Scalable Service Discovery for MANET

Françoise Sailhan, Valérie Issarny

Scalable Service Discovery for MANET

Françoise Sailhan, Valérie Issarny
INRIA-Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105,
78153 Le Chesnay Cédex, France
francoise.Sailhan, Valerie.Issarny@inria.fr

Abstract

Mobile Ad hoc NETworks (MANETs) conveniently complement infrastructure-based networks, allowing mobile nodes to spontaneously form a network and share their services, including bridging with other networks, either infrastructure-based or ad hoc. However, distributed service provisioning over MANETs requires adequate support for service discovery and invocation, due to the network's dynamics and resource constraints of wireless nodes. While a number of existing service discovery protocols have shown to be effective for the wireless environment, these are mainly aimed at infrastructure-based and/or 1-hop ad hoc wireless networks. Some discovery protocols for MANETs have been proposed over the last couple of years but they induce significant traffic overhead, and are thus primarily suited for small-scale MANETs with few nodes. Building upon the evaluation of existing protocols, we introduce a scalable service discovery protocol for MANETs, which is based on the homogeneous and dynamic deployment of co-operating directories within the network. Scalability of our protocol comes from the minimization of the generated traffic, and the use of compact directory summaries that enable to efficiently locate the directory that most likely caches the description of a given service.

1. Introduction

Ad hoc networks enable spontaneous connectivity among wireless devices, without requiring any particular infrastructure. In addition, ad hoc routing protocols increase connectivity by offering multi-hop communication, while supporting a continuously changing topology, leading to the automatic configuration of infrastructure-less Mobile Ad hoc NETworks (MANETs). However, effective exploitation of networked services within a MANET requires enabling the delivery of services that best match the application's requirements, still taking into account the MANET's dynamics.

The major challenges in enabling service delivery, i.e., service discovery and service invocation, within MANETs lie in:

- Enabling resource-constrained, wireless devices to discover services dynamically, while both minimizing the traffic generated by the discovery process and tolerating the intermittent connectivity of devices.
- Enabling service delivery to a wide spectrum of devices, regardless of their hardware and software platforms.
- Enabling service requesters to differentiate service instances according to provided non-functional properties, so that services can be matched against the applications' quality of service requirements.
- Enabling service discovery in large MANETs, composed of hundreds of nodes, so as support the large class of applications envisioned for MANETs, in areas as diverse as military operations, rescue missions, pervasive computing and ambient intelligence.
- Enabling bridging MANETs with infrastructure-based networks since Intranet and/or Internet connectivity remains the primary source of service provisioning. Such a feature allows devices that are located in the MANET to transparently access services offered in the interconnected networks (Internet or Intranet), and devices located outside the network to discover services provided in the interconnected MANETs.

This paper introduces a scalable service discovery protocol that meets the above requirements. The next Section first defines the mobile environment targeted by our service discovery protocol, and then evaluates existing protocols against the above requirements. Basically our service discovery protocol relies on a backbone of directories constituting a *virtual network*. Each directory performs service discovery in its vicinity (Section 3). Global service discovery within the overall network is then performed by the backbone which is composed of directories that cooper-

ate and are dynamically deployed (Section 4). Performance evaluation of our protocol through simulation shows in particular the scalability of our protocol, considering generated traffic (Section 5). Finally, we conclude with a summary of our contribution (Section 6).

2. Service Discovery in the Mobile Ad Hoc Environment

MANET was initially mainly targeted at crisis applications[1] (e.g., military, emergency/rescue operations). Spurred by progress of technologies and their coming availability at low cost, ad hoc networks appear much valuable for civilian applications as well. They are in particular central to the vision of pervasive computing, due to their support for ubiquitous networking. Independent of the application domain, the mobile environment now builds upon hybrid networks, bridging MANETs and infrastructure-based networks (§ 2.1), and provides access to a rich variety of services, including services hosted by wireless nodes (§ 2.2). However, existing service discovery protocols for the wireless environment are mainly suited for small-scale MANETs. This leads us to introduce a discovery architecture structured as a *virtual network* in which directories cooperate towards scalable service discovery (§ 2.3).

2.1. Network architecture

We assume a MANET composed of nodes holding the same network interface[2], with IP-level connectivity using the underlying routing protocol. We do not assume any specific routing protocol for the MANET, which may be reactive, proactive or hybrid. We simply consider that the protocol maintains routing information for nodes reachable in 1 hop, as implemented by proactive (e.g., OLSR [4]) and hybrid protocols (e.g., ZRP [8]). In case of reactive protocols (e.g., AODV [5]) information concerning nodes reachable in 1 hop are maintained by our service discovery protocol. Note that in case of scalable and dense ad hoc networks, proactive and hybrid protocols provide enhanced performances compared to reactive protocols. Nodes of the MANET are further heterogeneous in terms of resource capacities, and include today's wireless devices such as smart phones, PDAs, laptops, etc. We also consider that some nodes hold several network interfaces, and act as gateways with other networks, either ad hoc or infrastructure-based. This defines a hybrid network bridging MANETs and infrastructure-based networks, according to the specific networking capabilities of the wireless nodes (see Figure 1).
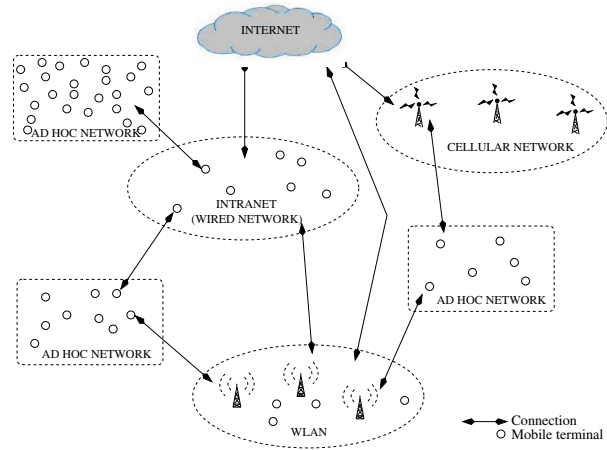

