A Simulation Study of Integrated Service Discovery

Gertjan P. Halkes, Aline Baggio, and Koen G. Langendoen

Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology, The Netherlands
{g.p.halkes, a.baggio, k.g.langendoen}@tudelft.nl

Abstract. The research in the field of service discovery in mobile ad-hoc networks is characterised by a lack of quantitative research. Many ideas have been put forward but few have been tested, either in simulation or real life. This paper fills part of that void, by comparing through simulation a simple broadcast-flood protocol, an integrated routing and service-discovery approach, and a global-knowledge based approach. The results show that using an integrated approach can achieve a similar level of performance as a global-knowledge based approach.

Keywords: Service discovery, Mobile Ad-Hoc Networks, Simulation.

1 Introduction

The field of service discovery is gaining more and more attention in the mobile ad-hoc network (MANET) research community [1,2,3,4,5]. However, the field is characterised by a lack of quantitative research. Many ideas have been put forward but few have been tested, either in simulation or real life. This paper fills part of that void, by comparing three different schemes through simulation: a simple broadcast-flood protocol, an integrated routing and service-discovery approach, and a global-knowledge based approach.

The simple flood protocol is an unoptimised service-discovery protocol. It is important as it is a natural extension of the Service Location Protocol (SLP) [6], which was developed for fixed-infrastructure networks, into the MANET domain.

Integration of routing and service-discovery is an idea that has been put forward by Koodli et al. [5]. By performing service discovery in the same way as route discovery, nodes can accumulate routing information while performing service discovery. If a service provider wishes to reply to a received service request it does not have to perform a route discovery for the originator of the request, because it already has the required routing information. This is a big advantage over a purely application-layer based approach, like the simple flood protocol, where all service providers that wish to reply to a service request have to do route discovery.

The global-knowledge approach uses an oracle to determine which service providers are available in the network and to locate the service provider that is most suitable to communicate with, that is, the closest one. The oracle serves

P. Havinga et al. (Eds.): EUROSSC 2006, LNCS 4272, pp. 39–53, 2006.

as an upper bound on the performance of any (integrated) service discovery protocol; similarly, the simple flood protocol provides a lower bound. In our experimental evaluation we have studied the effect of node mobility, service request rates, node density, and lifetime of cached service entries.

The contribution of this paper is twofold: firstly, it provides a comparison through simulation of integrated service-discovery with an unoptimised service-discovery protocol and a global-knowledge approach. Secondly, it provides a benchmark for further comparisons.

The rest of this paper is organised as follows: Section 2 gives a short overview of the research efforts in service discovery in MANETs. It is followed by background information on routing in Section 3. Section 4 details the service discovery protocols used in our simulation study. Section 5 presents the design of our experiments. Section 6 shows the results from our experiments, analyses the major trends, and discusses their significance. Finally, Section 7 lists our conclusions.

2 Related Work

Service discovery in fixed-infrastructure networks has received quite some attention. Standards are now being developed, the most important of which is the Service Location Protocol (SLP) [6]. The SLP protocol has two modes of operation: centralised and distributed. The centralised mode uses one or more service directories. Service providers register their services with the service directories. If a client wants to discover a server it contacts the service directories and requests a list of matching servers.

In distributed mode, service directories are not used. To find a service, a client simply broadcasts a service query on the network. If a service provider receives such a query, it sends a unicast message to the originator. Optionally, one can use multicast instead of broadcast, so as to limit the network traffic.

The centralised mode of SLP does not match with the ad-hoc nature of MANETs. Using the distributed mode, however, is feasible. Reusing an implementation of SLP meant for fixed-infrastructure networks is possible, by replacing the local broadcast with a broadcast flood. The resulting protocol is similar to the Nom [3] protocol. The main difference is that the Nom protocol also implements a cache of previously-seen service bindings. This service cache reduces the number of service requests sent to the network by allowing the reuse of previously gathered information.

Many papers have been written on the field of service discovery in MANETs. We now present the most important proposals.

The Intentional Naming System (INS) [1] is one of the first proposals for service discovery in multi-hop ad-hoc networks. INS integrates routing and service discovery, but does so using an overlay network. To create the overlay network, a central component is used. The overlay network is used both as a replicated distributed service directory, and as a network of forwarders. INS is a proactive protocol in the sense that services are advertised to and stored in a service directory, before the service information is requested. INS has a number of drawbacks.

First it has a single point of failure in the form of the central component used to build the overlay network. Second, keeping the service directory up to date, even when it is not used, can incur significant network traffic.

