Optimized Service Selection for MANETs using an AODV-based Service Discovery Protocol[*]

Sotirios E. Athanaileas, Christopher N. Ververidis and George C. Polyzos
Mobile Multimedia Laboratory, Department of Computer Science,
Athens University of Economics and Business,
Athens 10434, Greece
{sath, chris, polyzos}@aueb.gr

***Abstract*- Mobile Ad hoc Networks (MANETs) are networks, which operate without the need of an infrastructure. In such networks, mobile nodes must rely on their own capabilities and on other peers' capabilities to perform their tasks. In this context, many service discovery protocols have been proposed in order for mobile peers to be able to discover and take advantage of each other's services. A basic building block for service discovery protocols is their service selection strategy. Especially in service-rich environments with many (possibly resource poor) mobile nodes, the service selection strategy employed is of major importance. In this paper we investigate the impact of two basic and easy to implement service selection strategies on the lifetime of mobile servers. The first strategy takes into account hop-based server proximity and promotes the selection of the nearest mobile server, while the second strategy takes into account the remaining energy of service providers and promotes the selection of the mobile server with the maximum remaining energy. Through extensive simulations, we show that the provider's remaining energy yields the best performance when used as a service selection criterion under most situations, and that the shortest path selection criterion presents very competitive performance. The performance metrics used for evaluating both service selection strategies are service and network lifetimes, service success and service discoverability ratios.**

I. INTRODUCTION

Service discovery for Mobile Ad hoc Networks has recently attracted much attention by researchers, as it presents very challenging issues that need careful attention. MANETs are highly dynamic networks and this dynamism along with their infrastructure-less nature make service discovery difficult. A basic building block for service discovery protocols for MANETs, often neglected, is their service selection strategy. In this paper we investigate and discuss the performance of two simple service selection strategies that can be easily integrated to any service discovery protocol in terms of server and network lifetime duration and successful service invocation ratios[1]. What differentiates these two strategies is the type of metric used by each for selecting a service provider. More specifically:

- The first service selection strategy follows a classic approach commonly used by many service discovery protocols i.e. it minimizes the hop-count (network-specific metric) between a service requestor and a service provider. In this strategy when there are many service providers, capable of satisfying a user's request, the closest server to the requesting node is always selected.
- In the contrary the second service selection strategy utilizes a server-specific metric instead of a network-specific metric. This server-specific metric is the remaining energy of the service provider. In this strategy the most energy rich server is always selected among the set of discovered servers offering matching services. This strategy is the natural strategy that should be followed in order to prolong the lifetime of service providers. What needs to be investigated is how it affects service discoverability and the success of service invocations.

These two strategies have different goals. The first strategy aims at localizing traffic and avoiding long paths between servers and clients, while the second strategy aims at extending server lifetimes by performing implicit load balancing. In our investigation

---

[*] This research has been supported in part by the project Autonomic Network Architecture (ANA, #IST-27489), which is funded by the IST FET Program of the European Commission.

[1] The successful service invocation ratio is the ratio of the number of successfully completed service transactions to the number of total service transactions requested.

we take into account several factors like node density, node speed and client to provider ratio, in order to evaluate the two strategies under a great variety of possible MANET environments.

The structure of the remainder of the paper is as follows: Section II presents related work on service selection mechanisms, in Section III we describe our implementation of two representative service selection strategies embedded to a service and route discovery protocol, sections IV and V discuss simulation setup and experimental results respectively and Section VI concludes the paper.

