geographic position.

3.1.1. *Modeling of the location concept*

The location framework we implemented in our middleware platform provides several representations of the concept of location. It makes it possible to define

– way points, identified by their geographic coordinates;

– geographic location, identified by a symbolic name (defined as an address composed of text fields: street, zip code, city, etc.) or/and coordinates;

– points of interests, which make it possible to classify locations identified by a symbolic name (e.g. hotels, museums);

– and composite geographical areas (i.e. areas that can be composed from elementary areas).

In our framework, a building can thus be represented by its address, its name and its geographic location, or in a more complex way as an assembly of elementary areas (floors, rooms). Geographic areas can be used by service providers in order to define the areas where they are relevant and where they can be discovered and invoked. Such areas can typically be specified by a position and a radius expressed in meters. We qualify such an area as a proximity area. Similarly, a service client can define such an area in a service discovery request in order to identify the providers present in its vicinity.

3.1.2. *Determination of the location*

Several location determination methods have been implemented in our framework. These methods can be used in order to estimate the location of both mobile devices and fixed infostations. Several methods aiming to process location information have also been implemented. For instance, some of these methods make it possible to compute the distance between two given locations either following a set of intermediary way points or not, or to estimate if a location is included in another one.

A location can be determined using various methods and technologies. Some of these methods and technologies have been classified in [HB01] according to properties such as the physical supports used to capture the location information (e.g., acoustic, video, electromagnetic), the location computing method, the type of information, the cost and point of computing. In [HB01], three location determination methods have been identified: the proximity, the triangulation and the analysis of scenes or of patterns. These methods can be computed either on the client or on the server side. The location information returned by these methods can be a geographic position or a symbolic information, both can be defined in an absolute manner or relatively to another location.

Our framework has been designed with such a classification in mind. Thus, for a given location, it is possible to know what method and technology have been used in order to obtain it, as well as to know if the location estimation has been computed locally or on a remote device. Four types of location determination methods have been identified in our framework: the direct-measure-based method, the identification of patterns based on symbolic names, the proximity-based method and the triangulation-based method (the latter is not yet included in the current implementation). The direct-measure-based method is a simple method that uses the coordinates returned by a GPS receiver. The proximity-based method is used for wireless technologies, and has been implemented using a linear signal attenuation model.

Finally, our framework provides methods to perform map projections. It is thus possible to model an environment and to store the resulting representation in a database, and then to deploy such databases on mobile devices in order to correlate symbolic information with GPS coordinates and to project the user's position (or providers' location) on a map.

3.2. *Service management layer*

The service management layer implemented in our middleware platform is mainly composed of a service registry, of an element responsible for processing the service advertisements received from the network, of an entity performing the advertisement of local services, and of several elements dedicated to the asynchronous invocation of remote services and to the processing of responses received in return (see Figure 2).

The local services are expected to register themselves with the local service registry at startup, and to unregister themselves from this registry when they are stopped. The service registry maintains both a list of local services and a list of remote services that have been discovered and that are considered as being still valid. In order to invoke a remote service (or another local service), a local service must invoke the local service registry in order to obtain a reference to the required service. The service registry selects the services according to their functional (i.e. of their interface), their non-functional and their location properties. If it does not have any information about a service compatible with the required one, the service registry initiates a service discovery process by sending in the network a discovery request including the description of the required service. Once the client has obtained a reference to the service it requires, it can call this service using the elements dedicated to the asynchronous invocation of remote services. In Section 4, we provide more details about the definition and the processing of location properties in the discovery, the selection and the invocation processes.

3.3. *Communication layer*

The communication layer allows mobile devices and infostations to forward messages opportunistically. This layer is mainly composed of four elements, namely a message receiver, a cache of messages, a message sender and a orchestrator that controls the emission of messages. This orchestrator can be configured with different message forwarding and processing policies. Two policies are currently implemented: a forwarding policy based on a simple epidemic model, and a policy supporting a geographically-constrained and location-driven message forwarding. These two policies are further discussed in Section 5.

