A Proxy-based Model for Service Provision in Opportunistic Networks

Nicolas Le Sommer, Romeo Said, Yves Mahéo

A Proxy-based Model for Service Provision in Opportunistic Networks*

Nicolas Le Sommer, Romeo Said, Yves Mahéo
Valoria Laboratory
Université Européenne de Bretagne, France
{Nicolas.Le-Sommer|Romeo.Said|Yves.Maheo}@univ-ubs.fr

Abstract

This paper presents a proxy-based model for an enhanced provision of application services in opportunistic networks. This model makes it possible for service providers to specify what services can be delivered by proxies, and what mobile devices are expected to act as proxies. The proxy selection can be performed either in a distributed and autonomic manner or in a centralised manner. This paper also presents some results of the evaluations of this model we have performed using simulation techniques.

1 Introduction

Nowadays, handheld devices equipped with wireless interfaces supporting ad hoc communications are widespread (e.g., smartphones with a Bluetooth interface, PDA with a Wi-Fi interface). The prospect of using such a communication mode to provide nomadic people with application services appears attractive, especially for the institutes, the companies and the local authorities that cannot (or that do not want to) resort to licenced frequency bands (e.g., UMTS, GPRS) in order to provide end-users with a wide service access area. Opportunistic networking has recently emerged as a solution to support communication in large scale mobile ad hoc networks (MANETs), which are in realistic conditions fragmented in several distinct communication islands due the sparse and irregular distribution of the mobile devices and the fixed infostations that composed them. Opportunistic networking exploits ad hoc communication and device mobility to exchange data, and does not make any assumptions about the existence of a complete path between two nodes wishing to communicate. Furthermore, nodes are not supposed to have or acquire any knowledge about the network topology, which is instead necessary in the traditional MANET routing protocols –which aim at defining and maintaining end-to-end routes between the nodes wishing to communicate. Spatial and temporal paths are defined dynamically, while messages are forwarded to the neighbouring destination(s), and any possible host can opportunistically be used as next relay, provided it is likely to bring the message closer to the final destination(s). An additional delay is usually observed in message delivery, since messages are often buffered in the network waiting for a path towards the destination to be available. Although many application services can tolerate this additional delay, it can be detrimental to the quality of service offered to end-users.

In this paper, we propose a new proxy-based model providing an enhanced multi-hop service discovery and delivery in opportunistic networks. This model allows service providers to specify which services can be delivered by a proxy or a set of proxies, and to select which mobile devices are expected to act as proxies either implicitly using conditional rules or explicitly by specifying the addresses of the mobile hosts. The explicit selection of the proxies enables a centralised management of proxies, whereas the implicit selection of proxies allows a distributed and autonomic management of proxies. This model is based on the "store, carry and forward" principle and on a content-based management of service messages. This model has been implemented in a service-oriented middleware platform supporting opportunistic communications, and has been evaluated using simulation techniques.

The remainder of this paper is organised as follows. Section 1 first introduces the background of our model using a scenario, and then describes this model. Section 2 gives details about the implementation of this model in a service-oriented middleware platform. Section 3 presents some results we obtained by running our middleware on a network simulator. Section 4 presents related works. Finally, Section 5 provides a summary of our contribution and gives some perspectives.

*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

2 Opportunistic service provision using proxies

2.1 Background scenario

In opportunistic networks, messages are not simply routed in the network. While travelling from host to host in the network they can also be stored temporarily on certain hosts, and be forwarded later when the circumstances are favourable. Service messages (i.e., service discovery requests, service advertisements, service invocation requests, service responses) are thus stored locally in the caches of mobile devices, and should be returned on demand by these devices if needed. If the mobile hosts were able to process messages according to their content, they could correlate the service discovery requests with the service advertisements, and the service invocation requests with the service responses, and therefore could act as proxies for stateless application services, which are the most important type of services used in ubiquitous and pervasive computing environments (e.g., information services).

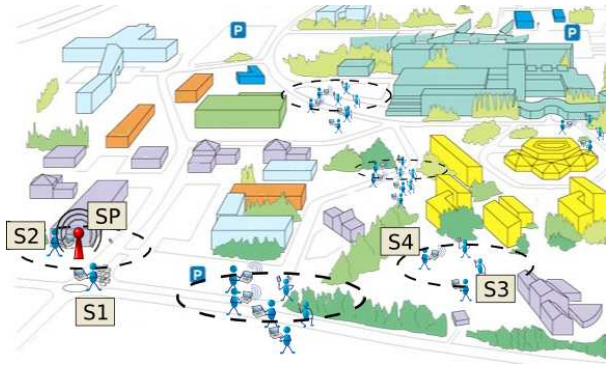

Figure 1: Example of a disconnected mobile ad hoc network.