II. RELATED WORK

Many service discovery approaches have been proposed in the literature, with the most efficient in terms of energy consumption being cross layer approaches, which try to embed service discovery functionality into routing protocols [2][9][10][11][13]. These cross layer approaches aim to minimize energy consumption by combining service and routing information into routing packets. This way, redundant transmissions of service discovery packets at the application layer are avoided and a lot of energy is saved (see results of [9]). However, besides the basic service discovery process another process that can be modified in order to take into account (and possibly save) energy is service selection. Service selection can be categorized to *automatic* and user *assisted* [4]. User assisted service selection requires the active participation of the user in the selection process. In such case the user has to run through a list of discovered services and select the best service that satisfies his/her needs. However, pervasive devices (e.g. PDAs) forming MANETs impose many limitations in such a process. On the one hand such devices have limited capabilities (i.e, small screen size, limited Graphical User Interfaces) and on the other hand it is hard for users on the move to concentrate (and also loose time) on reviewing service lists for selecting the most appropriate one. In a fast changing environment, where services appear and disappear in an unexpected way, it is crucial to employ a fast and efficient service selection process, which will also not distract the user.

This has lead researchers to investigate automatic service selection mechanisms, mainly based on service ranking systems with the ranking function requiring only an initial parameterization by the user. This parameterization regards assigning weights to various desirable service characteristics, so that user preferences can be reflected by a ranking function. A representative example of such a mechanism is found in [1], in which authors propose that users customize their selection algorithm and embed it in a mobile agent. Agents are then transferred to the service providers and compute a rank based on the specified metrics. They then send back to the requestor these ranks and based on a local user policy the desired service is selected.

However, simpler approaches have also been developed. For example, in [6] authors propose the exploitation of past service interaction information (i.e. number of previous interactions, length of interaction etc.) in order to select services based on previous experience. Services with a better ‘historical record’ are preferred among a discovered set of similar services. Another simple mechanism for automatic service selection mechanism could employ a keyword similarity function to measure the ‘distance’ between keywords given for a service request and the keywords in the discovered services, in order to determine the best match (in terms of degree of similarity).

In general a service selection strategy is based on certain criteria or metrics. These metrics can be either route (e.g. hop-count, bandwidth, delay) or service (e.g. server mobility, load, remaining energy, capacity) specific. In the discovery protocols proposed in [2] and [3] authors employ the lowest hop-count metric for selecting the service that is the closest to the requesting node. In [5] the discovery protocol proposed (based on proactive server advertisements) selects a service instance based on two metrics, the hop-count between service requestor and service provider and also the capacity of service (CoS), which expresses the nominal capacity of a service instance.

In this paper our contribution is to investigate how a service selection strategy based on a route specific metric (hop-count) and a service selection strategy based on a service specific metric (remaining energy) affect the discovery process in a MANET by using an AODV-based service discovery protocol. We select the hop-count metric since it is the most representative of route-specific metrics and also commonly used by service selection mechanisms [3][5][7][8]. For the second service selection strategy, the remaining energy is selected, among other candidate service specific metrics, since energy preservation is of major importance in energy-constrained environments like MANETs.

III. ANALYSIS OF THE PROPOSED SERVICE SELECTION STRATEGIES

In this paper we evaluate two different service selection mechanisms embedded on the extension of the AODV [12] routing protocol for service discovery as proposed in [13].

Ad hoc On demand Distance Vector (AODV) is an on demand, source initiated routing protocol for mobile ad hoc networks. All routes in AODV are discovered by a request/reply cycle: each time a node requires a route to another node of the network; it creates and broadcasts a route request (RREQ) message. A node that receives this RREQ can reply (unicast) with a route reply (RREP) message if it is the destination node or if it has a valid and 'fresh' enough route to the destination node.

One interesting feature of AODV is the use of an expanding ring search technique to prevent unnecessary network-wide dissemination of RREQs. In an expanding ring search, the originating node initially uses a time to live (TTL) named TTL_START in the RREQ packet IP header and sets the timeout for receiving a RREP to a fixed time represented by the protocol's RING_TRAVERSAL_TIME parameter. If the RREQ times out without a corresponding RREP, the originator broadcasts the RREQ again with the TTL value incremented by TTL_INCREMENT. This continues until the TTL set in the RREQ reaches a threshold (AODV's TTL_THRESHOLD parameter), beyond which the TTL is set equal to the estimated network diameter (represented by AODV's parameter called NET_DIAMETER) in number of hops. When it is desired to have all retries traverse the entire ad hoc network, this can be achieved by configuring TTL_START and TTL_INCREMENT both to be the same value as NET_DIAMETER [12].

