Service Provision in Disconnected Mobile Ad Hoc Networks

Nicolas Le Sommer

Service Provision in Disconnected Mobile Ad hoc Networks

Nicolas Le Sommer
Valoria Laboratory, University of South Brittany
Nicolas.Le-Sommer@univ-ubs.fr

Abstract

With the proliferation of mobiles devices equipped with communication interfaces such as IEEE 802.11 or IEEE 802.15, the mobile ad hoc networks have been recently arousing an increasing interest, and are considered from now as relevant supports for the ubiquitous and pervasive computing. The service provision in such networks arises as a critical issue for the success of these new computing paradigms. In this paper, we present both an approach and a middleware platform for the discovery and the invocation of services in disconnected, partially connected or intermittently connected mobile ad hoc networks. This approach relies on a content-based management of messages, on a flexible service invocation and on context-awareness.

1. Introduction

The generalisation of powerful portable devices capable of ad hoc communication are opening up new possibilities for service provision for nomadic people. Indeed beyond the application services considered so far for mobile ad hoc networks (i.e. services for emergency and/or military domains), new kinds of services relying on wireless opportunistic communications should be deployed and put at the civilians' disposal in the future.

Most of works investigating the discovery and the invocation of services in Mobile Ad hoc NETworks (MANETs) make strong assumptions regarding the communication between nodes by considering that this one is possible only if these nodes are both simultaneously active, and if a transmission route can be established between them whenever needed. This is actually equivalent to assuming that communication can only be made using a point-to-point synchronous communication paradigm, and that it can only occur in a fully connected network. By the way, a large number of works aim to define and to implement routing algorithms for MANETs in order to reuse the service provision mechanisms designed for wired-networks (e.g. Jini, UPnP, SLP). Yet when considering real mobile ad hoc networks, one can easily observe that their topology is extremely dynamic and frequently fragmented into small disconnected islands. The above-mentioned assumptions are therefore somewhat contradicted by the nature of these networks.

A number of recent research papers address specifically the problem of message delivery in disconnected, partially connected, or intermittently connected mobile ad hoc networks by focusing either on point-to-point routing in "delay-tolerant networks" [7, 13, 9] or on data flooding in such networks using epidemic [14, 10] or probabilistic [12] propagation schemes. Nevertheless, none of these works address specifically the service provision issues in such networks. The lack of solution for service discovery and delivery thus motivated us to propose a new service provision approach for these networks.

The remainder of this paper is structured as follows. In Section 2 we identify the functionalities suited for service provision in disconnected, partially connected, or intermittently connected mobile ad hoc networks using a scenario, and we present our approach. In Section 3 we detail the mechanisms we have implemented in a service-oriented middleware platform in order to evaluate our proposition. In Section 4, we compare our approach with related works, and in Section 5 we summarise our contribution and we conclude this paper.

2. Service provision in disconnected mobile ad hoc networks

2.1. Motivating scenario

So as to point up problems inherent in the discovery and the invocation of services in disconnected, partially connected or intermittently connected mobile ad hoc networks, let us consider the scenario illustrated in Figure 1. In this scenario, students having portable devices (e.g. PDA, laptops) equipped with IEEE 802.11 network interfaces (a.k.a WiFi) are supposed to use application services while moving in a campus. These application services –that can deliver information such as course and exam timetable, campus maps, extra-scholar activities– are provided by specific

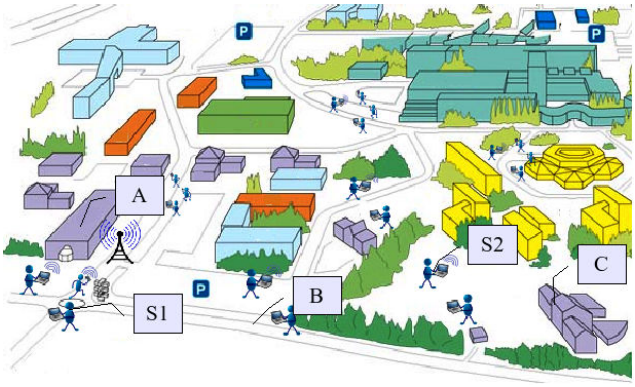

Figure 1. Example of a disconnected mobile ad hoc network.

devices deployed within the campus. The short communication range of WiFi interfaces and the mobility of devices generate frequent and unpredictable changes in the network topology –which is often partitioned into small disconnected islands.