When receiving a new message from the network, the message receiver is expected to store the message in the local cache and and to notify the orchestrator of the reception of this one. The message receiver can be configured in order to store in the local cache only the messages that match a predefined pattern, or, conversely, to reject messages that match a specific pattern, thus implementing a selective behavior regarding the message storage and the message forwarding. All the messages have a lifetime. Once they have expired, the messages are removed from the local cache. The size and the policy of management of the cache can be configured. LRU (Least Recently Used), MRU (Most Recently Used) and FIFO (First In First Out) cache algorithms are currently implemented.

The message forwarding and processing policy also enables to define whether the device is expected to act as a relay of messages or not. If so, the mobile will store the messages it receives in the local cache and will forward them to its neighbor nodes immediately. Moreover, according to the policy it receives in parameter, the orchestrator can periodically produce a list of messages that must be sent in the network. The list of messages is then sent by the message sender when the circumstances are favorable (i.e. when new neighbor nodes appear in the vicinity of the local node). This periodicity can obviously be configured.

Finally, our communication layer exhibits a publish/subscribe API. This programing paradigm can thus be used to develop application programs that do not need to be designed following the service-oriented programing model. Moreover, this publish/subscribe model makes the design of the service management layer easier. Indeed, the main elements of this layer can be developed as subscribers or publishers of service messages. For instance, the entity responsible for processing service advertisements received from the network can be implemented as a subscriber of service advertisements. Similarly, the element responsible for advertising local services can be designed as a producer of service advertisements.

3.4. *Implementation features*

Our middleware platform is designed as an extension of an OSGi platform. The opportunistic communication layer and the service management layer are implemented as OSGi services. These middleware-level services are expected to be used by application-level services in order to discover, to select and to invoke remote services. In addition to these services a service called *LocationProvider*, has been developed in order to obtain the location of the local device. The localization framework we developed is provided as a simple OSGi bundle, and is used by these three services.

The messages processed by the service management layer and by the communication layer, and exchanged opportunistically by mobiles devices, are structured in two parts: a header part and a content part (see Figure 3). These messages have compulsory headers, such as the address of the sender, the address of the recipient, the number of hops, the date of emission and a lifetime. Additional properties can also be specified by application services as optional headers in order to help in the selection, the discovery and the invocation of services, as well as in the forwarding of services messages. The content of the messages depends on their type. For instance, the content of a discovery request will include a pattern describing the main characteristics of the required service, whereas for a service advertisement, the message will include a descriptor of the considered service (see Figure 3). This descriptor will define location properties such as the position of the provider and the areas where the service can be discovered and invoked. According to the location properties specified by the service management layer and by the local service itself, the communication layer will define a specific

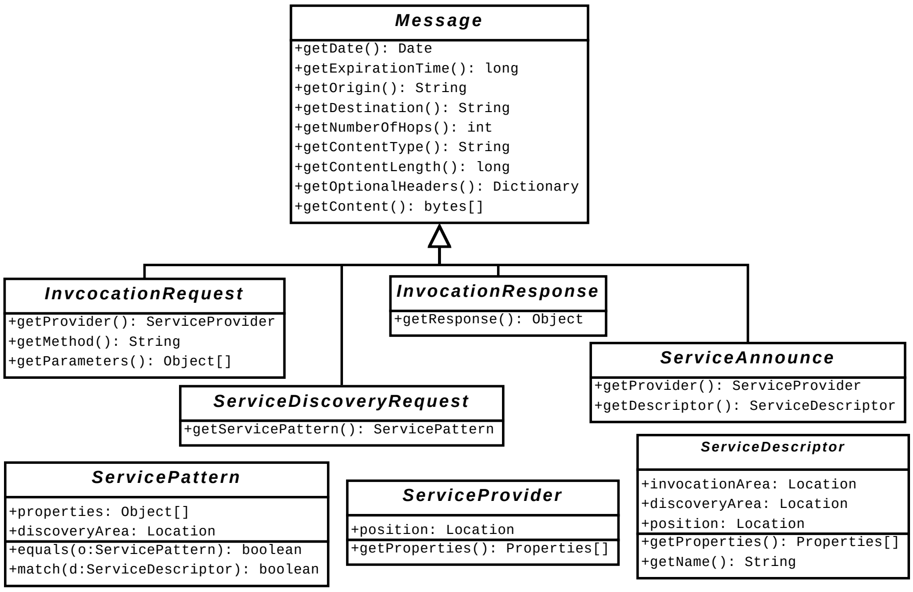

Figure 3: UML modeling of service messages.

header, call *restriction area*, that will be used by this layer to control geographically the propagation of the messages. Thus, the service advertisement generated by a service provider will be broadcast only in the area where the service is relevant.

