



HAL
open science

Vers une ubiquité d'accès aux services dans les réseaux mobiles ad hoc

Nicolas Le Sommer

► **To cite this version:**

Nicolas Le Sommer. Vers une ubiquité d'accès aux services dans les réseaux mobiles ad hoc. 7e conférence internationale sur les nouvelles technologies de la répartition (NOTERE'07), Jun 2007, Marrakech, Maroc. pp.1-12. hal-00498366

HAL Id: hal-00498366

<https://hal.science/hal-00498366v1>

Submitted on 7 Jul 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Vers une ubiquité d'accès aux services dans les réseaux mobiles ad hoc

Nicolas Le Sommer

*Laboratoire Valoria, Université de Bretagne Sud
Nicolas.Le-Sommer@univ-ubs.fr*

RÉSUMÉ. Depuis peu, les réseaux mobiles ad hoc font l'objet d'une attention particulière de par les perspectives qu'ils offrent en termes de services. En effet, au delà des services envisagés jusqu'à présent pour des opérations militaires et/ou de secours, des services utiles dans le monde civil et dans la vie quotidienne sont désormais considérés. La prestation de service est une question cruciale pour le succès de ce nouveau paradigme informatique. Dans cet article, nous présentons une nouvelle approche pour la prestation de service dans des réseaux mobiles ad hoc connectés par intermittence, ainsi qu'une plate-forme intergicielle mettant en œuvre cette approche. Cette approche repose sur un paradigme de communication asynchrone qui offre une gestion contextuelle et orientée contenu des messages utilisés dans la prestation des services.

ABSTRACT. Mobile ad hoc networks have been recently receiving much attention due to the important prospects they offer in application terms. Indeed, beyond the application services envisioned so far in emergency and military scenarios, civilian mobile ad hoc services usable in everyday life are considered from now. Service provision is a critical issue to the success of this new computing paradigm. In this paper, we present a novel approach for the service provision in intermittently connected mobile ad hoc networks, as well as a middleware platform implementing this approach. This approach relies on an asynchronous communication paradigm supporting a content and context based management of messages used in the service provision.

MOTS-CLÉS : plates-formes de services, informatique diffuse

KEYWORDS: service-oriented middleware, pervasive computing

1. Introduction

La prolifération croissante de terminaux mobiles équipés d'interfaces de communication sans-fils de type IEEE 802.11 ou IEEE 802.15 et présentant de réelles capacités de calcul et de stockage ouvre d'intéressantes perspectives en terme de services pour l'informatique diffuse et pour l'intelligence ambiante. Ces équipements devraient en effet favoriser l'émergence de réseaux mobiles ad hoc pair-à-pair de taille importante, réseaux qui trouvaient jusqu'à lors une application dans le domaine militaire et dans le domaine des secours du fait de leur facilité de déploiement.

La plupart des travaux visant à offrir un accès à des services dans ce type de réseaux font des suppositions vis-à-vis des communications. Ils considèrent par exemple qu'une communication entre deux équipements ne peut être établie que si ceux-ci sont présents simultanément dans le réseau et si une route entre eux peut être définie. Faire de telles hypothèses revient à considérer que les communications se font de manière synchrone et en mode point-à-point, et qu'elles ne sont possibles que dans un réseau routé. De nombreux travaux ont ainsi pour objectifs de définir et de mettre en œuvre des algorithmes de routage dans de tels réseaux afin de permettre la réutilisation des technologies de découverte et d'invocation de services conçues pour les réseaux filaires (e.g. Jini, UPnP, SLP, HTTP/SOAP). Pourtant, lorsque l'on observe ces réseaux, on s'aperçoit que leur topologie est très dynamique, et qu'ils se trouvent souvent partitionnés en plusieurs îlots de communication disjoints. De fait, les suppositions précédentes se trouvent remises en question par un certain nombre d'articles récents traitant des communications dans les réseaux déconnectés, partiellement connectés ou connectés de manière intermittente. Certains de ces articles se focalisent sur le routage point-à-point dans les réseaux tolérants les délais (Li *et al.*, 2000, Shah *et al.*, 2003, Musolesi *et al.*, 2004), d'autres au contraire abordent les problèmes d'inondation et de dissémination de messages dans de tels réseaux (Harras *et al.*, 2005, Musolesi *et al.*, 2005b). Certains de ces articles se focalisent en outre sur la recherche et le contrôle de motifs de mobilités (Zhao *et al.*, 2004, Li *et al.*, 2000), d'autres articles a contrario ne font aucune supposition vis-à-vis de cette mobilité et proposent d'exploiter la redondance d'information afin d'améliorer la fiabilité des transmissions. Dans cette catégorie, il est souvent proposé de s'appuyer sur une approche épidémique (Vahdat *et al.*, 2000, Eugster *et al.*, 2004, Musolesi *et al.*, 2005b) ou probabiliste (Sasson *et al.*, 2002) pour estimer les modalités de propagation de l'information. Cependant, aucun de ces travaux n'aborde spécifiquement le problème de la prestation de service dans ce type de réseaux.

