A Routing Layer Based Approach for Energy Efficient Service Discovery in Mobile Ad Hoc Networks

Christopher N. Ververidis and George C. Polyzos

Mobile Multimedia Laboratory
Department of Computer Science
Athens University of Economics and Business
47A Evelpidon & Lefkados str., Athens 11362, Greece

Tel: +30-2108203693, Fax: +30-2108203686, email: chris@aueb.gr

Abstract

Service discovery can be greatly enhanced in terms of efficiency, both regarding service discoverability and energy consumption, by piggybacking service information into routing messages. Thus, service discovery does not generate additional messages and a node requesting a service, in addition to discovering that service, it is simultaneously informed of the route to the service provider. We extended the Zone Routing Protocol in order to encapsulate service information in its routing messages. Our extended protocol, E-ZRP, may be seen as a representative of routing layer protocols providing service discovery functionality. Simulations demonstrate the superiority of this routing layer-based service discovery scheme over that of a similar, but application layer based service discovery scheme. In order to have a thorough evaluation of our approach we introduced a new metric, called Service Availability Duration (SAD), which characterizes the "quality" of discovered services and experimentally examined the implications of network density and node mobility on the availability of services discovered with E-ZRP, as a typical representative of routing layer based service discovery protocols.

I. Introduction

Much research has been devoted to Service Discovery in static networks, applied mostly to the (fixed) Internet. The emergence of wireless communications and mobile computing devices has created the need for developing service discovery protocols and architectures targeted to mobile environments. Especially, the proliferation of Mobile

Ad-Hoc Networks (MANETs) has introduced new requirements to service discovery due to the nature and inherent characteristics of these networks.

MANETs are extremely dynamic due to the mobility of their nodes, the wireless channel's adverse conditions and the energy limitations of small, mobile devices. The great majority of service discovery protocols developed for MANETs deal with the above issues at the application layer. In this paper we argue that by implementing service discovery at the routing layer, instead of the application layer, the resulting communication and energy consumption overheads are significantly reduced. Our approach is to implement service discovery in the routing layer by piggybacking the service information into the routing protocol control messages, thus enabling the devices to acquire both service and routing information simultaneously. This way a node requesting a service (henceforth called service requestor) in addition to discovering the service, it is also informed of the route to the service provider at the same time.

In our previous work [1], we proposed the piggybacking of service information in routing messages, in order to decrease communication overhead, save battery power and minimize discovery delays. This way, besides these savings, we can also achieve smooth service discovery adaptation to severe network conditions (e.g. network partitions). Smooth adaptation occurs because service availability is tightly coupled with route availability to serving nodes. Hence when all routes towards a node fail, this is immediately translated to a loss of service availability for the services that this node provides. We demonstrated the benefits of our approach (i.e. routing layer based service discovery) versus traditional application based service discovery, by extending the proactive part of the Zone Routing Protocol (ZRP) so that it is capable of encapsulating service information in its messages. ZRP is a hybrid routing protocol, i.e. proactive for a number of hops around a node called the node's zone and reactive for requests outside this zone. In this paper we perform additional simulations for the reactive part of ZRP as well. We also extend our work by evaluating the quality of service of the services discovered, so that a richer performance evaluation of our approach can be provided. With the term quality of service we refer here to the usability characteristics of a service and not its inherent characteristics (e.g. precision of the provided information). The study of the inherent characteristics of discovered services is beyond the scope of this paper. So, in order to measure the quality of discovered services we define a new metric called SAD (Service Availability Duration), which measures the availability of a discovered service. SAD is defined as the length of time that elapses from the moment the service is discovered until that time when the service is lost as a result of mobility or interference. It should be noted that if the path to the original service provider is lost, but there exists another provider for the same service-type in the node’s routing table, then the service is still considered ‘alive’. Only when all the routes from a node to all the available providers of the service are lost, this particular service is considered not to be available any more to that node. In the literature [17][18], a similar metric, called Path Duration has been widely used to measure the impact of mobility on routing protocols for MANETs. However these studies mainly focus on reactive routing protocols and do not consider service discovery. Moreover, they focus on node availability and not service availability, which is a different concept. In general a good discovery protocol should be able to adapt to different network conditions in order to effectively discover as many long-lived services as possible.

The remainder of this paper is organized as follows. Section II provides the essential background on service discovery by presenting the most significant research results. Section III presents the proposed approach of routing layer based service discovery, and section IV provides its evaluation showing simulation results along with their analysis. Finally section V summarizes our conclusions.

II. Related Work

Significant academic and industrial research has led to the development of a variety of protocols, platforms and architectures for service discovery such as JINI [2], Salutation [3], UPnP [4], UDDI [5], Bluetooths' SDP [6] and SLP [7]. All these approaches, except SDP, are mainly targeted towards the discovery of services in fixed infrastructure networks. They are mostly centralized approaches that assume that reliable communication can be provided by the underlying network. Most of these approaches utilize nodes acting as (central) service directories-repositories, where service providers register the services they offer. Service requestors submit their queries to these 'special nodes' in order to discover services and information about the nodes that actually host these services. It is clear that such assumptions are not consistent with MANETs' inherent features due to their volatile nature.

This has motivated some recent approaches in the field, namely Allia [8], GSD [9], DEAPspace [10], Konark [11] and SANDMAN [12]. These approaches were developed with pervasive computing environments in mind, and are briefly presented in the next paragraphs. One aspect of the discovery approach which we consider significant and we pay particular attention to is energy consumption.

Allia is an agent based service discovery protocol, centered on peer-to-peer caching of service information. Every node in the network periodically broadcasts service advertisements. Nodes with similar types of services form alliances by caching each other's services. So, when a node receives a service request, which it cannot fulfill (doesn't have an appropriate service), it checks whether it has cached information about other nodes (allies) that offer similar services. In case such information is indeed cached, this node sends back the appropriate reply. If there is no cached information, then, depending on its policy, the node either broadcasts this request to the other nodes in its vicinity or forwards it to the members of its alliance. When a node caches service information sent by another node, then this node automatically becomes a member of the caching node's alliance. Allia uses Unique Universal Identifiers (UUIDs) for services, which should be a-priori known to all nodes. However, Allia is entirely agent based and hence it is too demanding in terms of computational power and resources in general. It also does not address energy consumption, and no related measurements or metrics are provided.

Another approach is the Group-based Service Discovery Protocol (GSD). GSD is also based in peer-to-peer caching of service advertisements and selective forwarding of service requests. GSD generates fewer messages compared to a simple broadcasting scheme, since service requests are not broadcast but instead forwarded only to those nodes that have already cached information about similar services. However, GSD uses DAML-based service descriptions in the advertisement messages (instead of simple UUIDs) and performs semantic matching, thus increasing energy consumption.

Similarly to GSD, Konark is a distributed service discovery protocol based on peer-to-peer caching of service information. In Konark, every node maintains a service registry, where it stores information about its own services and also about services that other nodes provide. This registry is actually a tree-structure with a number of levels that represent service classification. Upon receiving a service advertisement, a node updates its registry by classifying that service under the appropriate leaf of its tree. Service advertisements are in an XML-like language (similar to WSDL but smaller), hence allowing semantic matching, leading to increased energy consumption, but more precise resolutions. Konark uses multicasting for service requests and unicasting for service replies; hence it is more efficient than simple broadcasting schemes in terms of messaging overhead.

DEAPspace employs a periodic broadcast scheme for service advertisements. Each node sends the full list of services that it is aware of in its one-hop vicinity. Hence DEAPspace is targeted to smaller networks than Konark. In DEAPspace each node listens to its neighbors' broadcasts. In case the node doesn't find its own services in these messages, it schedules a broadcast sooner than usual, informing all the others about its presence and the services it can provide. In contrast to the aforementioned approaches, DEAPspace deals with the problem of energy consumption explicitly, by forcing weak nodes to go into idle mode during pauses between (the periodic) broadcasts.

SANDMAN, like DEAPSpace, is another service discovery protocol that implements power savings. This is done by grouping nodes with similar mobility patterns into clusters; in each cluster, one of the nodes (called clusterhead) stays awake permanently and answers discovery requests. The rest of the nodes periodically wake up to provide the actual services and also inform the clusterhead about their presence and services. The clusterheads are re-elected periodically to avoid draining a single node's battery. Simulation results show energy savings up to 40% for low numbers of service requests. Increasing the size of a cluster can attain even higher savings. However, this results in a dramatic increase of the average interaction latency due to the fact that a requesting node has to wait the sleeping node to wake up in order to interact with its services.

It is clear from the above discussion that only the latter two approaches take into account energy consumption and provide related metrics and comparisons. A key difference of our approach from those is that we do not expect or allow the nodes to go into sleep mode, since we target environments where continuous communication is necessary. The other aforementioned approaches do not provide specific results regarding energy consumption. Our approach explicitly deals with power savings resulting from a routing layer supported service discovery scheme by modeling, simulating and recording energy consumption. Finally, in [13] an architecture called CARD is developed, using a zone-based protocol for service discovery and energy measures are provided for its performance. The focus of CARD is more on out-of-zone resource discovery and is specific to short-term transactions only. Also the provided measurements do not take into account MAC layer issues, like collisions. In our case measurements are the result of simulating the whole protocol stack from the application layer down to the physical layer. What also differentiates our work is that we use ZRP both for service and route discovery for all kinds of transactions and especially focus on intra-zone transactions. In the next section we present our approach in detail and justify our design decisions.

III. ROUTING LAYER BASED SERVICE DISCOVERY

Our motivation for seeking a routing layer solution for service discovery stems from the fact that any service discovery protocol implemented above the routing layer will always require the existence of some kind of routing protocol for its own use. Hence, two message-producing processes must coexist: the first one communicates service information among service providers and service requestors; the second one communicates routing information among them. As a result, a node is forced to perform multiple times the battery-draining operation of receiving and transmitting (control) packets. Our approach exploits the capability of acquiring service information along with routing information (from the same message) by piggybacking service information onto routing messages. This way, redundant transmissions of service discovery packets at the application layer are avoided and energy is saved.