Figure 1. Network architecture

2.2. Service architecture

While service discovery protocols were initially introduced to deal with services provided by devices like printers or cameras, the increasing popularity of handheld devices has led to consider a rich variety of mobile services, as illustrated by scenarios in the area of pervasive computing, ambient intelligence or mobile-commerce. This further requires service-level interoperability among mobile devices, despite the heterogeneity of the supported software platforms. Interoperability among mobile devices is a long term research goal. However, pervasiveness of the Web allows assuming that a significant level of interoperability can already be achieved using the Web services architecture[3]. We thus consider a service-oriented architecture based on Web services, with wireless nodes hosting Web services, as enabled by, e.g., the WSAMI open source middleware[4] [10] developed at INRIA in the context of the Ozone IST project[5]. We then build on standardization effort of the Web service activities that has in particular introduced XML languages for describing Web services' interfaces. Specifically, we use WSDL for service description. We further enrich WSDL to allow the specification of Quality of Service (QoS) parameters, as QoS-awareness is a key requirement in the mobile environment [11]. Thus, service providers proactively provide QoS parameters together with the service's functional interface, to service requesters. Based on the work of [15], QoS specification includes both service-related and resource-related QoS parameters. Service-related QoS parameters decompose into: (i) security mechanisms supported by the service, (ii) reliability relating to the availability of the service over

---

[1]http://w3.antd.nist.gov/wctg/manet/
[2]i.e., using to the same communication-related technology (e.g., Bluetooth, IEEE 802.11)

[3]http://www.w3.org/2002/ws/
[4]Web Service for AMbient Intelligence.
[5]http://www.extra.research.philips.com/euprojects/ozone

a specified period of time, and (iii) scalability specifying the capacity of serving additional requests for a specified period of time. Resource-related QoS parameters specify performance and capacity of the host, stating memory capacity, available energy, computation capacity, geographical localization support, connectivity (expressed in terms of the number of neighbors in the ad hoc network), and network capacity (expressed in terms of network type, mean bandwidth, and packet loss rate). Specification of the above QoS parameters is optional, and each parameter provides the following QoS attributes: name, value and expiration. The QoS expiration attribute is provided for parameters that evolve over time, and denote the point in time at which the provided QoS value is no longer valid. The QoS value is in the range [*not supported, null, bronze, gold, platinum, infinite*], from lowest to highest QoS level.

2.3. Discovery architecture

The major structural difference between existing service discovery architectures is the utilization or not of a central directory. For instance, well known solutions, such as Jini[6], UDDI[7], Salutation[8], rely on a central directory that stores information about services available in the network so as to enable service discovery and invocation. Discovery of the directory by service seekers and providers is in general based on multicasting. Then, service providers advertise their services to the central directory using a unicast message. And, to access a service, a client first contacts the central directory to obtain the service description, which is then used to interact with the service provider. Centralized resource discovery is much suited to wireless infrastructure-based networks. However, this makes the service discovery process dependent upon the availability of the central directory, which further constitutes a bottleneck. In order to increase the scalability of the discovery protocol, solutions presented in [2, 6] employ a distributed set of directories, which are deployed over base stations (or gateways) and are responsible for a spatial region (e.g., corresponding to a cell).

Decentralized resource discovery protocols appear better suited to ad hoc networks. Indeed, services and clients discover each others directly, as in, e.g., SLP[7] and UPnP[9]. Specifically, when a client wants to access a service, it broadcasts a service query that is handled by available service providers, which is known as *pull-based* discovery. Alternatively, under *push-based* discovery, service providers broadcast service advertisements, which are possibly cached by nodes for later service invocation. The ma-

jor drawback of the above decentralized solutions is that broadcasting leads to flood the network. Another approach to disseminating service information while not relying on broadcasting is to use geographic information for routing [12]. Nodes periodically send advertisements along a geographic trajectory (basically, north-south and west-east), and nodes on the trajectory cache the advertisements. Then, when a client seeks a service, it sends a query that eventually intersects the advertisement route at a node that replies to the client. This solution assumes that the density of nodes is high enough and further leads to cache service advertisements on a large number of nodes. Hence, it incurs resource consumption that may not be accommodated by wireless, resource-constrained nodes. Resource consumption is further increased by the required support for geographical location (e.g., GPS). Yet another approach is to couple the service discovery protocol with network-layer multicast dedicated to MANET [13, 9]. However, multicast protocols generate a large number of control messages, and are thus primarily suitable for data-intensive applications like multimedia streaming. Hence, the use of a multicast protocol for service discovery cannot be undertaken in general and should rather be adopted when the MANET is dedicated to data-intensive services (e.g., home networking). Alternatively, one may limit service discovery to the local area and thus broadcast messages over a limited number of hops, as in GSD [3]. GSD is a group-based service discovery protocol that performs local discovery of available services. To achieve global discovery, GSD classifies services into several groups, according to a service hierarchy expressed in DAML[10], and includes in each advertisement that is broadcasted, the list of groups that the sender has seen in its vicinity. Then, the group information is used to selectively forward a service request to other nodes. This solution requires the group information to be exchanged and included in each advertisement. As a result, it increases the size of the exchanged messages and further relies on a metric that quantifies the probability of presence of the corresponding group in the vicinity. As briefly surveyed above, existing service discovery protocols for the wireless environment are mostly aimed at small-scale MANETs (i.e., comprising less than 30 nodes according to published performance). The following introduces a service discovery protocol aimed at large MANETs (i.e., comprising at least about 100 nodes). Our primary goal is to keep to a minimum the traffic generated by the service discovery process, so as to minimize consumption of resources and in particular energy. Following the above survey, our design is based on a centralized discovery architecture, as it induces less traffic. Directories are further distributed and deployed dynamically for the sake of scalability. Specifically, our discovery architecture is structured as a virtual network (see Figure 2). A virtual network