In order to show the service provision process we want to provide, let us consider for the sake of illustration that a fixed device provided services such as those mentioned previously is deployed in point A (see Figure 1), and that at the time T, only three students are in the vicinity of this equipment. At this time only these students, including the student S_1 who is going to C, is able to discover and to invoke these services. Likewise, using traditional service provision mechanisms, the student S_2, who is going to A, will be able to discover and to invoke these services only when it will be in the vicinity of the device deployed in point A. Yet, it is suitable to allow the student S_2 to discover and to invoke these services as soon as possible even if no route can be established between it and the service provider. Therefore, at the time $T + \Delta t_1$, when the students S_1 and S_2 will be both at B, S_1 should spontaneously advertise S_2 of the availability of the considered services, or should reply to a discovery request sent by S_2 for such services. If at the time $T + \Delta t_2$, S_2 invokes a service thus discovered with a request similar to that addressed by S_1 at the time T, and if S_1 is always in the vicinity of S_2, S_1 should send to S_2 the response it obtained itself from the service at the time T, if this response is still valid obviously. To perform such tasks, the middleware platform deployed on the devices of the students S_1 and S_2must support asynchronous communications relying on the "store, carry and forward" communication principle, and must implement a content-based management of messages so as to correlate service discovery requests with service advertisements, as well as service requests with service responses. These mechanisms should also help at reducing the service provision latency and at deciding what messages can be stored locally. This middleware platform must also take into account spatial, temporal and contextual properties in order to improve service provision and to reduce the propagation of messages in the network. The spatial, temporal, and in a less measure, the contextual properties associated with massages are used in routing protocols[8, 3] in order to improve the message delivery in ad hoc networks. The properties are rarely accessible by the middleware level. Yet in these networks, these information are relevant for the elements performing the discovery and the invocation of services. Indeed, these properties should help them to choose the service providers, to estimate the probability to obtain a response from a remote service provider, to enable client and providers to specify the lifetime and the range of their requests and responses, etc.

2.2. Proposition for service provision

In the remainder of this section, we describe our proposition to address the above-mentioned issues by specifying the notion of service and service provider, as well as by giving a specification of the messages exchanged in the service provision process.

A service is traditionally described by both its interface and its non-functional properties. A service S_i implementing an interface I_i and having a set NP_i of n non-functional properties can thus be defined as

$$S_i =< I_i, NP_i = \{np_j\}/j \in [1..n] >$$

A same service, or a similar service, can be provided by several hosts in the network. A service S_j can be considered as being similar to a service S_i if and only if

$$I_i = I_j \, and \, \forall x \in NP_i, x =\in NP_j$$

So as to provides a flexible service invocation scheme, a context-based service selection, hosts must be characterised by a unique identifier, which can be for instance an IMEI number or a MAC address, their contextual properties (e.g. hardware characteristics, position), the services they provide and they have discovered, as well as with the groups of which they are members. The concept of group allows a host to send the same message to a set of remote service providers without invoking them successively. A host, whose the identifier is φ, can be described as

$$\begin{aligned} D_\varphi \quad = \quad & < \varphi, LS_\varphi = \{S_i\}/i \in [1..p], RS_\varphi = \{S_j\}/j \in [1..q] \\ & CP_\varphi = \{cp_k\}/k \in [1..r], G_\varphi = \{g_l\}/l \in [1..t] > \end{aligned}$$

where LS_φ is the set of services provided by the device, RS_φ the set of remote services considered as available in the network by the device, $CP_\varphi$the set of contextual properties exhibit by the device, $G_\varphi$the groups of which the device is member.