By modifying the values included in the headers, in the service descriptors or in the service discovery patterns, the service management layer and the communication layer can act on one another. For instance based on the information it knows (e.g. the number of intermediary nodes between the client and the provider), the communication layer can update the default value of number of hops specified by the service management layer in order that the message does not persist in the network unnecessarily. This modification will be notified to the service management layer, enabling this one to use this new default value in its next invocation requests. Conversely, by defining or updating the value of messages' headers, the service management layer can influence the way in which the messages are processed by the communication layer.

In the current implementation, the messages and the elements that process them are designed using the visitor design pattern. Thus the processing algorithms are clearly separated from the hierarchy of messages. Moreover, messages are serialized as XML documents and compressed before being sent in the network. Figure 4 shows a service advertisement serialized as a XML document.

```
<message id="fb0097820f0b371" type="service-advertisement">
 <headers>
  <header name="origin" value="00:0F:1F:C5:2F:F5"/>
  <header name="destination" value="*"/>
  <header name="number-of-hops" value="5"/>
  <header name="date" value="Nov 29 16:09:47 CET 2008"/>
  <header name="lifetime" value="12:00:00"/>
  <header name="restriction-area"
          value="proximity, N 47.64588 W 2.74516,300m"/>
  ....
 </headers>
 <content>
  ...
 </content>
</message>
```

Figure 4: Example of a service advertisement serialized as a XML document.

4. Service management

In this section, we show how application services can express in our platform service discovery requests and service invocation requests, and how they can process the responses they receive in return.