The idea of providing routing layer support for service discovery was first introduced by Koodli and Perkins in [14]. They argue that for proactively routed MANETs, a service reply extension added to topology updating routing messages is enough for providing both service discovery and route discovery concurrently. In reactively (or on-demand) routed MANETs, the service discovery process follows the traditional route discovery process by using its message formats for route requests (RREQ packets) and route replies (RREP packets) extended to carry also a service request or reply respectively. However, as far as we know, no experimental assessment of Koodli's and Perkins' proposal in terms of energy efficiency and quality of discovered services has been published until now.

In this paper we present experimental results using service discovery extensions both on the proactive part and reactive part of the Zone Routing Protocol (ZRP). Next, we describe the basic operation of ZRP and the extensions we have introduced in order to enhance it with service discovery capabilities.

ZRP

We proceed to describe the ZRP's structure and operation. ZRP actually consists of three sub-protocols, namely:

- The Neighbor Discovery Protocol (NDP), through which every node periodically broadcasts a "hello" message to denote its presence.

- The Intra Zone Routing Protocol (IARP), which is responsible for proactively maintaining route records for nodes located inside a node's routing zone (for example records for nodes located up to 2-hops away). This is depicted in Fig.1 where nodes B to H are inside the routing zone of node A; hence node A is proactively aware of all the routes to these nodes through IARP.

- The Inter Zone Routing Protocol (IERP), which is responsible for reactively creating route records for nodes located outside a node's routing zone (e.g. records for nodes located further than 2-hops away).

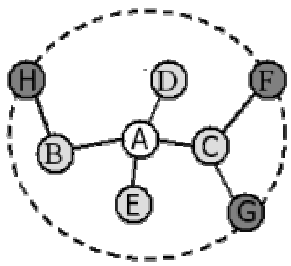


Fig.1: ZRP 2-hop zone

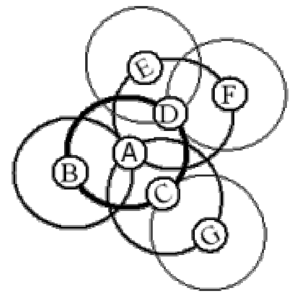


Fig.2: ZRP bordercasting

In ZRP, a node in search of a route towards a node outside its zone, unicasts the route request only to nodes located at the borders of its zone. This method is called bordercasting and is depicted in Fig.2. The border nodes check their IARP tables to find if the requested node is included in their respective routing zones; if not they also bordercast the request to their own border-nodes. When the requested node is found, a reply is unicasted back to the node that initiated the request. This way, global flooding is avoided and distant resources are discovered in an efficient and scalable manner.

As stated in the introduction we have extended the Zone Routing Protocol (ZRP) [15] so that it provides service discovery functionality. Services provided in a mobile ad hoc network will most probably have a local nature (especially when requiring physical interaction--imagine for example a user in need of a printing service). Furthermore services far away from the requestor are very likely to disappear frequently (causing severe service disruptions) due to the mobile wireless network's dynamics. Hence, on the one hand continuous monitoring and state maintenance of far away services will incur high cost and, on the other hand, interaction with such services is risky since it is highly likely that the service will disappear before the interaction has been completed.

Considering the above issues we have chosen the Zone Routing Protocol (ZRP) for adding service discovery functionality since:

a) ZRP proactively and continuously maintains (routing and with our extensions also service) information available in the vicinity of a node (through the notion of zones described further on) in a highly dynamic and energy efficient way, and

b) ZRP may reactively discover and collect information available at distant network areas through the use of intelligent forwarding instead of global flooding (explained later on).

Finally, ZRP was our selection for performing routing layer based service discovery also for another reason. In contrast to classic (monolithic) routing protocols for MANETs, ZRP can be also seen as a routing framework consisting of one reactive and one proactive part. Any existing purely reactive routing protocol (e.g. AODV or DSR) can be used as the IERP and any existing purely proactive protocol (e.g. DSDV) can be used as the IARP. Also, depending on ZRP's zone size, ZRP can be transformed to a purely reactive protocol (when zone radius equals 0) or to a purely proactive protocol (when zone radius is equal to the network diameter). Hence, ZRP may be considered as the best candidate for routing layer based service discovery (and in some sense a framework for a parameterizable class of protocols).

E-ZRP

In order to add service discovery capabilities to ZRP we embedded an extra field in NDP "hello" messages for storing service IDs. We used the concept of Unique Universal

Identifiers (UUIDs) instead of service descriptions, keeping packet lengths small for the routing messages and minimizing the effects on the network (the bigger the messages the larger the delays and the possibility of transmission errors). Such an approach implies that all nodes know a-priori the mappings between services offered in the MANET and UUIDs. This is a common assumption and is justified by the fact that most MANETs are deployed for certain purposes where there is lack of fixed communication infrastructure (e.g. a battlefield or a spot of physical disaster). In such environments, the roles of every participating node are concrete and can be easily classified in types of services. For example, in a battlefield one node may offer radar information to the rest, while another one may offer critical mission update information. In the case of a disaster such as an earthquake, an on-site relief team usually consists of members having different missions (e.g. one may be able to provide information about trapped people under ruins, another may provide information about terrain stability, and others may try to find and provide valuable structural information about the collapsed buildings etc.). In such environments the mapping of services to UUIDs is more than sufficient for service discovery. Semantic matching of rich service descriptions is of no particular use in these cases, not to mention that these techniques lead to increased energy consumption (a scarce and valuable resource in the above scenarios). Thus, by extending "hello" messages with service UUIDs, a node is able to denote both its presence and the services it provides.

ZRP was further extended in order to include service information in every routing entry of the IARP and IERP routing messages and tables. IARP listens to information gathered from NDP messages, updates its table and then periodically broadcasts its table to its neighbors. A node broadcasting this IARP update packets sets the TTL (Time To Live) field in these packets equal to its routing zone diameter, so that they will be dropped at border nodes. This way each node knows the routes to all the nodes in its zone and also the services that these nodes offer; thus adding the service discovery capability to the proactive part of ZRP. IERP is responsible for routing towards resources that are not available in a node's zone. When IARP fails to discover a service then an IERP message with a NULL destination address and a service field with the service requested is bordercasted. When a node receives such a message it first checks if it provides the requested service or if it is aware of another node that provides the service; and if it does, it generates an IERP reply message. Otherwise it re-bordercasts the message adding its own address to the previous hop list, so that a reverse route to the requestor can be established and used when the requested service is found.

The extended version of ZRP we implemented (henceforth called E-ZRP) is capable of providing routing layer support for proactive and reactive service discovery. In the following section we present our simulation results from applying E-ZRP in multiple scenarios.

IV. Performance Evaluation of E-ZRP

Our simulations were conducted using the Qualnet Simulator [16], which has a ZRP module. In the first 4 sets of experiments a basic assumption for evaluating the energy efficiency of E-ZRP is that each node hosts a unique service, which can be provided to other nodes. This was done for simplicity and in order to facilitate the analysis of the results. At the physical and data-link layer the IEEE 802.11b protocol was used.

As previously stated our goal was to compare E-ZRP with a traditional application layer based scheme for service discovery. Most such schemes utilize flooding for the propagation of messages. To be more specific the application layer protocol, which we use (henceforth called Flooding) does not involve global flooding but only range bounded flooding (using hop counters for its messages). To name but a few examples: flooding (and especially range bounded flooding, like the one used in our simulations) is used in [10], [11] and [8] for service advertisements, hence it is considered a well established and also representative mechanism for service discovery approaches at the application layer. We also note that the range (in hops) defining Flooding's bounded area is set to be equal to E-ZRP's zone radius for achieving a fair comparison of the two protocols. Measurements regarding out of zone service discovery using IERP and Flooding, show that both protocols expend almost the same amounts of energy, with Flooding (being more lightweight and stateless) giving energy savings of about 5%. However the delay imposed by Flooding in order for a node to discover out-of-zone services is an order of magnitude larger than the delay imposed by IERP. These findings are presented at the end of this section. In the following paragraphs we will focus on the experiments regarding IARP and Flooding comparisons for intra-zone service discovery, which are the more interesting ones, since IARP proved to outperform Flooding giving energy savings of 45% on average.

Initially, we conducted 4 sets of experiments, all of which deal with intra-zone service discovery using the service-extended IARP. In these first 2 sets the parameter settings for configuring both protocols were chosen to be identical, so that a fair comparison between the two schemes (i.e. application layer and routing layer based service discovery) is feasible. In the last 2 sets we modified these parameters so as to "favor" application layer based service discovery by employing larger update intervals compared to these used in the routing layer, hence minimizing the overhead as much as possible. Table I summarizes the settings for the first 2 sets of experiments.

TABLE I
PROTOCOL SETTINGS

Parameter	Value
IARP Zone Radius	*3 hops*
IARP broadcast interval	*10 seconds*
IARP deletion interval	*40 seconds*
Flooding Radius	*3 hops*
Flooding broadcast interval	*10 seconds*
Service deletion interval	*40 seconds*

The IARP Zone Radius is equal to the Flooding radius; this implies that range bounded flooding is performed, as opposed to global flooding. The broadcast interval is used by IARP in order for a node to send at regular time intervals all the information it has (zone routing information in the original ZRP, zone routing and service information in E-ZRP) to neighboring nodes. The same interval is used in Flooding as well, with the difference that Flooding messages are much shorter containing only a node's own service UUID and no routing information or other nodes' service UUIDs. The IARP deletion interval and the Service deletion interval, define the time after which a node erases records that haven't been updated. The size and contents for an IARP packet and a Flooding packet are presented in Appendix I.

In our first set of experiments, the two schemes are tested in a static context (i.e. nodes do not move). In the static context and in order to facilitate the analysis, we designed a “chain topology,” where nodes are placed in a row, each one of them having exactly one neighbor to the left and one to the right (except from the first and the last node of the chain). One could also consider other simple topologies. In fact we have obtained similar results for a cross-shaped topology, a snowflake-shaped topology and a star topology showing that the ratio of average energy consumption per node (also the average number of discovered services per node) when using E-ZRP to the average energy consumption (the average number of discovered services per node respectively) using Flooding remains the same with that obtained with the “chain topology” (see Appendix II). We decided to work with the “chain topology” because it is simple and allows us to easily come to conclusions regarding the performance of Flooding versus that of E-ZRP and estimate the theoretical maximum for the number of services that can be discovered over a given static network topology. Random topologies in a static context would not be appropriate for coming to such conclusions with a high degree of confidence. We conducted several experiments, altering each time the number of the participating nodes. Each experiment had duration of 1000 seconds (simulation time). The results of these experiments are presented in Fig. 3 and Fig. 4.