All the messages exchanged asynchronously by devices involved in a service provision process must include a unique identifier, temporal properties so as to decide if a message can be still considered as being valid, the identifiers of the sender and of the destination, both spatial and contextual information to control the propagation of messages in the network, the content of the message, as well as a set of properties describing this content in order to make the content-based management possible. A message can be a service discovery request, a service advertisement, a service request or a service response. These messages can be specified as follows:

$$M_i =< i, O_i, D_i, C_i, CD_i, T_i, L_i, H_i, CP_i >$$

The message M_i is identified by i. The origin, the destination and the content are specified by O_i, D_i and C_i respectively. The properties describing the content C_i is specified by the field CD_i . A message returned in response to another message also includes the identifier of this one. T_i is the time when the message is sent. L_i is the lifetime of the message. H_i is the spatial property expressed in number of hops. CP_i is the contextual properties exhibited by the sender O_i at the time T_i. A UML representation of messages is given in Figure 3.

Using the contextual, spatial and temporal properties specified in messages, each hosts is able to maintain its own view of the services available in the network, to estimate the probability to invoke a remote service provider successfully, to choose the better provider among several providers, as well as to select the most relevant invocation mode to obtain a response (invocation of a particular provider *vs* invocation of a group of providers).

3. A middleware platform for service provision in mobile ad hoc

This section presents the middleware platform we have designed in order to evaluate the proposition specified in the previous section. This platform, whose architecture is presented in Figure 2, is based on an OSGi gateway[1]. Application-level services deployed on this middleware platform are expected to use the two middleware-level services we have designed in order to discover and to invoke remote services asynchronously. The first service (the communication service) implements the "store, carry and forward" communication principle, and offers a publish-and-subscribe interface for the management of messages. The second middleware-level service provides functionalities to manage services (e.g. description, selection, discovery and invocation of services), and uses the publish-and-subscribe mechanisms defined by the first middleware-level service in order to perform the content-based management of messages. These services are further detail in the remainder of this section.

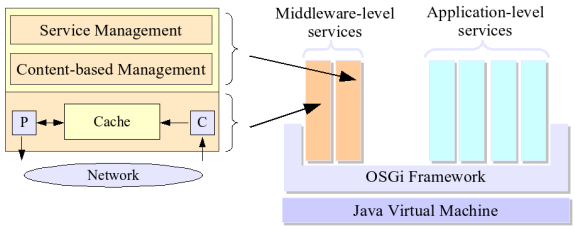

Figure 2. The middleware architecture.

3.1. Service Management

The core of our framework is based on the OSGi framework. In this framework, services are characterised by both the Java interface they implement and their non-functional properties. Since, this framework only supports the registration, the discovery and the invocation of local services, we have defined a set of extensions in order to support a service provision process addressing the issues pointed up in the previous section. These extensions, whose a partial UML representation is given in Figure 3 (only most relevant elements and methods are specified), make it possible to describe local and remote services, to characterise service providers, to define messages exchanged in the service provision process, to perform a proactive and/or reactive service discovery and an asynchronous service invocation.

Service discovery The proactive service discovery is performed using objects implementing the interfaces *ServiceRegister, ServicePublisher* and *ServiceTracker* respectively. The service register is responsible for maintaining locally a list of information about services and service providers. The service register is expected to be invoked by local services in order to discover remote services and to obtain references on service providers. When a service requested by a local service was not discovered yet, the service register creates itself a service discovery request (i.e. an object implementing the interface *ServiceDiscoveryRequest*) using the information specified by the local service (i.e. using the object implementing the interface *ServicePattern*), and sends it in the network. An object implementing the interface *ServicePattern* makes it possible to select services according to their description. When a service discovery request is received from the network by the service register of another node, this one checks if this request is matched by a descriptor of a local service, and if it is true, it asks to the service publisher to send in the network a service advertisement for this service. Such an advertisement

