

# Proximity-Based Service Discovery in Mobile Ad Hoc Networks

René Meier, Vinny Cahill, Andronikos Nedos, and Siobhán Clarke

Distributed Systems Group, Department of Computer Science,  
Trinity College Dublin, Ireland  
{rene.meier, vinny.cahill, andronikos.nedos,  
siobhan.clarke}@cs.tcd.ie

**Abstract.** Existing approaches to service discovery have been developed primarily for environments with a fixed network backbone and typically rely on centralized components being accessible to potential service clients at any given time. The characteristic lack of a designated service infrastructure in combination with the highly dynamic nature of the underlying network topology renders such discovery mechanisms unsuitable for mobile ad hoc environments. This paper presents an approach to the discovery of ad hoc services that exploits the fact that the relevance of such services is often limited to specific geographical scopes. Service providers define the areas (proximities) in which their services are available. Clients register interest in specific services and are subsequently informed whenever they enter a proximity within which these services are available. Since ad hoc services can be stationary or may be moving with the location of their mobile providers our approach supports discovery of services with fixed locations as well as of those that migrate with their providers. Our approach has been implemented as a push-based proximity discovery service and its evaluation demonstrates that it is well suited for highly dynamic networks as it maintains neither routes nor overlay network topologies.

## 1 Introduction

Many mobile applications can be characterized as collaborative in the sense that mobile nodes use the wireless network to interact with other mobile nodes that have come together at some common location. Collaborative nodes typically come together in some area, establish associations with other collaborative nodes dynamically, and make use of common services already available in that locality or provided by members of the group. The members of such a group may migrate together, like a fleet of vehicles or the participants in a guided tour. Although these applications may use infrastructure networks, they will often use ad hoc networks since they are immediately deployable in arbitrary environments and support communication without the need for a separate infrastructure. This collaborative style of application may be useful in the ubiquitous [1] and sentient computing [2] domain allowing loosely-coupled, highly-mobile components to communicate and collaborate in a spontaneous manner.

Entities that comprise this style of application are inherently mobile and may move together and apart over time. Service providers and service clients characteristically come together at a certain location, use services, and later come together at a different location to use other services. Hence, providers require a means to advertise their services and clients need be able to locate services of interest depending on their current location.

Existing approaches to service discovery [3-5] have been developed primarily for environments with a fixed network backbone and typically rely on centralized components being accessible to potential service users at any given time. The characteristic lack of a designated service infrastructure in combination with the highly dynamic nature of the underlying network topology renders such discovery mechanisms unsuitable for ad hoc environments. This paper presents a decentralized mechanism for the discovery of services in ad hoc environments that exploits the fact that *the relevance of such services is typically limited to specific geographical scopes*. Example services from the traffic management domain might include an accident warning service provided by a crashed car to approaching vehicles and a warning service provided by an ambulance rushing to an accident site to inform nearby vehicles to yield the right of way. Providers define the geographical scopes, called proximities, in which their services are available. Our Proximity Discovery Service (PDS) allows such providers to advertise their services in proximities, thereby bounding advertisements to the areas in which the services are relevant. Interested clients are informed whenever they enter a proximity where these services are available. The PDS therefore enables clients to dynamically discover available services (and their proximities). Clients subsequently use the discovered information to establish logical connections to the associated providers. The connections between the entities residing in a particular proximity are then used by providers to deliver their services hence, *allowing clients to use them at the specific location where they are valid*. This concept of proximity-based service discovery enables collaborating entities to come together at a certain location and to spontaneously discover the services available at that location.

Clients must register interest in specific services with the PDS in order to be informed whenever they enter a proximity where these services are available. Such clients may move from one proximity to another without re-registering their interest. Thus, registrations are persistent and will be applied transparently whenever a client enters a new proximity. This implies that a registration to a specific service type applies to all proximities in which these services are provided even though the client may only use a subset of these services at any time. A single registration may result in a service by different providers in multiple proximities being discovered. For example, a client representing a vehicle that registers interest in an accident warning service will be informed whenever it enters the proximity of such a service.

Services can be stationary or may be moving with the location of their mobile providers. A stationary service is attached to a fixed point in space whereas a mobile service is mapped to a moving position represented by the location of its mobile provider. Hence, a mobile service moves with the location of the provider to which it has been attached. An example of such a mobile service might include an information service provided by an entity representing an ambulance on a call. This implies that proximities may be moving with their service. The PDS therefore supports *discovery*

*of services (and proximities) with a fixed location as well as of those that migrate with their providers.*

The PDS has been implemented as part of an event-based middleware for wireless ad hoc environments [6] where it is used for the discovery of relevant event producers. However, the discovery paradigm of the PDS can be used as a means for locating arbitrary services. The evaluation of the PDS demonstrates that using proximity to bound multicast-based service discovery is well suited for highly dynamic networks as it limits the complexity of the underlying dynamic network topology without introducing overhead for maintaining discovery routes and overlay network structures.