Le manque de solutions nous a donc conduit à étudier les fonctionnalités pouvant permettre d'offrir une certaine ubiquité d'accès aux services dans ces réseaux, à proposer une approche et à valider celle-ci en concevant un cadre de conception pour la découverte et l'invocation de services, ainsi qu'une boîte à outils contenant des implantations prédéfinies des fonctionnalités de ce cadre de conception. Ces différents points sont présentés respectivement dans les paragraphes 2, 3 et 4. Des travaux apparentés au nôtre sont présentés dans le paragraphe 5. Le paragraphe 6 conclut cet article et présente quelques travaux futurs.

2. Proposition pour une ubiquité d'accès aux services dans des réseaux mobiles ad hoc connectés par intermittence

Dans ce paragraphe, nous illustrons à travers un scénario l'ubiquité d'accès aux services recherchée pour les réseaux mobiles ad hoc partiellement connectés, connectés par intermittence ou déconnectés que nous considérons. Nous y présentons également une proposition pour tendre vers cette ubiquité.

2.1. Scénario

Dans le scénario présenté dans la figure 1, nous considérons un réseau mobile ad hoc pair-à-pair composé de terminaux embarqués dans des véhicules (e.g. GPS) et d'équipements déployés dans l'environnement, ces équipements étant tous dotés d'interfaces de communication radio à courte portée (e.g. IEEE 802.11). En outre, nous nous intéressons aux modalités d'accès à des services d'information routière. Ces services, qui sont susceptibles d'être déployés de manière temporaire par des équipes réalisant des travaux de voiries ou par des équipes de secours durant les accidents, sont conçus pour être découverts et être invoqués par les systèmes embarqués dans les véhicules (e.g. GPS) en vue d'informer les conducteurs des éventuelles difficultés de trafic.

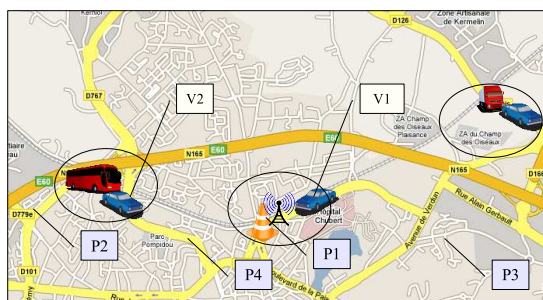


Figure 1. Exemple de réseau mobile ad hoc pair-à-pair de grande taille.

Afin de mettre en évidence les problèmes inhérents à l'accès aux services dans ce type de réseaux, et afin de présenter l'ubiquité d'accès aux services à laquelle nous souhaitons prétendre, supposons qu'une équipe réalisant des travaux de voirie en P_1 (voir Figure 1) a installé un équipement offrant un accès à un service comparable à ceux mentionnés ci-dessus. Considérons en outre qu'à un instant T donné, seul un véhicule, le véhicule V_1 , soit à proximité de cet équipement. A ce moment précis, dans ce réseau partitionné en plusieurs îlots de communication distincts, seul V_1 , qui se rend au point P_2 , est en mesure de découvrir et d'invoquer ce service. Dans un schéma de communication sans fils traditionnel, le véhicule V_2 , qui se rend au point P_3 , doit attendre d'être à proximité du point P_1 pour pouvoir découvrir et invoquer à son tour ce service. Il nous semble souhaitable d'aller au delà de ce schéma de communication