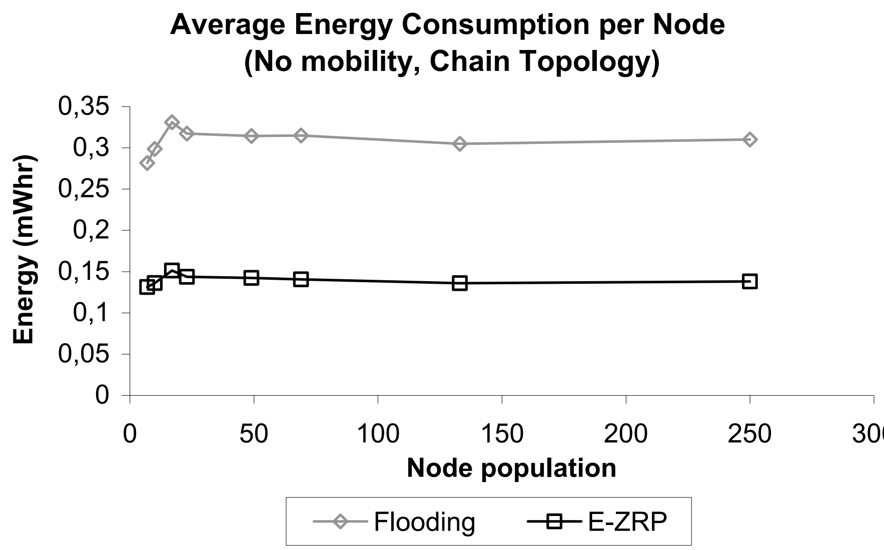


Fig. 3: Average Energy Consumption per node in a static context (E-ZRP versus Flooding).

Fig. 3 clearly shows that the energy consumption for E-ZRP is almost always 50% less than that for Flooding, irrespectively of the number of participating nodes. This happens because in the Flooding experiments, ZRP is also used at the routing layer to actually route packets. So, in the case of the Flooding scheme there are two processes creating messages: one at the application layer for service discovery and another one in the routing layer for route discovery. This application layer overhead in messages, leads to the observed dramatic difference of energy consumption between the two schemes. Also, it is evident for both schemes that energy consumption remains almost the same irrespectively of the node population. This is explained by the fact that the average number of every node's neighbors remains the same. In this static chain topology, every node exchanges information only with those nodes located inside its zone, and so energy consumption remains almost constant.

Fig. 4 depicts the average number of services discovered per node. What is worth noting is that a node using E-ZRP is able to discover on the average almost the same number of services, as compared to Flooding. The range bounded flooding scheme employed, performs slightly better than E-ZRP because Flooding packets (containing information about 1 service only) are shorter than IARP packets (containing information for all services provided in a node's zone) and hence are less susceptible to transmission errors. On the average (over all node populations) we get only 9,2% fewer services discovered when using E-ZRP, which is a small price to pay compared to the achieved energy savings of 47% on the average.

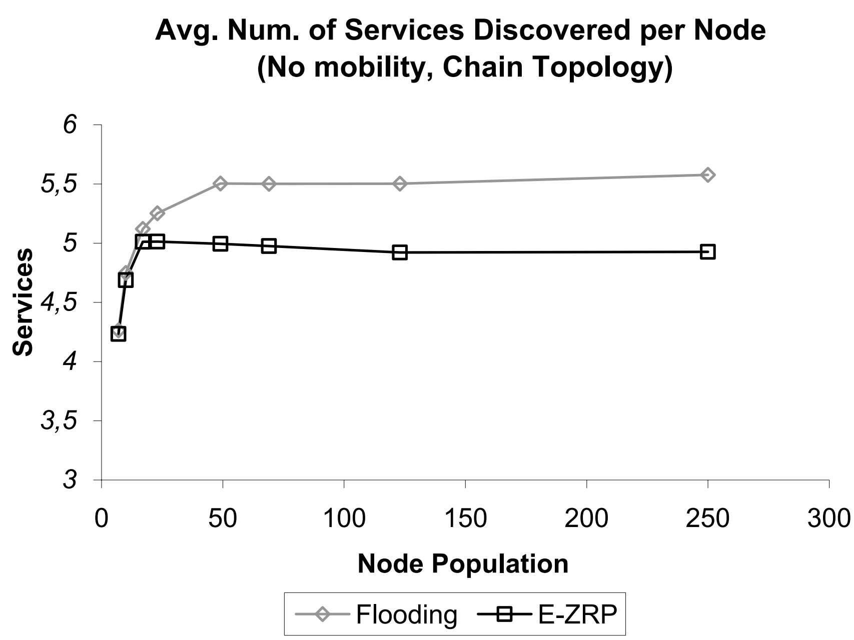


Fig. 4: Average Number of Services Discovered per node proactively in a static context (E-ZRP versus Flooding).

Considering the above results, it is clear that E-ZRP is more efficient than Flooding when there is no node mobility and both protocols have the same parameter settings (especially their update interval). In the following paragraphs we also test the two schemes under mobility conditions.

In the second set of experiments, the two service discovery schemes are tested in a mobile context (i.e. nodes do move). It is important to note that for stability reasons the density is kept fixed when varying the number of nodes (node population) by resizing the terrain in which they are allowed to move (however, later on we provide experimental results also regarding the effects of density in a mobile context). Every node in the simulated scenarios uses the random waypoint model with the following parameters:

- Minimum Speed = 0meters/second (m/s)
- Pause Time = 30 seconds.
- Maximum Speed takes the following values: 0.5m/s, 1m/s, 2m/s, 5m/s, 7.5m/s, 10m/s and 12.5m/s in order to test service discovery and energy consumption under different speeds.

Fig. 5 and Fig. 6 depict the results for service discovery and energy consumption respectively in this mobile context. Each spot in the diagrams represents an average value obtained by running the experiment over 8 different randomly chosen node populations (spanning from 10 nodes to 250 nodes).

Regarding service discoverability (Fig.5) the two protocols give almost identical results. We observe that both protocols perform better when speed increases (this means that each node will meet more nodes throughout its lifetime), with E-ZRP being better only when the maximum speed is set at 7.5m/s or more, hence giving 2% more services on average (across all speeds). The main reason is that in E-ZRP, IARP packets contain much more information about available services in a node's zone, compared to Flooding packets that only contain information about the service that their sender provides. Hence, when speed increases and successful packet transmissions are decreased (nodes remain much less time in each others transmission range), one IARP packet that successfully reaches a node is much more informative than several Flooding packets that may reach this node.

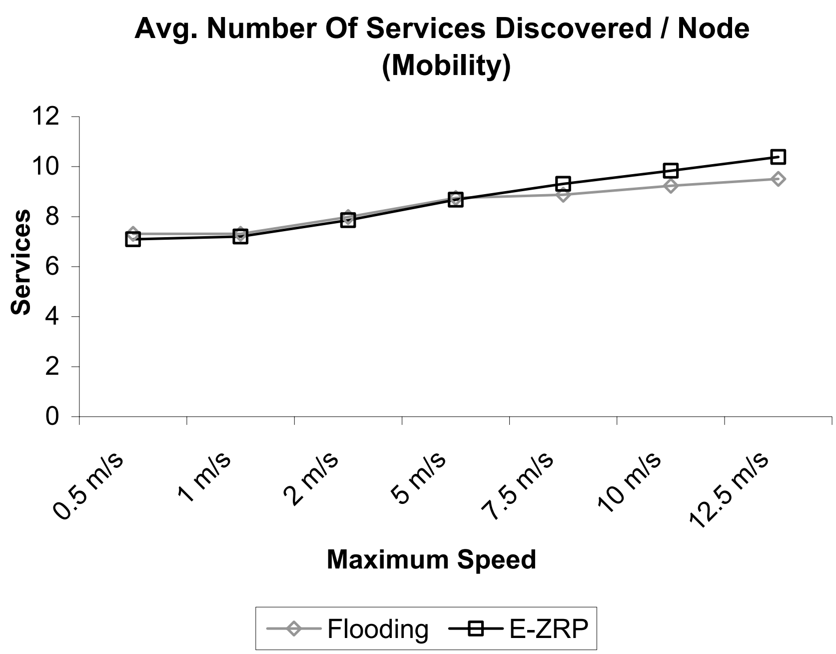


Fig. 5: Average Number of Services Discovered per node proactively in a mobile context (E-ZRP versus Flooding). The random way point mobility model is used with pause time 30 seconds and minimum speed 0m/s.

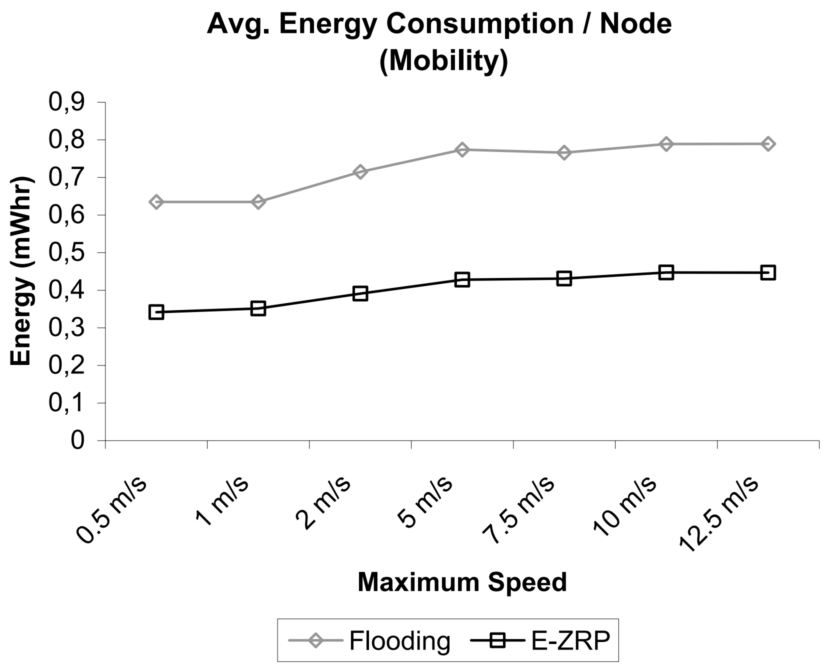


Fig. 6: Average Amount of Energy Consumed per node in a mobile context (E-ZRP versus Flooding). The random way point mobility model is used with pause time 30 seconds and minimum speed 0m/s.

