Asynchronous Document Dissemination in Dynamic Ad Hoc Networks

Frédéric Guidec, Hervé Roussain

Asynchronous Document Dissemination in Dynamic Ad Hoc Networks

Frédéric Guidec, Hervé Roussain

VALORIA Laboratory – University of South Brittany – France
{Frederic.Guidec|Herve.Roussain}@univ-ubs.fr

Abstract This paper presents a document-oriented model for information dissemination in dynamic ad hoc networks, such as those composed of highly mobile and volatile communicating devices (e.g. laptops and PDAs). This model relies on an asynchronous, peer-to-peer propagation scheme where documents can be cached on intermediate devices, and be later sent again –either spontaneously or on demand– in the network.

1 Introduction

Today most laptops and personal digital assistants (PDAs) feature wireless interfaces, many of which are capable of ad hoc communication. Our work aims at fostering the design, the implementation, and the deployment of application services capable of running specifically on devices participating in a dynamic ad hoc network, that is, a network in which nodes are highly mobile and volatile. Node mobility in a dynamic network is the consequence of the fact that devices are carried by users, which are themselves mobile. Node volatility results from the fact that, since mobile devices have a low power-budget, they are frequently switched off and on by their owners. An additional problem with dynamic ad hoc networks is that in many realistic scenarios such networks present themselves as disconnected networks. As a consequence, direct transmissions between any pair of devices is not always feasible, as such transmissions require that both devices are active simultaneously in the network, and that a connected-path can be established between these devices at transmission time.

The problem of delivering messages in disconnected ad hoc networks has been approached several times and following different lines in the past few years. For example, a new network architecture relying on the general principle of message switching in store-and-forward mode has been proposed in [1]. With this approach pieces of information are transported as so-called *bundles* between *bundle forwarders*, which are capable of storing messages (or bundles) before they can be sent again in the network.

With *Epidemic Routing* [6,4,3], messages are buffered in mobile hosts, and random pairwise exchanges of messages among these hosts are expected to allow eventual message delivery in partially-connected networks.

The service we present in this paper compares with the models proposed in the abovementioned papers. However it can be observed that these papers mostly address the

problem of message delivery in disconnected networks from a theoretical viewpoint: they propose new algorithms and heuristics for delivering messages in such networks, and they report the results of simulations that are meant to demonstrate how these algorithms should perform in realistic conditions. In contrast, our approach is more practical, since it consists in actually implementing a service for document dissemination in ad hoc networks, and then using this service as a building block with which application-level services can be developed and tested in realistic experimental conditions.

2 Service overview

The general architecture of the service we propose is shown in Figure 1. This service is not meant to be used directly by end-users. Instead it is meant to serve as one of the basic building blocks with which higher-level services can later be developed. Moreover this service is 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. Besides providing a caching system where documents can be maintained in mobile devices, our service also provides facilities for document advertisement, document discovery, and document transport between neighboring devices. For example, a device can sporadically or periodically notify its neighbors about all or part of the documents stored in its cache. It can also look for specific documents in its neighborhood, and either push documents toward –or pull documents from– its neighbors.

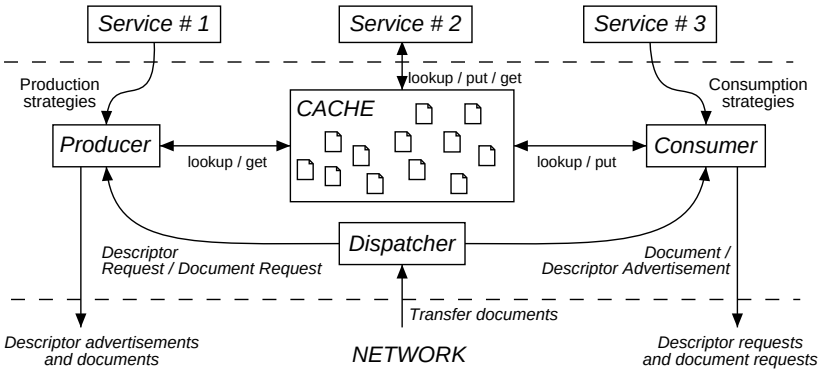

Figure 1. Illustration of the document caching service.