---

[6]http://www.sun.com/software/jini

[7]http://www.uddi.org

[8]http://www.salutation.org

[9]http://www.upnp.org

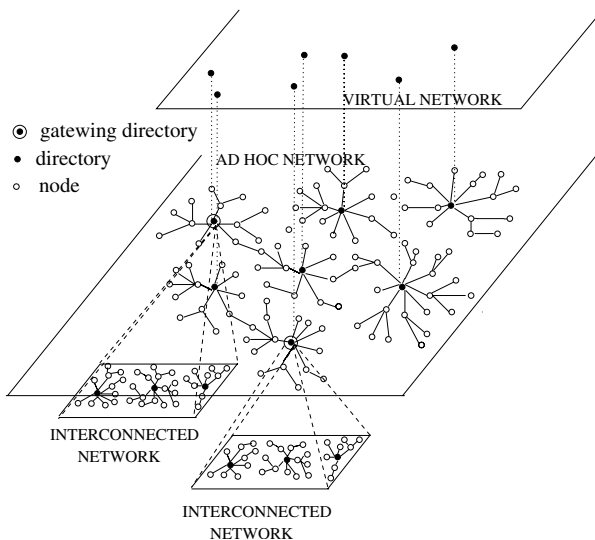[10]http://www.daml.org

IEEE
COMPUTER
SOCIETY

Figure 2. Discovery architecture

is composed of a subset of nodes of the MANET acting as directories. These directories represent a backbone of nodes responsible for performing service discovery. These directories are deployed so as at least one directory is reachable in at most a fixed number of hops, $\mathcal{H}$, whose value is dependent upon the nodes density. Directories cache the descriptions of Web services available in their vicinity which is defined by $\mathcal{H}$. Hence, wireless nodes that do not act as directories do not have to maintain a cache of service descriptions, and the network is not flooded by service advertisements. A service seeker simply sends a query to the directory for local service discovery (§ 3). If the description of the requested service is not cached by the local directory, the directory selectively forwards the query to other directories so as to perform global discovery (§ 4). Selection of directories to which service queries are forwarded, is based on the exchange of profiles among directories. The directory profile provides a compact summary of the directory's content and a characterization of the host capacity. Directory profiles allow both guaranteeing that service queries are issued to directories that are likely to cache the description of the requested service and to keep to a minimum the generated traffic. Most of the ad hoc networks deployed today are hybrid networks instead of pure ad hoc networks. Then, another critical issue lies in providing convenient features so as to enable discovery of services over a hybrid network. This is supported through mobile nodes acting as gateways between networks.

3. Local Service Discovery

As briefly sketched above, directories within our discovery protocol are responsible for: (i) caching the description of the services available in the directory's vicinity to answer to service queries, and (ii) advertising their presence to nodes within their vicinity.

A directory caches service advertisements that are published proactively by service providers to the directory(ies) over $\mathcal{H}$ hops, which define the coverage of the directory (with respect to the specific sender/directory). Management of the directory content further relies on a cache replacement policy, which is not detailed here due to the lack of space. Briefly stated, the cache is managed so as to remove first the description of the services that are the less likely to be accessed in the near future or may be retrieved from the other directories deployed in the MANET. Each service provider sets its periodicity of publicity according to: (i) the number of mobile terminals that are present in its vicinity, as known from the routing table managed by the underlying routing protocol or by the directory advertisements, (ii) the traffic load, congestions and collisions, as known by monitoring the traffic sent on the channel and the error rate, and (iii) the service's dynamic QoS attributes, as known from the service description.

When a directory receives a service query (including the service's WSDL-based functional interface and QoS parameters), it checks its cache content and sends back a hit message embedding the matching service description(s) –if any. In order to provide users with better quality of service and balance the load among service providers, directories select services that: (i) best match the query in terms of both functional interface and provided QoS, and (ii) are the less loaded, which is evaluated using a pricing-based scheme such as the one discussed in [14]. If the directory does not cache the description of the requested service, it initiates a global discovery, as detailed in Section 4. Directories are
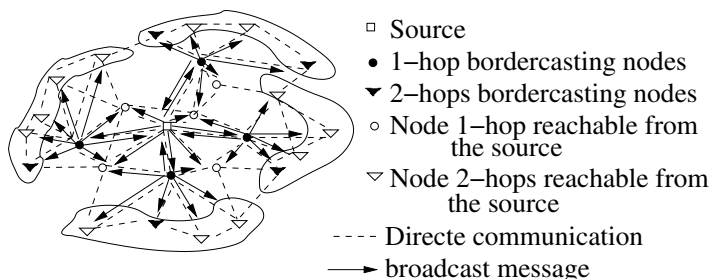

Figure 3. 2-hops bordercasting

responsible for being known to services providers and requesters, i.e., for advertising their presence to nodes over $\mathcal{H}$ hops. As directory advertisement relies on broadcasting, network flooding is avoided by limiting the broadcast to the directory coverage (i.e., $\mathcal{H}$ hops), and further reducing the related traffic as follows. First, when a packet has already been received, it is not forwarded. Second, we use *2-hops bordercasting* to drastically reduce network traffic (see Fig-

ure 3). Basically, the broadcast initiator broadcasts the message to all the nodes reachable in 1 hop, and selects among those nodes, the minimal number of nodes (called bordercast nodes) that allow reaching all the nodes accessible in 2 hops and whose zone of influence does not overlap. The selected nodes will then forward the message and the target nodes will perform the same algorithm as the broadcast initiator. The process is repeated until nodes at $\mathcal{H}$ hops are reached.