traditionnel et de permettre à ce véhicule de pouvoir découvrir et invoquer ce service avant d'atteindre le point P_1 , offrant ainsi au conducteur la possibilité de changer de direction en cas de difficultés de trafic. Ainsi selon nous, lorsque les véhicules V_1 et V_2 se trouveront tous deux au point P_4 à un l'instant $T + \Delta t_1$, le véhicule V_1 devrait être capable d'annoncer spontanément la disponibilité du service considéré, ou devrait être en mesure de répondre à une requête de découverte de service émise par V_2 concernant un tel service. En outre, lorsque le véhicule V_2 invoquera au temps $T + \Delta t_2$ le service ainsi découvert avec une requête comparable à celle émise par V_1 au temps T , le véhicule V_1 devrait être en mesure de retourner la réponse qu'il a lui même obtenu du service au temps T , si cette réponse est encore valide et s'il se trouve toujours à proximité de V_2 . Un tel processus de découverte et d'invocation de service devrait se reproduire de manière récurrente dans le réseau au gré de la mobilité des véhicules.

Outre les problèmes inhérents aux communications dans les réseaux ad hoc (e.g. la limitation du trafic généré par les noeuds), la découverte et l'invocation de services exige de disposer

- de disposer d'un mécanisme d'adressage flexible afin de pouvoir adresser un message à un noeud particulier, à un groupe de noeuds, ou à tous les noeuds à la fois afin de réduire le nombre de messages disseminés dans le réseau.
- de supporter des communications asynchrones reposant sur le principe du *store-and-forward* afin de faire face à la fragmentation du réseau, ainsi qu'à la volatilité et à la mobilité des équipements.
- de supporter une gestion des messages orientée contenu afin de réduire la latence lors du processus de découverte et d'invocation des services, et pour aider les noeuds à décider quels messages ils acceptent de stocker localement.
- de prendre en compte des propriétés spatiales, temporelles et contextuelles afin d'améliorer la prestation de service et de réduire la propagation des messages dans le réseau.

2.2. Proposition

Dans la suite de ce paragraphe, nous présentons notre proposition pour répondre aux problèmes évoqués précédemment en formalisant la notion de service, de fournisseur de services (i.e. en caractérisant les noeuds dans un réseau mobile ad hoc pair-à-pair), ainsi que les messages échangés dans le processus de découverte et d'invocation des services.

Un service est défini traditionnellement par son interface et par ses propriétés non-fonctionnelles. Un service S_i implémentant une interface I_i et ayant un ensemble NP_i de n propriétés non-fonctionnelles pourra donc être décrit de la manière suivante :

$$S_i = \langle I_i, NP_i = \{np_j\} / j \in [1..n] \rangle$$

Un service S_j pourra être considéré comme similaire à un service S_i si et seulement si

$$I_i = I_j \text{ et } \forall x \in NP_i, x \notin NP_j$$

Dans notre approche, un fournisseur de service sera quant à lui caractérisé par un identifiant unique (e.g. un numéro IMEI, une adresse MAC), par ses propriétés non-fonctionnelles (e.g. sa localité, ses caractéristiques physiques), par les services qu'il met à disposition des autres noeuds du réseau, par les services qu'il est en mesure de fournir « virtuellement » grâce aux messages qu'il a stockés localement, et par les groupes de communication auxquels il appartient. Un fournisseur de service ayant comme identifiant φ , peut donc être défini comme suit :

$$D_\varphi = \langle \varphi, LS_\varphi = \{S_i\}/i \in [1..p], VS_\varphi = \{S_j\}/j \in [1..q] \\ CP_\varphi = \{cp_k\}/k \in [1..r], G_\varphi = \{g_l\}/l \in [1..t] \rangle$$

où LS_φ est l'ensemble des services locaux fournis par le terminal, VS_φ l'ensemble des services fournis virtuellement par le terminal, CP_φ les propriétés contextuelles exhibées par celui-ci et G_φ les groupes de communication auxquels appartient le terminal.

Les différents messages échangés par les noeuds du réseau, à savoir les annonces de services, les requêtes de découverte de services, les requêtes d'invocation et les réponses de services, présentent tous des propriétés communes. Chacun de ces messages, identifié de manière unique, définit en effet l'identité de l'émetteur et du destinataire, ainsi que des propriétés temporelles, spatiales et contextuelles permettant de déterminer la validité et la pertinence des messages, et d'en contrôler la propagation dans le réseau. Chaque message intègre également des propriétés caractérisant leur contenu permettant ainsi de les traiter aisément en fonction de leur contenu. Plus formellement un message sera donc défini de la manière suivante :