Our approach is to implement service discovery in the routing layer by piggybacking the service information into the AODV's control messages, thus enabling the devices to acquire both service and routing information simultaneously, as proposed in [13]. The necessary extra fields for this purpose are: a) the Service Type Identifier Length, the Service Type Identifier [14] and an optional Attributes List in every service request (SREQ) message, and b) the Length of the extensions, the service Lifetime, the Length of the service URL and the URL itself in each service reply (SREP) message.

In our implementation every node caches the services it provides and the services that other nodes provide in its Service Table (along with all the necessary service information, such as lifetime, provider address etc.) and if more than one providers of the same service exists, then it selects one according with the service selection strategy that it uses.

In order for a node to invoke a service, it must have a valid service binding (a binding associating a service id to address(es) of provider node(s)) and a route to the resolved provider. If the node has a service binding but no route to the resolved address it creates a SREQ with the destination address set equal to the resolved (from the binding) IP address, otherwise it creates a SREQ message with a void destination address and broadcasts the message.

When a node receives a SREQ it checks all the fields of the message and executes the following actions: If the message does not contain a destination address, then the node will check if it has a service binding and a route to the resolved provider address and if it has it will send a SREP message back to the requestor. If the node has a service binding but no route to the provider, it will create a new SREQ with the destination address field set equal to the resolved provider address. If it has more than one provider for the requested service it will fill the destination address field with the selected provider, according to the service selection strategy that the node uses. In any other case the node will rebroadcast the SREQ unchanged.

If the message contains a destination address then the node will check if it has a valid route to that destination and if it has, it will send a SREP message including the service type and lifetime that is cached in its Service Table. If the node does not know a route to the destination, it will look up whether it has any alternative service bindings (along with a route to the provider address) for the requested service. If so, it sends an SREP including service time and the service lifetime cached in its table, if not it just rebroadcasts the SREQ.

We have implemented two versions of the AODV based service discovery protocol, each with a different service selection strategy embedded in its logic. The first service selection strategy is based on selecting the provider that is closest to the service requestor. The proximity between the service requestor and the service provider is calculated in hops and not in meters, so, when we say that we select the closest service provider we do not mean that we select the physically closest provider, but the one that is the least hops away. If a node discovers many providers offering the requested service and which have the minimum hop count, it selects one of them randomly.

To determine the number of hops that each service requestor is away from the service providers, we use the hop_count field of the SREP that is sent by the service provider (or an intermediate node that has a service binding and a route to the resolved provider) as a reply to the SREQ that the service requestor has sent.

The second service selection strategy is based on selecting the service provider with the highest remaining energy. To achieve this goal we added an extra field to the default SREP message, that we name provider_remaining_energy and which contains the remaining energy of the service provider. When a service provider receives a SREQ message from a node requesting the provider's service, it creates an extended SREP message, containing the remaining energy of the provider.

Whenever a node receives a fresher SREP message (the freshness is determined by the Destination Sequence Number of the SREP message) from a provider that is already cached in the Service Table of the node, it renews the service provider's entry with the fresher remaining energy of the provider. In this way the Service Table of each node in the MANET is kept up-to-date.

IV. SIMULATION SETUP AND ANALYSIS

In this section we discuss the simulation setup that we used in order to evaluate the performance of the service selection strategies that we presented in the previous section. For running our simulation scenarios we used the J-Sim simulator [15][16] with J-Sim's Wireless Package [17]. We have implemented in J-Sim the extension of AODV [12] for service discovery as proposed in [13] along with the two service selection strategies already presented. We have also discovered and fixed a bug in J-Sim's implementation of AODV, which has to do with the TTL values that are assigned to each packet sent to the network and which affects the expanding ring search technique used by AODV.