Another early proposal on how to implement service discovery in mobile ad-hoc networks has been put forward by Koodli et al. in a now expired draft RFC [5]. The authors propose to integrate service discovery and routing. By doing so, one leverages the existing experience with routing protocols to create an efficient service discovery protocol.

Several other protocols have been suggested that integrate a limited form of routing in the protocol itself, such as (GSD [2], CARD [4]. GSD reuses many of the ideas of the AODV protocol [7], while CARD implements large parts of the TRANSFER [8] routing protocol. However, these protocols are in essence duplicating some of the work of the routing layer which is inefficient. Moreover, the routing information gathered by the service discovery protocol cannot benefit other traffic in the network.

In a follow-up paper the authors of GSD [2] extend the integration with routing to also include subsequent communication with the service provider [9]. As an extra feature, the authors propose automatic re-routing to another available service provider if the route to the selected service provider breaks. Although this approach uses the routing information gathered during service discovery in subsequent communications with the service provider, other types of traffic still cannot benefit from this routing information. Moreover, the re-routing of traffic to another service provider only works when there are multiple providers delivering an indistinguishable service.

To demonstrate the advantages of integrated service discovery, a thorough study of the proposed protocols is needed, nevertheless performance has received only limited attention. Varshavsky et al. [10] have done a worst-case packet-count analysis and an experimental (in simulation) comparison with variants of the Service Location Protocol (SLP) [6]. In a later paper [11], Varshavsky et al. evaluate service-selection mechanisms, but only compare with centralised SLP variants. A paper by Garcia-Macias et al. [12] provides a very limited case study of integration with AODV vs. the Nom protocol. Both studies conclude that using integrated service-discovery can significantly reduce the number of messages needed for service discovery. This is also confirmed by the results reported in this paper, which includes a set of simulation experiments covering a wide range of parameters, for example, node density and speed.

3 Routing Background

As we will be presenting service-discovery protocols based on routing protocols, we now present some background information. Readers familiar with basic routing protocols like AODV [7] and DSR [13] may proceed with Section 4 immediately.

Routing protocols can be categorised into reactive and proactive routing protocols. Reactive routing protocols do not maintain routing information for the entire network, but only start communicating when a route is required.

Proactive routing protocols do maintain routing information for the entire network, and therefore don't have to communicate to find a route at the time the route is needed. On the other hand, maintaining this route information means that proactive routing protocols need to communicate constantly, even if no routes are needed.

The most well-developed reactive routing-protocol for MANETs at this time is the Ad-hoc On-demand Distance Vector (AODV) protocol. Nodes using AODV maintain a routing table. This routing table contains a next-hop address for all nodes to which a route is known. When a node using the AODV routing protocol needs to know a route to another node in the network, it first checks its routing table. If an entry is present, the message is forwarded to the node mentioned in the routing entry. Otherwise, a route discovery procedure is initiated.

A node performs a route discovery by broadcasting a Route Request (RREQ) message to its one-hop neighbours. Sending such a message is achieved by sending an RREQ message with a Time-To-Live (TTL) of one. If one of them has routing information for the requested address, it replies with a Route Reply (RREP) message. If, after a timeout, no neighbour has replied with an RREP the RREQ message is resent, this time with an increased TTL. On receiving a message with a TTL greater than one, a node rebroadcasts the message if it can not supply the originator with the desired routing information. A node records the sender of the message as the next hop for sending messages to the originator of the RREQ. This way a so-called reverse route is set up. This reverse route can then be used to send the RREP message. As long as the originator does not receive an RREP message within the timeout period for the TTL set in the RREQ message, it increases the TTL in the RREQ up to a certain maximum. This technique is called expanding-ring search.

Another prominent reactive routing-protocol is the Dynamic Source Routing (DSR) protocol. In contrast to the AODV protocol, the routing table of the DSR protocol contains route information for entire routes and not just the next-hop address for the different destinations. This can be implemented in several ways. Originally, the authors proposed to use a so-called route cache, whereby the routing table contained full routes to each known destination. This was later deemed impractical for larger networks, so the link cache was proposed. In this scheme, a node stores which links are available between nodes. When a route is needed, the link information is used to build a complete route to the destination.

The DSR protocol uses source routes in its messages. This means that each message contains the untraveled part of the route to the destination. Like AODV, the DSR protocol uses RREQ messages to gather route information. The DSR RREQ messages contain the route back to the originator so that the receiver of the message can also send a message back to the originator. By default the DSR protocol does not use the expanding-ring search. Instead a node starts with asking its one-hop neighbours first, and if they don't reply within a preset timeout the node sends an RREQ message with the TTL set to a predefined maximum. The DSR protocol uses overhearing (aka. promiscuous mode) to allow nodes that are near but not on a path to gather route information. Note that the use of source routes is beneficial in this situation.