For the sake of illustration, let us consider a disconnected mobile ad hoc network deployed in a campus. An illustration is shown in Figure 1. This network is composed of several mobile devices used by students and of a service provider *SP* offering application services to students such as course schedules, campus maps, etc. Moreover in this network, the mobile devices are expected to act as relays.

At time T, only the students who are in the same network island as *SP* can discover and invoke the services offered by *SP* (S_1 and S_2 can access *SP* directly because they are in its vicinity). Let us consider that at time $T+\Delta t$ S_2 leaves the network island of *SP* and joins the network island including S_3 and S_4, after having discovered and invoked one of the services offered by *SP*. Since S_2 has stored locally the service messages it has exchanged with *SP*, S_2 may either spontaneously send an advertisement for the services it has discovered in the network, or reply to a discovery request sent by students S_3 and S_4 for such services. Indeed, since S_2 can process service messages stored locally, it should be able to correlate the service discovery requests sent by S_3 and S_4 with the service discovery request it sent itself previously or with the service advertisement it received previously from *SP*. As far as the service invocation is concerned, when S_3 and S_4 try to invoke *SP* with the same requests as the ones S_2 has previously addressed to *SP*, S_2 is expected to reply with the responses it obtained from *SP* if these ones are still valid. Thus S_3 and S_4 could access the services offered by *SP* even if they are not in the same island as *SP*. In this scenario the mobile host S_2 thus acts as a proxy for the service provided by *SP*. In contrast to the mobile hosts that only implement the "store, carry and forward" principle, the mobile hosts that act as proxies can process messages according to their content, and can establish a correlation between the requests and the responses sent in the network by the mobile nodes.

In the remainder of this section, we present a taxonomy of services and we show how service providers can select proxies either explicitly or implicitly, how they can manage them, and how the service discovery and delivery are performed.

2.2 Overview of the model

	Parameter-dependent	*Parameter-independent*
Stateless services	Partially Proxyable	Proxyable
Stateful services	Non proxyable	Non proxyable
Time-dependent stateful services	Partially and temporary proxyable	Temporary proxyable

Table 1: Taxonomy of application services for a delivery using proxies.

A taxonomy of application services Two kinds of application services are likely to be deployed in a pervasive environment and to be provided using opportunistic communications: *(1)* the stateless application services, whose behaviour is recurrent and independent of the time and that can be cloned by some mobile hosts (i.e., the proxies)*; (2)* and the stateful application services whose behaviour cannot be cloned. Some of stateful applications can have a time-dependent behaviour, which can be reproduced temporarily by proxies. An example of such an application is an application service delivering weather forecast information. Moreover, the responses returned by some application services can depend on the parameters they received at invocation time. Hence, the behaviour of such application services cannot be fully reproduced by proxies. The application service taxonomy given in Table 1 is expected to be used by service providers in order to characterise the services they offer, and to help mobile hosts in deciding how to process the messages they receive for these services. In this taxon-

omy, five kinds of application services have been identified: 1) the services whose behaviour can be reproduced without any specific constraints (the proxyable services); 2) the services whose behaviour cannot be reproduced (the non-proxyable services); 3) the services whose behaviour depends on parameters and which therefore can only be partially cloned (the partially proxyable services); 4) the services whose behaviour is time-dependent and which can be reproduced during a given lease (the temporary proxyable services); 5) and the services whose behaviour is time-dependent and parameter-dependent (the partially and temporary services).

Proxy selection and management A proxy is expected to clone as faithfully as possible the behaviour of the service provider for which it acts as a proxy both from the viewpoint of the service discovery and invocation during a given lease –such a lease is expected to be defined by a service provider either explicitly or implicitly. In this context, it must be able to publish service advertisements spontaneously or to send service advertisements in response to service discovery requests. Similarly, it must be able to respond to the invocation requests it receives.