The remainder of this paper is structured as follows: Section 2 surveys related work. Section 3 presents our architecture for discovering services in mobile environments and describes how this architecture maps to the underlying ad hoc network. Section 4 outlines our evaluation of the PDS. Finally, section 5 concludes this paper by summarizing our work.

## 2 Related Work

The concept of service discovery is well established in distributed systems [7], since networked entities need to locate available remote resources in order to use them. Work on service discovery in mobile ad hoc networks focuses on using decentralized architectures to overcome the limitations of traditional discovery mechanisms, such as SLP [3], Jini [4], and UPnP [5], that rely on fixed infrastructure. Research in service discovery architectures for such mobile environments can be classified into lower-layer service discovery protocols [8-11] and higher-layer service platforms [12-14].

Service discovery protocols place emphasis on efficiency and distributed infrastructure while service platforms focus on providing a middleware layer that enables applications to use a service oriented programming model. The PDS bridges this gap by providing a programming model for service discovery based on location while exploiting a proximity-based multicast protocol that limits the complexity of the underlying ad hoc network topology without introducing overhead for maintaining routes and overlay network structures.

Research in the area of discovery protocols has produced encouraging results with some protocols exploiting location to achieve distributed discovery, while others rely on the formation of a virtual backbone of nodes which can subsequently act as service brokers. Protocols that use location for service discovery include the Grid Location Service [10] and the Distributed Location Management [9] protocol. Despite primarily being protocols that enable the distributed lookup of any node's location, in the context of service discovery, location can be considered as just another service. The use of location in such protocols offers nodes a kind of "topological awareness" and aids the distributed execution of their respective algorithms. The PDS differs in its use of location, as the above protocols require stricter assumptions such as knowledge of a geographic grid that must be shared by all operating nodes. The PDS has no such requirement and uses location as a constrain to enhance scalability and relevancy.

Protocols like [11] and [8] provide a fully distributed service discovery mechanism based on using selected mobile nodes that form a virtual backbone. The nodes in the

backbone subsequently act as service brokers, permitting service registration and thus, facilitate service discovery. Such protocols have proved to perform well under simulation compared to broker-less service discovery mechanisms that use multicast or anycast. However, they require specialized backbone formation and communication protocols which, coupled with the cost incurred by maintaining the virtual backbone, can result in reduced performance under different mobility scenarios. In contrast, the PDS requires no maintenance phase other than maintaining the multicast tree and increases efficiency by limiting packet forwarding to the proximity area.

Konark [14], GSD [13], and DEAPspace [12] provide service discovery platforms. Konark is a service discovery mechanism that is based on a hierarchical classification of services. Its architecture supports both push and pull style discovery, but is limited to locally scoped multicast. In comparison, the PDS hides the complexity of the underlying communication system by providing an asynchronous advertise/discover programming model to the application layer that is also independent of the structure of service representation. This has the extra benefit of allowing the PDS to work with multiple service representations. GSD uses caching and semantic matching of services to efficiently discover services in a distributed manner. GSD uses a predefined ontology to group similar services and forwards service requests efficiently by exploiting this grouping. The PDS uses a more sophisticated forwarding mechanism that relies on the notion of proximity rather than hop count. DEAPspace has been designed for pervasive environments and shares the same discovery paradigm as the PDS, namely push based discovery, but has a more narrower scope than the PDS as it is directed towards personal area networks. It was designed for small devices and provides energy efficient mechanisms for message dissemination. However, there are no clear guidelines as to how applications interact with DEAPspace. The PDS on the other hand, supports a concise application programming interface and its use of proximity implies support for single-hop and multi-hop ad hoc networks.

### 3 The Architecture for Service Discovery

The design of the architecture for service discovery is motivated by the hypothesis that there are services that are more likely to be used by clients once they are in close proximity. This means that the closer clients are located to a provider the more likely they are to be interested in the services that it offers. Significantly, this implies that services are relevant within a certain geographical area surrounding a provider.

The architecture is inherently distributed in order to support ad hoc environments. As shown in Fig. 1, the middleware that includes the PDS is exclusively collocated with application components and depends neither on centralized or intermediate components nor on the presence of a designated infrastructure. Every mobile device hosting the middleware therefore offers identical capabilities to its application without accessing remote components. Applications may connect a variable number of providers and clients to the middleware on each mobile device, thereby allowing individual devices to offer and to use one or more services. Each of these entities regulates its use of the PDS by exploiting the concepts of advertisement and registration. Providers advertise the proximities in which their services are available.