4 Service-Discovery Protocols

For our performance study, we have chosen two reactive routing protocols, namely AODV and DSR. These represent the most well-developed protocols currently available. Their operation is sufficiently different to warrant separate treatment. We have implemented integrated service discovery for both protocols and use the prefix SD to distinguish the integrated service discovery protocol from the original routing protocol.

As for comparison, we have chosen two extremes. A simple broadcast-flood protocol, which is the natural extension of distributed SLP into the MANET domain and also resembles the Nom protocol. This represents completely unoptimised service discovery. The other extreme is represented by a global-knowledge approach where each node knows all available services and the physical distances to each of these services. Communication is kept as local as possible, and service discovery is essentially a non-operation.

Our simulations do not include any proactive routing protocols as the effect of integration can be easily estimated analytically. Service information is not as volatile as routing information, therefore service information updates can be sent much less frequently than routing information updates. This means that the impact of disseminating service information in a manner similar to the dissemination of routing information gives a small and constant overhead.

In the following sections we will give more detail on each of the service-discovery protocols included in our simulations.

4.1 SD-AODV and SD-DSR

The SD-AODV and SD-DSR protocols have been implemented in the spirit of the routing protocols they extend. Two extra message types have been introduced, i.e. Service Request (SREQ) and Service Reply (SREP). The difference between an SREQ and an RREQ is that the target specified in the message is not an address, but a service description. An SREP differs from an RREP in that it also includes a service description of the offered service.

The forwarding and handling of SREQ and SREP messages is implemented like the forwarding and handling of RREQ and RREP messages in the original protocols. For example, in SD-AODV, SREQ messages use the same expanding-ring search technique used for RREQ messages. As the AODV protocol does not use overhearing, neither does SD-AODV. Conversely, DSR does use overhearing, therefore so does SD-DSR. Note that the SREQ and SREP messages also create entries in the routing tables in the same way as RREQ and RREP messages do.

The main difference between the handling of service related messages vs. route related messages is in the dissemination of SREP messages by so-called intermediate nodes. An intermediate node is a node that receives an SREQ message, but is neither the source nor the target of the SREQ message. SD-AODV and SD-DSR impose an additional constraint on the dissemination of SREP messages by an intermediate node: both a valid service description must be cached and a valid route to the target must be available. In SD-DSR only a one-hop neighbour of the source of the SREQ message may issue an intermediate-node reply. Experiments showed that allowing all intermediate nodes to issue replies increases the total number of messages sent, instead of decreasing it.

In both protocols, the handling of the service information differs slightly from the handling of the routing information. Each node has a service cache for storing service bindings. If an SREP message is received, the service description from the SREP message is used to create or update a service cache entry. The lifetime of the service-cache entry is determined from the received message. If the service cache entry already exists, the maximum of the lifetime of the existing entry and the lifetime in the message is taken as the new lifetime. Otherwise, the lifetime is copied form the received message.

The service cache is used to check for known service bindings, before initiating an SREQ message. If the service cache contains a valid and matching service description, no SREQ message is sent and the cached binding is returned to the application. However, should the application find that none of the bindings retrieved from the service cache could be used to contact a server after repeated attempts, the node requests a true service discovery to be initiated by the service-discovery protocol.

4.2 Flooding

The flooding protocol is the simplest of the service-discovery protocols. It uses the same service-caching regime as the SD-AODV and SD-DSR protocols. When a service query needs to be injected into the network, it simply initiates a network-wide broadcast flood. Intermediate nodes only pass on the request, even if they do have a valid service binding in their cache. When a request reaches a server that offers a matching service, this server sends a unicast message back to the source of the request.

As an optimisation, intermediate nodes that forward a service reply can inspect messages and cache the service binding. This ensures that a future request generated at the intermediate node can be satisfied by inspecting the cache, thus preventing a broadcast flood.

The flooding protocol is used in combination with both the AODV and the DSR routing protocol. When using the flooding protocol in combination with the DSR routing protocol, overhearing becomes a realistic option. The DSR protocol already uses overhearing itself, therefore it would not cost extra energy to also allow other protocols access to the gathered information. We have implemented a variant of flooding that incorporates overhearing.

