

Toward a Platform for Service Discovery and Invocation in Disconnected Mobile Ad Hoc Networks

Romeo Said and Yves Mahéo

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

Abstract

This paper presents a middleware platform that supports the provision of services in disconnected MANETs. This kind of MANET shows specific mobility, volatility and density characteristics that lead to the absence of end-to-end connectivity and thus to the fragmentation of the network. The presented platform exploits content-based communications (through a publish/subscribe paradigm) and follows a store-carry-and-forward approach for the network-wide opportunistic dissemination of messages. Service discovery and invocation are implemented on top of these communication features. The paper first describes the two layers of the platform and then gives simulation results obtained when evaluating the discovery and invocation satisfactions as well as the network load in a realistic scenario.

Keywords: Disconnected Mobile Ad Hoc Networks, Service-oriented Computing, Service Platform, Content-based Communication, Opportunistic Networking

1 Introduction

Mobile ad hoc networks (MANETs) are formed spontaneously by a number of mobile devices that communicate thanks to short-range wireless communication capabilities, using for example Wi-Fi or Bluetooth interfaces. A main advantage of this kind of network is that it can be used without deploying a specific –and sometimes costly– infrastructure (such as an interconnected set of Wi-Fi base stations). MANETs actually cover a wide variety of situations depending to the density of nodes in the network, their volatility –that is the fact that they may temporarily be switched off– or their mobility scheme. In this paper we address a particular class of MANETs we will refer to as *disconnected* MANETs. Disconnected MANETs show a low density and/or a high mobility of nodes. As a consequence, a temporaneous path cannot always be established between any pair of nodes in the network: end-to-end connectivity

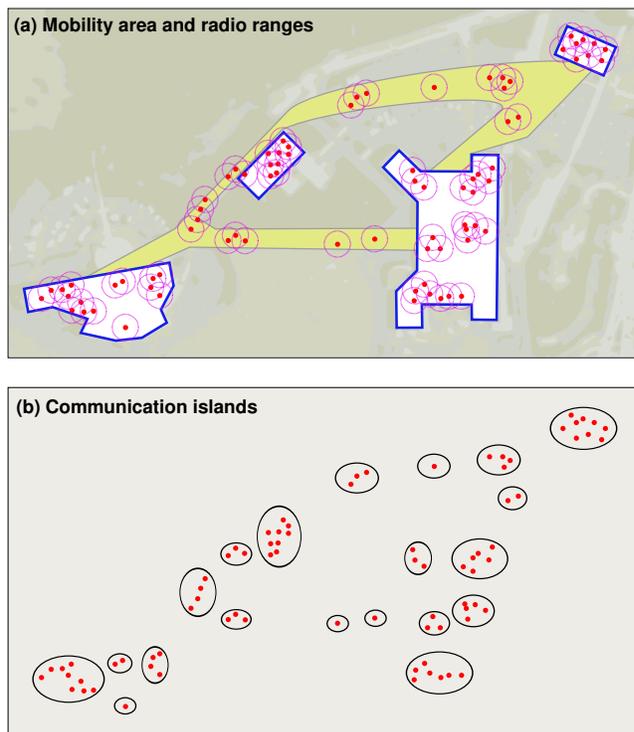


Figure 1. Disconnected MANET

is thus not guaranteed. Figure 1 shows an example of such a disconnected MANET, in which some laptops and handheld devices with Wi-Fi interfaces are scattered in several buildings. Because of their mobility, their limited radio-range and their volatility, the devices in this network form so-called “islands” whose topology evolves continuously. The lower part of Figure 1 highlights the fragmentation of the network into communication islands at a given time. Temporaneous communication between two islands is not possible. As a consequence, network-wide communication itself in such a network is still a challenge, namely because routing techniques designed for fully connected MANETs cannot be applied.

There is however a need for building applications that target disconnected MANETS. Some examples of application domains include information sharing during disaster relief interventions, tourist services in infrastructure-less sites or traffic information in town centers. Service-oriented applications seem well suited for this purpose. Building applications based on software services is now well mastered and supported by many techniques and tools, among which the most popular Web Services [9]. Moreover this approach fosters decoupling between interacting applicative entities. It should therefore accommodate well with the connectivity constraints of disconnected MANETS.