4.1. *Exploitation of the location in the management of services*

As mentioned in Section 3, each mobile device is responsible for maintaining its own view of the services available locally and in the network thanks to its service registry. At startup a local application service is thus expected to register itself with the service registry by

```
//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("BatA,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("print service","fr.
    ubs.casa.PrintService","color=black,dpi=600x600",position,
    invocationArea, discoveryArea);

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

Figure 5: Example of the creation of a descriptor of a location-aware service.

specifying if possible its geographic discovery and invocation areas. Figure 5 shows how a print service, similar to those considered in the scenario presented in Section 2, can describe its location properties and constraints in its own descriptor. This descriptor will be used by the service management layer in order to build the service advertisement, which will be broadcast in the network if the local platform supports the reactive service discovery scheme. Indeed, two discovery schemes are considered: the proactive and the reactive discovery. The reactive discovery relies on the reception and on the processing of unsolicited advertisements sent periodically or sporadically by service providers in the network. The proactive discovery relies on the emission of discovery requests in the network and in the processing of the advertisements return in response by the providers able to deliver the required service. In our middleware platform, these two discovery schemes are implemented. It is thus possible to perform a reactive discovery, a proactive discovery or both.

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

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

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

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

Figure 6: Example of location-based service discovery request.

The first block of instructions in Figure 5 shows how a location can be defined. In the present case, the building whose symbolic name is *"BatA, UEB Campus"* is defined as an assembly of more elementary areas (`floor 0`). The second block of instructions defines the descriptor of the service. It includes the name of the Java interface of the service (the functional properties of the service), the non-functional properties of the service, as well as its location properties. The last block of instructions show the registration of the service with the local service registry. This print service can be discovered in the whole campus, and can be invoked only in floor 0 of building *BatA*.

In order to discover what print services are available within a radius of 200 meters, a client service can create a pattern similar to that described in Figure 6. The discovery area is defined using an object of type *ProximityArea*. This pattern will be included by the service registry in a discovery request if it has no information about a provider that matches the criteria expressed by this pattern. This request will then be sent in the network. The selection process implemented in the service registry makes it possible to choose the closest provider among a set of providers that satisfy the functional and non-functional properties of the required service. Once this selection step is achieved, a service reference to the provider is returned to the client. This reference can then be used at the invocation time. If no provider has been found, the *null* value is returned.

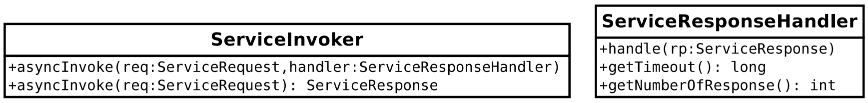

Figure 7: Elements supporting the asynchronous invocation of remote services.

In our middleware platform, the asynchronous invocation is performed using object of types *InvocationRequest*, *InvocationResponse*, *ServiceInvoker* and *ServiceResponseHandler* respectively. In order to invoke a remote service asynchronously, a service client must use the method *asyncInvoke()* defined by the *ServiceInvoker*. The first method takes in parameters an object of type *InvocationRequest* and an object of type *InvocationResponseHandler*. The latter is used by the client in order to process the responses, following an event-based programing model. This object takes in parameter a delay –which can be equal to the lifetime of the request– and the maximum number of expected requests. When the delay has expired, or when the number of messages is reached, this service response handler unregisters itself from the communication layer in order to not process new messages. The second *asyncInvoke()* method takes only one parameter of type *InvocationRequest*. This method is a blocking method. It returns a result when a response is received from the network, or returns a null value if the expected delivery delay has expired.

5. Evaluation results