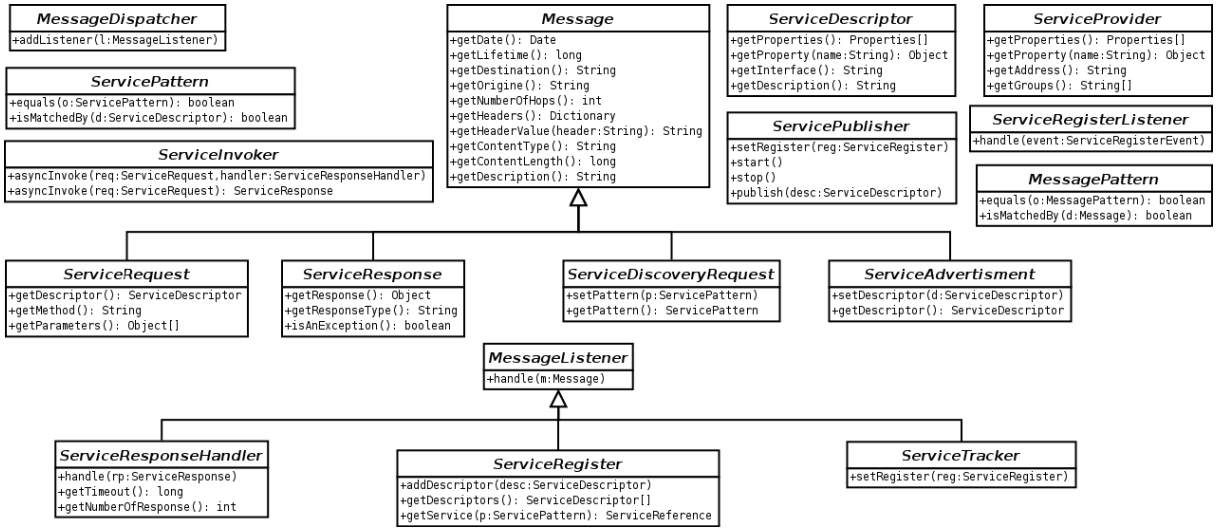

Figure 3. A partial UML representation of the service management API.

should be received by the service tracker of the service requester. This one is responsible for handling service advertisement from the network and to register them with the service register.

The reactive service discovery relies on the reception of unsolicited service advertisements and on their management by the service tracker. These advertisements can be sent periodically or sporadically by a service publisher. As shown in Figure 3, service advertisements include objects implementing the interface *ServiceDescritpor*. Service descriptors provide information about services (e.g. the name of the interface, the properties of the service).

When the lifetime of a service advertisement is not valid, then the service register removes this descriptor and notifies the local services using the considered service that this one is not reachable from now. Thus local services can adapt dynamically their behaviour in order to take these changes into account. Like the fields source and destination, the contextual properties exhibit by hosts are specified in messages as header fields. (see Figure 3).

Service invocation The asynchronous service invocation is achieved using notably the objects of type *ServiceRequest*, *ServiceResponse*, *ServiceInvoker* and *ServiceResponseHandler*. To invoke asynchronously a remote service (or a set of remote services), a local client is expected to use the methods *asyncInvoke()* defined by a *ServiceInvoker* object. The first method *asyncInvoke()* takes in parameters a *ServiceResquest* 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 in parameters a timeout –which should be equals to the lifetime specified in the request– and a number specifying how many responses must be handled. This handler uses the publish-and subscribe functionalities provided by the first middleware-level service in order to receive responses from the network. When the timeout it received in parameter is triggered, the handler is expected to unregister itself from this first middleware-level service.

Since temporal, spatial and contextual properties are intrinsically service-dependent, client services and service providers are responsible for specifying themselves these properties in the *ServiceRequest* and *ServiceResponse* objects respectively.

3.2. The communication service

With the service performing an epidemic dissemination of messages, any message sent in the network is maintained as long as possible in a local cache by as many devices as possible, so it can remain available for devices that could not receive it at the time it was sent originally. The underlying idea is that the dissemination of multiple copies of the same message may contribute to overcome the volatility of devices, while the mobility of these devices can itself help transport information between islands in a fragmented network or in distinct networks. Besides providing a caching system where messages can be maintained in mobile devices, our service also provides facilities for message advertisement, message discovery, and message transport between neighbouring devices. For example, a device can sporadically or periodically notify its neighbours about all or part of the messages stored in its cache. It can also look for specific messages in its neighbourhood, and either push messages toward or pull messages from its neighbours. In

the current implementation, message dissemination is performed by broadcasting UDP datagrams. We thus assume, for the time being, that each message is small enough to fit in a single datagram. In the future we plan to improve this communication service so as to address issues pertaining on message segmentation and reassembly.