Service-oriented applications involve distributed components that can either play the role of service providers or service clients. Service provision is generally performed in two main steps: *service discovery*, during which services advertised by providers can be discovered by potential clients, and *service invocation*, during which a given client actually interacts with the provider of a previously discovered service. A selection phase may precede the invocation, when the opportunity is given to the service client to choose among several providers.

Although the service-oriented approach seems relevant for disconnected MANETS, implementing distributed services for such networks still poses several challenges. Not only network-wide communication features must be provided, in spite of constant network fragmentation, but aspects such as the unpredictable reachability of the providers or potential communication delays must be taken into account at the service level. Our overall objective is to build a service platform that will support the execution of service-oriented applications in disconnected MANETS. We describe in this paper the main features of this platform, focusing on service discovery and service invocation.

As far as communication is concerned, the absence of end-to-end connectivity precludes traditional routing techniques that strive to identify a succession of nodes that forms a path between a sender and a receiver, in order to be able to transmit a message temporarily along this path. In our platform, we will rather apply a *store-carry-and-forward* approach, in which each node maintains a cache of messages. Hence, a message can be stored temporarily on a node, in order to be forwarded later by this node when circumstances permit, that is, when one or several devices susceptible of being appropriate carriers of the message pass in the neighborhood. Moreover, mobility can facilitate message propagation, as devices can carry a message when moving from an island to another. Another important impact of the fragmentation of the network is that the reachability of hosts is very fluctuating. So relying on destination-based communication (e.g. by manipulating addresses of service providers) for delivering requests and replies is likely to be inappropriate. A more suitable

communication model is *content-based* communication. In content-based communication, the flow of information is interest-driven rather than destination-driven [4]. Receivers specify the kind of information they are interested in, without regard to any specific source. Senders simply send information in the network without addressing it to any specific destination. It is noteworthy that content-based communication does not preclude destination-based communications which can be simply implemented thanks to a destination attribute included in the content of the messages and serving as a key-interest by the receivers. Our platform includes an implementation of this content-based communication paradigm based upon message *dissemination*. Both service discovery and service invocation exploit this dissemination through a publish/subscribe API.

At the service level, disconnected networks are challenged environments as far as interaction opportunities between clients and providers are concerned. A centralization of service descriptor management is therefore not applicable. On the contrary, our platform includes fully distributed descriptor repositories. Moreover we enrich these descriptors with semantic attributes to facilitate the selection of a service provider. Finally, as service clients have no guarantee about the immediate availability of –even already known– service providers at a given time, it is desirable that a client be able to invoke more than one specific provider when possible. We address this aspect by implementing a form of grouping of providers that can be invoked interchangeably.

The remaining of this paper is organized as follows. The different features of our service platform for disconnected MANETS is detailed in Section 2. We namely detail in this section the lower layer of the platform that serves as a communication support and the upper layer that focuses on service level aspects. Section 3 presents simulation results. The paper ends with some related works listed in Section 4 and a conclusion in Section 5.

2 Service platform

The service platform we designed is structured in two layers. Figure 2 gives an overview of its architecture. The first layer is a high-level service layer that is in charge of all service oriented processing, enabling discovery and invocation interactions between clients and providers. The second layer is a communication layer taking care of all data level exchanges, resulting in a communication system fully adapted to disconnected network environments. From the service layer point of view, a message is sent from a source node in the network using a message publication. The message is disseminated in the network according to its content; it is received at a node that has previously provided a pattern matching this message, via a subscription operation.

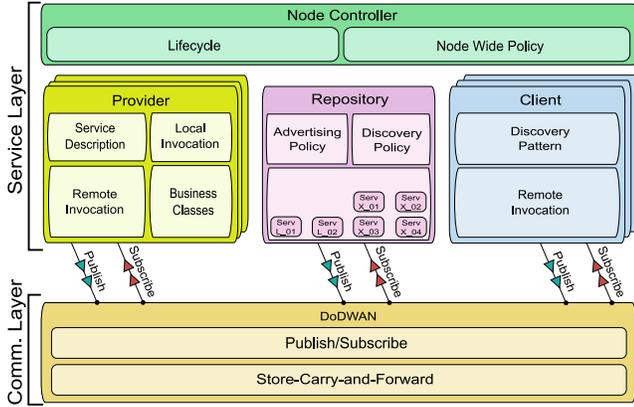


Figure 2. Architecture of the service platform

One service platform is operational on each network node. It enables a node to act as a provider and/or a client. The service platform can host several providers: each provider module exposes an installed business service, acting as a container for installed business classes. Likewise the service platform can host several clients. Each client module needs to consume an exposed service corresponding to a defined service pattern. The repository module acts as a decentralized storage of local and collected descriptions of services, and it applies global discovery and advertising policies. The communication layer implements a publish/subscribe module over a store-carry-and-forward module in order to enable content-based message dissemination. The node controller manages the lifecycles of the previously mentioned modules and applies node-wide preferences.

