

Analysis of Decentralized Resource and Service Discovery Mechanisms in Wireless Multi-Hop Networks

Jeroen Hoebeke, Ingrid Moerman, Bart Dhoedt and Piet Demeester

Department of Information Technology (INTEC), Ghent University - IMEC vzw
Sint-Pietersnieuwstraat 41, B-9000 Ghent, Belgium
{jeroen.hoebeke, ingrid.moerman, bart.dhoedt, piet.demeester}@intec.UGent.be

Abstract. The last few years, research in wireless multi-hop networks or ad hoc networks is mainly driven by a search for efficient routing protocols. From an application point of view, it is obvious that nodes (users) will only setup connections with a specific goal, i.e. in order to use services and resources that are available in or reachable through the ad hoc network. Consequently, resource and service discovery protocols that allow nodes to learn available services in the network are indispensable. In this paper we compare the performance of two basic decentralized resource and service discovery techniques, namely proactive and reactive discovery, through simulations and theoretical analysis. Our results show that the choice between them is not straightforward. It highly depends on the network and service characteristics and on the interaction with the underlying routing protocols. Therefore, our analysis provides protocol developers with some guidelines for developing new or extending existing resource and service discovery protocols for operation in mobile ad hoc networks.

1 Introduction

Mobile ad hoc networks are self-organizing mobile, wireless networks that do not rely on any fixed infrastructure for their operation [1]. Nodes that lie within each other's send range can communicate directly. In order to enable connections between nodes that are not directly within each other's send range, intermediate nodes act as routers that forward packets to the destination. In addition, nodes are free to move arbitrarily, resulting in topology changes the protocols have to deal with. Also, the available bandwidth is scarce and places limitations on the protocols' amount of control information. During the last few years, a lot of research efforts were, and still are, focused on the development of efficient routing protocols, as establishing connections between nodes is one of the primary functions the network has to perform. From a higher level, it is clear that nodes will only setup connections with a specific goal, i.e. in order to use services and resources that are available in or reachable through the ad hoc network. Possible services or resources include data storage, database access, files, network printer, Internet gateway... Therefore, resource and service discovery (R&SD) protocols, which allow nodes to automatically locate available resources and

services or to advertise their own capabilities, are also major component of mobile ad hoc networks, which operates in close relation with the routing protocol.

This article presents a comparison of two decentralized R&SD techniques, namely reactive discovery, in which nodes request a service when needed, and proactive discovery, in which nodes periodically announce their services. Section 2 presents an overview of related work and the basic principles of proactive and reactive R&SD mechanisms, followed by a general discussion of the main advantages and disadvantages of centralized and decentralized R&SD protocols in section 3. In section 4, we present a performance evaluation through simulations. In addition, we assess the performance through an analytical approach in section 5. Finally, conclusions are made in the last section of the article.

2 Related Work

R&SD architectures can be dichotomized into centralized and decentralized architectures [2]. In centralized architectures, nodes register their services with service brokers or service agents. When a node needs a service, a request is sent to a broker, which sends back a reply message containing the requested information (see Fig. 1a). The use of agents or brokers improves the scalability, reduces the response time and can be used for load balancing. Most existing resource and service discovery protocols such as the Service Location Protocol, Jini, Universal Plug and Play and Salutation protocol basically rely on directories for their operation, although some of them are able to function without central agents [3].

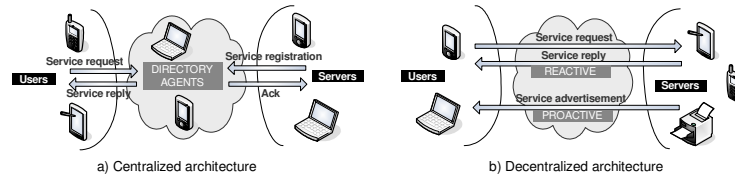


Fig. 1. Logical overview of centralized and decentralized architectures

Another approach is the use of a decentralized architecture that uses reactive discovery or proactive discovery. When using proactive R&SD, nodes periodically announce the services they offer by broadcasting service announcements (SANN) in the network. On reception of these SANNs, nodes extract the information from which they learn about the available services in the network and forward the SANNs to their neighboring nodes. During reactive R&SD, nodes that need a service send out service requests (SREQ), which are propagated in the network. Nodes offering services actively listen for such messages. If they receive a SREQ for a service they support, a reply message (SREP) is generated and sent back to the requesting node. Fig. 1b presents the concept of reactive and proactive R&SD. The Bluetooth Service Discovery Protocol is an example of a decentralized architecture specifically designed for small-scale Bluetooth networks [4].