As expected, energy consumption (Fig. 6) follows the same pattern (i.e. it increases when speed increases), which is explained by the fact that every node meets more nodes when moving at higher speeds; hence more bytes are received, leading to increased energy consumption. Energy consumption is on average 45% less for E-ZRP compared to Flooding (across all speeds).

The above simulation results prove the superiority of the routing layer based service discovery scheme compared to a traditional application layer based service discovery scheme when both layers work with identical parameter settings. This superiority is expressed in terms of significantly improved energy efficiency in both mobile and static environments with almost the same number of services discovered.

The last 2 sets of experiments were conducted in order to investigate the performance of the application layer based service discovery versus the routing layer based service discovery scheme, when the update intervals used at the application layer are larger. In these cases the application layer sends messages in larger time intervals and hence decreases the energy consumption. However this comes at the cost of decreased capability of discovering services. The purpose of these experiments was to show the optimal configuration of an application based service discovery scheme (based on updates in a bounded zone), so that service discoverability is equal or better to that achieved by a routing layer based approach. Table II summarizes these new settings. Note that service deletion interval will always be 4 times the broadcast interval for fairness reasons.

TABLE II
PROTOCOL SETTINGS

	Flooding broadcast interval	*Service deletion interval*
A	*200 seconds*	*800 seconds*
B	*160 seconds*	*640 seconds*
C	*80 seconds*	*320 seconds*
D	*40 seconds*	*160 seconds*
E	*20 seconds*	*80 seconds*
F	*15 seconds*	*60 seconds*
G	*10 seconds*	*40 seconds*

So, in our third set of experiments, the two schemes are again tested in a static context with a "chain node topology," Since in the previous similar experiments we showed that results do not vary much over different network sizes, we conducted experiments, over a network with 250 participating nodes. Each experiment had duration of 1000 seconds (simulation time). The results of these experiments are presented in Fig. 7. Each point on the curve corresponds to different parameter settings for the update and service deletion intervals (those presented in Table II) for the Flooding protocol. The vertical and horizontal dotted lines denote the energy consumption and the number of services discovered respectively, for E-ZRP with a broadcast interval of 10 seconds.

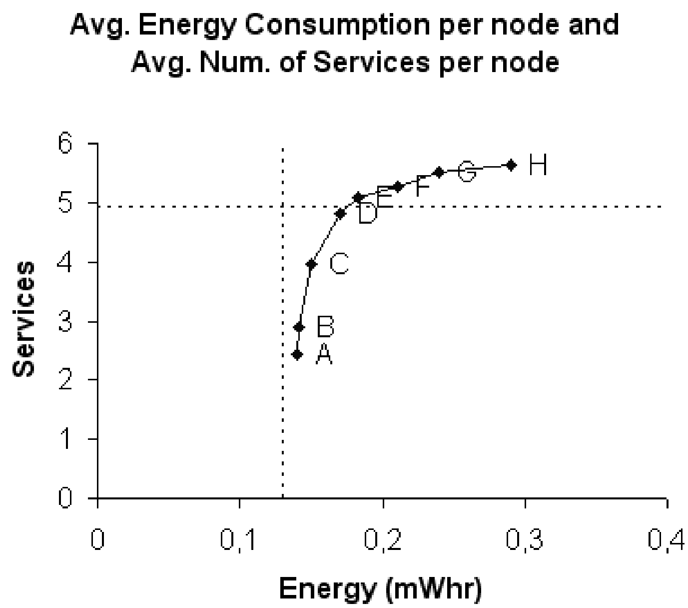


Fig. 7: Relating Average Energy Consumption per node to Average Number of Services discovered per node without mobility for Flooding.

It is evident that the application layer based service discovery scheme (Flooding) may perform better than the routing layer based scheme in terms of service discoverability for broadcast intervals lower than 40 seconds. However, this comes at the cost of energy consumption, which is increased 30% or more compared to the routing layer based scheme with the original broadcast interval of 10 seconds. This is again explained by the fact that the messages of the application layer scheme are much shorter (in order to be more economic) and hence less informative than those of the routing layer based scheme. So, service discoverability is reduced by reducing the number of broadcasted messages (bigger intervals means fewer messages transmitted, hence every node receives less information about services).

In the fourth set of experiments, the two schemes are tested in a mobile context. All the parameters (e.g. regarding node mobility) besides flooding broadcast interval and service deletion interval are the same as those used at the second set of experiments analyzed in previous paragraphs.

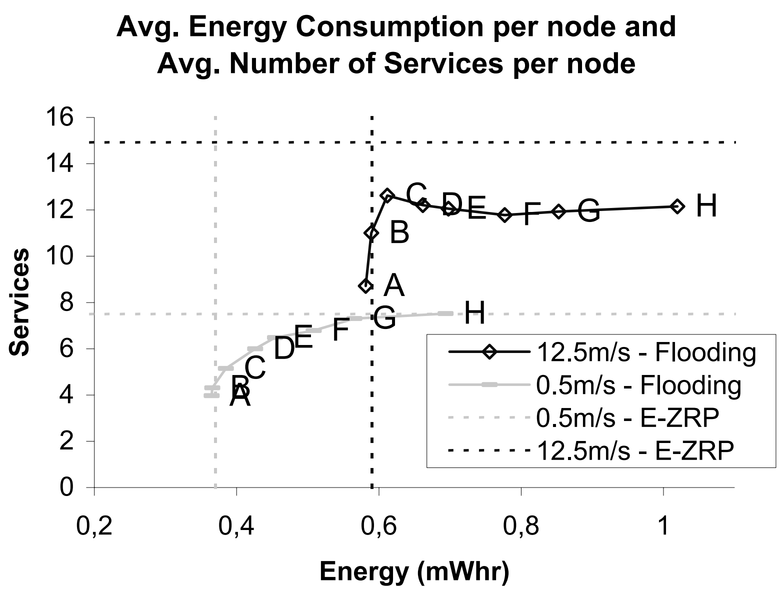


Fig. 8: Relating Average Energy Consumption per node to Average Number of Services discovered per node for high and low mobility.

Fig. 8 depicts the results for service discovery and energy consumption respectively in this mobile context. Each experiment was run over a network of 250 nodes (density remains fixed as in previous experiments). We study 2 extreme cases of mobility. The first case is for low mobility, where nodes move according to the random waypoint mobility model with minimum speed 0m/s, maximum speed 0,5m/s and pause time 30 seconds. The second case is for high mobility, where the mobility parameter of maximum speed changes to 12,5m/s. Each point on both curves corresponds to different parameter settings for the update and service deletion intervals for the Flooding protocol (those presented in Table II).

As shown in Fig. 8 the application layer based service discovery scheme performs better in terms of energy consumption (compared to the routing layer based scheme – dotted lines) when the broadcast interval is equal or more than 160 seconds (point B) saving 3% more power but discovering 43% fewer services for low mobility cases and 22% fewer services for high mobility cases.

In order to evaluate the quality of discovered services using E-ZRP we also conducted the following experiments. In this case we assumed that each node may host one out of three possible services, which can be provided to other nodes, and runs E-ZRP as its routing and discovery protocol. The selection of any of these 3 services has the same probability for any node, hence at the end of the allocation 1/3 of the node population hosts the first service, another 1/3 hosts the second service and the last 1/3 hosts the third service. In this context we replace the ‘number of discovered services’ metric with the ‘number of discovered service sessions’ metric. The last metric is more meaningful in an environment where each service provider doesn’t host a unique service, but a service belonging to a common set of service types. A service session begins from the moment a node discovers one or more service providers of a given service type until the moment it looses communications with all the service providers of that specific service type (while there is at least one service provider of the requested service type the visible to the node, the session for the specific service is considered alive). In this context the SAD metric measures the service session lifetime instead of the service lifetime as in previous experiments.

We simulated a network comprising 20 nodes uniformly dispersed in a 4000x4000 meters square area. We used a random waypoint mobility model. First, we tested the sensitivity of Service Availability Duration (SAD) at different speeds. We simulated five different scenarios. In the first scenario each node’s speed (in meters/second) was distributed between 0 and 3,5m/s (low mobility), in the second scenario between 0 and 7m/s (medium mobility), in the third scenario between 0 and 9m/s (medium mobility), in the fourth scenario between 0 and 11m/s (high mobility) and in the last scenario between 0 and 14m/s (high mobility). The zone radius for E-ZRP was set to 3 hops. In order to capture the effects of mobility per se on the performance of E-ZRP and Flooding we have used a perfect channel (at the end of the section we also evaluate under a noisy channel). The simulation duration was 2000 seconds in every experiment (each scenario was run 10 times with different simulation seeds and the results represent averages).

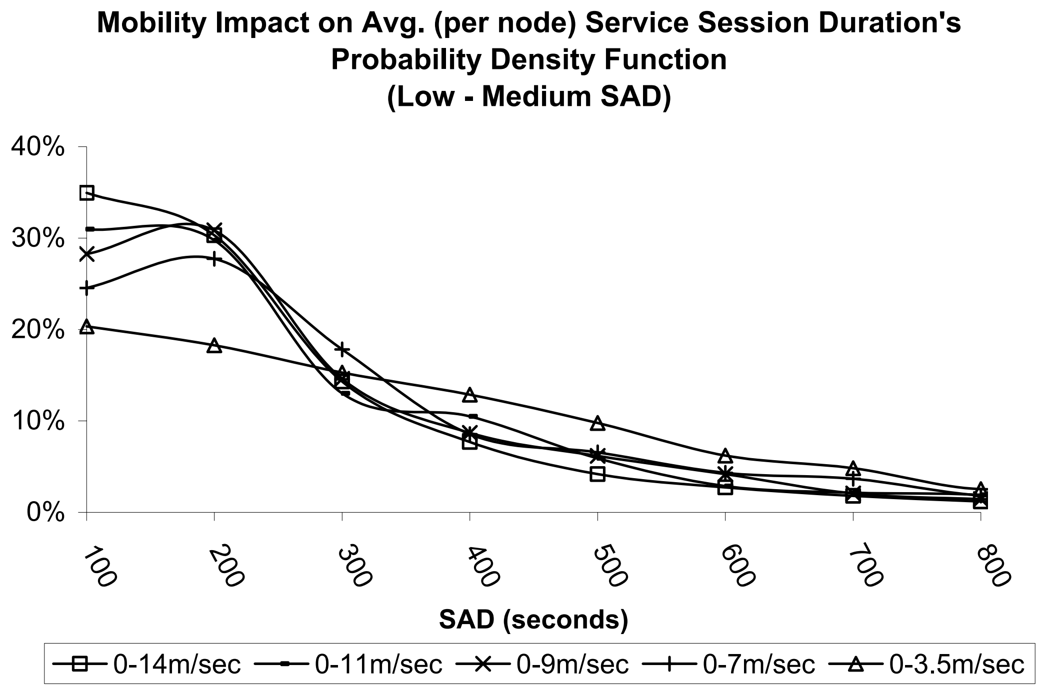


Fig.9: Avg. Service Session Duration PDF vs. Speed for E-ZRP (Low-Medium SAD).