Structure of documents. In the model our service relies on, each document can be associated a document descriptor, which provides information about its type, author,

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 itself be encapsulated in a *transfer document*, whose descriptor specifies transmission parameters for this document, such as its type, origin, and destination, as well as indications about how long and how far this document should propagate in the network. Examples of document descriptors can be found in [5].

Caching documents. Once a document has been received by a device, it is expected that this document be stored for some time on this device, and possibly sent again later in the network. Each device thus maintains a cache, whose capacity can of course be adjusted depending on the resources available locally. Local strategies can additionally 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, lifetime, etc. The caching service is not itself responsible for deciding how it should behave with respect to documents. Instead, it provides interfaces (not detailed in this paper) with which higher-level services can specify strategies regarding how one or another category of document should be managed locally. Moreover, attributes found in a document's descriptor can help determine how this document should be managed by the caching service. For example, attributes may indicate how long a document should be considered as being valid in the network, and how often the availability of this document should be announced in the network.

Document producers and consumers. Each device that participates in the dissemination of documents can play several distinct roles with respect to these documents. A device is 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. Symmetrically, a device is considered as being a potential *consumer* for 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. It 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.

Advertisement and request documents. Specific kinds of documents have been defined in order to allow the advertisement, discovery, and transmission of documents between neighboring devices. For example, 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 special kind of transfer document whose payload is composed of one or several document descriptors, corresponding to the descriptors of the documents whose availability is being announced.

Another special kind of document, called a ”request document”, can likewise be sent by a device to ask for the transmission of a document, or that of several documents. 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. 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. Selected documents can then be sent in the network.

3 Implementation details and ongoing work

The service for asynchronous document dissemination presented in the former section has been implemented in Java. Documents and document descriptors are also reified as standard Java objects. They can be transported in the network either as serialized Java objects, or as XML-formatted documents (examples can be found in [5]). It is worth mentioning that the code we developed can be deployed equally on a single-hop network, or on a multi-hop network relying on algorithms for dynamic routing and flooding.

The development of several application-level services is also under way in our laboratory. These services all rely on the facilities offered by the document dissemination service, but each of them defines its own strategy regarding what documents must be disseminated, and in what conditions. Among these application-level services are a peer-to-peer messaging service, a presence announcement service, and a service for the distribution and the deployment of software packages. Details about the latter service can be found in [2].

4 Conclusion

The service presented in this paper permits the asynchronous dissemination of documents in dynamic ad hoc networks, such as those composed of highly mobile and volatile communicating devices. It proposes an asynchronous, peer-to-peer, document-oriented propagation model, where each document received by a device can be maintained in a local cache in this device, so it can later be sent again in the network, either spontaneously, or after a request for this document has been received from another device. This approach is expected to help do with the volatility of devices, since it permits that documents reach devices that are only active sporadically in the network. It is also expected to permit information dissemination in a fragmented network, taking advantage of the mobility of devices which can serve as carriers between disconnected parts of the network.

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

References

1. Kevin Fall. A Delay-Tolerant Architecture for Challenged Internets. Technical Report IRB-TR-03-003, Intel Research, Berkeley, February 2003.
2. Nicolas Le Sommer and Hervé Roussain. JASON: an Open Platform for Discovering, Delivering and Hosting Applications in Mobile Ad Hoc Networks. In *International Conference on Pervasive Computing and Communications (PCC'04)*, pages 714–720, Las Vegas, Nevada, USA, June 2004.
3. Q. Li and D. Rus. Sending Messages to Mobile Users in Disconnected Ad Hoc Wireless Networks. In *Proceedings of the Sixth ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom 2000)*, pages 44–55, August 2000.
4. Anders Lindgren, Avri Doria, and Olov 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.
5. Hervé Roussain and Frédéric Guidec. A Peer-to-Peer Approach to Asynchronous Data Dissemination in Ad Hoc Networks. In *International Conference on Pervasive Computing and Communications (PCC'04)*, pages 799–805, Las Vegas, Nevada, USA, June 2004.
6. Amin Vahdat and David Becker. Epidemic Routing for Partially-Connected Ad Hoc Networks. Technical Report CS-2000-06, UCSD, July 2000.