Middleware Support for Delay-Tolerant Service Provision in Disconnected Mobile Ad Hoc Networks

Yves Mahéo, Romeo Said, Frédéric Guidec

HAL Id: hal-00341711
https://hal.science/hal-00341711

Submitted on 25 Nov 2008

Middleware Support for Delay-Tolerant Service Provision
in Disconnected Mobile Ad Hoc Networks

Yves Mahéo, Romeo Said and Frédéric Guidec

Valoria Laboratory, European University of Brittany

{Yves.Maheo,Romeo.Said,Frederic.Guidec}@univ−ubs.fr

Abstract

The service-oriented approach is known to be adequate for building distributed applications for mobile ad hoc networks, and a number of successful research works using this approach in routed MANETs have already been conducted. Yet, providing support for service-oriented applications poses specific problems in disconnected MANETs where end-to-end connectivity is not guaranteed. This paper presents a prototype Java middleware support for delay-tolerant service provision in disconnected MANETs. It first describes the lower layer of this middleware, which implements a protocol for document dissemination that leverages on delay-tolerant, opportunistic and content-based communication. It then explains how service discovery and service invocation mechanisms can be implemented on top of this protocol. Finally, simulation results are presented, which show that this approach gives good results compared with an alternative approach that consists in exploiting single-hop interactions between service providers and service clients.

1. Introduction

The service-oriented approach becomes more and more popular for building distributed applications. Service-oriented applications involve distributed components that can either play the role of service providers or service consumers. Service provision is generally performed in two steps: *service discovery*, during which services advertised by providers can be discovered by potential consumers, and *service invocation*, during which a given consumer actually interacts with the provider of a previously discovered service.

This decoupling between interacting entities makes this model suitable for mobile ad hoc networks (MANETs) as it can help cope with their dynamicity. Indeed, in contrast with an infrastructure-based wireless network (such as one relying on fixed access points), a MANET is a network

formed spontaneously by a group of mobile, wireless devices.

In the literature MANETs are often assumed to be dense enough, so they can be modelled as connected graphs in which end-to-end dynamic routing algorithms can be implemented [12]. For such networks, the research on service delivery mainly focuses on supporting service discovery, for service invocation can be performed just like in wired networks. A number of discovery protocols have indeed been developed for connected MANETs, most of them being closely related to the underlying routing protocols [2].

Yet, in MANETs showing a low density and/or a high mobility of hosts, there is no guarantee that a temporaneous path can ever be established between any pair of nodes in the network. Figure 1 shows a typical example of such a disconnected MANET,in which some laptops and hand-held devices with Wi-Fi interfaces are scattered in several buildings. In this figure empty circles denote inactive hosts (typically hosts in suspend-mode) and edges represent connectivity (and thus some potential for communication) between neighbour hosts. Because of their mobility, their limited radio-range and their volatility (they are not active all the time), the devices in this network form so-called "islands" whose topology evolves continuously. Temporaneous communication between two islands is not possible. As a consequence, network-wide communication itself in this kind of network is still a challenge, as routing techniques designed for fully connected MANETs cannot be applied. A method must thus be devised in order to bridge the gap between network islands. Delay-tolerant networking (DTN) is an approach that can help with this respect [5]. In a delay-tolerant network, a message can be stored temporarily on a host, in order to be forwarded later by this host when circumstances permit. Moreover, mobility can facilitate message propagation, as devices can carry a message when moving from an island to another. Disconnection also has an impact on the way service provision can be implemented. Both service discovery and service invocation must thus be revisited in such conditions.

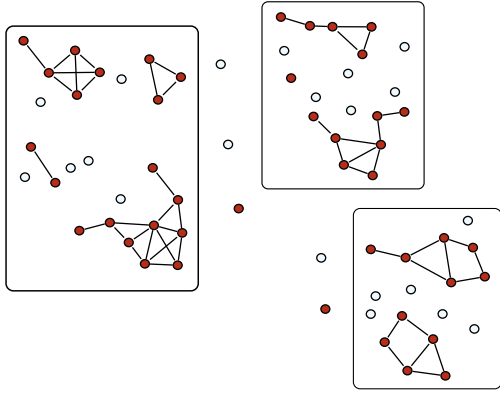This paper presents a two-layer middleware support for