In order to decide which mobile devices are likely to act as proxies, the service provider can currently use information such as the frequency and the number of the invocations made by the mobile hosts. In the future, we plan to support other kinds of information such as the location or the mobility degree of the mobile clients. In our model, service providers can select and manage explicitly a collection of proxies for the services they offer. This management mode is well suited for mobile ad hoc networks where some devices are relatively static and reachable at any time by the service providers in order to be invalidated easily. For instance, a relatively static mobile host invoking a service frequently can be considered as a good candidate for playing the role of proxy by the provider that offers this service and that performs a centralised management of proxies. In the centralised proxy management mode, the service provider selects and invalidates proxies by sending control messages in the network explicitly. Such messages notably contain the name of the service, and also the period after which the proxy is expected to renew its lease with the provider. If the proxy does not renew its lease by sending a request, the provider considers that the proxy is not reachable and thus removes it from its list of proxies. On its side, the mobile host stops to act as a proxy. The provider can also invalidate this proxy by sending a control message. Figure 2 presents a grammar describing how the providers can define in their control messages the characteristics of the service, the proxy management rules and the service management policy in terms of service discovery and invocation. For instance, a provider can send to a particular mobile host a control message specifying that this host must act as a proxy for a proxyable service for a given lease period. Moreover it can also specify, that the mobile host is expected to support a proactive service discovery by sending periodically an advertisement, and that the responses it returns do not depend on the time.

---

proxy:= **service-status proxy-management discovery-management**
service-status:=*non-proxyable* | *proxyable* | *partially-proxyable*
| **temporary-proxyable-r** | **temp-part-proxyable-r**
temporary-proxyable-r:= *temporary-proxyable* **lease-period**
temp-part-proxyable-r:= *temp-part-proxyable* **lease-period**
proxy-management:=**centralised** | **distributed**
centralised := **address**(, **address**)* **lease-period**
address:= *byte**
lease-period:= *int*
distributed := **naming-rules**
naming-rules:=**number_of_requests method_coverage_rate**
number_of_requests := *int*
method_coverage_rate := *float*
discovery-management:= **proactive_discovery** | *reactive* | **both_discovery**
proactive_discovery:= *proactive* **period**
both_discovery:= *both* **period**
period := *int*

Figure 2: Grammar rules for implicit and explicit proxy naming and management.

In addition to a centralised selection and management of proxies, it could be convenient to have an autonomic and distributed selection and management of proxies. Indeed, in opportunistic networks composed only of mobile devices that appear and disappear dynamically, the centralised selection and management of proxies cannot be extremely difficult. The autonomic and distributed management scheme of proxies is not achieved explicitly by the service provider itself, but implicitly by specifying rules in the service advertisement messages These rules will be used by mobile hosts in order to decide if they can act as proxies or not. These rules pertain to the number of different invocations and on a method coverage rate (i.e. the number of the methods invoked by the clients divided by the number of methods offered by the service). Such rules are presented in Figure 2, and an example of the utilisation of these rules is given in the next section. In the future, such rules should also refer to other contextual properties such as geographical properties and mobile host characteristics (e.g. CPU, Memory). Indeed, some services can be relevant only in a given geographic area, or can be provided by mobile hosts having specific hardware characteristics.

As far as the distributed management of time-dependent application services by proxies is concerned, no additional properties are required. Indeed, in opportunistic communication, messages generally include spatial and temporal properties so as to control their propagation in the network, and to avoid that messages disseminate eternally in the network. Typically these messages include a number of hops

–which is equivalent to the TTL of IP– and a lifetime. Such information can be used by a proxy in order to decide if a message can be considered as still valid or not, and thus whether it can return this message in response to a request sent by a client. Proxies can also exploit this information in order to decide how many times they can act as providers for a given service. The capability of a mobile host to act as a proxy for a time-dependent application service thus decreases naturally in time. In the distributed mode, each mobile node acting as a proxy is expected to autonomously evaluate if it must stop to act as a proxy.

3 Implementation overview

3.1 Middleware architecture

In order to validate our model, we have developed a middleware in Java over an OSGi platform. This middleware is mainly structured in two layers, namely an opportunistic message switching layer and a service management layer (see Figure 3). Actually, both layers are implemented as OSGi services, which can be used by the local application services in order to discover and invoke the application services offered by remote devices opportunistically. The first layer implements a *Publish/Subscribe* API and is composed of four main elements: a message publisher, a message receiver, a cache of messages and a message switching orchestrator that can be configured using strategies. For example, strategies using periodic message emission, or strategies implementing gossip mechanisms can be considered.