Structure of messages As shown in the previous section, all messages used in the service provision process are reified as standard Java objects. These messages can be sent in the network as XML-formatted documents. A message is structured in two parts: the content, and a set of headers providing information useful in message handling. Figure 4 shows a typical XML-formatted message sent by a service provider in return of a service invocation.

```
<message id="fb0097820f0b371" type="service-response">
 <headers>
  <header name="origin" value="00:0F:1F:C5:2F:F5"/>
  <header name="destination" value="casa"/>
  <header name="number-of-hops" value="5"/>
  <header name="date" value="Nov 29 16:09:47 CET 2006"/>
  <header name="lifetime" value="12:00:00"/>
  <header name="md5sum-content"
          value="186c322f04579a179f1cce23b78e7a555"/>
  <header name="request-id" value="db0192810f0b21"/>
  <header name="md5sum-request-content"
          value="186c322f04579a179f1cce23b78e7a555"/>
  ....
 </headers>
 <content>
  ...
 </content>
</message>
```

Figure 4. An XML-formatted message exchanged by application-level services.

The value of attribute *id* in a message should be unique. This condition makes it possible for a device receiving a message from the network to verify if a copy of this message is already available in its local cache. The attribute *type* precises the type of the message. The type of a message can be *service-discovery-request*, *service-advertisement*, *service-request*, *service-response* and *cancel-message*. The message also specifies that it has been sent by the node whose IP address is "00:0F:1F:C5:2F:F5" (header *origin*)and that is is addressed to the nodes that are members of the group "casa" (header *destination*). As mentioned in the previous sections, destination and origin address can be either the address of a given service provider (i.e. of a given node) of a group , or the wildcard "*". The header *number-of-hops* plays approximately the same role as the field TTL *(Time-To-Live)* in IP packets. It ensures that a message will not be propagated eternally in the network. The header *date* specifies the date when the message was originally sent in the network, and the header *lifetime* specifies that this message should be considered as being valid for only twelve hours after this date. The header *md5sum-content,* which is the md5 sum of the content, is expected to help node in the content-based management. In both service response and service advertisement, the md5 sum of the request associated with the response or with the advertisement is also specified. With such an information a mobile host receiving a service invocation request pertaining to a service that it not provides itself, but for which it has already played the role of forwarder, can reply itself to this message by sending the response that it has stored previously in its local cache, whether this one is still valid obviously. The headers can also include contextual properties such as the GPS position of the sender.

4. Related work

Existing middleware solutions, such as DEAPspace [6], Konark [5], ReMMoC [4] and INDISS [2], have investigated the service provision issues in ad hoc networks. DEAPspace provides a support for the discovery and the delivery of services in wireless single-hop ad hoc networks. The middleware Konark targets environment and goal similar to DEAPspace, but considers multi-hop wireless ad hoc networks. Like in DEAPspace, each device in Konark maintains in a directory a vision of services available in the network, and acts both as client and provider of services. Konark makes some assumptions regarding the network by considering that a transmission route can be established between a client and a service provider whenever needed. Thus it does not support the service provision in intermittently connected mobile ad hoc networks. ReMMoC and INDISS have investigated the dynamic discovery and interaction in pervasive environments, particularly focusing on configuring and using the appropriate discovery protocol to find services in the current environment. These systems have so far only considered infrastructure-based wireless networks in which standardised discovery protocols such as Jini, Service Location Protocol, and Universal Plug and Play are used.

There is now emerging a suite of discovery protocols for mobile ad-hoc networks (MANETs), which have been developed to deal with the limitations of nodes in such environments (i.e. unpredictable network topology, variable number of participating nodes, different levels of dynamicity, and nodes with restricted knowledge of their neighbours). Scalable Service Discovery (SSD) [11] is an example of such protocols. In SSD, the discovery relies on a set of directories elected on the fly by mobiles nodes. Each directory is responsible for performing service discovery in a *n*-hops range. The global service discovery is achieved by a collaboration of the directories. This ap-

proach aims at reducing the service discovery traffic. SSD assumes IP-level connectivity using the underlying routing protocol (e.g. OLSR, AODV, ZRP).