Clients register interest in specific services in order to be informed whenever they enter a proximity where these services are available until they un-register. The PDS augments these well-known and widely-used concepts in order to support scoped service discovery and ultimately mobility. The proximity information announced by providers is exploited primarily to establish communication relationships between mobile entities while registrations are used locally to map client interests to the currently available services.

The PDS has been designed to discover services for entities that can be either stationary or mobile. This implies that the PDS as well as the entities hosted by a particular mobile device are aware of their geographical location at any given time. The middleware therefore includes a Location Service (LS) that uses sensor data to compute the current geographical location of its host device and entities. To suit outdoor applications, for example in the traffic management domain, the middleware exploits a version of the LS that uses a GPS satellite receiver to provide latitude and longitude coordinates.

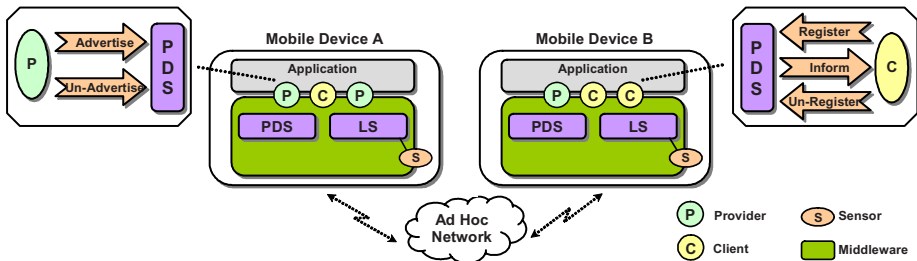


Fig. 1. Discovery service architecture

### 3.1 Stationary and Mobile Services

The PDS supports both stationary and mobile entities and consequently must be able to handle stationary and mobile services and their proximities. The proximity definition below shows that this notion of proximity is defined firstly by the covered area, which is described as a geometric shape with associated dimensions and a reference point that is relative to this shape, and secondly by a naval location. The reference point of a stationary proximity is attached to a naval represented by a fixed point in space whereas the reference point of a mobile proximity is mapped to a moving naval, which is characteristically represented by the location of a specific mobile provider. Hence, a mobile proximity moves with the location of the provider to which it has been attached. For example, a group of vehicles heading in the same direction may cooperate to form a platoon in order to reduce their consumption of fuel. These vehicles might use a mobile service provided by the leading vehicle. The proximity of such a service might be attached to a naval represented by the position of the leader thereby moving with its location. Proximities may be of arbitrary shape and may be defined as nested and overlapping areas. Nesting allows a large proximity to contain a smaller proximity subdividing the large area.

$$\text{Proximity} = [\text{Area (Shape, Dimensions, Reference Point), Naval}]$$

### 3.2 Identifying Services

There are two essential issues that need to be addressed when discovering services. Firstly, an addressing scheme for uniquely identifying services is required and secondly, a means for clients to obtain the correct identifiers needs to be provided. An approach to addressing these issues, based on statically generating a fixed number of unique and well-known identifiers, has been described by Orvalho et al. [15]. Another approach might involve using a centralized lookup service for generating and retrieving identifiers. However, neither of these approaches suffices for applications that accommodate a dynamically changing number of services and depend on an inherently distributed architecture.

The PDS exploits a decentralized addressing scheme based on a Distributed Hash Table (DHT) in which identifiers representing services can be computed from discovered service descriptions. Each combination of service name and proximity (shape dimensions, reference point, and naval location) is considered to be unique throughout a system assuming that there is no justification for providers to define multiple identical service name and proximity pairs for different services. A textual description of such a pair is used as stimulus for a hashing algorithm to dynamically generate hash keys that represent identifiers using node local rather than global knowledge. Clients compute the corresponding identifier upon discovery of a service and subsequently use these identifiers to connect to services.

This distributed addressing scheme replaces the kind of centralized approach traditionally used for identifying services of interest and therefore represents a key enabling mechanism for the inherently distributed architecture of the PDS. It enables mobile devices to recognize services of interest and to *locally* compute identifiers from serialized service descriptions.

**Computing Service Identifiers.** The PDS uses a hashing algorithm that generates 24 bit service identifiers from variable length character strings. The implemented algorithm is based on a combination of a hash function for such stimuli proposed by [16] with the use of a hash function multiplier [17] in order to reduce the chance of collisions. Fig. 2 depicts the structure of the character strings that describe services

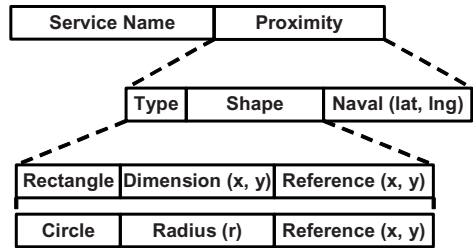


Fig. 2. Computing service identifiers

with rectangular and circular proximities. Such strings comprise the service name and the particulars of the associated proximity. Proximities are described by the dimensions and reference points of their shapes and by the coordinates that specify the location of their navals. Note that the proximity of a mobile service always describes its initial naval location since its actual naval location might change over time therefore enabling clients to generate consistent identifiers.