3 General Discussion of Centralized and Decentralized R&SD Techniques

Management Overhead. A centralized R&SD protocol requires that one or multiple nodes are assigned the task of collecting the service information, exchanging this information amongst each other, keeping it up-to-date and providing service information to nodes that request services. Deploying a centralized solution in an ad hoc network environment can create considerable overhead in managing the central agent(s). In such an environment, no dedicated central agents are present. This implies that a distributed agent selection algorithm is needed to select the most appropriate device that can take up the role of service agent. As the devices can be mobile, can leave and join the network, and are battery-powered, service agents need to be reselected from time to time and their information needs to be transferred. In addition, all nodes in the environment need to become aware of these changes. In a decentralized R&SD solution, whether proactive or reactive, no management overhead is present, as each node independently decides on the actions taken.

Scalability. In centralized solutions most R&SD control traffic will be unicast (or multicast), whereas in decentralized solutions broadcasting is mainly used. The unicast (multicast) will put a lower burden on the wireless multi-hop network. In addition, in centralized solutions all control traffic is directed to the central agents and does not propagate throughout the entire network. Finally, using central agents allows better scalability in terms of the number of services that can be handled.

Resilience. With centralized R&SD protocols, resource and service discovery functionality entirely relies on central agents. This means that these agents can form a single point of failure. In dynamic wireless network environments where nodes are mobile and battery-powered or agent functionality can be reassigned, resilience can become an issue. Therefore, a decentralized solution can provide an alternative or backup method, as functionality is guaranteed at all times.

Network Load. In a centralized solution, all services are registered at a single location. This implies that the load on this location will increase strongly when the number of nodes that need services increases. In an ad hoc network this could result in an unfair network load in specific parts of the network. In decentralized solutions, the network load will be more equally spread over the network. Of course, the exact trade-off between the two solutions will strongly depend on the R&SD patterns in the networks.

Latency. In centralized solutions, the latency to find a requested service depends on the time to forward the request to the central agents and the time to receive a reply. Of course, this latency will mainly consist of routing and forwarding delays. In decentralized solutions, the latency to discover a service or resource is strongly dependent on the type of R&SD: reactive or proactive. The latency in reactive discovery mainly consists of the time to propagate the service request up to a node that offers a service and the time for the service reply to arrive at the requesting node. In proactive discovery, no latency to discover the service is involved.

The above discussion makes clear that decentralized R&SD mechanisms have some interesting characteristics that are highly suited for mobile ad hoc networks. In the following sections we will investigate their performance in more depth.

4 Performance Evaluation

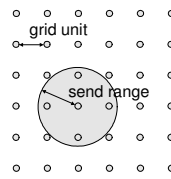


Fig. 2.
Reference
scenario

Table 1. Network parameters

| | |
|-------------------|---|
| Link/Radio layer | 802.11b radio (2.4 GHz, DCF), 11 Mbps, two-ray path-loss, accumulated noise, send range +/- 250 m. |
| Routing Protocol | WRP [6] (proactive) or AODV [7] (reactive) |
| R&SD | number of services, number of clients per service (per period of 150 s), number of servers per service, proactive R&SD: service announcement interval, announcement broadcast delay, reactive R&SD: starting hop count, hop count increment |
| Application layer | CBR traffic from client to server over UDP |
| Mobility | 36 nodes placed in grid, mobility: none or random way-point with constant speed and pause time 0 s |

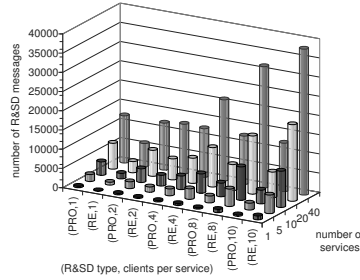
This section presents a simulation analysis of proactive and reactive R&SD techniques. This analysis is focused on the network aspects of the techniques and does not make any assumptions on message syntax, semantics... Fig. 2 and Table 1 present the reference scenario used in the simulations and the relevant network parameters.