This section presents simulation results obtained by executing our middleware platform on the MADHOC mobile network simulator [1]. This simulator is written in Java and allows the actual execution of the platform code and its evaluation in different scenarios. We focus here on an experiment whose objective is to assess the impact of the use of location information on the service discovery and invocation processes. To do so, we compare the results for four distinct configurations in a single scenario.

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

5.1. *Simulation parameters*

Like in the scenario presented in section 2, we consider a simulation environment of 10 km^2 that contains 4 buildings of different sizes, and in which move 80 pedestrians equipped with personal digital assistants endowed with Wi-Fi communication interfaces. These individuals move between buildings along predefined paths, and stay around 10 minutes outside the buildings. Moreover, 20 fixed infostations offering various services are dispatched in the four buildings. They periodically (every minute) advertise the services they provide. Besides, we consider that only 60 of the 80 people are interested in the provided services, and hence act as service clients, and that 70 of them accept to relay the messages they receive. After having discovered a provider that offers the services it requires, each of the 60 clients periodically issues (every 2 minutes), when possible, invocation requests. The total duration of the simulation is 3 hours.

As far as communication is concerned, each terminal sends the messages pertaining to its local services and the messages received from its neighbors every 15 seconds, if they have been validated. Each message has a lifetime of 4 minutes and a maximum number of 5 hops. Moreover, the nodes that play the role of relay re-send almost instantaneously the messages they receive.

During the simulations, we have used two types of opportunistic protocols, namely the protocol based on a simple epidemic model, and the protocol that implements a controlled epidemics according to location information. We studied the network load induced by these two different protocols, as well as the response delays and the failure rate, whether location is considered in the service selection process or not. To summarize, the four case studies are the following:

– Case 1: The provider selection does not take into account location information, and the communication protocol is based on a simple epidemic model.

– Case 2: The provider selection takes into account location information, and the communication protocol is based on a simple epidemic model.

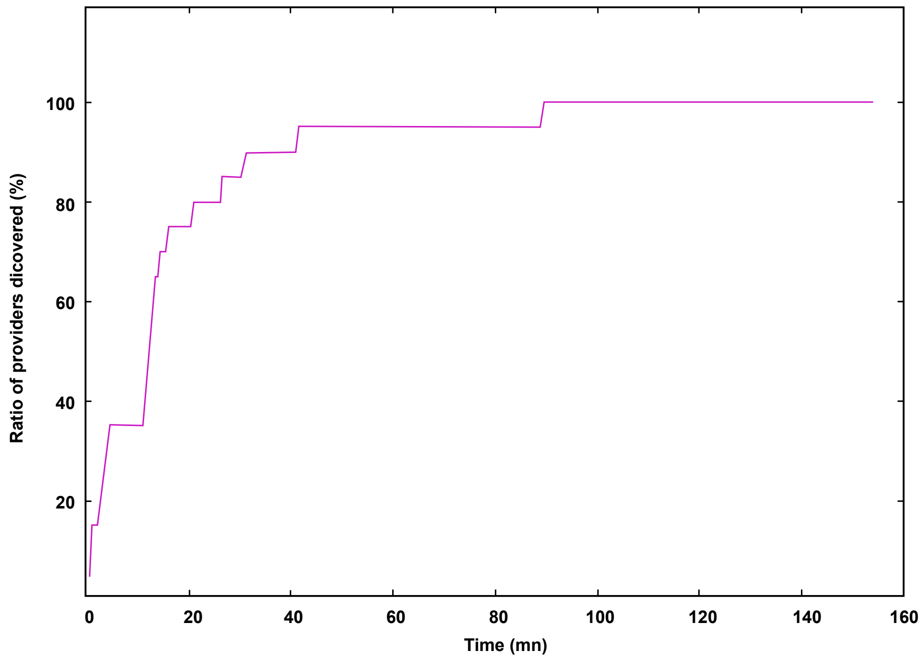

Figure 8: Evolution of the discovery process

– Case 3: The provider selection takes into account some location information, and the communication protocol is based on an epidemic model constrained by location information.

– Case 4: Similar to case 3, except that invocation is circumscribed to the invocation area specified by the provider.

Case studies 2 and 3 are intermediary cases introduced for the purpose of comparison, even if they are not useful in practice.

5.2. *Evaluation of service discovery and invocation*

Figure 8 depicts the evolution of the discovery process along time. After one hour and a half of simulation, a stationary state is reached: all clients have succeeded in discovering all the providers present in the environment.

	number of invocations sent	failure rate (%)	completion delay of successful invocations (seconds)	
			mean	std deviation
Case 1	4598	67.40	134.0	109.6
Case 2	4611	11.62	54.1	18.0
Case 3	4606	81.55	54.6	20.9
Case 4	2865	33.75	51.9	20.1

Table 1: Performance of service invocation

Table 1 shows the importance of service selection in function of the location. It gives performance figures for each of the four case studies, namely the number of invocation requests sent in the network, the failure rate (i.e. the percentage of invocation requests for which no response is received), and the completion delay for successful invocations (i.e. the duration between the time at which the request is sent and the time at which the response is received) .

For location-aware invocations (Cases 2, 3 and 4), one observe an average invocation delay of about 50 seconds, whereas a delay of more than 130 seconds is obtained for invocations that are performed without taking into account the location in the selection process (Case 1). Moreover, the standard deviation is reduced when location is considered. One can however notice a large variation in the failure rate of the three applications based on the location-oriented service selection process. This variation comes from the failures induced by the restriction of the message propagation to the geographical areas specified in the message headers. For example, for the lowest failure rate (11.62% for Case 2), messages are not geographically confined, in contrast with Case 3 in which invocation messages are sent by the service layer, but remain waiting in the communication layer until they are sent, when the device enters the specified invocation area. In Case 4, this problem is not present: the service layer waits for the device to be in the specified invocation area before actually invoking the service, which reduce the failure rate of invocations while sustaining the performances in terms of response time and network load (cf table 2). The delay between the time at which the service is discovered and the time at which the invocation