Figure 1. Example of a disconnected MANET

delay-tolerant services in a disconnected MANET. The lower layer implements a communication protocol for DTN which leverages on the combined principles of content-based communication, opportunistic networking and delay-tolerant networking in order to account for the absence of end-to-end connectivity in the network. This protocol is accessible to the programmer through a simple publish/-subscribe API. The upper layer of our middleware provides service delivery facilities. It allows service providers to describe and advertise services, using both functional and non-functional properties. Service consumers can discover services network-wide according to patterns matched against service descriptions. Besides, because of the delay-tolerant nature of the underlying network, service requests and responses are actually performed asynchronously.

The remainder of this paper is organized as follows: Sections 2 and 3 respectively describe the communication layer and the service layer of our middleware support. Simulation results are described in Section 4. Finally, Section 5 discusses related research works, and Section 6 concludes the paper.

2. DTN communication layer

2.1. Model and Protocol

Our communication middleware is called DoDWAN (Document Dissemination in Wireless Ad Hoc Network). It supports the content-driven, opportunistic dissemination of structured pieces of information in the network. Content-based communication is a style of communication whereby the flow of information is interest-driven rather than destination-driven [1]. Receivers specify the kinds of information they are interested in, without regard to any specific source. Senders can send information in the network without addressing it to any specific destination. In

our model, the structured pieces of information we consider are referred to as "documents". A document is composed of two parts: its descriptor, and its content. The descriptor 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.

DoDWAN provides support for storing documents in a host's local cache, so this host can serve as a mobile carrier for these documents while moving in the network. Each host 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. DoDWAN also defines a gossip-like communication protocol to orchestrate interactions between neighboring hosts, allowing them to exchange documents according to their respective interest profiles. Interaction between mobile hosts relies on a simple model, whereby each host periodically broadcasts its own interest profile and a catalog of document descriptors that are currently available in its local cache. When a host discovers that one of its neighbors can provide a document it is interested in (that is, a document descriptor 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 neighbours, the owner of this document broadcasts it on the wireless medium, so it can be received by all requestors simultaneously. Transient contacts between mobile hosts are thus exploited opportunistically for exchanging documents between these hosts, based on their respective interest profiles, and based on the documents they can provide each other on demand.

DoDWAN has been 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. It has been implemented in Java, in such a way that it can notably exploits the broadcast capabilities of different MAC layers, such as that of 802.11 (Wi-Fi). Further information about DoDWAN and about the content-based dissemination protocol it implements can be found in [7].

2.2. Publish/subscribe API

DoDWAN provides several kinds of API, including a publish/subscribe API we used to construct our service platform. With this API the programmer is provided a number of interfaces that allow to perform the following tasks:

- *Create and manipulate transfer documents*
  A *transfer document* contains an arbitrary payload and a descriptor consisting of a number of attributes (couples key/ string value). Some attributes are predefined

2

(such as one that specifies the expiration date of a document) and new ones can be added as reqired by the programmer. Attributes can be manipulated with traditional getters/setters.

- *Publish documents*
  Transfer documents that are supposed to be delivered to others hosts are published, that is, they are simply "pushed" into the network. According to the model described above, the effect of publishing a document is that this document is simply deposited in the local cache of the sender. It will subsequently be automatically and opportunistically transmitted to other hosts that have subscribed for this kind of document.

- *Subscribe for documents*
  In order to specify what kind of transfer documents a host wishes to receive from the network, this host provides a document pattern. A document pattern is simply a descriptor that contains a number of attributes (those attributes one wishes to base the selection upon), each of them carrying a value that is a standard regular expression that will be matched against actual values in the descriptors of accessible documents. The above-mentioned interest profile of a host is directly generated by DoDWAN from the different subscription patterns.

- *Receive documents*
  Document subscription triggers the installation of an event listener. When subscribing for a particular kind of documents, one specifies a handler (actually an object with an implicit handler method) that should be called when a matching document is received and deposited in the local cache. Handlers are therefore automatically invoked upon reception of "interesting" documents, that is documents that match the local host's interest profile.