$$M_i = \langle i, O_i, D_i, C_i, PC_i, T_i, L_i, H_i, CP_i \rangle$$

où O_i , D_i , C_i et PC_i représentent respectivement l'origine, la destination, le contenu et les propriétés caractérisant le contenu du message M_i . Le champ D_i prendra des valeurs différentes suivant les modalités d'adressage souhaitées (e.g. identifiant d'un noeud, nom d'un groupe de communication). Le champ O_i prendra quant à lui la valeur de l'identifiant du terminal émettant le message. Par ailleurs, dans le message M_i , les champs T_i , L_i et H_i indiquent respectivement la date d'émission, la durée de vie supposée du message, et le nombre de maximum de sauts pouvant être effectués par le message dans le réseau. Enfin, CP_i représente les propriétés contextuelles spécifiées par l'émetteur de message (i.e. O_i) à l'instant T . En outre, les messages retournés en réponse à un autre message incluent également des propriétés permettant d'établir une corrélation avec ce dernier. Ainsi, une annonce de service retournée en réponse à une requête de découverte de service inclura l'identifiant et les propriétés caractérisant le contenu de celle-ci.

Le contenu C_i d'un message M_i différera selon que celui-ci représente une annonce, une requête de découverte, une requête d'invocation ou une réponse de service. Par exemple, le contenu d'un message d'annonce de service sera une description du service considéré conforme à la spécification de service formulée précédemment. Pour une requête de découverte de service, un contenu identique à celui d'une annonce de service pourra être utilisé. En effet, seule la sémantique et le traitement du message différeront (les informations spécifiées ne seront pas considérées comme étant les caractéristiques d'un service disponible, mais comme étant les propriétés qui devront être satisfaites par le service recherché). Le contenu C_i d'une requête d'invocation de service M_i pourra quant à lui être défini de la manière suivante :

$$C_i = \langle S_x, M_y, P_z = \{p_l\} l \in [1..k] \rangle$$

où M_y est la méthode du service S_x qui doit être invoquée avec l'ensemble des paramètres P_z .

Les propriétés spatiales, temporelles, et dans une moindre mesure les propriétés contextuelles, associées aux messages sont utilisées par les protocoles de communication (Musolesi *et al.*, 2005a, Chen, 2001) afin d'améliorer l'acheminement des messages dans les réseaux mobiles ad hoc. Ces propriétés ne sont pas rendues accessibles aux couches supérieures (e.g. couche applicative). Pourtant dans ce type de réseaux, celles-ci constituent des informations pertinentes pour les éléments assurant la découverte et l'invocation des services. Elles devraient en effet leur permettre de mieux choisir les prestataires de service, d'estimer une probabilité d'accès aux services distants, de permettre aux clients et aux prestataires de services de définir la durée de validité et la portée de leurs requêtes et de leurs réponses, etc. En outre, ces éléments sont plus à même d'exploiter cette information que les protocoles de communications, qui eux ne sont pas en mesure de corrélérer ces informations avec le contenu des messages. De plus en déléguant le traitement de ces propriétés à une (ou plusieurs) entité(s) externes, les protocoles de communication seront plus simples à mettre en œuvre et plus efficaces. Partant de cette considération, nous avons conçu un cadre de conception pour la découverte et l'invocation de services qui offre des fonctionnalités capables de traiter ces propriétés. Ce cadre de conception est présenté dans le paragraphe suivant.

3. Un cadre de conception pour la découverte et la délivrance de service

Le cœur de notre cadre de conception est fondé sur la spécification OSGi. Dans cette spécification, les services sont caractérisés par l'interface Java qu'ils implémentent et par un ensemble de propriétés non fonctionnelles. L'intergiciel OSGi offre à chaque service local le moyen de découvrir les services et ressources (e.g. paquets logiciels) déployés localement *via* leur contexte d'exécution. La spécification OSGi ne permettant pas une découverte et une invocation des services distants satisfaisant aux spécifications proposées dans le paragraphe précédent, nous avons conçu et développé un cadre de conception et une boîte à outils pour la découverte et l'in-