The service management layer is composed of several elements collaborating in the service discovery and the service invocation process. Service discovery is mainly achieved by the service tracker, the service publisher and the service register. The service tracker is designed so as to be a subscriber to service advertisement messages and to register the discovered services in the service register. The service publisher is responsible for publishing the local services in the network using the communication layer.

The service invocation is mainly performed by the service invoker and the service response handler which are implemented so as to act respectively as a publisher of service invocation messages and as a subscriber to service response messages. These elements must be used by local services in order to invoke remote services. A further detailed description of these elements can be found in [7]. In our middleware platform, the proxy-based model we propose is implemented by a specific OSGi service behaving as a client of the communication layer in order to be notified about the service discovery requests and the service invocation requests for the services for which it must act as a proxy. It can also invoke the communication layer in order to look for the responses for the requests it received. If the responses are not available in the cache, or if they are not valid, the proxy forwards the request to the service provider in order to obtain a response. Each service provider can also run a proxy manager that is responsible for the centralised management of the proxies. The distributed and autonomous management of the proxies is, as for it, performed by the mobile hosts themselves. The conditional rules used for the proxy selection are specified by the local services when they register themselves in the service register.

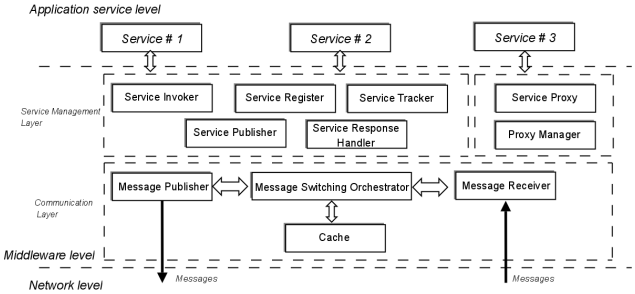

Figure 3: Middleware architecture

3.2 Content-based message management.

The local services considered are OSGi services. OSGi services are characterised by both the Java interface they implement and their non-functional properties. In the current implementation, the messages sent by these application services (or by the middleware platform) are serialised as XML documents structured in two main parts, namely the headers and the content of the messages. The headers provide information that can be used by both the communication layer and the service management layer. A header can contain for example the destination, the origin, the lifetime, the type of messages and a content description. In the current implementation of our platform the content-based management of messages is performed using a MD5 sum of the content. In the messages returned in response to a request, the MD5 sum of the content of the request is also specified in order to make it possible for proxies to quickly establish a correlation between a request and a response. In Figure 4, we present an example of a partial service advertisement returned in response to a service discovery request. The service advertisement messages also include additional information to support the distributed selection of proxies and the provision of services by these proxies. An example of the specification of such information is given in Figure 4. In this example, the service is considered as fully-proxyable. The mobile hosts that have a service method coverage rate higher than 0.9 and that have least 5 different invocation requests and responses are expected to act as a proxy, and to support a proactive and reactive service discovery with a period of service advertising of 30 seconds.

```
<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 2006"/>
  <header name="lifetime" value="12:00:00"/>
  <header name="md5sum-content"
           value="186c322f04579a179f1cce23b78e7a555"/>
  <header name="discovery-request-id" value="db019281f0b21"/>
  <header name="md5sum-discovery-request-content"
           value="186c322f04579a179f1cce23b78e7a555"/>
  <header name="proxy-selection-rules" value="5,0.9"/>
  <header name="proxy-service-status"
value="fully-proxyable,5,0.9"/>
  <header name="proxy-service-discovery" value="both,30"/>
  <header name="proxy-service-invocation"
value="time-independent"/>
  ....
 </headers>
 <content>
  ...
 </content>
</message>
```

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

4 Simulation results

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 discovery and invocation efficiently using proxies. For that, we compared our proxy-based service provision model with one implementing a 1-hop discovery and invocation model, and with one implementing a purely epidemic discovery and invocation model. In the 1-hop model, clients must be in the vicinity of a provider offering the service they require in order to discover and to invoke this service, while in the epidemic model, their service messages are forwarded by the other mobile hosts opportunistically (each mobile host forwards all the messages it receives in addition of its own messages).