5. Conclusion and future work

In this paper, we have presented both an approach and a framework for the service provision in partially connected, intermittently connected, or disconnected mobile ad hoc networks. This approach relies on a flexible addressing scheme, on a content-based management of messages and on asynchronous communications relying on the store-and-forward principle.

The middleware platform implemented using this framework will be evaluated in the future through simulation under different network conditions, varying the network size, node mobility level, the number of services and requests. Moreover, we plan to enable bridging MANETs with infrastructure-based networks since Intranet and/or Internet remains the primary source of service provisioning.

Acknowledgements This work is done in the project SARAH. This project is supported by the French ANR (Agence Nationale de la Recherche) in the framework of the 2005 ARA SSIA program.
`http://www-valoria.univ-ubs.fr/SARAH`

References

[1] O. Alliance. Osgi service platform, release 3, Mar. 2003. http://www.osgi.org/.

[2] Y.-D. Bromberg and V. Issarny. INDISS: Interoperable Discovery System for Networked Services. In *ACM/IFIP/USENIX 6th International Middleware Conference (Middleware'2005)*, Grenoble, France, Nov. 2005.

[3] L. Cheng. Service Advertisement and Discovery in Mobile Ad Hoc Networks. In *Workshop on Ad hoc Communications and Collaboration in Ubiquitous Computing Environments, in conjunction with the ACM 2002 Conference on Computer Supported Cooperative Work*, Nov. 2002.

[4] P. Grace, G. S. Blair, and S. Sam. ReMMoC: A Reflective Middleware to Support Mobile Client Interoperability. In *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, volume 2888 of *LNCS*, Catania, Sicily, Italy, Nov. 2003. Springer-Verlag.

[5] S. Helal, N. Desai, V. Verma, and C. Lee. Konark : Service Discovery and Delivery Protocol for Ad-hoc Networks. In *Third IEEE Conference on Wireless Communication Networks (WCNC)*, New Orleans, USA, Mar. 2003.

[6] R. Hermann, D. Husemann, M. Moser, M. Nidd, C. Rohner, and A. Schade. DEAPspace - Transient ad hoc networking of pervasive devices. *Computer Networks*, 35(4):411–428, Mar. 2001.

[7] Q. Li and D. Rus. Sending Messages to Mobile Users in Disconnected Ad-hoc Wireless Networks. In *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking*, pages 44–55, Boston, Aug. 2000. ACM Press.

[8] M. Musolesi, S. Hailes, and C. Mascolo. Adaptive Routing for Intermittently Connected Mobile Ad Hoc Networks. In *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM05)*, Taormina, Italy, June 2005.

[9] M. Musolesi, C. Mascolo, and S. Hailes. Adapting Asynchronous Messaging Middleware to Ad Hoc Networking. In *Proceedings of 2nd ACM International Workshop on Middleware for Pervasive and Ad Hoc Computing (MPAC 2004) in Middleware 2004 Companion*, pages 121–126, Toronto, Canada, Oct. 2004. ACM Press.

[10] M. Musolesi, C. Mascolo, and S. Hailes. EMMA: Epidemic Messaging Middleware for Ad hoc networks. *Personal and Ubiquitous Computing Journal*, 2005. To Appear.

[11] F. Sailhan and V. Issarny. Scalable Service Discovery for MANET. In *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom'2005)*, Hawai, USA, Mar. 2005. IEEE Press.

[12] Y. Sasson, D. Cavin, and A. Schiper. Probabilistic Broadcast for Flooding in Mobile Ad Hoc Networks. Technical Report IC/2002/54, Swiss Federal Institute of Technology (EPFL), 2002.

[13] R. Shah and N. C. Hutchinson. Delivering Messages in Disconnected Mobile Ad-Hoc Networks. In *Proceedings of ADHOC-NOW 2003*, Montreal, Oct. 2003.

[14] A. Vahdat and D. Becker. Epidemic Routing for Partially Connected Ad Hoc Networks. Technical report, Duke University, Apr. 2000.