	Case 1	Case 2	Case 3	Case 4
Global network load (number of messages)	3537988	1331732	2940699	1948290
Invocation load (number of messages)	2346738	127632	2940699	742582
Inter-zone traffic (invocation ratio)	40.11%	2.31%	40.52%,	2.32%
Intra-zone traffic (invocation ratio)	59.89%	97.69%	59.48%	97.68%

Table 2: Network load distribution

request is actually sent becomes of course significant (we measured a delay betwen 1 and 20 minutes, depending on the mobility of the device).

5.3. *Evaluation of the communication layer*

Table 2 shows a reduction of the global network load as well as a concentration of the traffic in the buildings when location information is exploited in the communication layer. Indeed, When comparing the case studies based on a controlled epidemics (Case 3 and Case 4) with the case studies based on a plain epidemics (cases 1 and case 2), one notices that the global traffic is halved when messages pertaining to the invocation process are confined to the access zone of the provider. Moreover the intra-zone traffic passes from around 60% to 97%, whereas the traffic intra-zone drops from around 40% to less than 3%.

6. Related work

6.1. *Service discovery and invocation*

Most of the research work done on service provision in mobile ad hoc networks focused on the discovery process, and proposed discovery protocols tightly coupled to routing protocols so as to minimize the cost of communication and energy consumption. In these protocols, requests and responses are often integrated into routing messages by way of piggybacking. Hence, a device can discover a service while being informed of the route towards the provider of this service. Proposals described in [ZMN04], [VP05] and [KT03] are examples of such protocols. However, these protocols assume that communication between two mobile devices are possible only if they are present simultaneously in the network and that there exists an end-to-end path between them. This is not always the case in dynamic and fragmented mobile ad hoc networks.

In discovery protocols developed at the application level, discovery is performed on top of a routing protocol, and no assumption is made on this one. For example, Konark [HDVL03], DEAPspace [HHM$^+$00] and PDS [MCNC05, NSC05] have implemented middleware platforms for service discovery and service invocation that exploit such protocols. DeapSpace allows discovering and invoking services in the immediate neighborhood (ie at one hop). It implements for this a discovery process based on a proactive approach. The Konark middleware has similar objectives as DeapSpace but considers multi-hop discovery and invocation. As in DeapSpace, each host is in charge of maintaining its own view of the services available in the network, and it can behave as a service provider or as a client. Konark offers in addition a discovery model combining both proactive and reactive approaches. However Konark makes stronger hypotheses on the lower layer as it assumes that a route can be used between the client and the provider for the invocation. Neither DeapSpace nor Konark take location properties into account. Yet, such information can help with the transmission of the messages and the control of their propagation in the network, as well as with the selection of services.

To our knowledge, few work has been conducted on the exploitation of location information at the service level. In [MCNC05] is presented a proximity-oriented service discovery protocol (PDS) for mobile ad hoc networks. PDS proposes to exploit the fact that services are often relevant in a precise geographical area. Service providers must define areas (named proximities) within which their services are available. In PDS, a client interested in a service can be notified when it arrives in the surroundings of the provider of this service. Proximities are defined by geometric forms and coordinates. Location information is included in discovery requests in order to search for relevant services. Nevertheless, PDS does not presently support other representations of the notion of location. For example, it does not consider a location identified by a symbolic name. Moreover, as in all the above-mentioned work, service invocations are implemented in a synchronous way, with end-to-end transfers.

6.2. *Opportunistic networking*