The simulation environment we consider is an open area about 1km^2 populated with 100 mobile devices equipped with Wi-Fi interfaces that evolve following a random way point mobility model. Each mobile host moves at a average speed between 0.5 and 2 m/s. Among these 100 hosts, 8 act as service providers, 60 are potential clients of the services offered by these providers, 32 are neither clients nor providers of services. In our simulation scenario, a provider offers only one service, and the same service is provided by two different providers. The clients are interested in only one service, and they are expected to initially send a service discovery request in the network in order to discover the providers of the service they look for (the clients do not send their service discovery request at the same time). When they have discovered a provider, the clients invoke this provider periodically (every 5 minutes) with a different request. The clients of the same service are designed so as to send the same invocation requests. Each host sends its messages with a periodicity of 20 seconds. Each message has a lifetime of 10 minutes. The services considered in this scenario are stateless application services. Each mobile host having relayed (and put in cache) at least a discovery request, an advertisement, and five invocation requests and responses is expected to start acting as a proxy for this service. The proxy lease is automatically managed by the mobile hosts themselves according to the lifetime of the messages they handle.

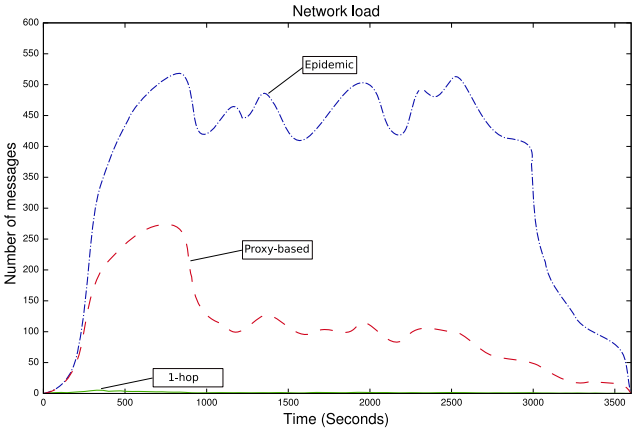

Figure 5: Network load.

In Figure 5, we present the network load for the simulations we have made, and in Table 2 we summarize the simulation results we have obtained. In contrast to the epidemic and the proxy-based provision models, the 1-hop service provision model has the advantage of offering a low network load, but to the detriment of the service provision quality since only 0.15% of the service invocation requests have been satisfied during the simulation (i.e., 1 hour). As shown in Figure 5 and Table 2, the proxy-based model has approximately the same invocation success ratio as the epidemic model, but with a better response delay and a lower network load.

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

	1-hop	epidemic	proxy-based
Number of clients discovery having discovered a provider	18/60	43/60	41/60
Average delay of service discovery (sec.)	198	133	88
Standard Deviation for delay of success invocation (sec.)	113	98	62
Average delay of success invocation(sec.)	202	122	78
Standard Deviation for delay of success invocation (sec.)	167	94	53
Average invocation success ratio	0.15	0,45	0,43

Table 2: Simulation results for service discovery and invocation.

5 Related work

Opportunistic networking has recently appeared as a promising approach to allow applications to communicate when synchronous communications based on end-to-end connectivity are not possible. Many ongoing works aim at defining communication supports for opportunistic and/or disruption/delay-tolerant networking [3, 9, 14, 16, 17]. The epidemic routing protocol [19], the disconnected transitive communication protocol [2], the asynchronous probabilistic protocol [10], the context-aware adaptive protocol [13, 12], the history-based routing protocol [1], the Opportunistic Spatio-Temporal Dissemination System [8] and the Time-Aware Content-based dissemination system [18] are some examples of protocols for opportunistic and/or disruption/delay-tolerant networking. Nevertheless, in contrast to our work such protocols do not offer content-based message management functionalities. Such functionalities are necessary in order to implement a proxy-based model for the service provision.

Service discovery protocols designed specifically for such mobile ad hoc networks can be divided into two categories: network layer-based and application layer-based service discovery protocols. In the service discovery protocols coupled with routing layer, service query and response messages are often piggybacked on to ad hoc routing messages. In this way, a host requesting a service in addition to discovering the service will also be informed of the route to the service provider at the same time. Examples of such protocols are [21], [20] and [6]. Protocols that implement service discovery at routing layer generally reduce the communication and energy consumption overheads. However, the proposals made along this line rely on the assumption that communication between two devices in a network is possible only if these devices are both simultaneously active, and if a transmission route can be established between them whenever needed using reactive or proactive techniques.