4.3 Global Knowledge

When using the global knowledge approach, all nodes know which services are available on which server. The minimum hop-count to all the servers is available to all nodes at all times as well. All this information is provided by an oracle. By using this information, a node can select the closest server that provides a desired service without performing any communication. However, the oracle does not provide routing information. The global-knowledge approach is therefore used in combination with both the AODV and the DSR routing protocol. By not providing routing information, we obtain a means to measure the service discovery overhead of the other protocols.

5 Simulation Design

Our experiments were conducted using the QualNet wireless network simulator [14]. Each simulation has been run 10 times with different random seeds, and simulates 1000 seconds. We used the 802.11 MAC and physical layer, with a radio range of 140 meters. The QualNet simulator includes models for the AODV and DSR protocols. Unfortunately the default AODV model contained a bug, whereby an expired route entry would be revived on reception of a packet from the node named in the routing entry. Our experiments use a fixed version of the AODV model. The default DSR model uses a path cache. We have replaced this with a link cache because of its smaller memory footprint.

For the basic stationary network experiments, we used an area of 1000×1000 meters. For the experiments with mobile networks, we increased the area size to 1200×1200 to counter the centring effect of the Random Waypoint mobility model. All our experiments involve 100 nodes, 50 of which are clients. The number of servers is three, except for the first experiment, where there is only one server.

When mobility is used, we use the Random Waypoint model [13]. We set minimum speed V_{min} and maximum speed V_{max} to 1 and 5 m/s respectively, and set the maximum pause time to 30 seconds. V_{min} is not set to 0 to avoid the nodes in the network slowing down to the point where it becomes almost a stationary network [15].

Client nodes repeatedly request and use a service provided in the network; a client is modelled as a parametrised Poisson process specifying the service request rate. After a service is discovered by a client, it chooses one of the servers it has heard of and sends it a unicast message. To select the nearest server, the client sorts the servers by hop count using information from the routing layer, or in the case of the global-knowledge approach by distance. The server responds, also with a unicast message. This step models the communication between a client and a server for which the service discovery was initiated. If this communication fails even after two retries at the application level, the client selects another server from its list. If there are no more entries in the server list and the client has so far used only cached results, it asks the service discovery protocol to update its cache by issuing a new service request. If, after trying the servers this last step yields, communication with a server has still not succeeded, the client gives up.

As described in Section 4 all the service discovery protocols simulated use a service cache. The lifetime of entries in the service cache is set to 120 seconds.

This setting strikes a balance between saving gathered information on the one hand, and the volatility of the network on the other hand. In one of our experiments, we investigate the effect of different values of the service cache lifetime (see Section 6.4).

6 Results

This Section presents the results of our experiments. We start with a simple stationary network with a single server, and gradually explore more complex situations. First, we increase the number of servers. Then, we add mobility (Section 6.2) and explore the effect of low node density (Section 6.3). Finally, in Section 6.4, we show the effects of varying the lifetime of entries in the service cache.

6.1 Stationary Network

For our first experiment, we start with a stationary network, with a single randomly located server. Figure 1 shows the absolute number of packets sent in the network. The protocols using AODV clearly require more messages to find service providers and communicate with them than the protocols running on DSR. This is a characteristic of the routing protocols themselves and is not specific to service discovery [16].

Figure 2 shows the number of packets sent for the protocols using AODV, normalised by the global-knowledge approach. It is clear that the integrated SD-AODV protocol has similar performance as the global-knowledge approach. In fact, it sometimes performs slightly better. This is due to minor implementation differences in the sending of SREQ and SREP messages with respect to the RREQ and RREP messages.

The flood protocol performs worse than either the global-knowledge approach or the integrated SD-AODV protocol. There are two reasons for this. Consider a node A that is close to a node B. When node A has performed a successful service discovery, it has service and routing information for reaching the server. If node B is not on the route from A to the server, it will not have any information for reaching the server. In the case of SD-AODV or the global-knowledge approach, node B is able to gather the required information using the expanding-ring search technique. As node B is close to node A, this requires only a few packets. However, in the case of the flood protocol node B would initiate a full network flood, which needs many packets.

The second reason is that once the flood initiated by node B reaches the server, and the server wishes to send a response, it does not have any routing information for node B. This means the server needs to initiate an RREQ, which again uses many packets. Avoiding these extra RREQs is the most important reason for implementing integration with the routing protocol.

