A Peer-to-Peer Approach to Asynchronous Data Dissemination in Ad Hoc Networks

Hervé Roussain, Frédéric Guidec

A Peer-to-Peer Approach to Asynchronous Data Dissemination in Ad Hoc Networks

Hervé Roussain, Frédéric Guidec

VALORIA Laboratory – University of South Brittany – France

{Herve.Roussain|Frederic.Guidec}@univ-ubs.fr

Keywords : ubiquitous computing, peer-to-peer computing, ad hoc networking, data dissemination

***Abstract*—This paper presents a model for the dissemination of information in highly dynamic ad hoc networks, such as those composed of mobile communicating devices. This model relies on an asynchronous, peer-to-peer propagation scheme. This approach is expected to help do with station volatility, as it allows communication between devices that are never –or only rarely– active simultaneously in the network. It is also expected to help transport information in a fragmented network, as a station roaming between network islands can provide its neighbors with information it has gathered previously while visiting another part of the network.**

I. Introduction

Nowadays we observe a multiplication of commercial offers for portable digital devices such as laptops and personal digital assistants (PDAs), which can be equipped –or are equipped natively– with Bluetooth or IEEE 802.11 wireless interfaces. For the owners of such devices, the prospect of exchanging information and services with other people without systematically resorting to some infrastructure network should obviously appear as an attractive one. Yet, although many devices are nowadays capable of ad hoc communication, this possibility is still rarely used. There are actually several reasons for this apparent lack of interest in ad hoc communication. Among these reasons is the fact that most of the application software we are nowadays familiar with (electronic mail, discussion forums, file transfer and file sharing, etc.) was designed and implemented so as to be used on traditional workstations connected to infrastructure networks. As a consequence most of the above-mentioned applications are usually implemented based on the client-server paradigm, with which services are actually offered by servers whose accessibility must be guaranteed to client programs. Moreover these applications often rely on chatty protocols, and they require synchronous interactions between clients and servers (a thorough analysis of these problems can be found in [1]).

Mobile communicating devices usually show characteristics that are quite different from those of standard workstations (slower processors, less memory, smaller or no hard disk, ”exotic” user interfaces, low power-budget, etc.). In the same time, ad hoc networking distinguishes itself from infrastructure networking by (usually) lower transmission rates, unstable network topology, transient –and often unpredictable– connectivity, etc. A piece of application code that has been originally designed to run on powerful workstations, and to communicate through a stable, almost fully reliable infrastructure network, can usually not be ported and used as is on a heterogeneous set of mobile devices participating in an ad hoc network.

Our work aims at fostering the design, the implementation, and the deployment of new application services capable of running on mobile, ad hoc communicating devices. In this work we address problems pertaining to the definition of such application services, to their deployment on mobile devices, and to their execution on these devices.

In this paper we present one of the services we have developed along this line. This service is meant to serve as one of the basic building blocks with which higher-level services can later be developed. It is dedicated to the dissemination of pieces of information in a dynamic ad hoc network, that is, a network composed of portable devices that are highly mobile and volatile. Such a network is illustrated in Figure 1, which shows a population of users that own portable devices capable of ad hoc communication. The devices in this net-

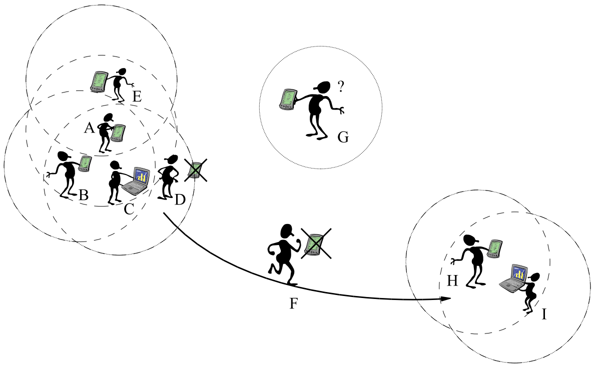

Figure 1. Illustration of a typical dynamic ad hoc network whose nodes are volatile (D and F are here shown in standby mode) and mobile (F is shown "travelling" in the network), which yields transient or chronic network fragmentation ({A, B, C, E}, {G}, and {H, I} constitute –perhaps temporarily– three distinct islands in this example).

work are mobile because the users themselves are mobile. Since the transmission range of a wireless interface is quite limited, the neighborhood of each device is limited (the circles in Figure 1 show the limited transmission range of each device). The neighborhood of a device can be loosely defined as the set of devices it can communicate with. Many current works aim at enlarging this neighborhood by implementing algorithms for dynamic data routing [2] and flooding [3], [4], [5] in ad hoc networks. However, these algorithms rely on the assumption that a connected-path can be established between the sender and the receiver(s) at transmission time. When the density and the geographical distribution of devices is not favorable, the multi-hop network thus obtained usually remains a non-connected one. Figure 1 shows such a fragmented network. Devices that are close enough to communicate together (possibly with multi-hop transmissions) are said to reside on the same "island", while the distances separating two islands prevent any communication between devices located in distinct islands.