The service delivery is carried out in two phases of client-provider interactions. The first phase is the service discovery during which a provider should advertise each of its service in the network for potential clients. The second phase is the service invocation during which a client can invoke a previously discovered provider.

2.1 Communication layer

The lower part of our service platform consists in a communication support adapted to disconnected MANETs. This layer, called DoDWAN (Document Dissemination in Wireless Ad Hoc Network), ensures the content-driven dissemination of information network-wide, despite the fragmentation of the network into isolated communication islands. It implements a model that manipulates structured pieces of information we referred to as “documents”. A document is composed of two parts: its header, and its content. The header can be perceived as a collection of attributes, which can provide any kind of information about the corresponding document, such as its origin, its topic, a

list of keywords, the type of its content, etc.

Some storage capacity in each node is dedicated to a local cache of documents, so this node can serve as a mobile carrier (thus it can be seen as a memorizing relay) for these documents while moving in the network. The dissemination model implemented in DoDWAN is not mere flooding. Indeed, each node in the network is associated an “interest profile”, that determines the kind of information it is interested in, and thus implicitly the kinds of documents for which it is willing to serve as a mobile carrier. Therefore, uninterested nodes do not participate in the dissemination.

A gossip-like communication protocol orchestrates interactions between neighboring nodes, allowing them to exchange documents according to their respective interest profiles. Interaction between mobile nodes relies on a simple scheme, whereby each node periodically broadcasts its own interest profile and a catalog of the document headers that are currently available in its local cache. When a node discovers that one of its neighbors can provide a document it is interested in (that is, a document header that matches its own interest profile and that is not already available in its own cache), it can request a copy of this document from this neighbor. Upon receiving one or several requests for a particular document from its neighbors, the owner of this document broadcasts it on the wireless medium, so it can be received by all requesters simultaneously. Transient contacts between mobile nodes are thus exploited opportunistically for exchanging documents between these nodes, based on their respective interest profiles, and based on the documents they can provide each other on demand.

DoDWAN provides a publish/subscribe API we used to construct our service layer. With this API the programmer is provided a number of interfaces for publishing documents and for subscribing for documents. Publishing a document comes down to depositing this document in the local cache so that it can automatically disseminate in the network, following the protocol described above. The subscription method takes two parameters: first a document pattern must be given for selecting a particular kind of documents (documents that match this pattern will be deposited in the local cache). The second parameter is a handler that will be called when the selected documents arrive in the cache. Specifying a null handler is a means to make the node behave as a simple relay for the selected documents, without actually consuming these documents (i.e. passing them to the upper layer via the handler).

DoDWAN is designed so as to maximize the document delivery ratio while remaining very frugal as far as the number and the volume of messages are concerned. Further information about DoDWAN and about the content-based dissemination protocol it implements can be found in [11].