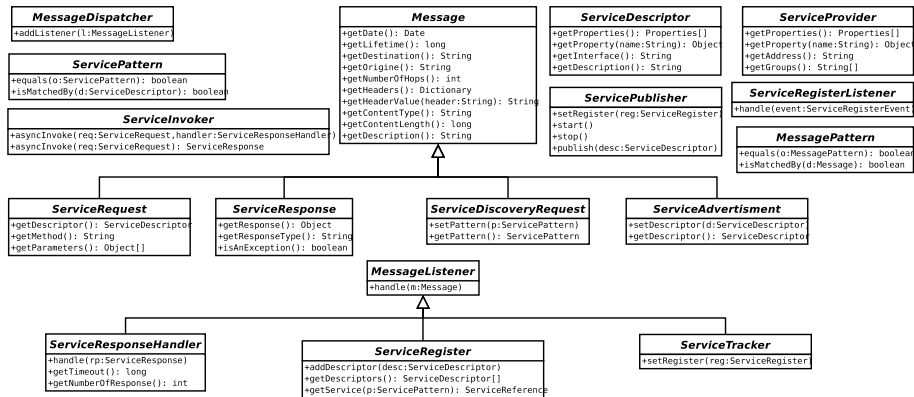


Figure 2. Représentation UML partielle du cadre de conception.

vocation de service. Ce cadre de conception a été utilisé pour mettre en œuvre un intergiciel avec lequel nous étudions la prestation de service dans des réseaux mobiles ad hoc.

3.1. La découverte et la publication des services

Plusieurs éléments du cadre de conception interviennent dans le processus de découverte et de publication des services. Les objets implémentant l'interface *ServiceDescriptor* permettent de décrire les services en précisant l'interface Java qu'ils implémentent, ainsi que leurs propriétés non-fonctionnelles. Ces descripteurs peuvent être inclus dans des objets de type *ServiceAdvertisement* afin d'être publiés dans le réseau via un objet implémentant l'interface *ServicePublisher*. Les objets de type *ServiceAdvertisement* publiés dans le réseau peuvent quant à eux être traités par un objet implémentant l'interface *ServiceTracker* en vue d'être enregistrés auprès d'un objet de type *ServiceRegister*, objet qui est chargé de maintenir une liste des services découverts et d'estimer une probabilité d'accès à ceux-ci en fonction des messages d'annonce de services reçus et des propriétés associées à ces derniers. Il détermine également quels sont les services les plus à même de répondre aux requêtes des clients locaux, et quelles sont les meilleures modalités d'invocation des services (invocation d'un service particulier vs invocation d'un groupe de services). Pour obtenir une référence sur un service distant, un service local doit invoquer ce registre de services en lui passant en paramètre un objet de type *ServicePattern*. Cet objet sera utilisé pour évaluer quel service est susceptible de répondre aux besoins du service local. Lorsqu'aucun service satisfaisant la demande du service local n'est trouvé, le registre de ressource peut initier une découverte proactive du service désiré en définissant un objet de type *ServiceDiscoveryRequest*, et en paramétrant celui-ci avec l'objet de type *ServicePattern* spécifié par le service local. Les objets implémentant les interfaces *ServiceAdvertisement* et *Service-*

DiscoveryRequest intègrent en outre les informations nécessaires à leur acheminement dans le réseau (adresse du destinataire, durée de vie, etc.), ainsi que des informations permettant de mettre en œuvre une gestion des messages reposant sur le contenu (e.g. type du contenu, description du contenu). Les objets de type *ServiceRegister* et *ServiceTracker* sont conçus pour s'enregistrer auprès d'un service OSGi implantant l'interface *DTNComService*, dont une représentation partielle est donnée dans la figure 2. Ce service est chargé d'acheminer les messages dans le réseau et d'aiguiller les messages en provenance du réseau vers les éléments concernés (i.e. vers des objets implantant l'interface *MessageListener*). Ainsi, l'objet de type *ServiceRegister* pourra réagir aux requêtes de découverte de service, et l'objet de type *ServiceTracker* aux annonces de service.