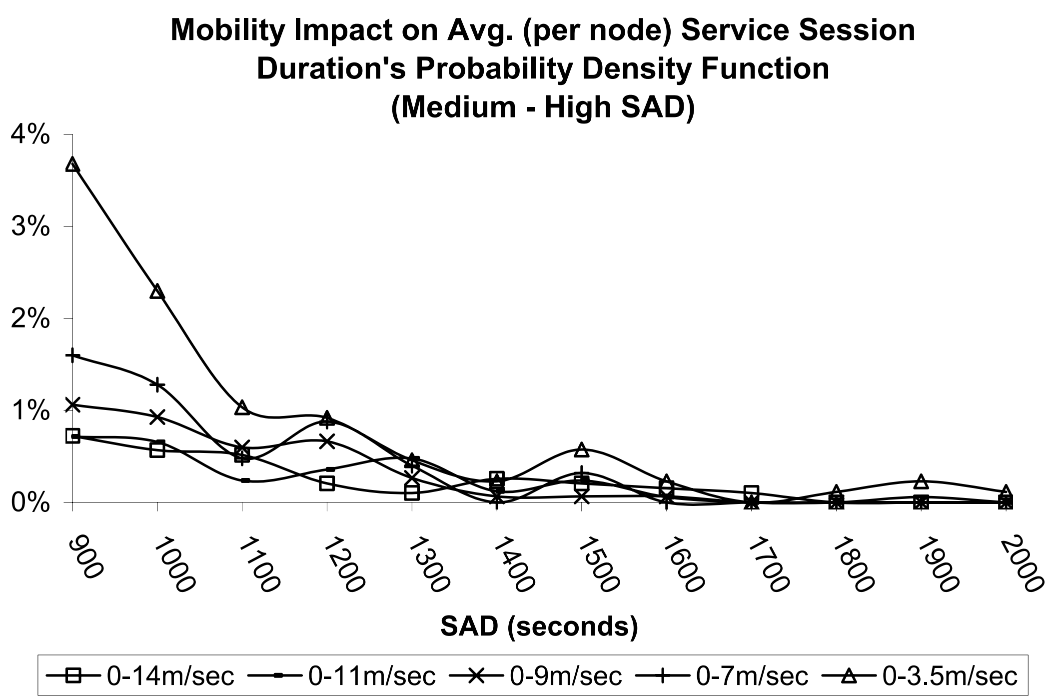


Fig.10: Avg. Service Session Duration PDF vs. Speed for E-ZRP (Medium-High SAD)

Fig.9 and 10, depict the results for these experiments. The X-axis represents the time for which a service remains visible to a node (SAD metric), and the Y-axis represents the average number of service sessions experienced per node. Since in the network every node is a service provider and there are 3 service types, the optimal results would be to have 2 service sessions per node each having duration of 2000 seconds (simulation duration). This implies that each node has discovered all the service types (excluding its own) and has kept connectivity to them until the end of the simulation. As we can see, it is more probable for E-ZRP to discovers short-lived services in highly mobile environments (due to node mobility and service rediscoveries), while more long-lived services can be discovered only in low mobility cases. This is explained by the fact that when the nodes are highly mobile, paths are difficult to be maintained and hence far-away services tend to last for a very short amount of time since the probability for a path break is larger when nodes move faster. When nodes move slower these paths tend to be more stable and hence services tend to be available for a longer time.

However, it is not obvious from these figures when we can achieve the maximum average SAD, which is a metric of great importance in analyzing the quality of discovered services.

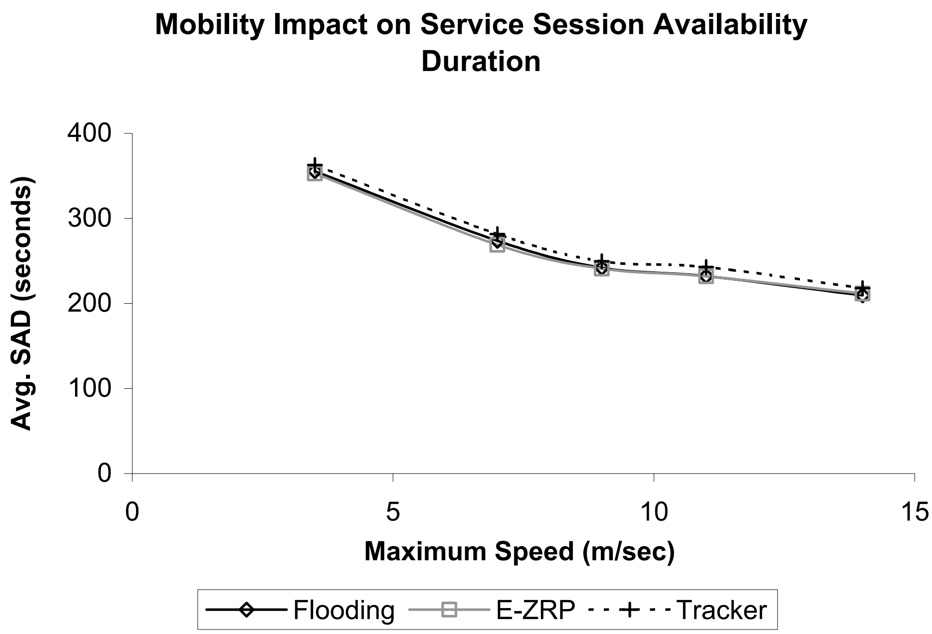


Fig.11: Average SAD vs. Speed.

The values of average SAD over low, medium and high mobility are presented in Fig.11, where we also present the SAD for the Flooding protocol given the same settings. The lines connecting the 5 spots in the figure do not correspond to results for speeds other than the five defined above, but are drawn for better viewing. We have also implemented a tracking protocol, which measures the realistic connectivity between the nodes in the network taking into account only their Euclidian distances. This protocol, called Tracker, checks the physical distances of nodes on the terrain and calculates the connectivity graph. Then, knowing the types of services offered by the nodes it calculates the realistic service duration time for all nodes of the graph. In order to allow the same service disconnection tolerance followed by Flooding and E-ZRP (40 seconds), the Tracker protocol considers a service active if connectivity to any of its providers has been detected at least once during the last period of 40 seconds. In case that no such connectivity has been detected it removes the service from the node's cache and keeps a record of its duration. Under the given density and the perfect channel assumption, both protocols closely follow the Tracker protocol and hence accurately reflect the realistic connectivity among nodes. It is also evident from this figure that the average SAD actually decreases when the speed increases both for E-ZRP and Flooding. However, it wouldn't be fair to compare the performance of the protocols with respect to service duration only. The amount of service sessions discovered is also important, since it is usually preferable for a node to discover a small number of service sessions with long durations, throughout its lifetime, instead of a high number of service sessions with small durations.

In Fig.12 we show the average number of service sessions discovered per node in case of low, medium and high mobility.

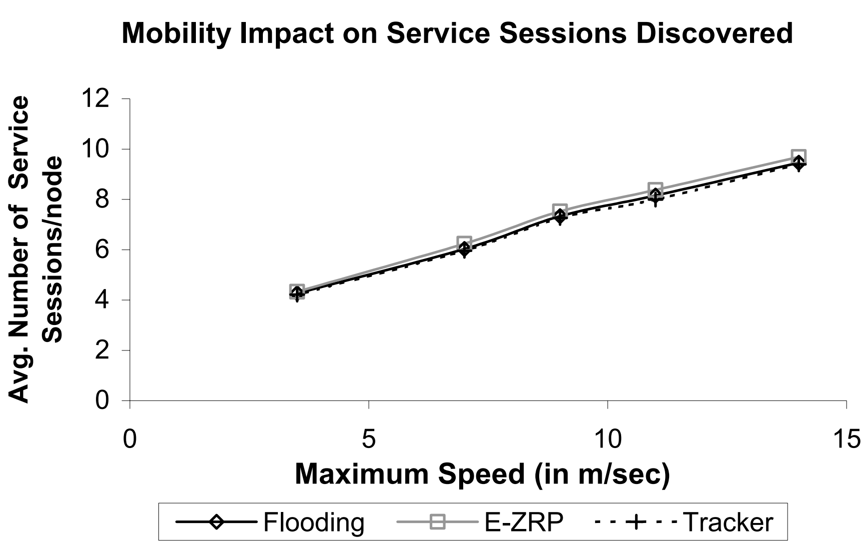


Fig.12: Avg. Number of Service Sessions Discovered vs. Speed.

As expected, the high mobility case (maximum speed = 14m/s) outperforms all the other in the number of service sessions discovered both for E-ZRP and Flooding. So, there is a tradeoff between average SAD and number of service sessions. In order to evaluate when our protocol performs better, we should be aware of the average transaction duration (ATD) between a node and any service. So, for high ATD, the discovery protocol would perform better in a low mobility setting. This is explained by the fact that the additional service sessions discovered in higher mobility settings would be of no use, because their average SAD would be inadequate to complete a transaction. However the discovery protocol would perform well even in a high mobility setting for low ATD.