**Collisions.** Depending on a number of factors, including the quality of the hash function and the ratio of stimuli to potential identifiers, this approach might lead to colliding identifiers. Such collisions occur when two distinct stimuli  $x$  and  $y$  generate

the same identifier. In other words, there exist stimulus pairs, such that  $x \neq y$ , for which  $h(x) = h(y)$ . Collisions may result in different services using the same identifiers. This does not affect a system provided that such services are used in different geographical scopes, i.e., that their proximities do not overlap. Overlapping proximities with colliding identifiers can lead to unwanted service messages being received by certain mobile devices. Such devices can be prevented from delivering unwanted service messages to their clients by a run-time type checking mechanism. Such a mechanism can detect and eventually discard these service messages. Hence, colliding identifiers may lead to additional use of communication and computational resources, but will not cause delivery of unwanted service messages and are in any case unlikely due to geographical separation.

### 3.3 Discovering Services

The PDS runs on every physical machine that hosts service users regardless of whether these local entities act either as providers or as clients, or indeed as both. The PDS uses beacons to periodically advertise relevant services (and the associated proximities) on behalf of providers. The discovery service advertises the service name and proximity pairs that have been defined by providers within the scope of a proximity. This implies that the location at which these advertisements are disseminated can change when the device hosting a PDS migrates and that the set of adjacent devices is likely to change as well. Moreover, this also implies that other stationary or mobile devices may need to forward advertisements for them to reach the boundaries of the proximity.

```

advertise(serviceName sn, proximity p)
un-advertise(serviceName sn, proximity p)
register(serviceName sn, discoveryHandler dh)
un-register(serviceName sn, discoveryHandler dh)

discoveryHandler {
    inform(serviceName sn, proximity p, status s) {
        serviceIdentifier = h(sn + p.toString())
        //use discovered service
    }
    ...
}

```

Fig. 3. The application programming interface of the PDS

**Using the Discovery Services.** Fig. 3 outlines the application programming interface supported by the PDS. These interface functions reflect the fact that the PDS is based on an implicit discovery model as they refer neither to explicit entities nor to designated components of any kind. Instead, the functions for advertising (un-advertising) and registering (un-registering) interest refer to a service name.

Providers and clients must use a common vocabulary defined by the application to agree on the name of a service. Such a *serviceName* typically describes a service type rather than referring to a service provided by a particular provider. This allows clients to discover services based on their availability (at a location) regardless of the specifics of their providers.

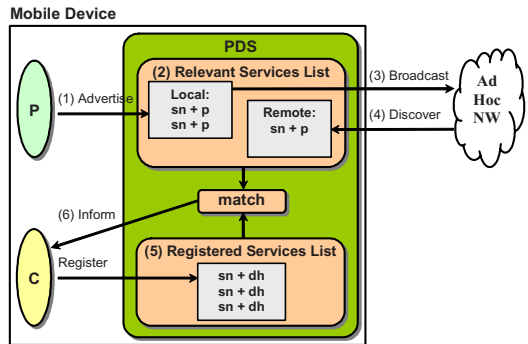


The PDS application programming interface also illustrates how providers advertise service proximities and how clients specify discovery handlers. Clients implement discovery handlers and subsequently register them. The PDS uses these handlers to inform clients of services that either have become available or have expired. Clients use *serviceName* and *proximity* pairs to generate service identifiers and, depending on the service *status*, commence or discontinue using the service. This approach enables clients to potentially register a specific discovery handler whenever they express their interest. Depending on their requirements, clients may therefore provide either a discovery handler for an individual service or for a set of services.

**The Discovery Mechanism.** The PDS allows its providers to advertise and its clients to discover relevant services according to the following discovery mechanism:

1. Initially, a proximity discovery service recognizes the services advertised by its local providers.
2. A proximity discovery service maintains a list describing all services that are relevant at its current location. A specific service description is discarded when the hosting mobile device leaves the proximity associated with this service description.
3. A proximity discovery service periodically broadcasts messages describing the services advertised by its local providers within the proximity of the advertised service relative to its current location. The broadcast period is application specific and hence, can be configured for each individual proximity discovery service.
4. A proximity discovery service receiving an advertisement adds a service description to its own list if it is currently located inside the associated proximity.
5. A proximity discovery service maintains a list describing the interests registered by its local clients.
6. Clients are informed whenever a service of interest is added to or discarded from the list of relevant services.