Another major characteristic of portable mobile devices is that they are quite volatile, that is, they are frequently switched off and on by their owners (if only to preserve their power-budget). Yet, traditional routing and flooding schemes only allow synchronous interactions between senders and receivers, as they require that the devices sending and receiving data are available simultaneously.

The service discussed in the remainder of this paper permits asynchronous information dissemination in a fragmented ad hoc network whose devices are highly mobile and volatile. This service is both application-oriented and document-oriented. Basically, we propose that any document sent in the network be maintained as long as possible in a local cache by as many devices as possible, so it can remain available for those devices that could not receive it at the time it was sent originally. The underlying idea is that the dissemination of multiple copies of the same document may help do with the volatility of devices, while the mobility of these devices can itself help transport information between islands in a fragmented network.

This propagation model is inherently a stochastic one, as eventual document delivery cannot be guaranteed. Interestingly, though, theoretical studies based on simulations (on variants of this model [6], [7]) show that in many realistic scenarios, the probability that documents actually reach their destination can be quite high.

In any case, in our opinion the main interest of this model is that it makes it possible to disseminate documents among devices that would not have had a chance to receive these documents otherwise. For example, assume that device F in Figure 1 has sent information to device D before leaving the island where devices A to E currently reside. Device D has not received this document since it is presently in suspend mode. However, some of its neighbors may have accepted to store temporarily this document, and D will thus have a chance to obtain the document when it wakes up. Similarly, before leaving the island where devices A to E reside, F may have collected some documents that will be of interest to devices H and I when it reaches their island (and provided F is reactivated when reaching this island, since Figure 1 shows that it is in suspend mode during this transfer).

II. Service definition

Our service for document dissemination relies on the general principle of peer-to-peer interaction: each device in the network can benefit from this service, while contributing itself to supplying this service.

Document addressing. We intentionally adopted a highly flexible, high-level addressing scheme. This scheme makes it possible to address a document equally to a user, to a device, to a group of users, to a group of devices, to a specific service running on a specific device, etc. The underlying idea is that, since a user may for example own several devices, it should be possible to send a document to a user, without imposing that this document be received on one or another device. Similarly, it should be possible to send a document to a group of users (such as all members of a laboratory) without having to specify the identity of each of these users.

Document caching. A cache must be implemented on each device, so that documents can be maintained in this cache when needed. The capacity of the cache can of course be adjusted on each device, depending on the resources available on this device. Moreover local strategies can be defined on each device in order to specify caching modalities for documents. Possible criteria for defining such strategies are document size, type, origin, destination, etc. Any high-level service that is liable to use the caching service –since this service is application-oriented– should be able to specify its own requirements regarding document caching.

Document flow. Each device that participates in the dissemination of documents can play several distinct roles with respect to these documents.

- A device can be considered as the *provider* of a document if this document is stored locally (in its cache), and if it can send this document in the network.
- A device can be considered as the *consumer* of a document if it can receive this document from the network, and either use this document immediately or store this document in its cache (or both).

Depending on circumstances, a device may play only one of these two roles, or both roles simultaneously. The behavior of a device may actually not be the same for all types of documents. For example the caching service may be configured so as to accept and receive only a certain category of documents from the network. Moreover these documents may be received and stored in the cache only for the benefit of other local application services. The device would thus behave as a consumer for certain documents, while refusing to disseminate these documents further. Conversely a device may be configured so as to provide its neighbors with documents produced locally, while refusing to consume similar documents received from the network.

Besides playing the role of a producer or consumer for a category of documents, a device can behave either proactively or reactively (or both) with respect to each of these roles.

A device that plays the role of a document provider can behave *proactively* by sending spontaneously this document in the network. Spontaneous transmission can for example occur when a document has just been produced locally, or when a document in the cache must be sent several times (periodically or sporadically). The latter strategy may help reach recipients that are suspected not to have received the document during former transmissions.