3.2. L'invocation des services

Dans notre cadre de conception, l'invocation de service repose sur des objets implantant les interfaces *ServiceRequest*, *ServiceResponse*, *ServiceResponseHandler* et *ServiceInvoker*. Un objet de type *ServiceRequest* précise le service à invoquer, la méthode à utiliser lors de l'invocation, ainsi que les paramètres de celle-ci. Un tel objet implémente également l'interface *Message* afin de permettre aux services de pouvoir spécifier le(s) destinataire(s) de la requête (un hôte particulier, un groupe de fournisseur de services ou tous les noeuds du réseau), de spécifier pendant combien de temps la requête peut être considérée comme valide, et comment elle doit être propagée dans le réseau. Pour invoquer de manière asynchrone les services distants, les services locaux doivent utiliser les méthodes *asyncInvoke()* d'un objet de type *ServiceInvoker*. L'une de ces méthodes prend en paramètres une requête et un objet implantant l'interface *ServiceResponseHandler*. Ce dernier sera utilisé par le service local pour traiter les réponses qui lui seront adressées. En effet, lorsqu'un service local adresse une même requête à un groupe de fournisseurs de services, celui-ci est susceptible de recevoir en retour plusieurs réponses de manière asynchrone. Il faut donc lui permettre de réagir à ces événements (i.e. de réagir à la notification des réponses). Ce gestionnaire de réponse pourra être paramétré par l'objet de type *ServiceInvoker* avec les informations temporelles spécifiées dans la requête. Ainsi, ce gestionnaire de réponse ne traitera plus de réponses lorsque la durée de validité de la requête sera atteinte. La seconde méthode *asyncInvoke()* est quant à elle conçue pour être bloquante et pour retourner un objet de type *ServiceResponse* lorsqu'une réponse émanant du réseau est reçue. Lorsque la durée de validité spécifiée dans la requête est atteinte, cette méthode retourne une valeur *null*. À l'instar des objets de type *ServiceRegister* et *ServiceTracker*, l'objet de type *ServiceResponseHandler* peut être conçu pour s'enregistrer automatiquement auprès du service implantant l'interface *DTNComService* afin de recevoir les réponses associées à la requête émise par le service qui l'a instancié.

4. Quelques éléments de mise en œuvre

Le cadre de conception présenté dans le paragraphe précédent a été mis en œuvre sous la forme d'un service système dans un intergiciel OSGi. Ce service est déployé sur chaque terminal mobile. Les services de niveau application sont censés utiliser ce service pour découvrir et invoquer les services distants de manière asynchrone. En outre, ce service s'appuie sur un autre service OSGi de niveau système afin d'acheminer les messages dans le réseau. Ce service, qui implante l'interface *DTNComService*, offre le moyen de maintenir chaque message émis dans le réseau aussi longtemps que possible dans un cache local par autant de terminaux que possible, afin qu'il puisse être délivré à une machine qui n'était pas présente dans le réseau au moment de son émission. La dissémination de multiples copies du même message doit contribuer à améliorer l'accès aux services, et à permettre aux terminaux de transporter eux-mêmes l'information entre les différents îlots du réseau. Dans sa mise en œuvre actuelle, ce service annonce périodiquement à l'ensemble de ses voisins les messages dont il dispose et les messages qu'il recherche. Ce service implante des mécanismes permettant de contrôler la propagation des messages dans le réseau (accusé de réception, gestion du nombre de sauts et de la durée de vie des messages, etc.)

4.1. Structures des messages

Les messages utilisés dans les processus de découverte et d'invocation de services sont réifiés sous la forme d'objets Java afin de pouvoir être manipulés facilement. Pour être émis dans le réseau, ces messages sont formatés sous la forme de document XML. Ces documents sont structurés en deux parties : une partie spécifiant les informations utiles pour l'acheminement et le traitement des messages, et une partie incluant le contenu de ces derniers. La figure 3 présente un exemple de message au format XML, message concernant une réponse émise par un fournisseur de service en retour d'une invocation.

La valeur de l'attribut *id* dans un message doit être unique afin de permettre à un noeud recevant un message de vérifier si une copie de ce message n'est pas déjà stockée dans son cache local. L'attribut *type* précise le type du message. Le type de message peut être *service-discovery-request*, *service-advertisement*, *service-request*, *service-response* ou *cancel-message*. Le message considéré spécifie également qu'il a été émis par une machine dont l'adresse MAC est "00 :0F :1F :C5 :2F :F5" (en-tête *origin*) et qu'il est adressé à l'ensemble des machines appartenant au groupe "casa" (en-tête *destination*). Le champ *number-of-hops* joue approximativement le même rôle que le champ TTL dans les paquets IP. Il permet d'assurer que le message ne sera pas propagé éternellement dans le réseau. Les champs *date* et *lifetime* indiquent quant à eux la date d'émission du message et sa durée de vie. Le champ *md5sum-content*, qui est la somme md5 du contenu, est utilisé ici pour décrire le contenu et pour permettre une gestion des messages orientée contenu. Par ailleurs, dans les réponses et dans les annonces de services, la somme md5 de la requête qui leur est associée est également

```
<message id="fb0097820f0b371" type="service-response">
<headers>
  <header name="origin" value="00:0F:1F:C5:2F:F5"/>
  <header name="destination" value="casa"/>
  <header name="number-of-hops" value="5"/>
  <header name="date" value="Nov 29 16:09:47 CET 2006"/>
  <header name="lifetime" value="12:00:00"/>
  <header name="md5sum-content"
    value="186c322f04579a179f1cce23b78e7a555"/>
  <header name="request-id" value="db0192810f0b21"/>
  <header name="md5sum-request-content"
    value="186c322f04579a179f1cce23b78e7a555"/>
  ...
</headers>
<content>
  ...
</content>
</message>
```