4.1 Protocol Overhead and Delay

Fig. 3 and Fig. 4 show the protocol overhead, expressed as the number of R&SD messages (SREQs, SREPs and SANNs), as a function of the number of services and the number of clients and servers per service. Fig. 3 clearly shows that proactive R&SD scales with the number of clients, as its overhead only depends on the number of servers. On the other hand Fig. 4 proves that reactive R&SD scales with the number of servers, as it only depends on the number of clients that request services. The choice whether to deploy proactive or reactive R&SD strongly depends on the service context. This means that developing a scalable decentralized R&SD mechanism should be a hybrid that deploys both reactive and proactive R&SD, where the choice between proactive and reactive depends on the service context and can even be different for different service types. Frequently used services should announce their presence proactively through SANNs, rarely used services should be requested by the clients through SREQs. This also implies that the preferred decentralized protocol is capable of adapting itself dynamically to the service context in the network.

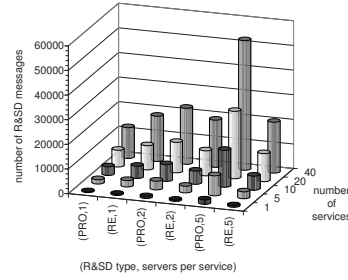
Wireless multi-hop networks require routing protocols, which also rely on proactive and reactive mechanisms. This observation motivates the need to investigate the interaction of decentralized R&SD mechanisms with the underlying routing protocols. To this end, we evaluated the performance of proactive and reactive R&SD on top of

a proactive (WRP) and reactive (AODV) routing protocol (Note that in this study the R&SD messages and routing messages are completely separated). In case a reactive routing protocol is used, different degrees of coupling with the reactive R&SD are possible, as depicted in Fig. 5.



Parameters: grid unit 200m, 1 server per service, announcement interval 60s, no mobility

Fig. 3. Overhead of proactive and reactive R&SD on top of AODV as a function of the number of services and clients per service



Parameters: grid unit 200m, 4 clients per service, announcement interval 60s, no mobility

Fig. 4. Overhead of proactive and reactive R&SD on top of WRP as a function of the number of services and servers per service

In case there is no coupling, AODV will create a route to the client when the server wants to send a service reply (SREP) to the requesting client. In this case, the SREP is sent directly to the client on IP level, which means that the SREP is transparent for the intermediate nodes. This option rules out the possibility of caching without violating the layered protocol structure. In case of loose coupling, the SREQ will interact with the routing protocol and will be used to create a backward path to the client. In this case, the SREP is sent to the client on a hop-by-hop basis and is processed by the R&SD of each intermediate node. Once the client wants to send data to the server, a forward path still needs to be created (but there is no additional routing overhead for sending back the service reply). The last option is strong coupling, where the SREQ will be used to create a backward path and the SREP will be used to create a forward path. In this case, the reactive R&SD protocol will create a bi-directional path if a client requests a service.

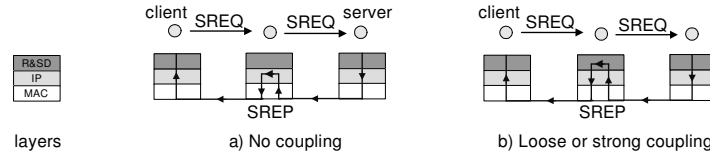
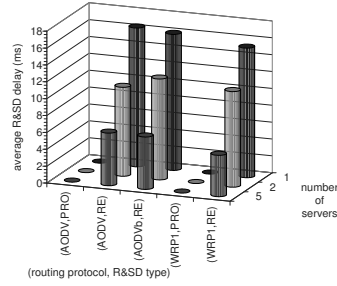


Fig. 5. Interaction of reactive R&SD with reactive routing

Fig. 6 and Fig. 7 illustrate the impact of R&SD and routing protocol combinations on the R&SD delay (time it takes to find the service) and the routing delay (time between finding the service and the delivery of the first data packet). Note that AODV means loose coupling and AODVb means strong coupling. It can be seen that by using proactive R&SD the R&SD delay is reduced to 0 as all nodes in the network

know all services. Also, as expected, the use of a proactive routing protocol results in much smaller routing delays. Fig. 7 also proves that a strong coupling between the R&SD protocol and AODV significantly reduces the routing delay. Again we can observe that the choice of decentralized R&SD mechanism and its interaction with the routing protocol strongly influences the delay. In the ideal case, the R&SD and the routing protocol should adapt their behavior to the network, not only to reduce the network load, but also to deal with the delay requirements imposed by the applications that need services.