Fig. 4 illustrates the components of our approach to maintaining lists representing relevant services and services of interest respectively. The proximity discovery mechanism essentially comprises two algorithms, one for disseminating (locally-defined) service advertisements and another for handling (remotely-defined) service advertisements upon receiving them. Both algorithms operate on the relevant services list of a particular mobile device and collaborate in order to maintain the relevant locally and remotely specified services. Hence, relevant services lists are maintained according to geographical relevance and are independent of the set of issued registrations.



**Fig. 4.** The discovery mechanism



**Advertising Services.** The algorithm for broadcasting and to a certain extent maintaining service descriptions on a particular mobile device periodically traces through all service descriptions stored in a relevant services list in order to advertise the services that have been defined by local providers and to verify the geographical validity of service proximities. Hence, the algorithm essentially identifies stationary and mobile services with proximities that are relevant at the actual location of the mobile device and periodically advertises those locally-specified services.

Stationary services remain in a list as long as their scope includes the current location regardless of whether they have been specified locally or remotely. A service is removed from the list once the device has left the associated proximity. Removing a relevant service may lead to a change to the set of services available to local clients. Clients that have registered with such a service are informed of its cancellation. However, their interest, expressed by a related registration, remains in the registered services list.

The dynamic aspect of mobile services causes some variation in the means by which they are maintained and advertised. Mobile services specified by local providers always (by definition) include the actual location of the mobile device and migrate together with the device. Consequently, the migration of a mobile device does not cause mobile services to expire. These services can therefore be advertised without verifying their validity. In addition to enabling remote devices to discover mobile services, these advertisements provide a means for disseminating location updates describing the migration of mobile services. The naval of the proximity of every mobile service is therefore updated with the latest device location prior to being advertised. The validity of remotely-specified mobile services is verified when updates on their latest locations are received. This prevents validity checks using cached and therefore potentially obsolete (due to migration) location information.

**Locating Services.** The algorithm for handling received service descriptions adds newly discovered services to the relevant services list of any mobile device residing inside the proximity while known services, whose proximities no longer include the device's location, are removed from the list. Clients with a corresponding registration are informed of newly discovered services as well as of cancelled services. Other service descriptions comprise information that is either irrelevant at the current location or already known and are therefore discarded.

Remotely-specified services are handled similarly regardless of whether they describe stationary or mobile services. Both stationary and mobile services that are no longer relevant at the current location are removed from the repository. Mobile services are removed based on their latest naval location while being identified according to their initial naval location. This enables the PDS to identify mobile services even though their actual naval location changes.

### 3.4 Supporting Ad Hoc Networks

The PDS implements a location-based routing protocol that exploits proximity to control multicast-based flooding of the underlying ad hoc network [18]. This approach uses a well-known multicast group to provide a means for disseminating service descriptions in a one to many manner within the boundaries defined by a

proximity without introducing extra overhead for maintaining routing information. Such routing information characteristically needs to be updated more frequently with increasing speed of mobile devices and thus, approaches that attempt to maintain routing information are less suited for applications comprising highly-mobile entities [19].

Providers may define proximities for their services independently of the physical transmission range of their wireless transmitters with the underlying communication system routing messages from provider to client using a multi-hop protocol. Nodes residing within the boundaries of a proximity forward and deliver these messages while nodes located outside a proximity discard them without forwarding.

The concept of proximity provides an application level abstraction for bounding the message dissemination scope. However, proximities can also be exploited in order to optimize the delivery of specific messages. The PDS uses a provider's geographical location and radio transmission range in conjunction with the proximity of its services to determine whether to use single-hop or multi-hop messages. The PDS therefore supports both single-hop and multi-hop dissemination and typically uses cost efficient single-hop messages for advertising services in proximities that are likely to be covered by a provider's wireless transmission range and employs multi-hop messages only when transmitting advertisements beyond this range.

## 4 Experimental Results

This section evaluates the proximity-based approach to the discovery of ad hoc services that is proposed in this paper. The main objective of the experiments is to assess the cost of service discovery and more specifically, to assess how exploiting proximity limits advertisement forwarding without the need for control messages. The experimental results demonstrate that discovery cost for stationary services are comparable to those for mobile services and that cost neither depends on the speed of clients nor on the speed of the service provider. This evaluation therefore demonstrates that using proximity to bound multicast-based service discovery is well suited for highly dynamic networks as it limits the complexity of the underlying dynamic network topology without introducing overhead for maintaining discovery routes and overlay network structures.