Opportunistic networking recently appeared as a promising approach for enabling communication in distributed applications when synchronous end-to-end communication is not possible. Indeed, many research efforts aim at designing opportunistic, delay-tolerant or disruption-tolerant protocols in this perspective [Fal03, LR00, MMH05, PPC06, SH03]. Examples of such protocols take the form of an epidemic routing protocol [VB00], a disconnected transitive communication protocol [CM01], an asynchronous probabilistic protocol [LDS04], a context-aware adaptive protocol [MMH04, MHM05], a routing protocol exploiting history information [BCIP07], an opportunistic spatio-temporal dissemination system [LM07] or a time-aware content-based dissemination system [SMM07]. Although positioning systems are now widespread, few protocols take benefit from location information to date. GeOpps [LM07] is however an example. It implements the store-carry-and-forward principle in the context of vehicular networks. GeOpps leverages on the location information offered by navigation systems in order to select vehicles that are liable to transport messages towards their destination. However, the protocol is not suited to human mobility. Indeed, in contrast to vehicles, individuals do not necessarily follow predefined paths and move more randomly, making a planned communication more difficult to implement. GPSR [KK00], Terminode routing [BLBG04], and GRA [JPS01] are also routing protocols that take location into account. They use the location information collected only from their neighborhood to relay data. Routing is performed in a economic way by forwarding a packet to the neighbor that is the closest to the destination. The local optimal choice is repeated by each intermediary node until the packet has arrived at its destination or the packet lifetime (expressed in a number of hops) has been reached. This model is more adapted to human mobility.

7. Conclusion

This paper presented a middleware platform that supports the discovery and the invocation of location-aware services in disconnected mobile ad hoc networks. This platform includes a layer that implements location-aware services and an opportunistic communication layer that can exploit location information to circumscribe the dissemination of messages in the network. Moreover, the platform offers several models for the notion of location and several methods for obtaining and managing this information. Through simulation results, we showed the effectiveness of an approach that takes into account location not only in the communication process but also in service-level operations such as service selection. In the future, we plan to improve the communication layer by applying heuristics that would enable mobile nodes to select among their neighbors the best message carriers in function of their orientation and they speed of movement. Another possible improvement is to integrate gossiping techniques so that the mobile nodes broadcast only messages of interest to their neighbors, and that they can insert they own information requirements in these broadcast messages.

References

[BCIP07] Chiara Boldrini, Marco Conti, Iacopo Iacopini, and Andrea Passarella. HiBOp: a History Based Routing Protocol for Opportunistic Networks. In *8th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'07)*, pages 1–12, Helsinky, Finland, June 2007. IEEE CS.

[BLBG04] 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.

[CM01] Xiangchuan Chen and Amy L. Murphy. Enabling Disconnected Transitive Communication in Mobile Ad Hoc Networks. In *Workshop on Principles of Mobile Computing, 20th Symposium on*

Principles of Distributed Computing (PODC'01), pages 21–23, Newport, RI, USA, August 2001. ACM.

[Fal03] Kevin Fall. A Delay-Tolerant Network Architecture for Challenged Internets. In *Conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM'03)*, pages 12–34, Karlsruhe, Germany, August 2003. ACM.

[HB01] Jeffrey Hightower and Gaetano Borriello. Location Systems for Ubiquitous Computing. *IEEE Computer*, 34:57–66, August 2001.

[HDVL03] Sumi Helal, Nitin Desai, Varun Verma, and Choonhwa Lee. Konark: Service Discovery and Delivery Protocol for Ad-hoc Networks. In *3rd Conference on Wireless Communication Networks (WCNC'03)*, New Orleans, LA, USA, March 2003. IEEE CS.

[HHM+00] Reto Hermann, Dirk Husemann, Michael Moser, Michael Nidd, Christian Rohner, and Andreas Schade. DEAPSpace: Transient Ad-Hoc Networking of Pervasive Devices. In *1st International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'00)*, pages 133–134, Piscataway, NJ, USA, August 2000. ACM.

[JPS01] Rahul Jain, Anuj Puri, and Raja Sengupta. Geographical Routing Using Partial Information for Wireless Ad Hoc Networks. *IEEE Personal Communications*, 8:48–57, 2001.

[KK00] Brad Karp and H. T. Kung. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In *6th International Conference on Mobile Computing and Networking (MobiCom'00)*, pages 243–254, Boston, MA, USA, August 2000. ACM.

[KT03] Ulaş C. Kozat and Leandros Tassiulas. Network Layer Support for Service Discovery in Mobile Ad Hoc Networks. In *22nd Joint Conference of the IEEE Computer and Communications Societies (IEEE/INFOCOM-2003)*, volume 3, pages 1965–1975, San Francisco, CA, USA, April 2003. IEEE CS.

[LDS04] Anders Lindgren, Avri Doria, and Olov Schelen. Probabilistic Routing in Intermittently Connected Networks. In *1st International Workshop on Service Assurance with Partial and Intermittent Resources (SAPIR 2004)*, volume 3126 of *LNCS*, pages 239–254, Fortaleza, Brazil, August 2004. Springer.

[LM07] Ilias Leontiadis and Cecilia Mascolo. GeOpps: Geographical Opportunistic Routing for Vehicular Networks. In *Workshop on Autonomic and Opportunistic Communications, 8th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'07)*, pages 1–6, Helsinki, Finland, June 2007. IEEE CS.