3.1. Directory deployment

In order to deploy dynamically and homogeneously directories in the MANET, directories are elected on the fly by mobile nodes. The election process accounts for directories that are already deployed so as to keep to a minimum duplicate directories' coverage. However, a given node can be covered by more than one directory, depending on the movement of nodes. Specifically, directories advertise periodically their presence every Δ_τ time unit. Then, if for a given period of time $\alpha \times \Delta_\tau$ (α being set according to the network's load), a mobile node (that is not hosting a directory) does not receive any directory advertisement, the node initiates the election of a directory.

The election process starts by broadcasting an election message over $\mathcal{H}$ hops, still using 2-hops bordercasting. Mobile nodes receiving the election message can either refuse or accept to act as a directory, depending on their capacity, considering that fair behavior of nodes is promoted using the aforementioned pricing-based scheme [14]. When a node is willing (resp. refusing) to act as a directory, it sends back an acceptance (resp. refusal) message to the initiator of the election, providing its capacity in terms of available resources and its context parameters characterizing the node's environment (i.e., stability level, a list of directories covering the node, a list of mobile terminals reachable in 1 hop and available bandwidth). Finally, the initiator elects a directory based on the capacity and context of the nodes that replied, and sends a notification message to the chosen node. The main selection criteria is the directory's coverage, i.e., the node that has the highest number of neighbor nodes and the smallest number of directories in its vicinity (i.e., $\mathcal{H}$). This mechanism allows electing the directory that offers the best physical properties and to distribute efficiently directories since directories are elected in the network area that is the less covered.

In case, two (or more) mobile nodes initiate simultaneously an election process, the one having the smallest IP address is chosen as the coordinator (i.e., the one that processes the election), while the others stop running the election process. Elected directories change periodically to fairly distribute the load of service discovery among mobile nodes, and to account for their disconnection, either volun-

tary or involuntary because of limited battery lifetime. A directory that passes on its role then initiates an election process, as defined above.

4. Global Service Discovery

The above directory-based service discovery allows discovering services that are (or were previously) in the vicinity of the local directory. However, nodes need to be able to locate services available in the overall MANET and even in the interconnected networks. This is achieved through cooperation among directories (§ 4.2), possibly relying on nodes acting as gateways to bridge with other networks (§ 4.3). However, efficiency of the discovery process both in terms of response time and generated traffic requires interacting with the directories that most likely cache the description of a service that is not known locally. Towards that goal, we introduce directory profiles that provide host characterization and a compact summary of the directory content (§ 4.1).

4.1. Directory profile

We use Bloom filters as a method for summarizing the content of a directory, i.e., the set of cached WSDL-based service descriptions. Bloom filters [1] are an efficient way of describing sets because: (i) they are compact summaries and thus induce limited storage requirements and bandwidth utilization, and (ii) unlike traditional profile-based approach to the characterization of cache content (e.g., [3]), they do not require to predefine a static set of keywords characterizing some categories or types of popular services that serve generating profiles. The main idea is to compute a vector v of m bits (initially set to 0), which corresponds to a Bloom filter. First, k independent hash functions h_1, h_2, ..., h_k, each with range $\{1, ..., m\}$ are chosen. From a practical point of view, the hash functions are built by: (i) calculating the md5 signature of 128 bits [16] although other hash functions may be used, (ii) dividing the 128 bits into k groups of $128/k$ bits words, and (iii) taking the modulo m of each 32 bits word. Then, for each service description w stored in the directory, the service description is hashed with the k independent hash functions. The bits of vector v whose positions are given by the results of the k hash functions are set to 1, i.e., the bits at positions $h_1(w)$ $h_2(w)$, ..., $h_k(w)$ are set to 1 in v.

Computation of the directory's Bloom filter is based on hashing service descriptions. However, a service query may contain a subset of the service's description rather than the whole service description. Computing a hash for the complete service description is thus not sufficient. An alternative would be to hash each possible subset of the service description but the resulting computation would be too costly.

Thus, we limit the number of subsets of the service description that are extracted for hashing by considering offered operations individually. Basically, we define a tree of depth 2 from the service description, where:

- The root of the tree corresponds to the service and its value is the service's name.
- The root has as many child nodes as operations provided by the service, and the value of each node is the operation name
- Each of the above node has as many child nodes as service-related QoS parameters defined by the service, and the value of each node is the name of the parameter followed by its value (i.e., the expiration attribute is not included).

The root and all the paths of length 1 (i.e., limited to an operation) and 2 (including a QoS parameter) of the above tree then define all the subsets of the service description. The value of each subset is given by the concatenation of the value of the nodes of the specific path, and it is hashed for inclusion in the Bloom filter.