The primary measurement of interest is an abstract quantity we refer to as *cost*. We assign a relative cost to the dissemination of a single advertisement. Cost describes the number of messages required when propagating an advertisement from a provider to the clients residing within its radio transmission reach and the forwarding of this message to clients beyond this range. Hence, cost depends on the number of clients residing within a particular proximity and provides a qualitative indication of the bandwidth required for discovering a service. For example, the cost of a provider disseminating an advertisement to three clients, each of which forwards the message that describes the service once, is described as 4; one message sent by the provider and 3 messages forwarded by the clients. This allows us to measure cost independently of the actual frequency of an advertisement while varying a number of application parameters. These parameters include the migration speed of the entities,

the range of the proximity within which services are advertised, and the number of clients.

All evaluation experiments have been conducted by implementing the selected scenarios as prototypical service applications. Each of the providers and clients that comprise these applications is represented by an independent PDS instance and interacts with other entities using a real ad hoc network. The entities and their PDS instances are hosted by a number of notebook computers running the Microsoft Windows XP operating system on a 1GHz Intel Pentium III processor. Each machine is equipped with a Lucent Orinoco Gold WiFi PCMCIA card with a channel capacity of 11 Mbit/s providing the ad hoc network connection for inter entity communication. One or more entities may reside on a notebook computer. Entity locations and mobility are simulated throughout these experiments using the location service of their respective PDS instances while the hosting notebook computers were placed within ad hoc communication reach of each other. The radio transmission range  $T_r$  for each machine was simulated to be  $T_r = 200$  meters. This ensures that an entity discards all communication messages received from entities located beyond  $T_r$ , even though the distance between the physical locations of their host machines is less than  $T_r$ . Multiple runs were conducted for each experiment and the data collected was averaged over those runs.

#### 4.1 The Application Scenarios

We have selected two application scenarios from the traffic management domain for this evaluation. *Scenario<sub>a</sub>* models a broken down car providing a stationary accident warning service to vehicles within its vicinity. This scenario is set on a two way road with a provider acting as the broken down car advertising its service to clients representing the vehicles. These vehicles are randomly distributed on the lanes of the road. *Scenario<sub>b</sub>* models an ambulance providing a mobile warning service to nearby vehicles for them to yield the right of way. This scenario is set similarly to the previous scenario on a two way road and comprises an ambulance advertising its service to a number of randomly distributed clients representing vehicles. Both scenarios include a circular shaped proximity with a central reference point that is mapped to the location of the respective provider and involve a road section of 1400 meters with the provider being located at the center of the section. All clients move on this road section and their number defines the saturation of the area.

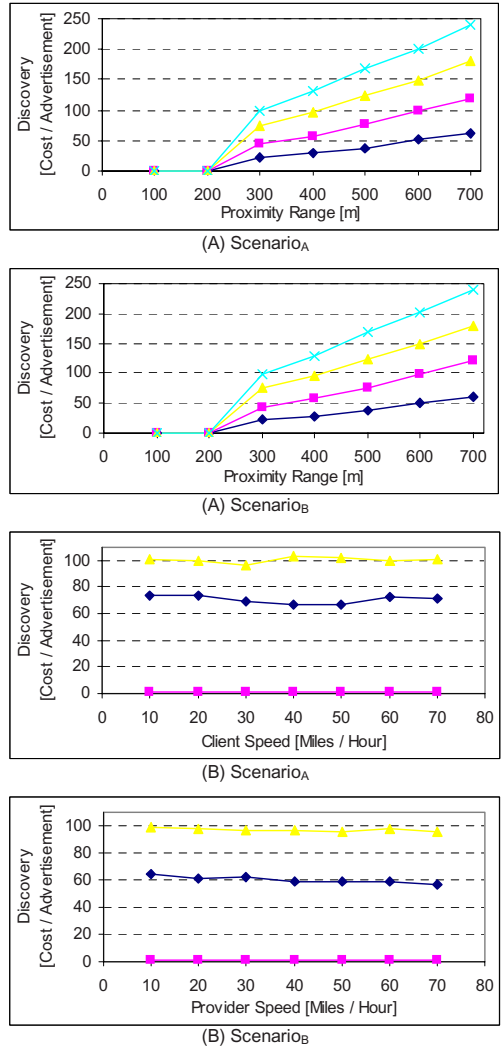
#### 4.2 Results and Analysis

Fig. 5 (A) shows the discovery cost as a function of proximity range  $P_r$ . The respective proximities define the set of clients residing inside the area of interest that discover the service. These ranges have been selected to include services with proximities in which all clients can be reached using a single-hop radio transmission ( $P_r \leq T_r$ ) as well as those that require multi-hop routing ( $P_r > T_r$ ). The largest proximity covers the whole scenario area enabling all clients on the road section to discover the service. The results essentially demonstrate how proximities bound service discovery cost by bounding the number of clients that forward a certain

advertisement. They show a significant difference when comparing discovery cost within the single-hop reach to the cost beyond this range.