Now related to density, we simulated three scenarios. The first scenario included 20 nodes moving on a terrain of 2000x2000 meters, following the random waypoint model with maximum speed ranging from 3,5m/s to 14m/s (minimum speed is 0m/s). The zone radius for E-ZRP was set to 3 hops. The second scenario (half density scenario) was identical to the previous one but included only 10 nodes. Both scenarios had duration of 2000 seconds each (each scenario was run 10 times with different simulation seeds and the results represent averages). The results are shown in Fig.13, where it is obvious that by reducing node density to one half, the number of long-lived service sessions in the half-density case is significantly smaller than the number of the long-lived service sessions found in the full-density case. This is due to the fact that re-discoveries of services are more frequent in a denser environment.

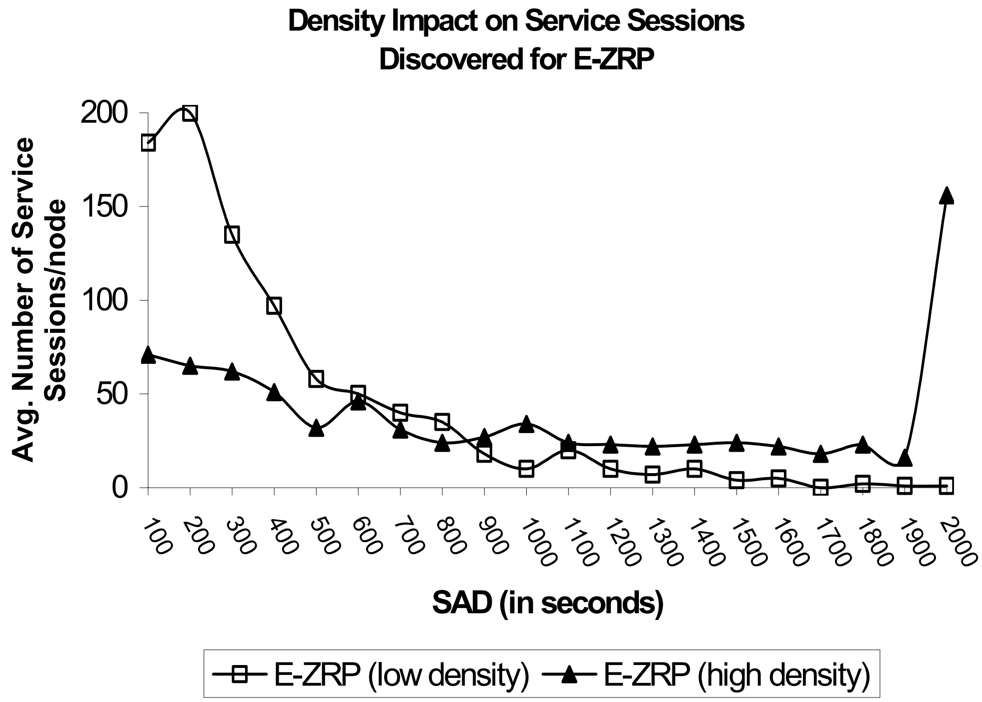


Fig.13: Service Duration Distribution vs. Density for E-ZRP.

One would expect that in the denser environment services would tend to last longer, since there are more alternative paths to a service provider through which a node can reach a service and also more alternative service providers, hence a failure of one or more paths doesn't necessarily mean that the node cannot access the given service. Simulation results presented in Table III validate this. Actually, when density increases, due to the existence of multiple paths and providers, the average service duration is increased. The total number of service sessions discovered, is also higher in denser environments (Table III).

TABLE III
AVERAGE SAD VS. DENSITY

	Full Density (20 nodes)	Half Density (10 nodes)
Avg. SAD	*963 s*	*352s*
Avg. Service Sessions/node	*3,97*	*8,87*

The third scenario included 20 nodes moving in one case on a terrain of 2000x2000 meters (high density case) and on a second case on a terrain of 4000x4000 meters (low density case), following the random waypoint model with maximum speed ranging between 3,5m/s and 14m/s (minimum speed is still 0m/s). Both E-ZRP and the Flooding protocol are evaluated under these two different densities using a zone (respectively flooding) range of 3 hops. Fig. 14 presents the performance of E-ZRP and Flooding regarding SAD for the two aforementioned densities under varying speeds while Fig. 15 presents the performance regarding average service sessions for each protocol.

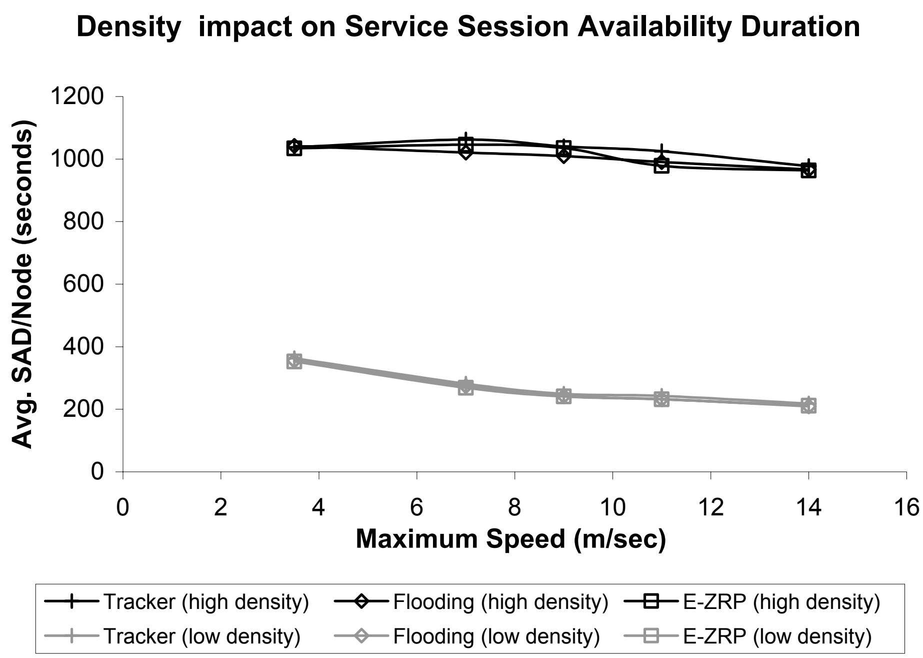


Fig.14: Density impact on SAD.

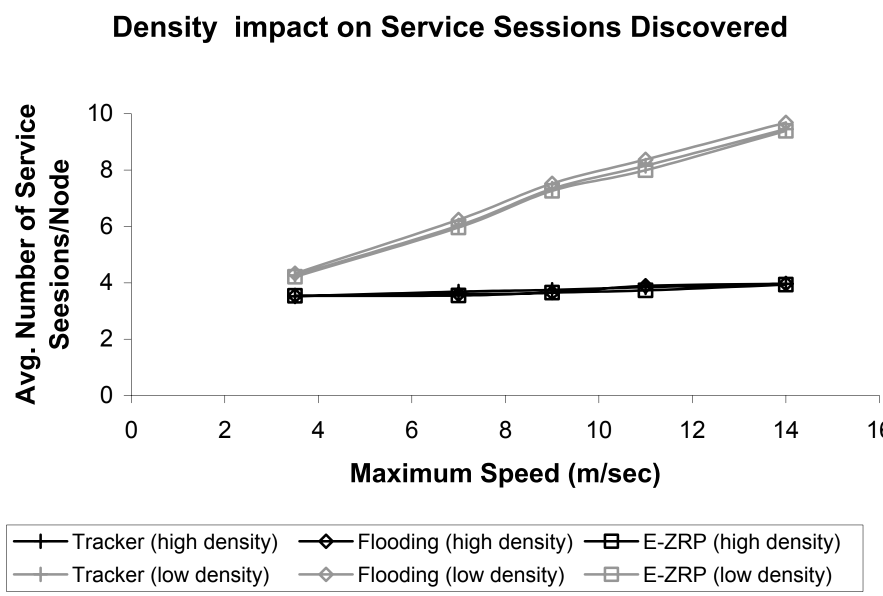


Fig.15: Density impact on Service Sessions.

Both protocols provide increased SADs for denser environments and tend to discover a lower number of service sessions for such environments, which is explained by the fact that better connectivity is provided and fewer service session breaks occur.

As stated earlier, the above simulations used a perfect channel in order to reveal the effects of mobility on the performance of both protocols. In the following experiment we assume a realistic (affected by noise) channel in order to also see the effects of the channel on the performance of the two protocols. For this we have simulated a network consisting of 20 nodes moving on a terrain of 2000x2000 meters, following the random waypoint model with maximum speed ranging between 3,5m/s and 14m/s (minimum speed is 0m/s). The zone (respectively flooding) range is set to 3 hops. In Fig.16 and Fig.17 we present the results.

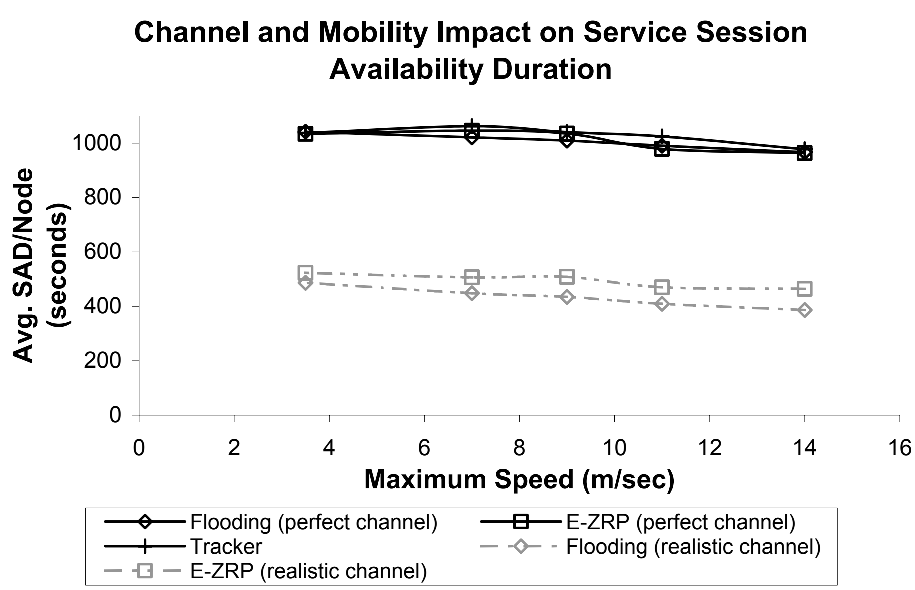


Fig.16: Channel and mobility impact on SAD.