3   Service Layer

The upper-layer of our middleware support is an aggregation of service-oriented concepts and techniques forming what we refer to as "a service platform". This service platform was built while always trying to take into account all the specificities of disconnected mobile ad hoc networks. The two main characteristics that have to be taken into account are the communication delays and the absence of end-to-end connectivity, the latter leading to an unpredictable availability of service providers. These aspects have obviously an impact on the application logic. But because this impact is essentially application-specific, it will not be discussed further in this paper. Instead our aim is rather to show how middleware mechanisms can be implemented at the service-level so we can cope with the induced delays on both the provider and the client sides, and how we can give the client the means to be aware of the available services in its environment (discovery) and the means to asynchronously interact with these discovered services (invocation).

3.1   Architecture

The service platform we developed covers the main aspects of service-oriented programming. It therefore implements protocols for service description, service discovery, and service invocation. This platform is implemented in Java. Each mobile node in the network is meant to run only one instance of the service platform, that will take care of the needs of both service providers and service clients.

The architecture of our service platform is depicted in Figure 2. It is composed of the following modules:

- The *Service Manager* module exposes a service logic to be consumed by remote Client modules. A Service Manager is a service container. It includes service business classes and their descriptions, and it supplies translation between local and remote method invocations. Many instances of Service Manager modules can be operational, each instance corresponding to one installed service.

- The *Client* module wraps a service discovery pattern and supplies a user interface and remote method invocation functionality. Many Client modules can be operational, each module corresponding to a declared service pattern.

- The *Repository* module is a self managing storage space for service descriptors. It is also responsible for a platform-wide application of the advertising and discovery policies.

- The *Controller* module mainly handles the life-cycles of the Service Manager and Client modules.

Communications are carried out by message passing. Messages encapsulate transfer documents that can be sent or received using the publish/subscribe API detailed in Section 2. The content-based nature of the publish/subscribe paradigm is clearly appropriate for service advertisement and discovery. It namely allows semantic discovery when matching a client service pattern against service descriptors. Besides using the publish/subscribe model for service advertisement and discovery, we also rely on this model to support invocation requests and responses. Indeed, the underlying support for content-based communication makes it possible to address a document to a specific host by inserting a destination attribute in the document's descriptor.
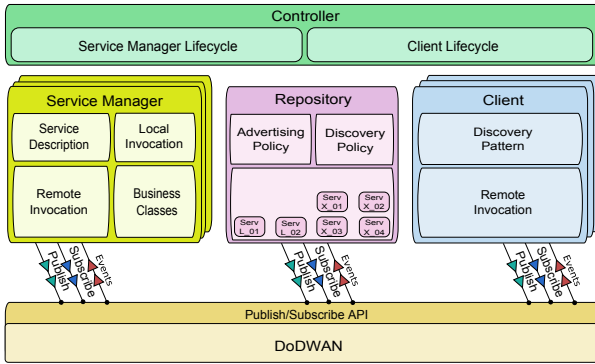
Figure 2. Architecture of the service platform

3.2 Behavior of Service Providers

On the service provider side, the administrator of the host installs for each service the needed service business classes and supplies the corresponding service descriptor. The descriptor of a service is an XML document containing all the information needed by the Client modules to select and then invoke this service. The functional interface of the service is described using the WSDL description language [3]. Non-functional properties (such as the service name, description, author, etc.) and context-aware properties (e.g. availability, reliability, trust) can be taken into account in the selection process.

The service descriptor is submitted to the advertising policy of the local Repository module. This module regularly publishes all local descriptors. The Controller module reads the business classes implementing the service and creates a new instance of the Service Manager module. This module subscribes for incoming invocation requests. When receiving such requests it will translate them into local method invocations, and it will then translate the local results back by publishing responses containing these results. The payload of the invocation message is implemented in SOAP [6].

3.3 Behavior of Service Clients

On the service client side, the user defines a service pattern (written in XML) that is meant to match a service descriptor published by a provider. The Controller module creates a corresponding Client module instance that updates the discovery policy of the Repository module by submitting its service pattern in order to convey its wishes in the discovery process. The Repository module submits the service pattern for subscription and stores all the matching service descriptors it receives from the network. Thus, at any time, and for a given service pattern, the client has access to a list of matching services collected in the local repository.

The client can invoke the desired service by publishing a method invocation (with a SOAP message payload). It then subscribes and waits for the service response.