In order to determine whether a directory caches a service description w, using the directory's Bloom filter, we consider each subset $sub_i(w)$ of the service description, as defined above. For all the subsets, all the bits at position $h_1(sub_i(w))$, $h_2(sub_i(w))$, ..., $h_k(sub_i(w))$ are checked. If they are all set to 1, the directory is likely to contain w. However, there is a non-null probability for the directory to not cache w, which is known as *false positive*. The probability of a *false positive* is given by: $\left(1 - (1 - \frac{1}{m})^{knp}\right)^k \approx \left(1 - e^{knp/m}\right)^k$, with n (resp. m) corresponding to the number of service descriptions stored in the directory (resp. the size of the Bloom filter), and p corresponding to the average number of subsets. The value of k and m should thus be chosen so that the probability of a false positive is kept to a minimum, i.e., $k \approx m\frac{ln2}{np}$. However, in practice, the value given to k is less than the optimal one to reduce the computation overhead of hashing. As an illustration, consider a directory that stores the description of 100 services, a service description taking 3kbits. Figure 4 gives the effectiveness of Bloom filter for such a directory, giving the probability of false positive according to the Bloom filter size for $k = 1, ..., 8$. One can see that a Bloom filter size of 700 bits with $k \geq 3$ suffices to to have a probability of false positive that is close to 0. When the directory content is updated, the directory's Bloom filter needs to be updated accordingly. In order to minimize this cost, we use the following counter-based solution. We associate a vector c of counters with the directory's Bloom filter v, where $c(i)$ counts the number of services that have set to 1 the bit at position i in v. For instance, if the caching of 2 service descriptions has led to set to 1 the bit at position i in v, $c(i) = 2$. Similarly, when a service description, w, is removed from the cache, the coun-
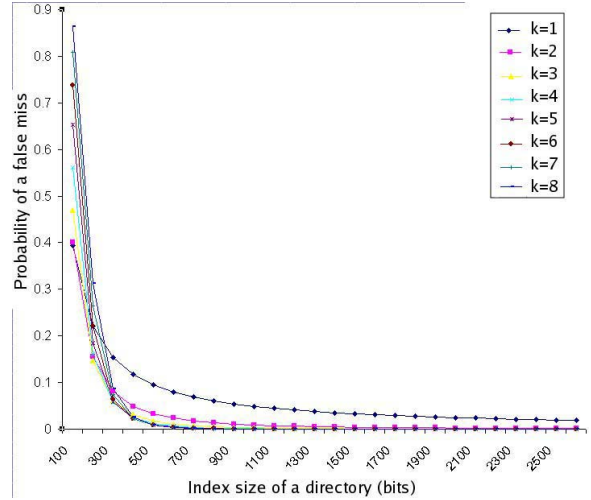

Figure 4. Probability of false positive using Bloom filters

ters that were incremented in c following the caching of w, are decremented. Then, when an element of c reaches 0, the corresponding bit in v is set to 0. As a result, the directory Bloom filter is updated at low cost.

4.2. Cooperative service discovery

When a newly configured directory D enters the network, if D is aware of at least one directory A (as known whether from the acceptation or refusal messages exchanged during the election period or from the previous vicinity of D), then D sends a request to directory A and A sends back the list of profiles (i.e., Bloom filter and host capacity) which it hold. Otherwise, directory D broadcasts an advertisement over the network using 2-hops bordercasting. Directories that receive the advertisement send back their profile (i.e., Bloom filter and host capacity). In both cases, directory D creates the corresponding list of peer directories, storing the directories' profiles. Then dually, D later sends its local profile to peer directories, once it caches a significant number of service descriptions. Each directory further periodically broadcasts an advertisement, including its profile, over $\mathcal{N} = 2 \times \mathcal{H}$ hops (i.e., 2 times the size of the directory vicinity), using 2-hops bordercasting. This allows directories to discover peers according to their respective mobility pattern.

When two formerly separate networks N and M merge following a new wireless link between two frontier nodes (b and c), then directory(ies) $Dir_{N1}, \cdots, Dir_{Nn}$ (respectively $Dir_{M1}, \cdots, Dir_{Mm}$) composing the virtual network of N and respectively M should exchange their profiles. Let Dir_{N1} be the directory covering node b, and Dir_{M1}

a directory covering node c. At least one directory (let Dir_{N1} be this directory, without lost of generality) detects the presence of directory Dir_{M1}. Dir_{N1} and Dir_{M1} then exchange the list of profiles they hold. Finally, Dir_{N1} sends the list of profiles of directory(ies) composing the virtual network of N (resp. M) to the peer directory(ies) composing the virtual network of M (resp. N). Changes to the directory content has to be forwarded to peer directories. Scalability of our solution in particular comes from the fact that directories profiles do not have to be accurate all the time, i.e., updated upon every change to the directory's content. Instead, a request for updated profile to a directory D is reactively sent by a peer storing the profile when the percentage of false misses[11] (and hence degradation of the hit ratio) reaches a given threshold. When a directory receives a profile request, it sends either the complete bit vector or the list of bits that changed, depending on their respective size. According to the directory deploy-
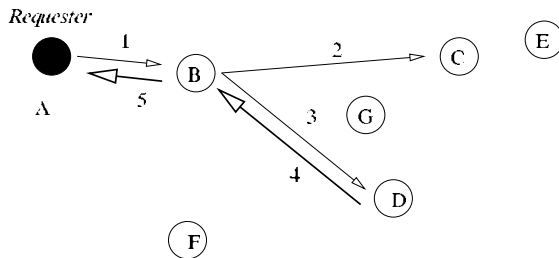

Figure 5. Global discovery

ment policy (§ 3.1), any mobile node A has at least one directory B. When A seeks a service, it sends a query message to B (see Fig 5 – message 1), leading to execute the following discovery process. When B receives the query message from A, it probes its cache and sends back a hit message to A if it stores the description of the requested service. Otherwise, cache miss at B leads B to probe the Bloom filters of known peer directories, and send a query message to the M ($M < Max$) peer directory(ies) that are likely to cache the description of the target service (messages 2 and 3). Specifically, let d be the number of known peer directories, at time t, directory B requests among the d directories, the l directories that minimize the following function for $1 \leq i \leq d$: $min_i(f_i(t) + g(t, n_i))$, with $f_i(t)$ giving the remaining battery lifetime at time t, which is computed from the battery discharge curve available from the operating system API, and the battery indicator previously provided by the directory i at time t'. This function is then curved so that at a given point of time (e.g., when reaching 80 % of the battery lifetime), the cost grows rapidly. Function g further accounts for the impact of forwarding the query message to n_i mobile terminals upon the

remaining battery lifetime. Selection of the directories to which the query message should be forwarded further takes into account the distance between the respective directories. Finally, requested directories (e.g., C and D in the figure) send back to B a Hit message (including the description of the service that matches with the request) if the requested service is stored (message 4). Notice that no miss message is sent to B to limit traffic. Finally, B sends back to A either a Hit or Miss message (message 5), the latter being sent if B does not receive any Hit message after a given timeout.