Parameters: grid unit 200m, 5 services, 2 clients per service, announcement interval 60s, no mobility

Fig. 6. R&SD delay of proactive and reactive R&SD on top of AODV (reactive) and WRP (proactive)

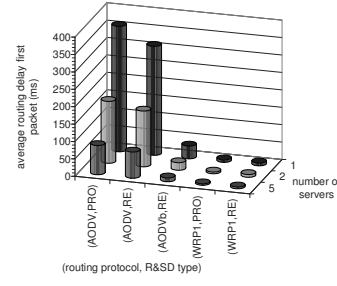


Fig. 7. Routing delay of proactive and reactive R&SD on top of AODV (reactive) and WRP (proactive)

In the above simulation results, basic versions, without any optimizations, of the reactive and proactive R&SD mechanisms were used. For reactive R&SD this means that the SREQ is propagated throughout the entire network. For proactive R&SD this means that servers announce their presence each announcement interval by broadcasting a service announcement in the network. However, a number of optimizations that improve the scalability can be used and are discussed in the next section.

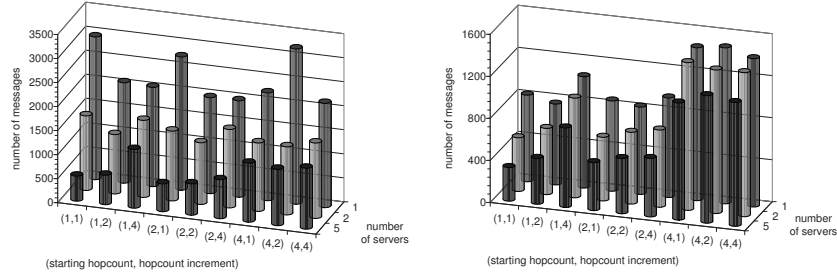
4.2 Scalability Improvements

Improvements to Reactive R&SD: Expanding Ring Search and Caching

Instead of broadcasting the SREQ throughout the entire network, an expanding ring search could be used, where nodes first start to look for the services they need in their immediate environment. If no service has been found, the search area is gradually expanded. Fig. 8 shows the overhead reduction that can be obtained by using the expanded ring search concept. The type of expanding ring search (combination of starting hop count and hop count increment) is influenced by the average number of hops to reach a server (and thus by the number of servers per service).

Another improvement to reactive R&SD could be the use of caches. Clients that have discovered a service or nodes that store service information contained in SREPs can answer the SREQ from other nodes. Of course, information within these caches is only valid for a limited time, which in turn depends on the type of service. Also, by using caches some important information, such as the distance to the server, will be

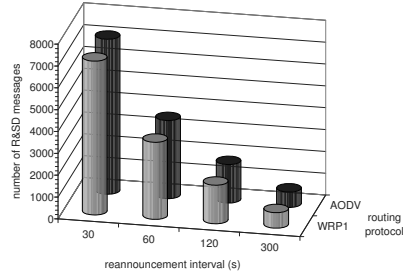
come unavailable. In Fig. 9, clients that have discovered a service will also answer SREQs, and thus constitute a sort of service cache. The results show that caching reduces the number of hops the SREQ needs to travel to find the server that offers the service, which in turn improves the efficiency of using an expanding ring search.