[LR00] Qun Li and Daniela Rus. Sending Messages to Mobile Users in Disconnected Ad-hoc Wireless Networks. In *6th International Conference on Mobile Computing and Networking (MobiCom'00)*, pages 44–55, Boston, MA, USA, August 2000. ACM.

[MCNC05] René Meier, Vinny Cahill, Andronikos Nedos, and Siobhán Clarke. Proximity-Based Service Discovery in Mobile Ad Hoc Networks. In *5th IFIP Interntational Conference on Distributed Applications and Interoperable Systems (DAIS'05)*, volume 3543 of *LNCS*, Athens, Greece, June 2005. Springer.

[MHM05] Mirco Musolesi, Stephen Hailes, and Cecilia Mascolo. Adaptive Routing for Intermittently Connected Mobile Ad Hoc Networks. In *6th IEEE International Symposium on a World of Wireless, Mobile, and Multimedia Networks (WoWMoM 2005)*, Taormina, Italy, June 2005. IEEE CS.

[MMH04] Mirco Musolesi, Cecilia Mascolo, and Stephen Hailes. Adapting Asynchronous Messaging Middleware to Ad Hoc Networking. In *2nd International Workshop on Middleware for Pervasive and Ad Hoc Computing, 5th International Middleware Conference (Middleware 2004)*, pages 121–126, Toronto, Canada, October 2004. ACM.

[MMH05] Mirco Musolesi, Cecilia Mascolo, and Stephen Hailes. EMMA: Epidemic Messaging Middleware for Ad hoc networks. *Personal and Ubiquitous Computing*, 10(1):28–36, 2005.

[NSC05] Andronikos Nedos, Kulpreet Singh, and Siobhán Clarke. Service*: Distributed Service Advertisement for Multi-Service, Multi-Hop MANET Environments. In *7th IFIP International Conference on Mobile and Wireless Communication Networks (MWCN'05)*, Marrakech, Morocco, September 2005. King's College - University of London.

[PPC06] Luciana Pelusi, Andrea Passarella, and Marco Conti. Opportunistic Networking: Data Forwarding in Disconnected Mobile Ad

Hoc Networks. *IEEE Communications Magazine*, 44(11):134–141, November 2006.

[SH03] Ritesh Shah and Norman C. Hutchinson. Delivering Messages in Disconnected Mobile Ad-Hoc Networks. In *2nd International Conference on Ad-hoc, Mobile, and Wireless Networks (ADHOC-NOW'03)*, volume 2865 of *LNCS*, pages 72–83, Montreal, Canada, October 2003. Springer.

[SMM07] Guiseppe Sollazzo, Mirco Musolesi, and Cecilia Mascolo. TACO-DTN: A Time-Aware COntent-based dissemination system for Delay Tolerant Networks. In *1st international MobiSys workshop on Mobile opportunistic networking, 5th International Conference On Mobile Systems, Applications And Services (Mobisys'07)*, pages 83–90, San Juan, Puerto Rico, June 2007. ACM.

[VB00] Amin Vahdat and David Becker. Epidemic Routing for Partially Connected Ad Hoc Networks. Technical report, Duke University, April 2000.

[VP05] Christopher N. Ververidis and George C. Polyzos. Routing Layer Support for Service Discovery in Mobile Ad Hoc Networks. In *Pervasive Wireless Networking Workshop, 3rd International Conference on Pervasive Computing and Communications (PerCom 2005)*, pages 258–262, Kauai Island, Hawaii, USA, March 2005. IEEE CS.

[ZMN04] Feng Zhu, Matt Mutka, and Lionel Ni. PrudentExposure: A Private and User-centric Service Discovery Protocol. In *2nd International Conference on Pervasive Computing and Communications (PerCom'04)*, pages 329–338, Orlando, Florida, USA, March 2004. IEEE CS.

***Nicolas Le Sommer** received in 2003 is PhD in computer science from the Université de Bretagne-Sud (UBS), France. Since 2004, he is been an assistant professor in computer science at the UBS, and a member of the computer science laboratory of the UBS (Valoria Laboratory). His main research interests pertain on component-oriented and service-oriented middleware platforms for pervasive and ubiquitous computing.*

***Salma Ben Sassi** holds an engineering diploma from the ENSEIRB (France) in 2006. Since 2007, she is a PhD student in the computer science laboratory of the Université de Bretagne-*

Sud. Her PhD subject pertains on the investigation and the development of a location-aware service-oriented middleware platform for disconnected mobile ad hoc networks.

Yves Mahéo *holds a master's degree and a Ph.D. in Computer Science from the University of Rennes 1, France. Since 1997, he's been an associate professor with the Université de Bretagne-Sud and a member of the Valoria Computer Science laboratory. His main research interests are distributed middleware for dynamic networks, distributed components and services, software deployment, and delay-tolerant and opportunistic networking. He currently works on service-oriented middleware for opportunistic mobile ad hoc networks.*

Frédéric Guidec *received his PhD in Computer Science from the Université de Rennes I (France) in 1995. He then joined the École Polytechnique Fédérale de Lausanne (Switzerland), where he took part in several projects involving scientific numerical computation and distributed and parallel computing. He is now an assistant professor at the Université de Bretagne-Sud (France), where he conducts research activities in the general field of ambient computing. Presently his major interests are in mobile networking in challenged environments, such as partially or intermittently connected ad hoc networks.*