4.3. Service Discovery in the hybrid network

Several types of networks (i.e., wired networks and different types of wireless networks) can coexist in an area without being interconnected since they are administered by distinct entities in different administrative domains. A classical solution to bridge those networks is to connect wireless nodes with the wired network using a wireless access point. Another solution is based on a mobile terminal holding several network interfaces and acting as a gateway to bridge different administrative domains (e.g., wired network and ad hoc network). More precisely, one of these network interfaces communicates with mobile terminals in the MANET, whereas the other interfaces are used to communicate with external networks (e.g., wired network) and to connect these latter with the ad hoc network. Interconnected networks form a new network called hybrid network. In a hybrid network, although a mobile terminal (with one network interface in ad hoc mode) can communicate with further networks through the gateway nodes, the service discovery process is commonly limited to services available in the ad hoc network. Enabling service discovery in all the networks composing the hybrid network requires mobile terminals that act as gateways to: (i) advertise their service (e.g., with a related gateway protocol, a zeroconf protocol, or using our protocol), and (ii) act as directories, implementing the same cooperative behavior as discussed above. A gateway directory then holds the description of services available in all the networks composing the hybrid network (but the Internet), and advertises itself to the networks it bridges (but the Internet) composing the hybrid network.

Furthermore, to support service discovery in a managed network in which a network administrator deploys directories, clients and service providers behave differently. Since clients and service providers do not need to elect a directory, they can listen for directories announcements or rely on the DHCP protocol.

With our approach, services provided within the ad hoc network are visible from mobile terminals in the interconnected and external networks forming the hybrid network. This is especially useful in an enterprise network where some ad hoc networks can be created and interconnected to

---

[11]also called *false positive*

the Intranet network. Notice that the goal of the proposed service discovery protocol is not to perform service discovery in large scale networks such as the Internet. As a result, we assume that a client willing to access some Web service located in the Internet relies on Internet-level service discovery, such as UDDI.

5. Assessment

The NS simulator[12] is used to evaluate the performance of our service discovery protocol. Simulations are run for a duration of 500s, and data are collected from the beginning of the simulation to provide the cost of directory deployment at initialization time. We consider an ad hoc network of mobile terminals with transmission range $t = 200m$, and that is such that mobile terminals are initially uniformly distributed, and randomly move at an average speed of $1\ m/s$. We use OLSR [4] as underlying multi-hop routing protocol on top of the IEEE 802.11 MAC protocol. We in particular take advantage of the routing protocol facilities for implementing 2-hops bordercasting: instead of computing bordercast nodes (also called multi-point relays), we use the bordercast nodes selected by the multi-hop routing protocol. This allows us to further reduce the traffic generated by our protocol, as electing bordercast nodes introduces extra traffic due to control messages sent periodically to indicate to a given node that it has been selected by another to be its bordercast node. In order to provide an accurate measure of the traffic generated by our service discovery protocol, we take into account both: (i) the traffic generated by the initial sender, and (ii) the traffic generated by message forwarding. The sum of both traffic gives the cost of broadcasting. In the following, each traffic is provided separately.

We first consider a MANET of N nodes over a surface $S = 600m \times 600m$. The MANET is structured around directory whose coverage corresponds $\mathcal{H} = 1$ hop. Figure 7 and 6 gives the traffic generated by our Web service discovery protocol for the above MANET, according to nodes density, considering each type of message separately. In the simulation, 10% of the mobile nodes are service providers that offer $E = 2$ services, and each client node periodically seeks a service among the $E \times N \times 10\%$ services that are provided in the network. As a result, the traffic generated by service queries and advertisements increases with the nodes density. The nodes density has a low impact on the traffic generated by the deployment of directories (i.e., *election, directory advertisement*, and *Bloom filter exchange* messages), which is stable, except for messages sent by nodes that accept to act as directories during the election process. The latter result was foreseeable since the number of election repliers is equal to: $\frac{S}{N} \times \pi \times t^2 \times \mathcal{H}^2$. The
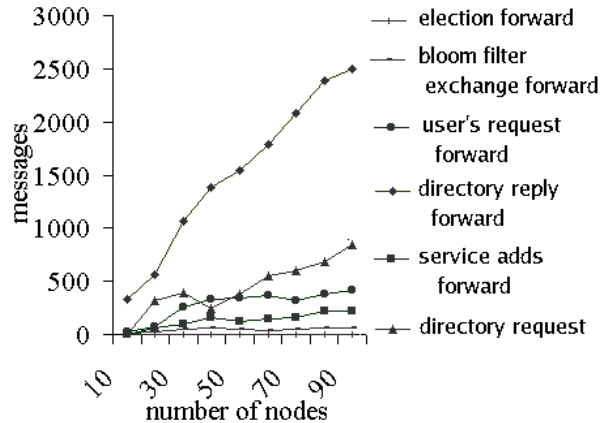
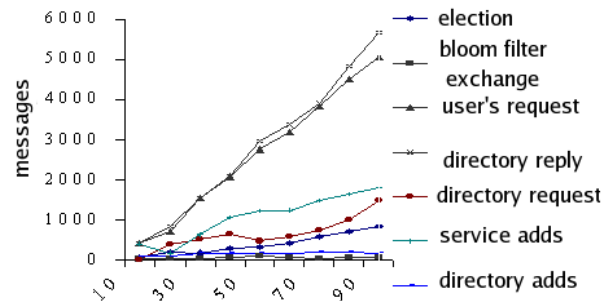**Figure 6. Traffic generated according to nodes density**