For higher request rates, the number of SREQs initiated by the flood protocol on AODV does not increase because of the use of caching. The total number of messages used for communication does increase as the clients communicate with

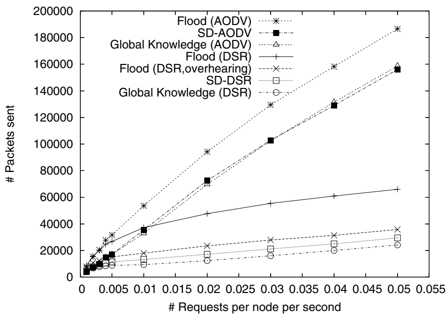

Fig. 1. Number of packets sent in a static network with one server

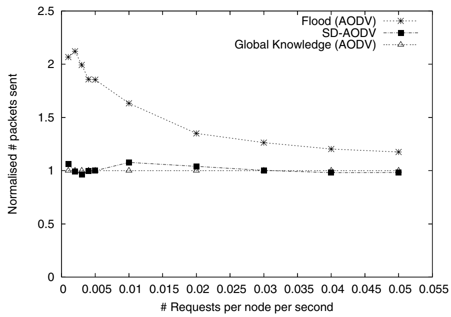

Fig. 2. Number of packets sent in a static network with one server for all protocols using AODV for routing, normalised by the global knowledge approach

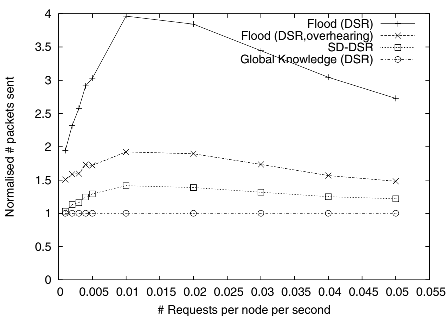

Fig. 3. Number of packets sent in a static network with one server for all protocols using DSR for routing, normalised by the global knowledge approach

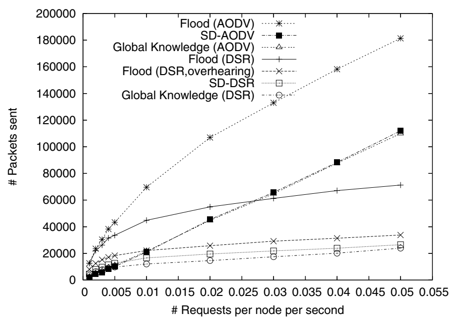

Fig. 4. Number of packets sent in a static network with three servers

the server more often. The net effect is that the flood protocols performs better with respect to the global-knowledge approach for higher request rates.

Figure 3 shows the number of packets sent for the protocols using DSR, again normalised by the global-knowledge approach. Note however, that this is now the global-knowledge approach on top of the DSR protocol and not on top of the AODV protocol as in Figure 2. In this case we see that the integrated SD-DSR protocol cannot reach the same level of performance as the global-knowledge approach. This is explained by the difference in handling and using routing and service information, in combination with the overhearing used in (SD-)DSR. As far as RREQs and SREQs are concerned, the handling is mostly similar. However, when a node that has overheard routing information subsequently uses that information to send a packet, it thereby disseminates routing information to its neighbours. However, when using service information to communicate with a service provider the node does not disseminate this service information. This

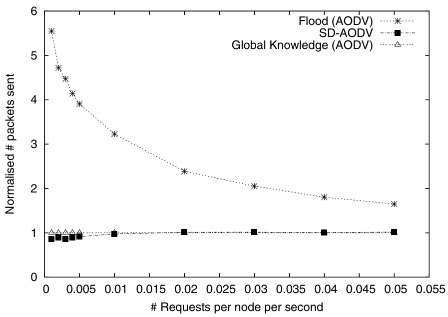

Fig. 5. Number of packets sent in a static network with three servers for all protocols using AODV for routing, normalised by the global knowledge approach

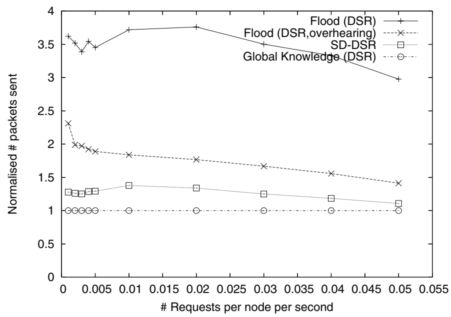

Fig. 6. Number of packets sent in a static network with three servers for all protocols using DSR for routing, normalised by the global knowledge approach

means that in some cases the global-knowledge approach has all the information it needs (i.e., the routing information) while SD-DSR still requires an SREQ to discover services.

At the lowest simulated request rate, the SD-DSR protocol achieves the same performance as the global-knowledge approach. This is because the request rate is so low that any route information spread by a communication between a client and a server times out before the next request is issued. Therefore, the clients will always have to initiate an RREQ or SREQ (SD-DSR) which uses the same number of packets.