Beyond the single-hop range, all results show similar tendencies of increasing cost with expanding proximities and rising saturations as every client residing inside a certain proximity forwards advertisements. This illustrates that exploiting proximity enables the PDS to transparently select the appropriate protocol when disseminating advertisements. The PDS uses its cost efficient single-hop protocol for advertising services with proximities that are covered by the provider's  $T_r$  and employs the multi-hop version only when transmitting advertisements beyond  $T_r$ . Other discovery service platforms typically use either a single-hop protocol with propagation range limitations or a more expensive multi-hop protocol for both short and long range discovery. However, Fig. 5 (A) most significantly illustrates that the results recorded for *Scenario<sub>A</sub>* and *Scenario<sub>B</sub>* are virtually identical demonstrating that, given a similar configuration, the discovery cost for stationary services are comparable to those of mobile services.

Fig. 5 (B) depicts the service discovery costs recorded for *Scenario<sub>A</sub>* and *Scenario<sub>B</sub>* respectively as a function of migration speed. These results show the effect of migration speed and thus, of a dynamically changing network topology that reflects entity movements, on discovery cost. Similar results were recorded for *Scenario<sub>A</sub>* where mobile clients discover a stationary service and for *Scenario<sub>B</sub>* in which a mobile service provided by an ambulance moving along the road is being discovered by stationary clients representing stopped vehicles. These results essentially show that the cost is low within single-hop reach and increases with



**Fig. 5.** (A) Discovery cost as a function of proximity range for saturations of 60, 120, 180, and 240 from bottom to top. (B) Discovery cost as a function of migration speed for proximities with  $T_r = 200, 400,$  and  $600$  meters from bottom to top and for a saturation of 120

service proximities expanding beyond single-hop range. Significantly, the results recorded for *Scenario<sub>A</sub>* and *Scenario<sub>B</sub>* demonstrate that service discovery cost does neither depend on the speed of clients nor on the speed of the provider. This is due to the fact that the PDS exploits proximities to control multicast-based flooding and consequently does not introduce extra overhead for maintaining routing information that needs to be updated more frequently with increasing relative migration speed. The study of flooding-based multicast protocols for ad hoc networks presented by Lee et al. [19] presents similar conclusions and hence, argues that neither the number of transmitted messages nor the associated delivery ratio is a function of the relative speed of the interacting entities.

## 5 Summary and Conclusions

This paper presented a decentralized approach to discovery of services in ad hoc environments. The approach supports discovery of services that are stationary as well as those that move with the location of their mobile providers and exploits the fact that the relevance of such services is often limited to specific geographical scopes. Providers and clients of this style of service characteristically come together at a certain location, use a service, and later come together at a different location to use another service. Providers define the geographical scopes, called proximities, in which their services are available. Our Proximity Discovery Service allows such providers to advertise their services in proximities, thereby bounding advertisements to the areas in which the services are relevant. Clients are informed whenever they enter a proximity where these services are available. The Proximity Discovery Service therefore enables clients to dynamically discover available services. Clients subsequently use the discovered information to establish logical connections to the associated providers. The connections between the entities residing in a particular proximity are then used by providers to deliver their services thereby allowing clients to use them at the specific location where they are valid.

This concept of proximity-based service discovery overcomes the characteristic lack of a designated service infrastructure in ad hoc environments and is well suited for highly dynamic network topologies. The architecture of the Proximity Discovery Service is inherently distributed as it depends neither on centralized nor on intermediate components. A decentralized addressing scheme represents a key enabling mechanism to this inherently distributed architecture allowing mobile devices to recognize services of interest and to locally compute identifiers from service descriptions. The evaluation of the Proximity Discovery Service demonstrated that exploiting proximity to bound multicast-based service discovery limits the complexity of the underlying dynamic network topology without introducing overhead for maintaining discovery routes and overlay network structures. Such information typically needs to be updated more frequently with increasing migration speed of providers and clients causing the use of communication and computational resources to increase with the frequency of network topology changes. The evaluation also illustrated that exploiting proximity enables the Proximity Discovery Service to transparently select the appropriate protocol when disseminating advertisements.

**Acknowledgements.** The work described in this paper was partly supported by the Irish Higher Education Authority's Programme for Research in Third Level Institutions cycle 0 (1998-2001) and by the FET Programme of the Commission of the European Union under research contract IST-2000-26031 (CORTEX).