A document provider may also behave *reactively* by sending a document in the network after this document has been explicitly requested. It can of course show a mixed behavior, sending for example one document periodically (with a rather long period so as not to load the network too much), and replying immediately to explicit requests for this document.

Similarly, a device that plays the role of a document consumer can behave either proactively or reactively, or show both kinds of behavior simultaneously. A document consumer can behave proactively by sending requests for this document in the network (thus soliciting a reactive behavior from devices that possess a copy of this document). It can also behave reactively by receiving a document from the network, and consuming this document even if it has not been explicitly requested before.

The document caching service is not itself responsible for deciding how it should behave with respect to one or another category of documents. Instead, any higher-level service that requires that certain documents be sent, received, and forwarded through the ad hoc network is expected to interact directly with the caching service in order to specify its own requirements with respect to these documents.

Document structure. Each document can be associated a document descriptor. The descriptor of a document is meant to provide information about the document (type, author name, keywords, content, etc.). A document may encapsulate its own descriptor, but the descriptor can also be handled separately (which means it can for example be transmitted, edited, stored, and displayed separately).

When an application-level document must be sent in the network, it must be encapsulated in a *transfer document*, whose descriptor specifies transmission parameters for this document.

```
<transfer-descriptor
   document-id="fb0097820f0b371"
   origin="principal:Pierre Dupont"
   destination="principal:VALORIA Team"
   type="service/email"
   number-of-hops="5"
   date="Jan 30 08:26:32 CET 2004"
   lifetime="12:00:00"
   advertisement-period="00:20:00"
/>
```

Figure 2. Descriptor of a document exchanged within the context of an email messenging service.

Figure 2 shows a typical XML-formatted transfer descriptor. The value of attribute *document-id* in a descriptor should be unique. This condition makes it possible for a device receiving a document from the network to verify if a copy of this document is already available in its local cache (in which case the document received is a duplicate that can be silently discarded).

The document descriptor in Figure 2 also specifies that this document has been sent by user Pierre Dupont (attribute *origin*) and that it is addressed to all members of the VALORIA laboratory (attribute *destination*). The attribute *type* specifies that the document was sent –and should thus be received and consumed– by an email service agent. The attribute *number-of-hops* plays approximately the same role as the field TTL *(Time-To-Live)* in IP packets. It helps preventing that a document propagates eternally in the network. The attribute *date* specifies the date when the document was originally sent in the network, and the attribute *lifetime* specifies that this document should be considered as being valid for only twelve hours after this date. The last attribute *advertisement-period* indicates that once the document has been put in the local cache of a device, this device may periodically announce that this document is available, with a periodicity of 20 minutes.

It is worth mentioning that a device that considers that its cache needs to be cleaned up has no obligation to maintain a document in this cache until it becomes obsolete. Likewise, a device has no obligation to announce that it owns a document, even if the descriptor of the document suggests an announcement period. In any case, attributes specifying a document's lifetime or suggesting a period for announcing the availability of this document are meant to serve as guidelines about how this document should be handled by devices. The actual behavior of each device with respect to the documents it maintains in its cache can be guided by such suggestions, but it can also conform to default strategies defined by higher-level services, or according to user-defined global strategies applying equally to all documents.

Advertisement and request documents. Specific kinds of documents are meant to contribute to document dissemination in the network, by allowing interactions between neighboring devices.

An "advertisement document" can be sent by a device to announce that it owns one or several documents in its cache, and that it can provide any of these documents on demand. An advertisement document is thus a transfer document in which attribute *type* in the descriptor characterizes a document sent by the caching service implemented on a device. Its payload is composed of one or several document descriptors, corresponding to the descriptors of the documents whose availability is being announced.

```
<transfer-descriptor
   document-id="fb55271h0f03hg5"
   destination="principal:VALORIA Team"
   number-of-hops="1"
   type="service/cache/descriptor/advertisement"
   origin="device:rama"
/>
<descriptor-list>
   <transfer-descriptor
      ...  />
   <transfer-descriptor
      ...  />
</descriptor-list>
```

Figure 3. Example of a document announcing the availability of two documents (whose descriptors are not detailed here).

An example of an advertisement document is shown in Figure 3. In this example a device called *rama* informs its neighbors that it owns a number of documents. In this particular case all neighbors are not supposed to pay attention to this advertisement, but only those that are interested in documents pertaining to the VALORIA laboratory (see attribute *destination*). Moreover this advertisement concerns two distinct documents, whose descriptors (not detailed in the figure) are encapsulated in the advertisement document.