Clearly the flood protocol is at a disadvantage when it does not use overhearing. However, even when using overhearing, it still suffers from the same problems as described for the combination of the flood protocol with the AODV routing protocol. Of course the exact effect is different as DSR does not use expanding-ring search, but a simple two-stage search. The better relative performance at very low request rates is again due to the route information timing out before a new request arrives.

Figures 4 through 6 show the results of simulations in the same static network, but now with three randomly located servers all providing the same service. In Figure 5, we can see that for the case with few service requests SD-AODV performs better than the global-knowledge approach. The fact that the global-knowledge approach chooses the closest server to communicate with is actually a slight disadvantage here. Where SD-AODV finds service and routing information that was spread by a previous service request of a nearby node, the global-knowledge approach is forced to find route information to the closest server. This routing information may not be found as nearby as the information that SD-AODV uses. Although SD-AODV communicates with a server further away, it uses fewer packets to find it. If communication were to continue between client and server, the shorter route that the global-knowledge approach sets up would ensure that the balance tips in its favour.

The DSR-based protocols behave mostly the same as for the single server case. The only difference is that for very low request rates the SD-DSR protocol and the flood protocol now perform worse, with respect to the global-knowledge approach. This difference is caused by the fact that the global-knowledge approach requests a route to a single server, while the service request from both the flood protocol and SD-DSR targets all servers in the network. As both SD-DSR and the flood protocol use broadcast floods, they will reach all the servers in the network. All these servers then have to send a reply, which is the cause of the extra messages.

We have also conducted experiment with three servers providing three different services, and with three servers each providing the same set of three servers, both with and without mobility. The results are so similar to the results already provided we do not show them here.

6.2 Mobility

As we are considering *mobile* ad-hoc networks, we introduce mobility into our experiments. Figure 7 shows the results of simulations with all nodes moving according to the Random Waypoint model. The most striking feature of this graph is its similarity to Figures 1 and 4.

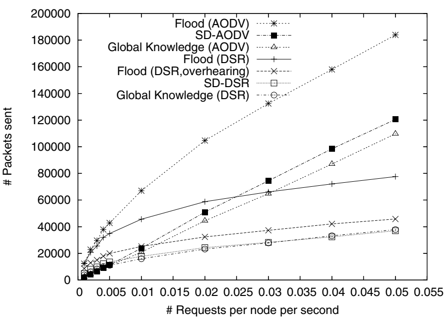

Fig. 7. Number of packets sent in a mobile network with three servers

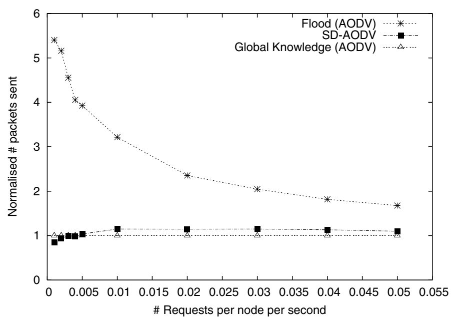

Fig. 8. Number of packets sent in a mobile network with three servers for all protocols using AODV for routing, normalised by the global knowledge approach

If we further compare Figures 5 and 8 we can see that for protocols using AODV, mobility changes only one thing about the relative performance of the service discovery protocols: the SD-AODV protocol starts to perform worse. The relative performance decrease is indirectly caused by the expanding-ring search technique. The expanding-ring search technique has the characteristic that it stops searching once a single answer has been found. For routing this obviously is not a problem, as there should be only one node with a given address. However, in the case of service discovery, knowing about more than one server is beneficial.

When a node moves around, a different server from the one previously discovered may come (much) closer, and thereby becomes a much more attractive partner for communication. SD-AODV may not know about the closer server, and will try to keep communicating with the server it does know about. However, longer routes are also more prone to breaking and maintaining these longer routes therefore increases the number of packets sent. The longer routes themselves add to the number of packets sent, simply by requiring more packets to be sent to let one message travel between client and server.

In Figure 9, we see that all the service-discovery protocols on top of DSR perform better with respect to the global-knowledge approach than for the static network with three servers in Figure 6. In fact, SD-DSR performs just as well for higher request rates. The global-knowledge approach always chooses the nearest server to communicate with. However, the communication with the nearest server may be very unreliable for an amount of time because of an unreliable link on the route. Then, each time a node tries to communicate with the nearest server it has a high chance of failure. If after several retries it determines that the chosen server is unreachable, it tries to communicate with the second nearest server. All of this also holds for the SD-DSR protocol, but where the global-knowledge approach will always try to communicate with the nearest server, the SD-DSR protocol will remove an unreachable server from its cache and will not find the same problem over and over again. For low request rates this problem does not exist, because the unreliable links disappear before a new request is issued.