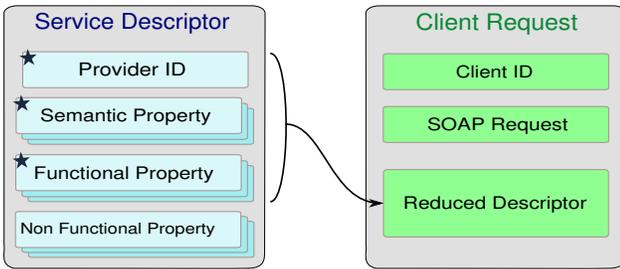


Figure 3. Components of a service descriptor and of a client request

2.2 Service discovery

Service description. The service discovery process begins with the description of services by the provider, using all the information needed by potential clients to select the appropriate service and to invoke its provider. The provider creates and advertises its service descriptors by publishing them in the network. A descriptor is disseminated by network nodes until it eventually reaches a client node. The descriptor is an XML document whose main components are depicted in Figure 3. Three mandatory components form a minimal descriptor: identifier, semantic property, and functional property (starred components in Figure 3). Functional properties (written in WSDL [6]) describe the programming interface used by clients to formulate invocation requests. Though, these functional properties may not be expressive enough for clients to choose the right service. Therefore, a set of semantic properties are included in the descriptor, that form a contract proposition of offers and requirements. An offer represents what the service proposes to its clients. A requirement represents what the service needs as input from the client in order to do its job. Providers can also add non-functional properties to enhance the description (e.g. author, version, language, cost, etc.).

Service selection. On the client side, another XML document called service pattern must be created to convey the wishes of a client hoping to discover a suitable service. The service pattern contains components similar to those of the service descriptor. A service pattern passes from the client module to the repository. The repository may itself append non functional properties to the pattern in order to take into account node-wide discovery preferences (e.g. a required security level). Finally the repository submits the resulting pattern for subscription in the publish/subscribe module.

The pattern subscription in the publish/subscribe module enables a content-based matching process between the client's preferences expressed in the pattern against the descriptors that are disseminated in the network. If a match

is made by the publish/subscribe module, the received descriptor is passed up to the repository module for service-level caching. The client can now select the descriptor cached in the repository and formulate its requests accordingly. In case the repository stores several matching descriptors for the same pattern, it is up to a higher application level to decide which one to select.

In the current implementation of the platform, we use simple property-value elements with regular expressions in the pattern to be matched against corresponding property-value elements of the descriptor. In order to enhance the discovery, we intend in future works to use and evaluate semantic ontologies for the description of services, taking advantage of the work done in the domain of semantic web services [2].

2.3 Service invocation.

The service layer is tolerant to communication delays induced by the very nature of disconnected network environments. The amount of delay that a client can tolerate depends on the interactivity level needed (at business level) for it to operate in reasonably good conditions.

After the selection process, the selected provider is invoked using request and response messages (currently implemented using SOAP [10]). The client formulates a request according to the functional properties of the descriptor; it publishes its request in the network to the attention of the provider (using the provider's ID) and then subscribes to the corresponding responses so he can be notified when they are received.

Service group. With the procedure described above, the final destination of an invocation request is restricted to the selected provider node. Yet, several compatible providers could be found in the network, all exposing the same business service. These compatible providers can therefore substitute each other when it comes to responding to a request, hence providing sometimes faster responses (if the originally intended provider has become distant or even unreachable). The choice of actually using this invocation mode depends on the client needs and on the service application scenarios (state-less or state-full service invocations). The multiplicity of providers could be the result of different providers hosting the same service implementation, or different providers hosting different service implementations with the same declared invocation interface.

We call a service group the set of these compatible service providers. In order for a client to benefit from a group of compatible providers, this client should use the content-based dissemination of the request message more effectively. We made this possible by appending a reduced version of the service descriptor to the header of the request

in order to let other providers decide whether or not they can answer. Hence, as illustrated in Figure 3, a client formulates its request not only with the SOAP part and the primary intended destination (i.e. the provider ID) but also with a description of the invoked service (i.e. some semantic and functional properties). Symmetrically, providers do not receive only the requests for which they are the primary destination. Instead, they use the publish/subscribe module so that they receive all the requests containing a reduced-descriptor part matching one of the descriptors of the services they provide.

This use of groups may yield multiple response messages for the same client request. We identified two possible policies that can be applied locally by the providers for managing these responses. The first one is an *isolation* policy: when a request is received, the provider simply replies to this request. The other one is a *competition* policy: providers attempt to reduce the replication of responses. When a provider receives a request, it replies and publishes an exclusivity control message for the other compatible providers in an effort to prevent them from replying to the same request. A provider makes sure that it has not already received any exclusivity message before replying to a received request. Notice that this policy is a post-response coordination attempt, because compatible providers do not rely on any form of a priori group affiliation and therefore the exclusivity does not need to be guaranteed. It is an attempt to reduce the extra load created by multiple responses that are sent in the network.

This describes a service group scenario where the compatible providers are acting rather independently. More advanced inter-provider coordinations and cooperations are envisaged in future works.