Figure 7. Traffic due to message forwarding

traffic resulting from the exchange of Bloom filters is independent of the nodes density, since the number of directories do not directly depends on that parameter. Furthermore, this traffic has a low impact on the overall traffic, while it enables discovering services known by remote directories and keeping keep to a minimum the number of requests exchanged between directories. The difference between the traffic generated by queries from service seekers and those from directory seekers comes from the fact that a directory reply includes the reply to both the directory and service query. The traffic corresponding to directory requests (i.e., service queries sent by directories to other directories) and the exchange of Bloom filters together with the difference between directories replies and service queries give the cost for enabling global discovery in the overall MANET, compared to local discovery. Figures 8 and 9 compares our protocol with a decentralized, pull-based service discovery protocol that floods the network on demand, i.e., service query at a node is broadcasted using 2 hops bordercasting facilities over the MANET. Simulation parameters are the same as above. Our service discovery protocol induces a
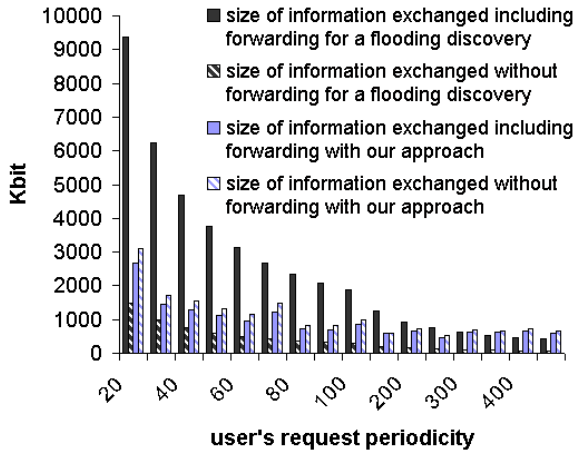
Figure 8. Comparison of messages length with decentralized, pull-based discovery

Figure 9. Comparison of traffic with decentralized, pull-based discovery

Figure 10. Traffic generated according to the directory's coverage

fixed cost, which corresponds to the deployment and update of directories. It appears that our protocol performs better than a decentralized, pull-based protocol when the rate of service queries reaches a given threshold. Notice that when the user's request rate increases, the performance of the flooding protocol decreases proportionally. We now consider a MANET of $N = 90$ nodes over a surface $S = 1000m \times 1000m$. Figure 10 evaluates the traffic generated by our service discovery protocol according to the number of hops (i.e., value of $\mathcal{H}$) defining a directory coverage. The cost of directory election, including the traffic generated by the broadcast message sent by the initiator of the election, and the replies sent by nodes accepting to act as a directory (in the simulation, all the nodes accept) is quite stable since there is a balance between the number of election initiators (which is conversely proportional to $\mathcal{H}$) and the number of nodes replying to the election initiator (which is proportional to nodes density and $\mathcal{H}$). The traffic induced by the exchange of Bloom filters is directly proportional to the number of directories deployed over the network. The figure also gives the cost associated with forwarding service queries and publicities (corresponding to unicast messages), which is proportional to $\mathcal{H}$ since the number of hops directly impact on the average distance between a client node and a directory. Taking into account the cost of directory deployment (election messages, Bloom filter exchange and directory assignment), it appears that an optimal value for $\mathcal{H}$ is 2. However, since the client's requests rate directly impacts on the forwarding of service queries, the value of $\mathcal{H}$ must take it into account and be changed on the fly, so as to be conversely proportional to the client's requests rate.

We provide in Figure 11 an evaluation of the average response time per node according to node velocity, consid-
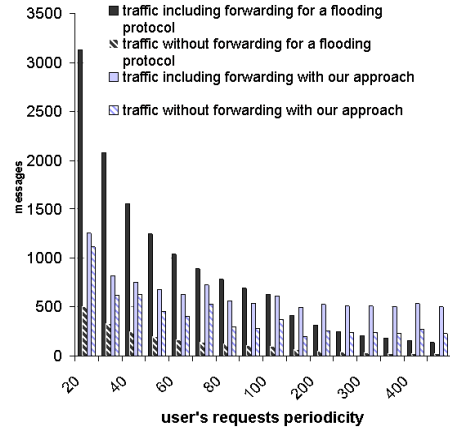
ering a topology of 1000m $\times$ 1000m composed of $N = 50$ and $\mathcal{H} = 2$ nodes. This evaluation demonstrates that the service discovery protocol is quite insensitive to nodes velocity. This result is due to the fact that an increased mobility of directories and service providers causes a higher dissemination of the service descriptions among directories.When a client requests a directory, the probability that the requested service description is held by the requested directory is higher. The overhead (in term of delay) caused by the link failures is balanced by this fact. Note that when path failure occurs, directories are still reachable (thanks to the multi-hop routing protocol).
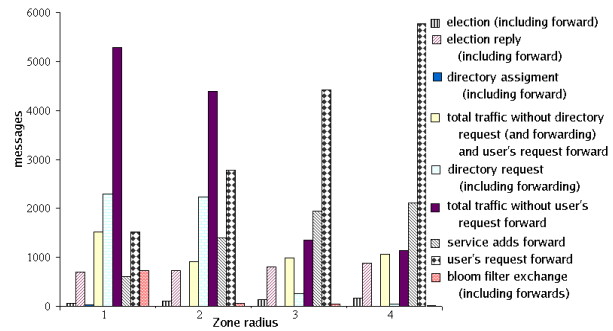
6. Conclusion