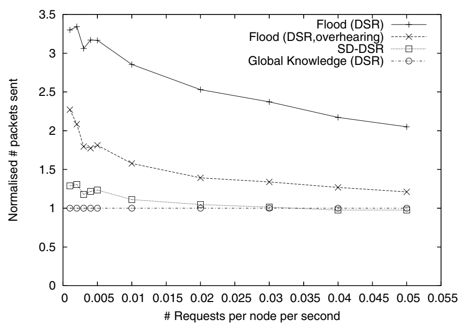

Fig. 9. Number of packets sent in a mobile network with three servers for all protocols using DSR for routing, normalised by the global knowledge approach

We have also conducted experiments with different settings of the pause time and maximum speed, and also with a different mobility model. The results showed the same effects as described above, so we omitted them for brevity.

6.3 Node Density

Our next experiment shows the effect of reducing the node density in the network. Figure 10 shows the result when the area size is increased from 1200×1200 to 1600×1600, thereby increasing the total area by a factor 1.8. The mobility induced problems are exacerbated by the decreased connectivity. In particular, the persistence to use the closest server for each new request by the global-knowledge protocol is causing a significant decrease in performance. Furthermore, the redundancy in routing is decreased, thereby increasing the number of route errors and packets sent. The best example of the increased problems is the DSR-based global-knowledge approach. It now performs worse than SD-DSR *and* the DSR-based flood protocol with overhearing for high request rates.

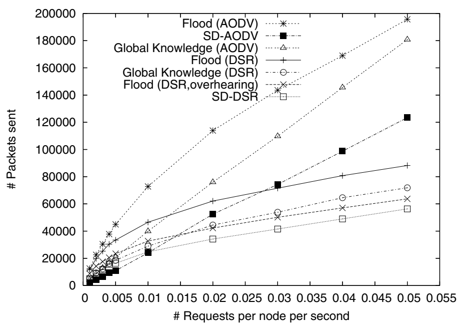

Fig. 10. Number of packets sent in a mobile network on a large area

6.4 Cache Lifetime

The final parameter we varied was the cache lifetime of the entries in the service cache. We performed this experiment in the 1200×1200 network with mobility. Figure 11 shows the results for the request rate of 0.01 requests per node per second. The graphs for the other request rates show the same trends.

For all but the SD-AODV protocol, increasing the cache lifetime decreases the number of packets sent. The flood and SD-DSR protocols know all the servers in the network, after performing one service-discovery phase. Therefore, it is not

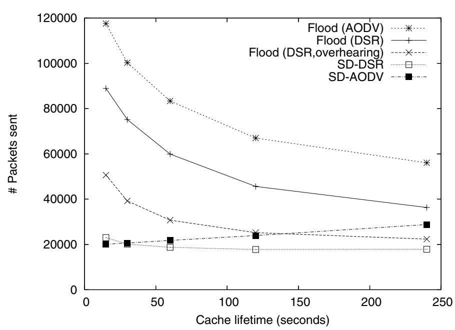

Fig. 11. Number of packets sent in a mobile network for different service-cache lifetimes at request rate 0.01 requests per node per second

beneficial for these protocols to discard the service information. The more often these protocols discard the service information, the more often they have to do service discovery, at the expense of more packets.

The SD-AODV protocol has one disadvantage. As described in Section 6.2, the SD-AODV protocol mostly finds only a single server. In combination with mobility, this can cause a node to persist using a previously found server while another server is now nearer by. Doing service discovery more often therefore ensures that SD-AODV will regularly rebind to the closest server. Keeping communication local means that fewer packets need to be sent. The extra overhead of the service discovery is offset by the reduction of route breaks and the shorter routes. Note that the service discovery overhead is also limited, as SD-AODV uses the expanding-ring search technique.

7 Conclusions

In this paper, we presented extensive simulations comparing a simple flooding-based protocol, integrated service-discovery protocols and a global-knowledge approach for service discovery. We tested each of these protocols in combination with both the AODV and DSR routing protocols. The results show that the integration of service discovery with the routing protocol out-performs the simple flooding protocol for a large set of parameters.