Parameters: grid unit 200m, 5 services, 4 clients per service, no mobility, AODV

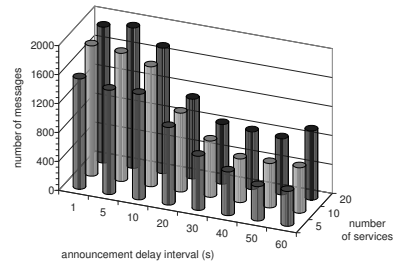
Fig. 8. Impact of expanding ring search on reactive R&SD

Fig. 9. Impact of expanding ring search and caching by clients on reactive R&SD



Parameters: grid unit 200m, 5 services, 2 servers per service, 2 clients per service, no mobility, AODV

Fig. 10. Impact of announcement interval on proactive R&SD



Parameters: grid unit 200m, 4 clients per service, 2 servers per service, announcement interval 60s, no mobility, AODV

Fig. 11. Impact of message aggregation on proactive R&SD

Improvements to Proactive R&SD: Announcement Interval and Message Aggregation

Fig. 10 and Fig. 11 show how the value of the announcement interval and the use of message aggregation help reducing the protocol overhead. The announcement interval determines with which interval servers announce the services they offer. Of course, the optimal value of this parameter cannot be increased as is, without taking into account the type of services and the network context. Message aggregation means that nodes store all SANs that arrive during a certain time period (i.e. announcement broadcast delay interval). At the end of the interval, these announcements are combined into one bigger message, thereby reducing the protocol overhead (or several messages if the value of the announcement broadcast delay is too high, which of course reduces the effect of the aggregation). Note that as we do not make

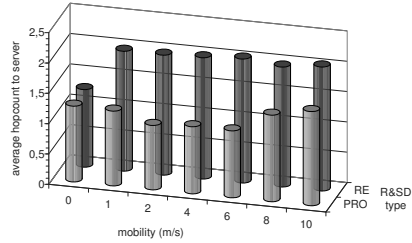
any assumptions on message syntax, we only take into consideration the number of messages, and thus the number of times access to the channel is needed, as a measure for the protocol overhead. Once a choice on the message syntax has been made, more detailed overhead calculations in terms of bytes are possible.

Improvements to Proactive and Reactive R&SD

Of course, more efficient broadcasting schemes than blind flooding can be used to reduce the protocol overhead. Using blind flooding, each broadcast costs N transmissions in a network of N nodes.

Impact of Mobility

If using reactive R&SD, a node that has found a service can start using it. However, the node will not detect if a better service (e.g. the same service but offered by another server closer to this node) has become available, unless it periodically performs a new discovery. When using proactive R&SD, nodes will become aware of better servers and can switch to a new server when appropriate. Fig. 12 shows how switching to a new server can improve the quality of the delivered service (in this case by reducing the hop count when switching to another server that delivers the same service). Of course, the service should support switching to a new server. An interesting example is gateway discovery.



Parameters: grid unit 150m, 2 services, 5 servers and 2 clients per service, announcement interval 30s

Fig. 12. Impact of mobility on proactive and reactive R&SD

5 Theoretical Analysis

In this section we theoretically derive the performance of proactive and reactive R&SD. A theoretical derivation is given for the message overhead of proactive R&SD, with and without message aggregation, and reactive R&SD, with and without expanding ring search. Analog expressions for the R&SD latency can be derived similarly based on the per hop latency. Table 2 shows the notations that are used.

Table 2. Notations

| Notation | Description |
|----------|---|
| N | number of nodes |
| $l(i)$ | Prob[route length between 2 random nodes is i hops]. For the reference scenario in Fig. 2 the values $l(0)$ to $l(10)$ are the following: 36/1296, 120/1296, 196/1296, 232/1296, 232/1296, 200/1296, 140/1296, 80/1296, 40/1296, 16/1296, 4/1296. |

| | |
|------------------|--|
| $L(i)$ | $\text{Prob}[\text{route length between 2 random nodes is } \leq i \text{ hops}] = \sum_{k=0}^i l(i)$ |
| $N(i)$ | number of nodes in a region of i hops around a random node. The following holds: $N(i) \approx L(i) N$ |
| l_{MAX} | maximum route length in hops ($\sum_{i=0}^{l_{\text{MAX}}} l(i) = 1$) |
| L | average route length ($\sum_{i=0}^{l_{\text{MAX}}} i l(i)$) |
| S | number of services |
| s | number of servers per service |
| f_{req} | number of service request per second and per service |
| f_{ann} | number of service announcements per second (per server and per service) for proactive R&SD |
| f_{del} | broadcast announcement delay frequency |
| M_{pro} | message overhead of proactive R&SD (messages per second) |
| M_{re} | message overhead of reactive R&SD (messages per second) |