3 Simulation Results

We conducted a number of simulations in order to evaluate our service platform. The main objective is to assess the performance of service discovery and invocation (in terms of response time and network load) in as realistic as possible conditions.

Environment. The environment we used for our simulations is an area of 500×900 meters populated with network nodes (see Figure 1). It represents a tourist site with 4 buildings where pedestrians equipped with laptops or hand-held devices can wander. Nodes can be stationary in buildings temporarily or move from one building to another through pre-defined footpaths, at a speed of 0.5 to 2 meters per second. A node can choose to pause for a period of time of 1 to 6 minutes. A node spends 50% of the time indoor and the rest of the time moving between buildings. A node represents a laptop, a PDA, or a smart-phone equipped with

a Wi-Fi interface and capable of interacting with its neighbors in ad hoc mode. We assume that nodes have a wireless coverage radius ranging from 20 to 30 meters. The simulations were performed with the MADHOC network simulator [13], which allows us to run the actual implementation code of the service platform.

We populated the simulation environment with 80 network nodes, among which 8 play the role of service providers and 32 act as service clients. The remaining 40 nodes are neither providers nor clients but are used as simple message relays in some simulation runs. Four different business services are deployed on the providers; each service is actually deployed twice in order to obtain 8 provider nodes. Four different types of clients are present, forming four sets of eight clients targeting the same service.

Scenarios. Globally, at the start of a simulation run, providers and clients are distributed in the 4 buildings, each building containing 2 service providers (one fixed provider and one mobile provider) and 8 mobile service clients.

The experiment consists in running three simulation scenarios, all with the same node mobility model but with various behaviors with respect to message dissemination:

1. In the first simulation scenario, only the 8 providers and 32 clients participate selectively in the message dissemination performed by the communication layer. The remaining 40 nodes are passive and do not even relay messages.
2. The second simulation scenario is similar to the first, but now, the 40 nodes that are neither providers nor clients become interested in relaying messages. So we obtain 40 provider and client nodes that relay the messages and effectively consume the messages relative to service discovery and invocation they are concerned with, and 40 additional nodes just helping in the message dissemination process, regardless of the type of service involved.
3. In the third simulation scenario, we consider that all 80 nodes have disabled their store-carry-and-forward module of the communication layer. As a consequence, no network-wide dissemination is possible. In this “proximity” scheme, providers and clients must be in wireless range to interact. This scenario is intended to model an approach that gives results very similar to what could be obtained with more classical one-hop broadcast-based discovery methods (e.g. with UPnP).

As far as the service layer is concerned, the behavior of the providers and the clients is defined for a discovery phase followed by a single invocation. At time zero, all the service providers publish their service descriptor and wait for incoming service invocations. Publication is made only once

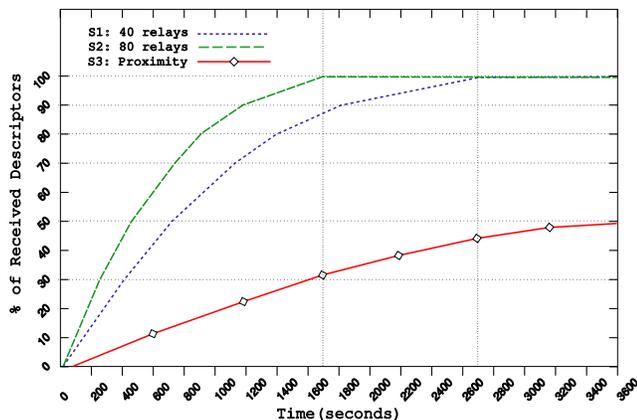


Figure 4. Discovery satisfaction

except in the third scenario in which it is repeated regularly because network-wide dissemination is not performed. Every client’s objective is to invoke a desired service once. For this purpose, a client defines a service pattern and waits until it discovers an adequate service descriptor. It then makes its invocation to the provider specified in the discovered descriptor.

Satisfaction ratios. We evaluated the satisfaction of clients for the discovery phase as well as for the complete discovery-invocation cycle.

The discovery satisfaction ratio is defined as the percentage of clients that have discovered the service they targeted at a given time. Figure 4 presents the results obtained for the three simulation scenarios described above. It shows that the use of network-wide dissemination has a drastic impact on the speed of discovery: after 1600 seconds, 90% of the clients have discovered the service they wanted when only half of the nodes (that is the providers and the clients) participate in the dissemination (scenario 1 with 40 relays). This percentage reaches 100% when all the nodes are ready to play the role of relay (scenario 2 with 80 relays). After the same period of time, only 30% of the clients are satisfied when communications are restricted to one-hop transmissions (scenario 3: proximity).