Figure 3. An XML-formatted message exchanged by application-level services.

spécifiée. Avec une telle information, un terminal mobile recevant une requête d'invocation de service concernant un service qu'il ne fournit pas lui-même, mais pour lequel il a déjà fait office de relais, peut jouer lui-même le rôle de fournisseur de service en envoyant la réponse qu'il aura préalablement stockée dans le cache local, si celle-ci est encore valide. Les champs définissant des propriétés contextuelles comme par exemple la position GPS du fournisseur ou du client peuvent également être ajoutés. Celles-ci peuvent également contribuer à limiter la propagation des messages dans le réseau. Dans l'implantation actuelle de notre cadre de conception, le contenu des messages sont des objets sérialisés Java.

5. Travaux apparentés

Les problèmes inhérents à la prestation de service dans les réseaux ad hoc ont déjà été abordés par le passé, et plusieurs solutions de type intergiciels ont été proposées. Ainsi, l'intergiciel DEAPspace (Hermann *et al.*, 2001) offre le moyen de découvrir et d'invoquer des services dans des réseaux ad hoc à 1 saut. L'intergiciel Konark (Hlal *et al.*, 2003), qui a des objectifs très similaires, considère quant à lui des réseaux ad hoc à plusieurs sauts. À la différence de notre proposition, dans laquelle nous ne faisons aucune supposition vis-à-vis du réseau, Konark et DEAPspace considèrent qu'une route peut toujours être établie entre un client et un fournisseur de service lorsque cela est nécessaire. Une telle supposition ne permet pas d'assurer une prestation de service dans des réseaux fortement dynamiques et connectés de manière

intermittente. ReMMoC(Grace *et al.*, 2003) et INDISS(Bromberg *et al.*, 2005) ont également étudié la prestation de service dans le contexte de l'informatique diffuse, en se focalisant particulièrement sur la configuration et l'utilisation des protocoles de découverte de service. Ces systèmes ne considèrent jusqu'ici uniquement les réseaux sans-fils reposant sur une infrastructure, et sur l'utilisation de protocoles de découverte de service standards tels que Jini, SLP (*Service Location Protocol*) ou UPnP (*Universal Plug and Play*).

Plus récemment, des protocoles de découverte de service adaptés spécialement aux réseaux ad hoc mobiles dynamiques ont vu le jour. SSD (*Scalable Service Discovery*) (Sailhan *et al.*, 2005) est en est un exemple. SSD aborde le problème de la découverte de service dans des réseaux ad hoc de grande taille. Dans ce protocole, la découverte de service repose sur la collaboration d'un ensemble d'annuaires élus à la volée par les noeuds mobiles Cette approche vise à réduire le trafic engendré par le processus de découverte. Les auteurs de SSD font l'hypothèse d'un réseau routé par des protocoles tels que OLSR, AODV ou ZRP. Dans (Musolesi *et al.*, 2005a), Muscolesi et al. abordent les problèmes posés par les communications asynchrones dans des réseaux mobiles ad hoc partiellement connectés. Ces travaux ne considèrent cependant pas une gestion des messages orientée contenu, ce qui nous semble être essentiel pour différencier les messages et améliorer la prestation de service.

Notre travail peut également être comparé avec celui de Vahdat et Becker (Vahdat *et al.*, 2000), travail dans lequel ils introduisent le concept de routage épidémique. Dans ce modèle, les messages sont sauvegardés sur les équipements mobiles et échangés de manière aléatoire entre les noeuds. Ce modèle offre la possibilité de délivrer des messages dans des réseaux partiellement connectés. Néanmoins dans cet article, Vahdat et Becker considèrent uniquement des transmissions unicast, ils suggèrent néanmoins que le routage épidémique serait également approprié pour les communications multicast.