## References

- [1] M. Weiser, "Ubiquitous Computing," *IEEE Hot Topics*, vol. 26, pp. 71-72, 1993.
- [2] P. Verissimo, V. Cahill, A. Casimiro, K. Cheverst, A. Friday, and J. Kaiser, "CORTEX: Towards Supporting Autonomous and Cooperating Sentient Entities," in *Proceedings of the European Wireless Conference*. Florence, Italy, 2002, pp. 595-601.
- [3] E. Guttman, C. Perkins, J. Veizades, and M. Day, "Service Location Protocol, Version 2," IETF, 1999.
- [4] K. Arnold, R. Scheifler, J. Waldo, B. O'Sullivan, A. Wollrath, B. O'Sullivan, and A. Wollrath, *Jini Specification*: Addison-Wesley Longman Publishing Co., Inc., 1999.
- [5] Microsoft Corporation, "Universal Plug and Play: Background," 1999.
- [6] R. Meier and V. Cahill, "Exploiting Proximity in Event-Based Middleware for Collaborative Mobile Applications," in *Proceedings of the 4th IFIP International Conference on Distributed Applications and Interoperable Systems (DAIS'03)*, LNCS 2893. Paris, France: Springer-Verlag, 2003, pp. 285-296.
- [7] S. J. Mullender and P. M. B. Vitanyi, "Distributed Match-Making for Processes in Computer Networks (preliminary version)," in *Proceedings of the 4th Annual ACM Symposium on Principles of Distributed Computing*. Minaki, Ontario, Canada: ACM Press, 1985, pp. 261-271.
- [8] Z. J. Haas and B. Liang, "Ad-Hoc Mobility Management with Randomized Database Groups," in *Proceedings of the IEEE International Conference on Communications (ICC 1999)*: IEEE Computer Society, 1999, pp. 1756-1762.
- [9] Y. Xue, B. Li, and K. Nahrstedt, "A Scalable Location Management Scheme in Mobile Ad-Hoc Networks," in *Proceedings of the 26th Annual IEEE Conference on Local Computer Networks*: IEEE Computer Society, 2001, pp. 102-112.
- [10] J. Li, J. Jannotti, D. S. J. D. Couto, D. R. Karger, and R. Morris, "A Scalable Location Service for Geographic Ad Hoc Routing," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*. Boston, Massachusetts, USA: ACM Press, 2000, pp. 120-130.
- [11] U. C. Kozat and L. Tassiulas, "Network Layer Support for Service Discovery in Mobile Ad Hoc Networks," in *Proceedings of IEEE INFOCOM*, vol. 23: IEEE Computer Society, 2003, pp. 1965-1975.
- [12] R. Hermann, D. Husemann, M. Moser, M. Nidd, C. Rohner, and A. Schade, "DEAPspace: Transient Ad-Hoc Networking of Pervasive Devices," in *Proceedings of the 1st ACM International Symposium on Mobile Ad Hoc Networking & Computing*. Boston, Massachusetts, USA: IEEE Press, 2000, pp. 133-134.
- [13] D. Chakraborty, A. Joshi, T. Finin, and Y. Yesha, "GSD: A Novel Group Based Service Discovery Protocol for MANETS," in *Proceedings of the 4th IEEE Conference on Mobile and Wireless Communications Networks (MWCN 2002)*. Stockholm, Sweden: IEEE Press, 2002.
- [14] S. Helal, N. Desai, V. Verma, and C. Lee, "Konark -- A Service Discovery and Delivery Protocol for Ad-hoc Networks," in *Proceedings of the 3rd IEEE Conference on Wireless Communication Networks (WCNC)*. New Orleans, USA: IEEE Press, 2002.

- [15] J. Orvalho, L. Figueiredo, and F. Boavida, "Evaluating Light-weight Reliable Multicast Protocol Extensions to the CORBA Event Service," in *Proceedings of the 3rd International Enterprise Distributed Object Computing Conference (EDOC'99)*. Mannheim, Germany, 1999.
- [16] B. Preiss, *Data Structures and Algorithms with Object-Oriented Design Patterns in C++*: John Wiley & Sons, Inc., 1999.
- [17] P. S. Wang, *C++ with Object-Oriented Programming*: PWS Publishing Company, 1994.
- [18] R. Meier, "Event-Based Middleware for Collaborative Ad Hoc Applications," Department of Computer Science, University of Dublin, Trinity College, Ireland, Ph.D. Thesis September 2003.
- [19] S.-J. Lee, W. Su, J. Hsu, M. Gerla, and R. Bagrodia, "A Performance Comparison Study of Ad Hoc Wireless Multicast Protocols," in *Proceedings of IEEE INFOCOM 2000*. Tel Aviv, Israel, 2000.