The satisfaction ratio is calculated in a similar way for service invocations: it is the percentage of clients that have received a reply from the provider (after the discovery phase and after having sent their invocation request). Figure 5 plots this ratio against time for the three scenarios considered. Scenario 3 shows an invocation curve that is almost identical to the discovery one. This is because the invocation request and response can almost always be transmitted immediately after the client has discovered the service, as in such cases the provider is a close neighbor (the slight loss is due to the fact that sometimes a client and a provider

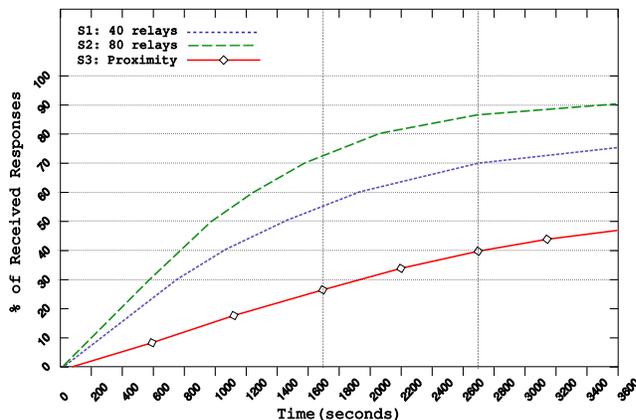


Figure 5. Invocation satisfaction

do not stay close to each other long enough to complete the invocation process). The performance in scenarios 1 and 2 –that exploit our network-wide dissemination model–remains significantly higher, in spite of the fact that point-to-point requests and replies are performed via publish/subscribe methods.

Network load. Resorting to network-wide dissemination for service advertisement yields a higher number of messages when compared to a one-hop broadcast-based method. Moreover, the fact that we exploit a publish/subscribe for point-to-point invocation requests and replies is likely to burden the network with an extra load. However, the gossiping protocol implemented in DoDWAN has been designed so as to minimize the number and the size of the exchanged messages, thus reducing the actual load. Figure 6 plots the measured load of the network over time in the three considered scenarios. This load is expressed as the cumulative amount of data (in kB) sent by all nodes in the wireless medium. The figure confirms that the load in the “proximity” scenario increases linearly, because of the periodic advertisements of service descriptors. The curves for the scenarios involving network-wide dissemination have a higher slope but tends to flatten after a while. It can be noticed that the number of relays can be chosen to obtain a good compromise between the client satisfaction and the network load. In this example, fixing the number of relays to 40 induced a still reasonable load while offering a fairly good satisfaction ratio (as shown in figure 4 and 5).

Impact of groups of services. We evaluated the use of service groups in the first and second simulation scenarios. In these scenarios, each business service is deployed in the network in two providers. Therefore, the use of groups consists in letting a service provider respond to a request that was initially sent to the other provider hosting the same

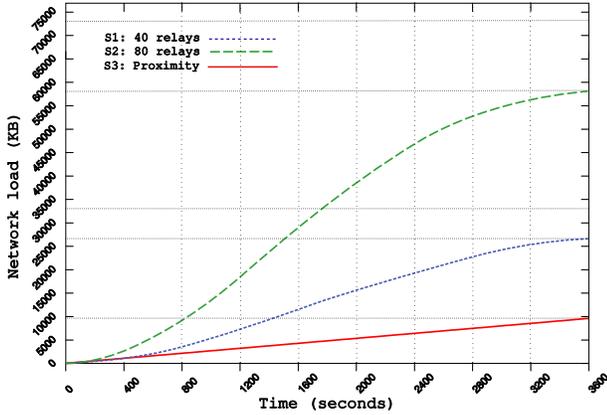


Figure 6. Cumulative network load

business service.

Figure 7 shows a comparison of the satisfaction ratios for the invocation, obtained while enabling and disabling the service groups. We can observe an increase in satisfaction because a client request is not send to a specific provider but rather addressed to the business service, so the response coming from the closest provider will arrive faster to the client. This preliminary experiment considered only groups of two providers. Nevertheless, the impact of provider replication is significant. It can be noticed that the two middle curves are pretty close: service groups can save us the use of extra relay nodes.