The integration of service discovery with the AODV protocol can yield results on par with the global-knowledge approach. However, we also showed that to reach this performance with SD-AODV in mobility scenarios we needed to decrease the service-cache-entry lifetime, instead of increasing it. The cause of this counter-intuitive setting lies in SD-AODV's expanding-ring search. The expanding-ring search causes service requests to find only a limited number of services. In combination with mobility this causes nodes to make a suboptimal choice on which server to use once a node has moved. Frequent rediscovery mitigates this effect.

DSR uses source routing and overhearing. Because of this, sending a message to a server spreads route information to all nodes within communications range. Nevertheless, that message does not spread service information. This means SD-DSR is at a disadvantage compared with the global-knowledge approach. However, the persistence to use the nearest server can prove a problem for the global-knowledge approach when routing to that server is problematic. The result is that SD-DSR can perform both better and worse than the global-knowledge approach depending on the exact network conditions, but the performance never diverges greatly.

More generally, we conclude that the integrated service-discovery protocols are efficient and are therefore a much more suitable standard for comparing the efficiency of service-discovery protocols than the flooding protocol. The integrated protocols achieves a level of performance close to that of the global-knowledge approach. The only drawback is that for the integrated service-discovery protocols an implementation is required per routing protocol. Yet, the extensions to the routing protocol are sufficiently straightforward to allow rapid implementation.

References

1. Adjie-Winoto, W., Schwartz, E., Balakrishnan, H., Lilley, J.: The design and implementation of an intentional naming system. In: Proc. of the 17th ACM Symposium on Operating Systems Principles (SOSP'99). (1999) 186–201
2. Chakraborty, D., Joshi, A., Yesha, Y., Finin, T.: GSD: a novel group-based service discovery protocol for MANETS. In: Proc. of the 4th IEEE Conf. on Mobile and Wireless Communications Networks (MWCN 2002). (2002) 140–144
3. Doval, D., O'Mahony, D.: Nom: Resource location and discovery for ad hoc mobile networks. In: Proc. of the First Ann. Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net 2002). (2002)
4. Helmy, A., Garg, S., Pamu, P., Nahata, N.: Contact-based architecture for resource discovery (CARD) in large scale MANets. In: Proc. of the 17th Int. Parallel and Distributed Processing Symposium (IPDPS 2003). (2003)
5. Koodli, R., Perkins, C.E.: Service discovery in on-demand ad hoc networks. Internet draft, IETF Manet Working Group (2002) draft-koodli-manet-servicediscovery-00.txt (expired).
6. Guttman, E., Perkins, C.E., Veizades, J., Day, M.: Service location protocol, version 2. RFC 2608, IETF Network Working Group (1999)
7. Perkins, C.E., Belding-Royer, E.M., Das, S.: Ad hoc on-demand distance vector (AODV) routing. RFC 3561, IETF Network Working Group (2003)
8. Helmy, A.: TRANSFER: Transactions routing for ad-hoc networks with efficient energy. In: Proc. of the IEEE 2003 Global Communications Conference (GLOBECOM 2003). (2003) 398–404
9. Chakraborty, D., Joshi, A., Yesha, Y.: Integrating service discovery with routing and session management for ad hoc networks. Ad Hoc Networks, Elsevier Science (2004)
10. Varshavsky, A., Reid, B., de Lara, E.: The need for cross-layer service discovery in manets. Technical Report CSRG-492, Department of Computer Science, University of Toronto (2004)
11. Varshavsky, A., Reid, B., de Lara, E.: A cross-layer approach to service discovery and selection in MANETs. In: Proc. of the Second Int. Conf. on Mobile Ad-Hoc and Sensor Systems (MASS 2005). (2005)
12. Garcia-Macias, J.A., Torres, D.A.: Service discovery in mobile ad-hoc networks: better at the network layer? In: Proc. of the Int. Conf. Workshops on Parallel Processing (ICPP 2005). (2005) 452–457
13. Johnson, D.B., Maltz, D.A.: Dynamic source routing in ad hoc wireless networks. In Imielinski, Korth, eds.: Mobile Computing. Volume 353. Kluwer Academic Publishers (1996)
14. Scalable Network Technologies, Inc.: QualNet 3.8 user's guide (2004) `http://www.scalable-networks.com/`.
15. Yoon, J., Liu, M., Noble, B.: Random waypoint considered harmful. In: Proc. of the Twenty-Second Ann. Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM 2003). Volume 2. (2003) 1312–1321
16. Broch, J., Maltz, D.A., Johnson, D.B., Hu, Y.C., Jetcheva, J.: A performance comparison of multi-hop wireless ad hoc network routing protocols. In: Proc. of the 4th Ann. ACM/IEEE Int. Conf. on Mobile Computing and Networking (MobiCom'98). (1998) 85–97