Basic Proactive and Reactive R&SD

The overhead when using proactive R&SD consists of the service announcements that are broadcasted periodically throughout the entire network. For basic reactive R&SD, the overhead consists of the SREQs that are propagated throughout the entire network and the SREPs that are sent back by all corresponding servers. This leads to the following simple expressions for the message overhead.

$$\begin{aligned} M_{\text{pro}} &= S s f_{\text{ann}} N \\ M_{\text{re}} &= S f_{\text{req}} N + S s f_{\text{req}} L \end{aligned} \quad (1)$$

Proactive R&SD with Message Aggregation

The amount of aggregation depends on the announcement broadcast delay interval. In the optimal case, the choice of f_{del} will be such that at the end of the interval multiple SANNs can be aggregated into one single message. Too small values of the interval will reduce the efficiency, as at the end of the interval there are not always SANNs available for aggregation; too high values will also reduce the efficiency as in this case the SANNs will not fit into a single message anymore. For the optimal case, the overhead is approximated by the following expression:

$$M_{\text{pro}} = f_{\text{del}} N \quad \text{with } f_{\text{del}} < S s f_{\text{ann}} \quad (2)$$

Reactive R&SD with Expanding Ring Search

In order to calculate the message overhead we introduce the notations in table 3.

Table 3. Notations (2)

| Notation | Description |
|--------------------|--|
| $S(k)$ | Prob[minimum 1 server within k hops] |
| $s(k)$ | Prob[nearest server is at k hops] |
| $M_{\text{re}}(k)$ | message overhead of reactive R&SD (messages per second) if the nearest server is at k hops |
| h_{start} | starting hop count |
| h_{inc} | hop count increment |

The probabilities $S(k)$ can be calculated as follows:

$$S(k) = \begin{cases} 1 & , \text{if } N - N(k) < s \\ 1 - \prod_{i=0}^{s-1} \frac{N - N(k) - i}{N - i} & , \text{if } N - N(k) \geq s \end{cases} \quad (3)$$

From $S(k)$ we can easily derive the probabilities $s(k)$.

$$s(0) = \frac{S(0)}{S(I_{MAX})}, \quad s(k) = \frac{S(k) - S(k-1)}{S(I_{MAX})} \quad \text{for } k = 1..I_{MAX} \quad (4)$$

Let us define i_{MAX} as the value for which the following equation holds:

$$h_{start} + (i_{MAX} - 1)h_{inc} \leq k \leq h_{start} + i_{MAX} h_{inc} \quad (5)$$

Now we can calculate the values of $M_{re}(k)$ as follows:

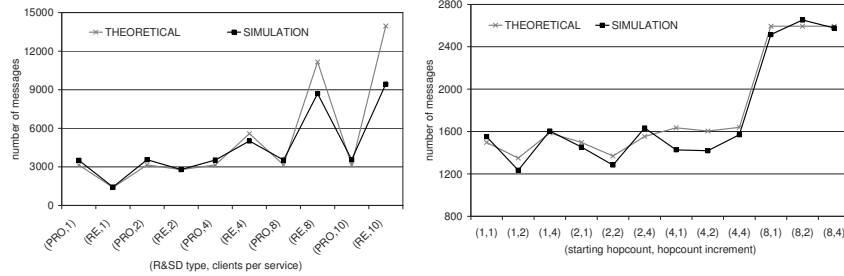
$$M_{re}(k) = N(\min(I_{MAX}, h_{start} - 1)) + k + (s - 1)S(\min(I_{MAX}, h_{start})) \frac{k + \min(I_{MAX}, h_{start})}{2}, \quad \text{if } k \leq h_{start} \quad (6)$$

$$M_{re}(k) = \sum_{i=0}^{i_{MAX}-1} N(h_{start} + i h_{inc} - 1) + N(\min(I_{MAX}, h_{start} + i_{MAX} h_{inc} - 1)) + k + (s - 1) \left[\frac{S(\min(I_{MAX}, h_{start} + i_{MAX} h_{inc}))}{-S(h_{start} + (i_{MAX} - 1) h_{inc})} \right] \frac{k + \min(I_{MAX}, h_{start} + i_{MAX} h_{inc})}{2}, \quad \text{if } k > h_{start} \quad (7)$$