A "request document" can be used by a device to ask for the transmission of a document, or that of several documents. A device can send a request after receiving an advertisement indicating that a specific document is available in the neighborhood. A request can also be sent spontaneously by a device that wishes to discover if a certain document is indeed available somewhere in the neighborhood. A request can be addressed specifically to a given device (for example after an advertisement has been received from this device), or it can be sent to all or part of the devices in the neighborhood. A request can ask for the direct transmission of a document, or it can ask that only the descriptor of this document be transmitted.

A request document is thus a transfer document in which attribute *type* in the descriptor characterizes a request addressed to the caching service implemented on the destination device(s). This attribute additionally specifies if this document requests that full documents be transmitted, or only their descriptors. The payload of a request document is composed of one or several descriptor patterns. The structure of a descriptor pattern compares with that of a descriptor, but for all or part of the attributes that can appear in a document descriptor, it specifies a regular expression to be applied to the corresponding attribute. A device receiving a descriptor pattern can thus use this pattern to examine the descriptors of the documents it maintains in its cache, and to decide which of these descriptors match the pattern. The device can then either send the selected documents, or only the descriptors of these documents, depending on the kind of reply requested.

An example of a request document is shown in Figure 4. In this example the request has been sent by the device called *rama*, and it has been addressed to all other devices in its close neighborhood (the number of hops is here set to 1, which precludes any propagation of the request beyond the devices that are direct neighbors of *rama*). The request asks that devices receiving this request send in return the descriptors of the documents they maintain in their cache, but only if these documents are addressed to user Pierre Dupont, and only if they were originally sent in the network after the date specified. Moreover, the attribute *reply-to* indicates that replies to this request should be addressed to all devices that take an interest in traffic pertaining to user Pierre Dupont, rather than being addressed to device *rama* only (remember that Pierre Dupont may own several devices, and that other users may have configured their devices so that they relay and forward documents addressed to Pierre Dupont). This example shows that a device behaving as a proactive consumer for some kind of document can formulate requests in such a way that its neighborhood can benefit from the transmission of documents it will itself have solicited.

```
<transfer-descriptor
   document-id="fb56971h0f0b455"
   destination="device:*"
   number-of-hops="1"
   type="service/cache/descriptor-request"
   origin="device:rama"
   reply-to="principal:Pierre Dupont"
/>
<descriptor-pattern
   destination="principal:Pierre Dupont"
   after="Jan 30 08:00:00"
/>
```

Figure 4. Example of a request asking for the advertisement of all documents addressed to user Pierre Dupont after the specified date and time.

As a general rule, advertisement and request documents make it possible to implement on each device some procedures ensuring the proactive and reactive behaviors mentioned earlier in this section. A device behaving as a proactive consumer can send its neighbors requests in order to ask for the direct transmission of specific documents, or in order to discover that some of its neighbors own documents it may be interested in. After receiving an advertisement for a document it is interested in, a reactive consumer can ask for the immediate transmission of this document. Similarly, a device behaving as a proactive producer can send advertisement documents in order to provide its neighbors with descriptors of the documents it can send them on demand.

III. Implementation details

The model presented in the former section has been implemented in Java. Documents and document descriptors are thus reified as standard Java objects, which can be sent in the network either as serialized Java objects, or as XML-formatted documents. In the current implementation, document dissemination is performed by broadcasting UDP datagrams. We thus assume, for the time being, that each document is small enough to fit in a single datagram. In the future we consider implementing a complementary service for document segmentation and reassembly.

The code we developed can be deployed equally on a multi-hop or single-hop network. It simply requires that a device can send a document to neighboring devices. In a multi-hop network (with data flooding), the neighborhood of a device is composed of all devices that reside on the same island. In a single-hop network, it is composed of those devices that can interact at link level with the sending device.

Several application-level services are also being developed in our laboratory. These services are designed in such a way that they can rely on the asynchronous document dissemination service discussed in this article. One of these services is a general messenging service, which features functionalities that are traditionally offered separately in the electronic mail service and in the newsgroup (or discussion forum) service. Indeed, messenging with this service can be performed either user-wise or group-wise (that is, a message can be sent equally to a certain user, or to a certain community of users).

Another application-level service under development is a presence announcement service. It relies on the dissemination of specific documents announcing periodically or sporadically the presence of a person, or that of a device. With this service a device can provide its owner with an up-to-date graphical representation of all neighboring entities (users and devices) that have announced their presence.