6. Travaux futurs et conclusion

Dans cet article, nous avons proposé une approche visant à offrir une certaine ubiquité d'accès aux services dans des réseaux mobiles ad hoc partiellement connectés, connectés par intermittence ou déconnectés. Cette approche s'appuie sur un mécanisme d'adressage flexible, sur une gestion des messages basée sur le contenu, et sur une communication reposant sur le principe du *store-and-forward*. Le cadre de conception et la boîte à outils mettant en œuvre cette approche ont été utilisés pour concevoir un intergiciel, intergiciel dont les performances sont en cours d'évaluation. Ces évaluations sont réalisées par simulation pour différents modèles de réseaux. Dans le futur, nous pensons établir des passerelles entre les réseaux d'infrastructure et les réseaux ad hoc, car les réseaux d'infrastructure demeurent la première source pour l'approvisionnement des services.

7. Bibliographie

- Bromberg Y.-D., Issarny V., « INDISS : Interoperable Discovery System for Networked Services », *ACM/IFIP/USENIX 6th International Middleware Conference (Middleware'2005)*, Grenoble, France, November, 2005.
- Chen X. and Murphy A. L., « Enabling disconnected transitive communication in mobile ad hoc networks », *Workshop on Principles of Mobile Computing (2001)*, p. 21-23, August, 2001.
- Eugster P., Guerraoui R., Kermarrec A.-M., Massoulié L., « From Epidemics to Distributed Computing », *IEEE Computer*, vol. 37, n° 5, p. 60-67, May, 2004.
- Grace P., Blair G. S., Sam S., « ReMMoC : A Reflective Middleware to Support Mobile Client Interoperability », *On The Move to Meaningful Internet Systems 2003 : CoopIS, DOA, and ODBASE*, vol. 2888 of *LNCS*, Springer-Verlag, Catania, Sicily, Italy, November, 2003.
- Harras K. A., Almeroth K. C., Belding-Royer E. M., « Delay Tolerant Mobile Networks (DTMNs) : Controlled Flooding in Sparse Mobile Networks », *IFIP Networking Conference, Waterloo, Ontario, CANADA*, May, 2005.
- Helal S., Desai N., Verma V., Lee C., « Konark : Service Discovery and Delivery Protocol for Ad-hoc Networks », *Third IEEE Conference on Wireless Communication Networks (WCNC)*, New Orleans, USA, March, 2003.
- Hermann R., Husemann D., Moser M., Nidd M., Rohner C., Schade A., « DEAPspace - Transient ad hoc networking of pervasive devices », *Computer Networks*, vol. 35, n° 4, p. 411-428, March, 2001.
- Li Q., Rus D., « Sending Messages to Mobile Users in Disconnected Ad-hoc Wireless Networks », *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking*, ACM Press, Boston, p. 44-55, August, 2000.
- Musolesi M., Hailes S., Mascolo C., « Adaptive Routing for Intermittently Connected Mobile Ad Hoc Networks », *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM05)*, Taormina, Italy, June, 2005a.
- Musolesi M., Mascolo C., Hailes S., « Adapting Asynchronous Messaging Middleware to Ad Hoc Networking », *Proceedings of 2nd ACM International Workshop on Middleware for Pervasive and Ad Hoc Computing (MPAC 2004) in Middleware 2004 Companion*, ACM Press, Toronto, Canada, p. 121-126, October, 2004.
- Musolesi M., Mascolo C., Hailes S., « EMMA : Epidemic Messaging Middleware for Ad hoc networks », *Personal and Ubiquitous Computing Journal*, 2005b. To Appear.
- Sailhan F., Issarny V., « Scalable Service Discovery for MANET », *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom'2005)*, IEEE Press, Hawaii, USA, March, 2005.
- Sasson Y., Cavin D., Schiper A., Probabilistic Broadcast for Flooding in Mobile Ad Hoc Networks, Technical Report n° IC/2002/54, Swiss Federal Institute of Technology (EPFL), 2002.
- Shah R., Hutchinson N. C., « Delivering Messages in Disconnected Mobile Ad-Hoc Networks », *Proceedings of ADHOC-NOW 2003*, Montreal, October, 2003.
- Vahdat A., Becker D., Epidemic Routing for Partially Connected Ad Hoc Networks, Technical report, Duke University, April, 2000.
- Zhao W., Ammar M., Zegura E., « A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks », *Proceedings of ACM Mobihoc 2004*, Tokyo Japan, May, 2004.