The first equation, which holds if only one iteration in the expanding ring search is needed, consists of three terms: the SREQ, the SREP of the closest server, the SREPs of other servers in the area covered by the SREQ. The second equation, which holds if there are $i_{MAX} + 1$ ($i_{MAX} > 0$) iterations needed in the expanding ring search, consists of four terms: the SREQs of the first i_{MAX} unsuccessful iterations, the SREQ of the last successful iteration, the SREP of the nearest server and the SREPs of other servers in the additional area covered by the last SREQ. Based on the probabilities $s(k)$ and the values of $M_{re}(k)$, we can compute the total overhead as:

$$M_{re} = \sum_{k=0}^{I_{MAX}} M_{re}(k) s(k) \quad (8)$$

In Fig. 13, a comparison is made between the theoretical and analytical results.



Overhead of proactive and reactive R&SD. Parameters: grid unit 200m, 1 server per service, announcement interval 60s, 10 services.

Expanding ring search. Parameters: grid unit 200m, 5 services, 4 clients per service, 2 servers per service.

Fig. 13. Comparison of analytical and simulation results (no mobility)

We can observe that the obtained results are quite similar, which makes the analytical approach useful and reliable for evaluating the overhead in larger ad hoc networks. The only requirement is the knowledge of the probabilities $l(i)$. For random ad hoc networks, the calculation of these probabilities is a research topic on its own.

6 Conclusion

In this paper we have discussed the tradeoffs between centralized and decentralized R&SD techniques. As decentralized solutions have some interesting advantages in mobile multi-hop networks, we have evaluated the performance of reactive and proactive R&SD both through simulations and analytically. Our results show that the choice between them is not straightforward. It highly depends on the network and service context and on the interaction with the underlying routing protocols. Ideally, the R&SD and routing protocols should be developed in close cooperation or even be integrated and should be able to adapt their behavior to the demands of the network and applications [8]. Therefore, our analysis provides protocol developers with some guidelines for developing new or extending existing resource and service discovery protocols for operation in mobile ad hoc networks.

Acknowledgements. This research is partly funded by the Belgian Science Policy through the IAP V/11 contract, by The Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT) through the contract No. 020152, by the Fund for Scientific Research - Flanders (F.W.O.-V., Belgium) and by the EC IST integrated project MAGNET (Contract no. 507102). Jeroen Hoebeke is Research Assistant of the Fund for Scientific Research – Flanders (F.W.O.-V., Belgium)

References

1. C-K. Toh, "Ad Hoc Mobile Wireless Networks: Protocols and Systems", Prentice Hall, 2002.
2. U.C. Kozat and L. Tassiulas, "Service discovery in mobile ad hoc networks: an overall perspective on architectural choices and network layer support issues", *Ad Hoc Networks*, vol. 2, no. 1, 2004, pp. 23-44.
3. C. Lee and S. Helal, "Protocols for Service Discovery in Dynamic and Mobile Networks", *International Journal of Computer Research*, vol. 11, nr. 1, 2002, pp. 1-12.
4. Specification of the Bluetooth System, available from www.bluetooth.com, December 1999.
5. X. Zeng, R. Bagrodia and M. Gerla, "GloMoSim: a Library for Parallel Simulation of Large-scale Wireless Networks", *Proceedings of the 12th Workshop on Parallel and Distributed Simulations (PADS'98)*, Banff, Alberta, Canada, May 1998.
6. S. Murthy and J.J. Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks", *Mobile Networks and Applications*, vol. 1, no. 2, Oct. 1996, pp. 183-197.
7. C.E. Perkins and E.M. Royer, "Ad-hoc On-Demand Distance Vector", *Proceedings 2nd IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, LA, U.S.A, Feb. 1999, pp. 90-100.
8. J. Hoebeke, I. Moerman, B. Dhoedt and P. Demeester, "Adaptive Multi-Mode Routing in Mobile Ad Hoc Networks", *Proceedings of the 9th International Conference on Personal Wireless Communications 2004 (PWC 2004)*, Delft, The Netherlands, September 2004, pp. 107-117.