We used the exclusivity policy described in Section 2.3 for managing the multiple providers' responses: when a provider answers to a request it also sends a control message to try to prevent other compatible providers from answering. The load measurements we have conducted do not show a significant overhead related to these messages in the scenarios considered.

4. Related Work

A number of research works are concerned with communication in disconnected MANETs, mainly with the objective of providing destination-based communication [20, 17] and sometimes proposing content-based approaches [3, 7]. A noticeable contribution in this area has been made in the European Huggle project [1]. All of these works however focus on communication issues and do not address the problems of the design and the implementation of a service platform targeting disconnected MANETs.

On the other hand service-oriented programming in pervasive environments has been the object of many research activities. Most of them primarily focus on service discovery protocols using traditional centralized service directories over infrastructure-based (LAN) or single-hop wireless

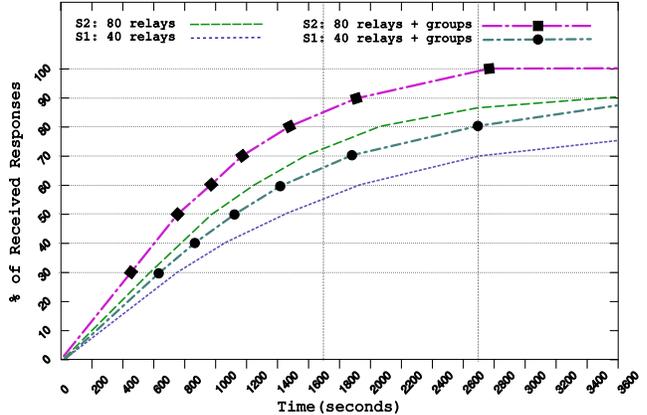


Figure 7. Invocation satisfaction using service provider groups

networks ([21] presents a survey of the main protocols).

Recently, the characteristics of MANETs (dynamicity, low resources, unreliable transmissions) have been taken into account, With the development of specific discovery protocols, as part of the routing layer [5, 8, 15, 19] or as part of the service layer with no dependency on routing [12]. In these protocols, communications are still considered achievable and to some extent sustainable (routes between hosts can always be established). GSR [5] enables group-based service discovery using discovery requests and advertisements, and reuses the paths of discovery to also handle data transmission for routing invocations (with session failure management). Konark [12] uses a distributed multi-hop advertisement/discovery scheme but relies on a local multicast and implements connected invocations that are not suitable for disconnected environments.

Other efforts try to create a virtual higher-level network in MANETs where some selected nodes are assigned special functionalities. Some of them implement semi-centralized directories for discovery [18] where each inter-connected network (small connected network island) has one directory, while others assign network service brokers [16] or create network backbones [14]. Our prototype does not assume any viable interconnections so each host is responsible for its own service repository. In addition, we do not use overlay network structures nor maintain discovery routes.

5 Conclusion

We presented in this paper the design and the main implementation features of a service platform that specifically targets disconnected MANETs, in which no end-to-end connectivity can be guaranteed.

Our service platform is structured in two distinct layers. The lower layer, called DoDWAN, is a communication middleware dedicated to the content-based dissemination of documents. It leverages on the store-carry-and-forward approach so as to free itself of the presence of independent communication islands when transmitting documents. DoDWAN offers a publish/subscribe API that is used in the service layer of the platform for exchanging service advertisements and discovery messages as well as for invocation requests and responses. A number of features are included in this service layer in order to improve the opportunities of interaction between service providers and service clients: service descriptors are enriched with semantic information, and a form of grouping of service providers is implemented so as to exploit possible redundant service provisions.

Simulation results demonstrate the feasibility of service provision in disconnected MANETs built upon a content-based dissemination middleware like DoDWAN and confirm the positive impact of the features we implemented at the service level.

Acknowledgements. This work is supported by the French Agence Nationale de la Recherche under contract ANR-05-SSIA-0002-01.