4 Simulation Results

In order to evaluate the applicability of our approach, we have conducted a series of simulations. We focus here on one particular experiment whose objective was to measure the ability of our service platform to satisfy client service discovery requests and service invocations. In this perspective, we compared our approach to one consisting in restricting communication to one-hop transmissions. In this alternative model, a service client must move within the radio range of a service provider in order to be able to discover the services it provides, and possibly invoke these services. In the remainder of this section, we will call this model PROXIMITY-based as opposed to the DODWAN-based model described in this paper.

We ran two simulations to establish this comparison. We first simulated our service platform stack running over DoDWAN as depicted in Figure 2, and second we simulated the same service platform stack running over a proximity module that replaced the DoDWAN module. This proximity module implements one-hop discovery and invocation. We tuned the parameters of the implementation (such as the advertisement frequency) so as to remain consistent with the DoDWAN implementation.

In both simulation runs, the simulation starts by the client defining its service discovery criteria. These criteria are assembled into a service pattern. Then, the scenarios slightly differ depending on the chosen model.

- DODWAN-based: The client sends a discovery request (i.e. it injects the service pattern in the network) and then waits for a discovery response (i.e. it waits for being notified when a matching service description issued by a provider has arrived). After processing a discovery response, the client sends an invocation request to the provider and waits until the invocation response is received. All along these interactions, the client and the provider do not necessarily have to be close neighbors, as messages are exchanged network-wide using the DoDWAN dissemination protocol.

- PROXIMITY-based: After the client has defined its service pattern, it waits for an encounter with a matching service provider (which regularly broadcasts service advertisements). When such a provider is in radio range, the client processes the advertised service description, sends its invocation request and waits until it receives a response. Of course, in this scneario the client and the provider must stay in radio range during the whole duration of the transaction.

The simulation environment is an open 1 km² simulation area, populated with hosts equipped with Wi-Fi interfaces evolving according to a random waypoint mobility model at speeds varying between 2 m/s and 3 m/s. We created a scarcely populated environment with 100 hosts. Among these hosts, only one is a service provider (which provides a unique service) and 20 of the remaining hosts are potential clients of this service. The remaining 79 hosts are neither producers nor consumers of the single service considered in this scenario. Yet they can serve as message forwarders and carriers, thus helping documents to disseminate network-wide. At time zero, all the clients begin searching for a provider of the desired service. Each of them will send an invocation request as soon as it receives the service description. We ran the simulations in the same environmental conditions for both the PROXIMITY-based and the DODWAN-based communication models. The simulations were performed with the MADHOC network simulator [9], which allows us to run the actual code of the service platform and the underlying communication middleware.

Figure 3 gives some information about the performance of the discovery process and the overall process (involving a discovery and an invocation) for the PROXIMITY-based and the DODWAN-based communication models. The curves show the percentage of client hosts that have received a response against time. A response is a discovery response (dashed curves) or an invocation response obtained after a discovery process (full-line curves).
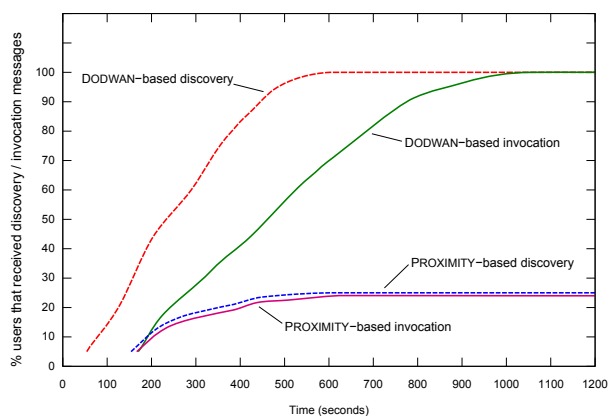

Figure 3. Discovery and invocation satisfaction ratios for the DODWAN-based and the PROXIMITY-based models

As expected, the two PROXIMITY-based curves are almost superimposed. This is because of the synchronous interactions of requests and responses this model permits. However, the global performance remains low: no more than 20% of the clients are satisfied, even after a 1200 s pe-

riod of time. The fact that a client and a provider can only count on their own move to be able to interact is indeed very penalizing.