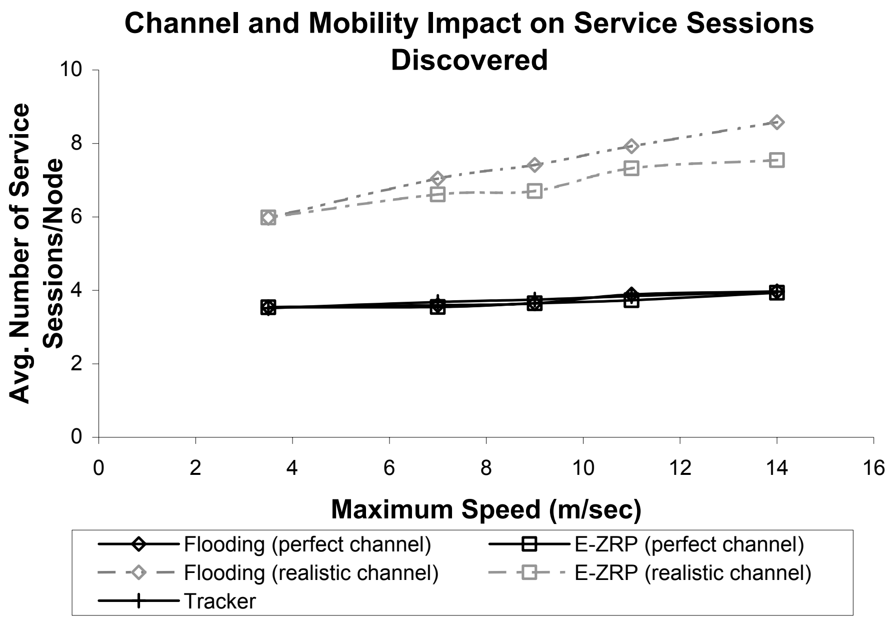


Fig.17: Channel and mobility impact on Service Sessions.

It is evident that E-ZRP performs better than the Flooding protocol under realistic situations (noisy channel). This is due to the fact that an IARP message encapsulates more information regarding the services available in the neighborhood of a node, compared to the information carried by a Flooding message, which only informs the receiving node about the service of one of its neighbors.

Hence, in a realistic (noisy) environment with packet losses, loosing a Flooding packet costs more in constructing an accurate view of the available services as compared to loosing an IARP packet (since IARP packets from different neighbors contain overlapping information for the zone of the receiving node). Combining the results presented in Fig.16 and Fig.17 this is validated, since the Flooding protocol is shown to discover more short-lived service sessions than the E-ZRP. Also in the case of the Flooding protocol the fact that there exists an additional and separate packet sending process at every node's routing layer, that of the routing protocol (traditional ZRP in our case) worsens the channel conditions. From the simulator's packet traces we identified increased packet collisions due to the existence of both protocols.

Another issue worth investigating is the impact of E-ZRP's zone radius both on SAD and number of discovered service sessions. We evaluate E-ZRP's performance using zones of 1 up to 20 hops for 20 nodes moving on a terrain of 2000x2000 meters for 2000 seconds both for perfect and realistic channels.

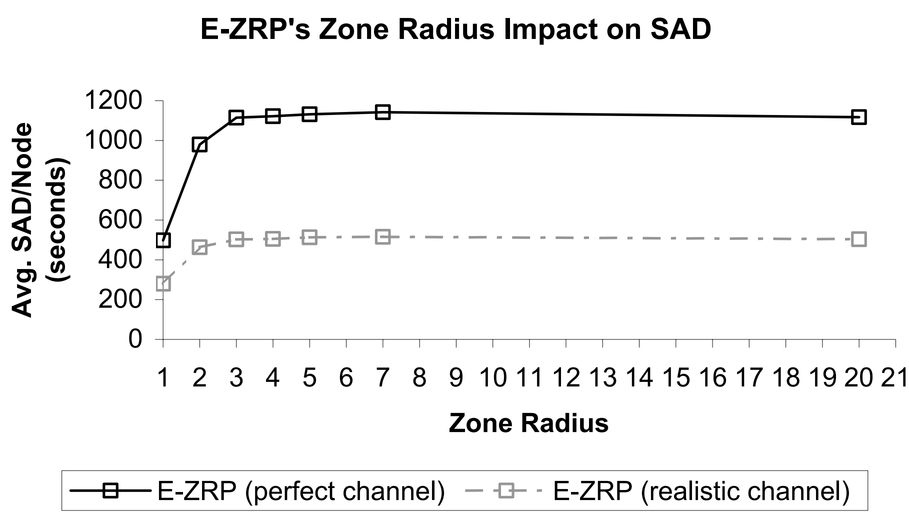


Fig.18: E-ZRP's Zone Radius Impact on SAD.

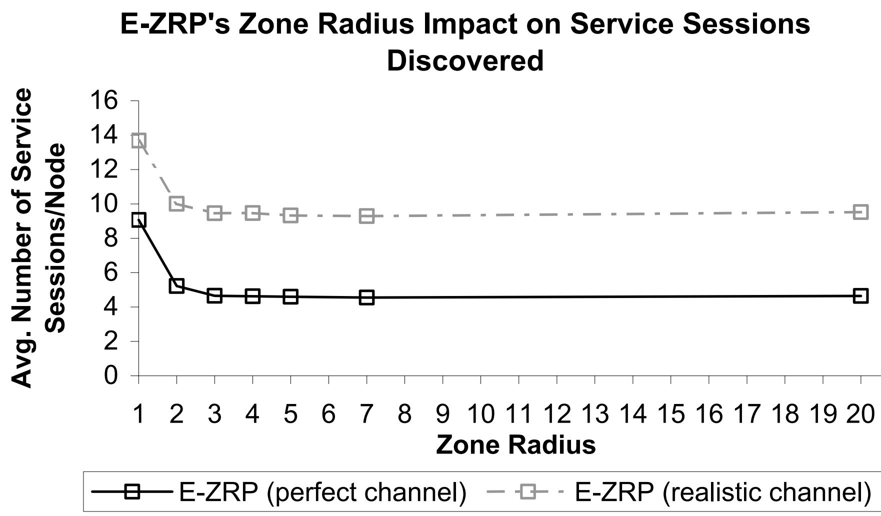


Fig.19: E-ZRP's Zone Radius Impact on Service Sessions Discovered.

It is obvious, from Fig.18 and Fig.19, that for the given network density, the node mobility and the degree of replication of the 3 available service types among nodes, increasing the zone radius more than a threshold (in our case 2 hops) does not provide any significant extra gains but leads to highly increased energy consumption as shown in Table IV below. It is part of our future work to investigate ways to optimally tune the zone radius based on the network conditions.

TABLE IV
AVERAGE ENERGY CONSUMPTION VS. E-ZRP ZONE RADIUS

Zone Radius (in hops)	Avg. Energy (in mWhr)
1	*0,485755*
2	*0,485755*
3	*2,223715*
4	*3,4445*
5	*4,320735*
7	*5,122245*
20	*10,49182*

All the above results concerned in-zone service discovery mainly. Regarding out-of-zone service discovery, we compared E-ZRP with a non-proactive Flooding protocol. Non-proactive flooding means that this kind of flooding doesn't issue constant updates for maintaining service discovery information. In the contrary it is a one-shot process. This kind of flooding proved to achieve minor energy savings (5%) overall compared to E-ZRP, since E-ZRP constantly updates intra-zone service information and hence spends more energy. However, if we take into account the energy consumption (for discovering an out-of-zone service) imposed only from the reactive part of E-ZRP (IERP), which uses bordercasting instead of broadcasting, then again E-ZRP proves to be more energy efficient than flooding. Flooding also imposes significant delays for discovering out of zone services. In Fig.20 we provide our experimental results showing the delays imposed by both protocols under different zone settings and different number of hops-to-provider. Each point on the diagram is an average obtained over 20 service discovery requests between different node pairs having the same hop distance. Giving delays to discover a service in the area of 10 to 50 milliseconds, it is clear that E-ZRP outperforms Flooding, where using the latter a node needs from 200 milliseconds up to 800 milliseconds to discover a service. Of course for both protocols the further the requested service is located (in number of hops) the larger the delay to discover it. However since E-ZRP uses the mechanism of bordercasting, it can efficiently and quickly "scan" distant areas of the network to find the requested service. Flooding takes a long time to "scan" the network since it relies on hop-by-hop broadcasting.

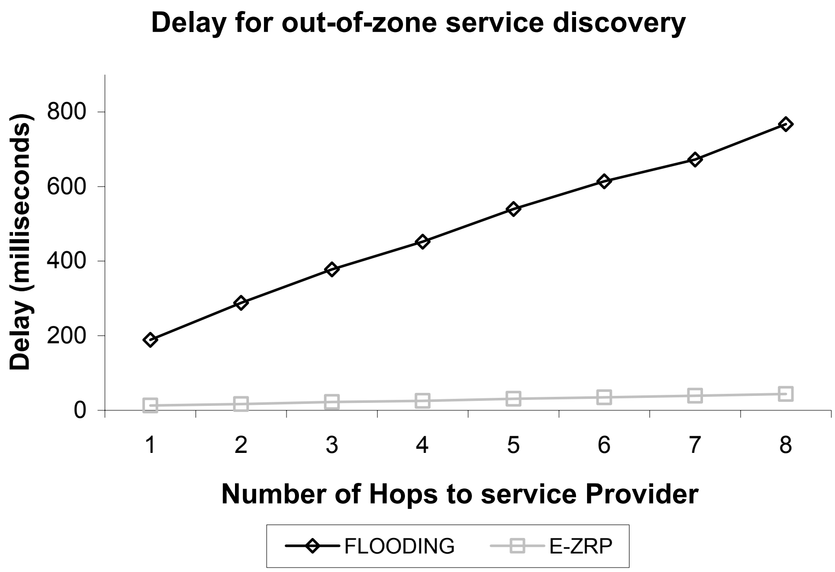


Fig.20: Delay for out-of-zone service discovery.

V. CONCLUSIONS

Most previous research efforts on service discovery do not investigate and do not report on energy consumption, neither do they comment on service availability. Also, existing application layer based service discovery architectures suffer from redundant packet transmissions in their effort to discover routes towards the services (in the sense that control messages for information discovery are required at both the network and application layers).