In service discovery protocols based on the application layer, the service discovery functionality is supported above the routing layer and usually do not make any assumption regarding this one, and thus seems to be well suited for context-aware and opportunistic networking. Most of these service discovery protocols are implemented in service-oriented middleware platforms. Konark [4], and DEAPspace [5] and PDS [11, 15] have investigated the service discovery and delivery issues in ad hoc networks with middleware platforms. DEAPspace provides a support for the discovery and the delivery of services in wireless single-hop ad hoc networks. DEAPspace implements a push based service discovery paradigm. The Konark middleware has objectives similar to those of 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 a client and a provider of services. It supports both push and pull discovery style, and it is limited to one-hop. Moreover, 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.

PDS, hides the complexity of the underlying communication system by providing an asynchronous advertise/discover programming model to the application layer that is also independent of the structure of service representation. PDS can therefore work with multiple service representations. Moreover, it 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. Clients register their interest in specific services and are subsequently informed whenever they come into a ”proximity” within which these services are available [15].

6 Conclusion

In this paper, we have presented a proxy-based service provision model for opportunistic networking. This model relies on a taxonomy of application services and a content-based management of services messages, and allow service providers to specify which mobile hosts should acts as proxies and which services can be proxified. Moreover it supports a centralised control or alternatively a distributed and autonomic management of proxies. This model was evaluated using simulations that show that it is a relevant model for the provision of services. In the future, we plan to improve this model by taking into account geographical properties. In this way service providers could specify in which area the services they offer are considered as relevant, as well as to control the spatial distribution of the proxies. We also would like to improve this model by supporting a proxy collaboration in the service provision through an overlay of proxies.

References

[1] C. Boldrini, M. Conti, J. Jacopini, and A. Passarella. Hibop: a history based routing protocol for opportunistic networks. In M. Conti, editor, *Proc. IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks WoWMoM 2007*, pages 1–12, 2007.

[2] X. Chen and A. L. Murphy. Enabling disconnected transitive communication in mobile ad hoc networks. In *Workshop on Principles of Mobile Computing*, pages 21–23, aug 2001.

[3] K. Fall. A Delay-Tolerant Network Architecture for Challenged Internets. In *Proceedings of ACM SIGCOMM03*, Aug. 2003.

[4] 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.

[5] R. Hermann, D. Husemann, M. Moser, M. Nidd, C. Rohner, and A. Schade. DEAPSpace: Transient Ad-Hoc Networking of Pervasive Devices. In *1st ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 133 – 134, Boston, Massachusetts, USA, 2000. ACM.

[6] U. C. Kozat and L. Tassiulas. Network Layer Support for Service Discovery in Mobile Ad Hoc Networks. In *Proceedings of IEEE/INFOCOM-2003*, Apr. 2003.

[7] N. Le Sommer. A Framework for Service Provision in Intermittently Connected Mobile Ad hoc Networks. In *8th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM 2007)*, Helsinki, Finland, June 2007. IEEE Computer Society Press.

[8] I. Leontiadis and C. 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.

[9] 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.

[10] A. Lindgren, A. Doria, and O. Schelen. Probabilistic Routing in Intermittently Connected Networks. In *Proceedings of the The First International Workshop on Service Assurance with Partial and Intermittent Resources (SAPIR 2004)*, Fortaleza, Brazil, Aug. 2004.

[11] R. Meier, V. Cahill, A. Nedos, and S. 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.

[12] M. Musolesi, S. Hailes, and C. 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.

[13] 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.

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

[15] A. Nedos, K. Singh, and S. Clarke. Service*: Distributed Service Advertisement for Multi-Service, Multi-Hop MANET Environments. In *Proceedings of 7th IFIP International Conference on Mobile and Wireless Communication Networks (MWCN'05)*, Marrakech, Morocco, Sept. 2005.

[16] L. Pelusi, A. Passarella, and M. Conti. Opportunistic Networking: Data Forwarding in Disconnected Mobile Ad Hoc Networks. *IEEE Communications Magazine*, nov 2006.

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

[18] G. Sollazzo, M. Musolesi, and C. 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.

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

[20] C. N. Ververidis and G. C. Polyzos. Routing Layer Support for Service Discovery in Mobile Ad Hoc Networks. In *Pervasive Wireless Networking Workshop of the 3rd IEEE International Conference on Pervasive Computing and Communications PerCom 2005*, pages 258–262, Kauai Island, Hawaii, mar 2005.

[21] F. Zhu, M. Mutka, and L. Ni. PrudentExposure: A Private and User-centric Service Discovery Protocol. In *Second IEEE International Conference on Pervasive Computing and Communications (PerCom'04)*, pages 329–338, Orlando, Florida, USA, mar 2004.