The vision of pervasive computing and/or ambient intelligence requires enabling ubiquitous computing and networking so that mobile users can seamlessly get access to
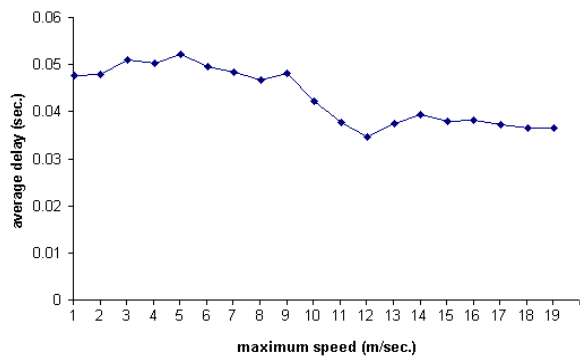
Figure 11. Average delay comparison

digital services anywhere, anytime. MANETs are one enabler of such a vision, providing networking capabilities to mobile devices without requiring any infrastructure. However, the specifics of MANETs such as potentially highly dynamic topology and networking of heterogeneous wireless nodes whose energy needs to be saved for enhanced autonomy, require special care in the handling of distributed service provisioning. In particular, the discovery of services must allow accessing services of the overall MANET for increased availability, while limiting resource consumption in the network. Existing discovery protocols, including the most recent ones specifically designed for MANETs, tend to induce significant traffic overhead, which make them primarily suitable for networks comprising a few number of nodes (i.e., less than 30 according to published performance). This paper has introduced a scalable service discovery protocol aimed at MANETs composed of up to a hundred of nodes, and further supporting bridging with other networks, either ad hoc or infrastructure-based. While decentralized service discovery is a priori more suited to the specifics of MANETs by not assigning any specific role to given nodes, it is too costly in terms of traffic. But, centralized service discovery based on service directory(ies) either makes the protocol too much dependent on specific node(s) or induces too much resource consumption in the network.

Our solution relies on the dynamic and homogeneous deployment of cooperating service directories within the MANET. The MANET is structured around a directory backbone in which directory coverage is defined with respect to a given number of hops, which is set according to the nodes density, as known from the underlying routing protocol. Dynamic deployment of directories is then carried out so that directories cover distinct areas. Scalability of our solution further comes from the fact that: (i) our protocol minimizes the generated traffic using border-casting, and (ii) the directory content is summarized using a Bloom filter, which provides a highly compact summary that may be exchanged with other directories, and is used

to accurately locate the directory that holds the description of a given service. We have evaluated our solution through simulation, which shows the efficiency of our protocol. We are currently implementing our protocol as a middleware service of the WSAMI middleware[13], which supports the composition of Web services deployed over mobile nodes in a hybrid network.

References

[1] B. Bloom. Space/time trade-offs in hash coding with alowable errors. *Comunication of the ACM*, 13(7), July 1970.

[2] P. Castro, B. Greenstein, and R. Muntz. Locating application data across service discovery domains. In *Mobile Computing and Networking*, pages 28–42, 2001.

[3] D. Chakraborty, A. Joshi, and T. Finin. GSD: a novel group-based service discovery protocol for MANETs. In *Proc. of IEEE MWCN*, 2002.

[4] T. Clausen and P. Jacquet. Optimized link state routing protocol. IETF RFC 3626, www.ietf.org, October 2003.

[5] S. Das, C. Perkins, and E. Royer. Ad hoc on demand distance vector routing IETF RFC 3929. www.ietf.org, November 2002.

[6] M. Denny, M. Franklin, and P. Castro. Mobiscope: A scalable spatial discovery service for mobile network resources. In *Proc. of ACM MDM*, 2003.

[7] E. Guttman. Service Location Protocol: Automatic discovery of IP network services. *IEEE Internet Computing*, 3(4), July 1999.

[8] J. Haas. A new routing protocol for the reconfigurable wireless networks. In *Proc. of IEEE ICUPC*, 1997.

[9] S. Helal, N. Desai, V. Verma, and C. Lee. Konark: A service discovery and delivery protocol for ad hoc networks. In *Proc. of WCNC*, 2003.

[10] V. Issarny, D. Sacchetti, F. Tartanoglu, and al. Developing ambient intelligence systems: A solution based on web services. *Journal of Automated Software Engineering*, 2004. To appear.

[11] V. Issarny, F. Tartanoglu, J. Liu, and F. Sailhan. Software architecture for mobile distributed computing. In *Proc. of the 4th IEEE/IFIP WICSA*, 2004.

[12] N. Klimin, W. Enkelmann, and H. Karl. A hybrid approach for location-based service discovery in vehicular ad hoc networks. In *Proc. of WIT*, 2004.

[13] U. Kozar and L. Tassiulas. Service discovery in mobile ad hoc networks: An overall perspective on architectural choices and network layer support issues. *Ad Hoc Networks*, 2, 2004.

[14] J. Liu and V. Issarny. On service allocation in selfish mobile ad hoc networks. In *In Proc. of EDBT PIM*, 2004.

[15] J. Liu and V. Issarny. QoS-aware service location in mobile ad hoc networks. In *Proc. of ACM MDM*, 2004.

[16] R. Rivest. The md5 message-digest algorithm. In *IETF RFC 1321, www.ietf.org*, 1992.

---

[13]http://www-rocq.inria.fr/arles/download/ozone

COMPUTER SOCIETY