We have presented a new architecture that integrates service discovery functionality with an existing routing protocol. In this paper we examined the implications of network density and node mobility on the availability of services discovered with a representative routing layer based service discovery protocol we designed, namely E-ZRP. We have experimentally shown that our scheme consistently outperforms an application-layer service discovery scheme based on range bounded flooding in terms of energy consumption, both in static and mobile environments. . E-ZRP leads to significantly lower energy consumption (approximately 50% less), but also, in certain cases, it achieves higher service discoverability. It was also shown that ‘favoring’ the application layer based service discovery protocol with larger flooding time intervals (in order to become more economical in terms of energy consumption, leading to savings of 3%), had a detrimental effect on service discoverability, reducing it by 22% or more, compared to the proposed routing layer based protocol. Our experiments for out-of-zone services revealed that E-ZRP consumes 5% more energy than (restricted-area) flooding, but achieves an order of magnitude lower delay for discovering services. Finally we introduced a new metric called SAD (Service Availability Duration) for measuring the quality of discovered services and examined the implications of network density and node mobility on the availability of services discovered with a representative routing layer based service discovery protocol (namely E-ZRP).

REFERENCES

1. C.N. Ververidis and G.C. Polyzos, “Extended ZRP: Performance Evaluation of a Routing Layer Based Service Discovery Protocol for Mobile Ad Hoc Networks,” Proc. Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous’05), pp. 114-123, San Diego, CA, July 2005.
2. Sun Microsystems, JINI Architecture Specification, Nov. 1999.
3. Salutation Consortium, “Salutation Architecture Specification,” http://www.salutation.org/specordr.htm, 1999.
4. Microsoft Corporation, "Universal Plug and Play: Background", http://www.upnp.org/ resources/UPnPbkgnd.htm, 1999.
5. Universal Description Discovery and Integration Platform, http://www.uddi.org/pubs/ Iru_UDDI_Technical_White_Paper.pdf, Sept. 2000.
6. “Specification of the Bluetooth System,” http://www.bluetooth.com, December 1999.
7. E. Guttman, C. Perkins, J. Veizades, and M. Day. “Service Location Protocol, Version 2,” IETF RFC 2608, June 1999.
8. O. Ratsimor, D. Chakraborty, S. Tolia, D. Kushraj, A. Kunjithapatham, G. Gupta, A. Joshi, T. Finin, “Allia: Alliance-based Service Discovery for Ad-Hoc Environments,” Proc. ACM Mobile Commerce Workshop, September 2002.
9. D. Chakraborty and A. Joshi, "GSD: A novel group-based service discovery protocol for MANETS", Proc. IEEE Conference on Mobile and Wireless Communications Networks, Stockholm, Sweden, September 2002.
10. M. Nidd, "Service Discovery in DEAPspace", *IEEE Personal Communications*, August 2001, pp. 39-45.
11. S. Helal, N. Desai, V. Verma, and C. Lee, "Konark - A Service Discovery and Delivery Protocol for Ad-Hoc Networks", Proc. 3rd IEEE Conference on Wireless Communication Networks (WCNC), New Orleans, Louisiana, March 2003.
12. G. Schiele, C. Becker and K. Rothermel,“Energy-Efficient Cluster-based Service Discovery for Ubiquitous Computing,” Proc. 11th ACM SIGOPS European Workshop, Leuven, Belgium, September 2004.
13. A. Helmy, S. Garg, N. Nahata and P.Pamu, “CARD: A contact-based Architecture for Resource Discovery in Wireless Ad Hoc Networks,” *Mobile Networks and Applications* (MONET), Vol 10, pp.99-113, 2005.
14. R. Koodli and C. E. Perkins, “Service discovery in on-demand ad hoc networks,” IETF Internet Draft, draft-koodli-manet-servicediscovery-00.txt, October 2002.

15. Z.J. Haas, M.R Pearlman, P. Samar, "The Zone Routing Protocol (ZRP) for Ad Hoc Networks," IETF Internet Draft, draft-ietf-manet-zone-zrp-04.txt, July 2002.

16. Scalable Network Technologies Inc., "Qualnet Simulator," http://www.scalable-networks.com, 2005.

17. N. Sadagopan, F. Bai, B. Krishnamachari, A. Helmy, "PATHS: Analysis of PATH Duration Statistics and their Impact on Reactive MANET Routing Protocols," Proc. ACM MobiHoc'03, Annapolis, MD, USA, 2003.

18. F. Bai, N. Sadagopan, B. Krishnamachary, A. Helmy, "Modeling Path Duration Distributions in MANETs and their Impact on Routing Performance," *IEEE JSAC*, Vol. 22, No. 7, pp. 1357-1373, Sept. 2004.

APPENDIX I

The structure of the IARP packet header is the following (12 bytes when not counting neighbour information):

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Link Source Address                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Link State Seq Num      |  Zone Radius  |      TTL      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          RESERVED             |   ServiceID   | Link Dest Cnt |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Link Destination 1 Address                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   ServiceID   |  Metric Type  |          Metric Value         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                              |  |
                              |  |
                             \|  |/
                              \  /
                               \/
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Link Destination n Address                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   ServiceID   |  Metric Type  |          Metric Value         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The structure of the Flooding packet is the following (12 bytes):

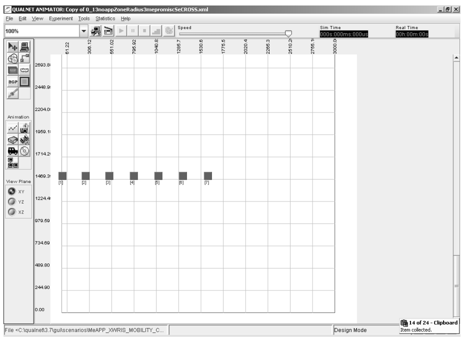
```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Link Source Address                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   ServiceID   |    HopCount   |           RESERVED            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            RESERVED                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The size of the Flooding packet was selected to be the same as the size of an IARP packet without including neighbor information for achieving a fair comparison of the two protocols.

APPENDIX II

In the following we present simulation results regarding the energy consumption of Flooding and E-ZRP for various static topologies:

Topology type: **Chain**

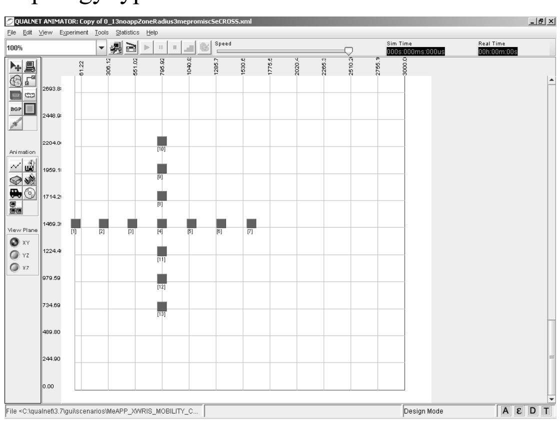


The node population for this topology was 7. Each node hosts a unique service. Zone radius and flooding range are set to 3 hops. The results on energy consumption and average number of discovered services per node are:

	E-ZRP	**Flooding**
Avg. Services Discovered/Node	4,23	4,26
Avg. Energy/Node (mWhr)	0,13	0,27

E-ZRP to Flooding Energy Consumption Ratio = 0,13/0,27 = 0,481

Topology type: **Cross**

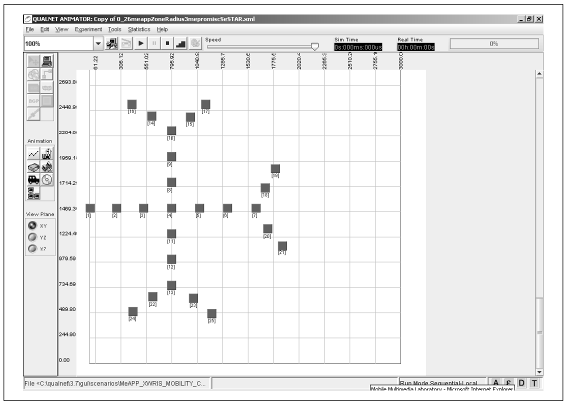


The node population for this topology was 13. The rest setup parameters are the same with those used in the chain topology. The results on energy consumption and average number of discovered services per node are:

	E-ZRP	**Flooding**
Avg. Services Discovered/Node	6,7	7,3
Avg. Energy/Node (mWhr)	0,20	0,43

E-ZRP to Flooding Energy Consumption Ratio = 0,20/0,43 = 0,465

Topology type: **Snowflake**

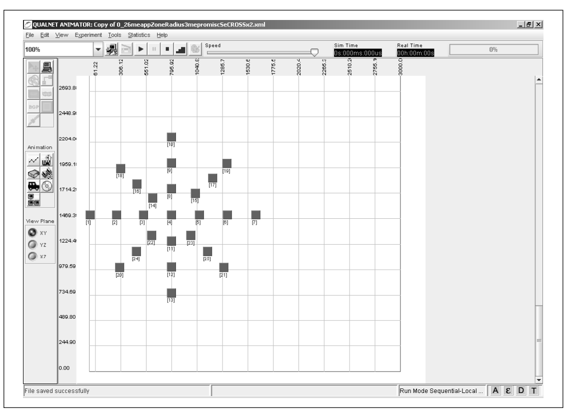


The node population for this topology was 26. The rest setup parameters are the same with those used in the chain topology. The results on energy consumption and average number of discovered services per node are:

	E-ZRP	Flooding
Avg. Services Discovered/Node	6,8	7,4
Avg. Energy/Node (mWhr)	0,23	0,48

E-ZRP to Flooding Energy Consumption Ratio = 0,23/0,48 = 0,479

Topology type: **Star**



The node population for this topology was 26. The rest setup parameters are the same with those used in the chain topology. The results on energy consumption and average number of discovered services per node are:

	E-ZRP	Flooding
Avg. Services Discovered/Node	14,58	14,56
Avg. Energy/Node (mWhr)	0,51	0,99

E-ZRP to Flooding Energy Consumption Ratio = 0,51/0,99 = 0,515

It is evident that the Energy Consumption Ratio of the two protocols remains approximately the same for any topology; hence selecting the simplest topology (chain topology) would be the best choice.