References

- [1] Huggle Project. <http://www.huggleproject.org>.
- [2] OWL-S: Semantic Markup for Web Services. W3C Member Submission. <http://www.w3.org/Submission/OWL-S/>.
- [3] R. Baldoni, R. Beraldi, M. Migliavacca, L. Querzoni, G. Cugola, and L. Migliavacca. Content-Based Routing in Highly Dynamic Mobile Ad Hoc Networks. *Journal of Pervasive Computing and Communication*, 1(4):277–288, Dec. 2005.
- [4] A. Carzaniga and A. L. Wolf. Content-based Networking: A New Communication Infrastructure. In *NSF Workshop on an Infrastructure for Mobile and Wireless Systems*, number 2538 in LNCS, pages 59–68, Scottsdale, Arizona, Oct. 2001. Springer-Verlag.
- [5] D. Chakraborty, A. Joshi, and Y. Yesha. Integrating service discovery with routing and session management for ad-hoc networks. *Ad Hoc Networks*, 4(2):204–224, Mar. 2006.
- [6] R. Chinnici, J. J. Moreau, A. Ryman, and S. Weerawarana. Web Services Description Language (WSDL) Version 2.0. Technical report, W3C, Mar. 2006.
- [7] P. Costa, M. Musolesi, C. Mascolo, and G. P. Picco. Adaptive Content-based Routing for Delay-tolerant Mobile Ad Hoc Networks. Technical report, UCL, Aug. 2006.
- [8] P. E. Engelstad, Y. Zheng, R. Koodli, and C. E. Perkins. Service Discovery Architectures for On-Demand Ad Hoc Networks. *Ad Hoc and Sensor Wireless Networks*, 1, 2006.
- [9] W. S. A. W. Group. Web services architecture. <http://www.w3.org/TR/ws-arch>.
- [10] M. Gudgin, M. Hadley, N. Mendelsohn, M. Jean-Jacques, and H. Frystyk Nielsen. SOAP version 1.2. Technical report, W3C, June 2003.
- [11] J. Haillot and F. Guidec. A Protocol for Content-Based Communication in Disconnected Mobile Ad Hoc Networks. In *IEEE 22nd International Conference on Advanced Information Networking and Applications (AINA'08)*, Okinawa, Japan, Mar. 2008.
- [12] S. Helal, N. Desai, V. Verma, and C. Lee. Konark : Service Discovery and Delivery Protocol for Ad-hoc Networks. In *3rd IEEE Conference on Wireless Communication Networks (WCNC)*, New Orleans, USA, Mar. 2003.
- [13] L. Hogie, P. Bouvry, and F. Guinand. The MADHOC simulator. <http://www-lih.univ-lehavre.fr/~hogie/madhoc>.
- [14] U. C. Kozat and L. Tassiulas. Network Layer Support for Service Discovery in Mobile Ad Hoc Networks. In *Joint Conference of the IEEE Computer and Communications Societies (IEEE/INFOCOM-2003)*, San Francisco, USA, Apr. 2003.
- [15] V. Lenders, M. May, and B. Plattner. Service Discovery in Mobile Ad Hoc Networks: A Field Theoretic Approach. In *International Symposium on a World of Wireless, Mobile, and Multimedia Networks (WoWMoM 2005)*, Taormina, Italy, June 2005.
- [16] A. Nedos, K. Singh, and S. 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, Sept. 2005.
- [17] L. Pelusi, A. Passarella, and M. Conti. Opportunistic Networking: Data Forwarding in Disconnected Mobile Ad Hoc Networks. *IEEE Communications Magazine*, Nov. 2006.
- [18] F. SAILHAN and V. ISSARMY. Scalable Service Discovery for MANET. In *International Conference on Pervasive Computing and Communications (PerCom'2005)*, Hawaii, USA, Mar. 2005. IEEE Press.
- [19] A. Varshavsky, B. Reid, and E. de Lara. A cross-layer approach to service discovery and selection in MANETs. In *2nd IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 05)*, Washington, USA, Nov. 2005. IEEE Press.
- [20] Z. Zhang. Routing in Intermittently Connected Mobile Ad Hoc Networks and Delay Tolerant Networks: Overview and Challenges. *IEEE Communications Surveys and Tutorials*, 8(1):24–37, Jan. 2006.
- [21] F. Zhu, M. Mutka, and L. Ni. Service Discovery in Pervasive Computing Environments. *IEEE Pervasive Computing*, 4(4):81–90, Dec. 2005.