Since documents exchanged in an ad hoc network are not necessarily meant to be received and consulted by all users of the network, we have also designed a cryptographic service that makes it possible to encrypt and to sign any kind of transfer documents. This service can be used liberally by higher-level services, such as the above-mentioned messenging and presence announcement services, whenever confidentiality or trust are needed during document dissemination.

IV. Related work

Our approach to document dissemination in ad hoc networks obviously compares with that proposed by Vahdat and Becker in [6], where they introduce the concept of *Epidemic Routing*. In their model messages are buffered in mobile hosts, and random pair-wise exchanges of messages among these hosts allow eventual message delivery in partially-connected ad hoc networks. Although Vahdat and Becker are mostly concerned with unicast transmissions in their paper, they suggest that Epidemic Routing is appropriate for supporting multicast traffic as well. A variation of Epidemic Routing has been presented in [7], where the authors introduce probabilistic heuristics in order to augment the chances that a message is indeed routed toward its actual destination.

Our approach also compares with that of IRTF[1]'s group working on delay-tolerant networks (DTNRG: *Delay-Tolerant Network Research Group*[2]). This group focuses on the deployment of networks in extreme environments in which end-to-end data connectivity cannot be guaranteed. It has defined a new network architecture relying on the general principle of message switching in store-and-forward mode. This architecture can itself be implemented above standard protocol stacks. The approach fostered by the DTNRG consists in transporting pieces of information as so-called *bundles*, and implementing *bundle forwarders* that are capable of storing messages (or bundles) before they can be sent again [1].

The mechanisms we propose for announcing the availability of a document, for requesting a

[1] IRTF: *Internet Research Task Force* (http://www.irtf.org).
[2] http://www.dtnrg.org

document, and for seeking a document obviously show similitudes with those that have otherwise been proposed to perform service advertisement and discovery in ad hoc networks (a good survey of discovery protocols can be found in [8]). Indeed, we consider that the description of a service (offered or desired) simply constitutes a particular kind of information, which can therefore be encapsulated in some type of document. In our opinion, service advertisement and discovery should thus be perceived as particular applications whose implementation could rely on the more general document dissemination service we have designed.

V. Conclusion

The model presented in this paper permits the asynchronous dissemination of documents in highly dynamic ad hoc networks, such as those composed of mobile communicating devices. This model proposes an asynchronous, peer-to-peer, document-oriented propagation scheme. It requires that each document received by a device be maintained in a local cache, so it can be sent again in the network either spontaneously, or after a request for this document has been received from another device. This approach helps do with the volatility of devices, since it permits asynchronous communications between devices that are never – or only rarely– active simultaneously in the network. It also permits information dissemination in a fragmented network, by taking advantage of the mobility of devices.

A Java-based implementation of this model is now available. The asynchronous document dissemination service thus obtained is expected to serve as a building block during the development of higher-level application services. A messenging service and a service for presence announcement are two such high-level services, whose implementation is now in progress in our laboratory.

Acknowledgements. This work is supported by the French "Conseil Régional de Bretagne" under contract B/1042/2002/012/MASC.

References

[1] K. Fall, "A Delay-Tolerant Architecture for Challenged Internets," Tech. Rep. IRB-TR-03-003, Intel Research, Berkeley, Feb. 2003.

[2] E. M. Royer and C.-K. Toh, "A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks," *IEEE Personal Communications Magazine*, pp. 46–55, Apr. 1999.

[3] P. Poupyrev, M. Kosuga, and P. Davis, "Analysis of Wireless Message Broadcast in Large Ad Hoc Networks of PDAs," in *Proceedings of the Fourth IEEE conference on Mobile and Wireless Communications Networks*, pp. 299–303, 2002.

[4] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint Relaying; an Efficient Technique for Flooding in Mobile Wireless Networks," Rapport de recherche 3898, INRIA, Mar. 2000.

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

[6] A. Vahdat and D. Becker, "Epidemic Routing for Partially-Connected Ad Hoc Networks," Tech. Rep. CS-2000-06, UCSD, July 2000.

[7] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic Routing in Intermittently Connected Networks," in *Proceedings of the Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2003)*, June 2003.

[8] A. Rakotonirainy and G. Groves, "Resource Discovery for Pervasive Environments," in *Proceedings of the 4th International Symposium on Distributed Objects and Applications*, pp. 866–883, Oct. 2002.