With the DODWAN-based model, we observe a delay between the discovery and the invocation due to the very nature of the communications. This delay increases (before the curve flattens) because of the two-way transmissions induced by the service invocation. Yet clients discover the service provider much faster than with the PROXIMITY-based model: the first client receives a service description in approximately 50 s (against 130 s with the PROXIMITY-based model). And after 550 s, all the clients have completed the discovery phase. The same kind of observation can be made when studying invocation responses: performance is comparable with the PROXIMITY-based until 200 s but the the curve clearly arises afterwards. After approximately 17 mn (1030 s), all the clients have received their invocation response (against only 20% for the PROXIMITY-based model).

These results confirm that using DoDWAN can effectively help in allowing interaction between service providers and service clients, especially as the network gets sparser. The global cost in messages has not been taken into account in the work reported here. However, several other simulations –not detailed for the sake of brevity– show that, when considering a greater number of service providers, the relative cost induced by the participation of relaying hosts in the DODWAN-based model decreases.

5. Related Work

Service-oriented programming in pervasive environments has been the object of many research activities. Most of them primarily focused on service discovery [16] and tried to use traditional centralised directory services in infastructure-based wireless networks.

Recently, the characteristics of MANETs (dynamicity, low resources, unreliable transmissions) have been taken into account, with the development of specific discovery protocols. For many of these protocols, communications are still considered achievable and to some extent sustainable (routes between hosts can always be established) [15, 4, 11, 2]. Konark [8] uses a fully distributed advertisement/discovery scheme but relies on a multi-hop multicast and implements session-based invocations.

Other efforts try to create a virtual upper layer in MANETs where some selected hosts are assigned special capabilities. Some of them implement decentralized directories for discovery [14] where each interconnected network (small connected network group) has one directory, while others assign network service brokers [13] or create network backbones [10]. Our prototype does not assume any viable interconnections so each host is responsible for its own ser-

vice repository. We do not use overlay network structures nor maintain discovery routes.

6. Conclusion

We presented in this paper the implementation of a two-layer service provision middleware targeted at disconnected mobile ad hoc networks. It provides service discovery and service invocation mechanisms that are built upon a communication layer called DoDWAN. DoDWAN implements a frugal protocol for disseminating documents network-wide, freeing itself of the presence of independent communication islands through the exploitation of the store-carry-and-forward paradigm. DoDWAN offers a publish/-subscribe API that is used in a service platform (the upper layer of our middleware) for exchanging service advertisements and discovery messages as well as for invocation and response messages.

We briefly described a simulation experiment that shows that this approach allows interesting service satisfaction ratios when compared to the alternative that consists in relying only on one-hop transmissions for service discovery and service invocation.

These preliminary results demonstrate the feasibility of service provision in a disconnected network built upon an opportunistic content-based dissemination middleware like DoDWAN. As future work, we plan to investigate how the fact that a group of hosts provides the same service can be exploited to speed up service discovery as well as service invocation. Indeed, the ability to take benefit of the redundancy of service providers could be crucial in sparsely populated networks.

References

[1] 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.

[2] 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.

[3] R. Chinnici, J. J. Moreau, A. Ryman, and S. Weerawarana. Web Services Description Language (WSDL) Version 2.0. Technical report, W3C, Mar. 2006.

[4] 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.

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

[6] M. Gudgin, M. Hadley, N. Mendelsohn, M. Jean-Jacques, and H. Frystyk Nielsen. SOAP version 1.2. Technical report, W3C, June 2003.

[7] F. Guidec and Y. Mahéo. Opportunistic Content-Based Dissemination in Disconnected Mobile Ad Hoc Networks. In *International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM 2007)*, pages 49–54, Papeete, French Polynesia, Nov. 2007. IEEE Press.

[8] 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.

[9] L. Hogie, P. Bouvry, and F. Guinand. The MADHOC simulator. http://www-lih.univ-lehavre.fr/~hogie/madhoc.

[10] 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.

[11] 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.

[12] C. Liu and J. Kaiser. A Survey of Mobile Ad Hoc network Routing Protocols. Technical report, University of Magdeburg, 2005.

[13] 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.

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

[15] 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.

[16] F. Zhu, M. Mutka, and L. Ni. Service Discovery in Pervasive Computing Environments. *IEEE Pervasive Computing*, 4(4):81–90, Dec. 2005.