Each node of our scenarios was equipped with the default wireless network card that J-Sim provides, which has 250m transmission range, 2Mbps transfer rate, 660mW transmission power and 395mW receiving power as implemented in the WirelessPhy component of J-Sim. All our simulation scenarios used distributed coordination function (DCF) of IEEE 802.11 as the MAC protocol. To simulate a service transaction we used the file sharing protocol (fsp), as implemented in the fsp component of J-Sim. Each service provider had a copy of the same file (with a fixed size of 34Kb), which represents the data that have to be transferred to a client as the result of a service call/invocation. Clients request a service transaction every 20 seconds.

Our simulated network consists of a total number of nodes, varying from 25 to 39, which are randomly and uniformly placed on a square flat space, varying from (400m x 400m) to (1000m x 1000m). The initial energy of all the nodes is set equal to 50J. All the nodes of the scenarios move following the Random Waypoint Model (RWM) with no pause time. In J-Sim's implementation of RWM, a node randomly chooses a position in the simulated area as its destination and moves in a straight line to that destination point at a constant speed which is uniformly distributed between 0 and Max_Speed (we have experimented with various values for Max_Speed, from 1,5m/s to 18,5m/s). Whenever the node arrives at the destination, it chooses the next position and repeats the procedure again.

We simulated several scenarios with various client/server ratios (from 10clients/15servers to 24clients/15servers). At the start of the simulation of each scenario, every client of the scenario waits for a random amount of time T (in sec) before it starts requesting the file sharing service and after that time has passed, it starts making a request every 20 seconds. The T period before the initial request of a client is used because we wanted to minimize the possibility of all the nodes requesting the service at the same time.

In all of our scenarios, the mobile nodes that formed the network were either clients or providers. When a client sends a SREQ message, requesting for the file sharing service, it waits for 2 seconds in order to discover all the available providers in its vicinity (as specified by the TTL value of AODV). Then according to the service selection strategy of the scenario, it selects the closest provider (if the closest-provider selection strategy is used) or the provider that has the most remaining energy (if the energy-aware selection strategy is used). For every simulation scenario, the results we present are averages over 5 experiments.

All scenarios started with a 200 second 'initialization' phase, where all the nodes just move around the simulation terrain, without requesting any service. After 200 seconds of simulation, clients start to request services. Each simulation scenario runs for 1200 seconds (200 sec 'initialization' phase and 1000 sec 'service discovery' phase).

V. SIMULATION RESULTS

We have evaluated the performance of the service selection strategies in terms of:

- Service Provider Lifetime (SPL): the total amount of time that every service provider 'lives'. In all of our scenarios the only reason why a provider can 'die' is because of depleting its battery resources (by sending/ receiving network messages).
- Number of Discovered Providers (DP): the total number of times that the service requestors of the network have discovered at least one provider of

the desirable service. The main reasons why a service provider cannot be found is a) because all the providers of the network have ‘died’ (have depleted their battery resources) or b) the network has been divided into two or more segments and no path between the requestor and a provider can be found.

- Number of Successful Service Invocation (SSI): the total number of times that the service requestors of the network have successfully invoked the desirable service, which practically means that they have successfully downloaded the entire file. The reasons why a requestor might not successfully download the entire file is a) because of the mobility of the nodes, the path that connects the requestor and the provider may break or b) because a provider that serves a service requestor may deplete its energy resources and ‘die’ before it has sent the entire file. Case b is applied to intermediate nodes of a path as well.

In every subsection that follows we present three figures, which depict the results from each set of the simulation scenarios. The first figure of the set always shows the service providers’ lifetime, as the percentage of the total time that the simulation lasted (providers’ lifetime/total simulation time). Each dot in this figure represents the average value of the participating service providers’ lifetime for each set of scenarios. The second and the third figure of each set depict each strategy’s performance in terms of DP and SSI respectively. Every dot in these Figures represents the average value of the DP and SSI for each set of scenarios. Our confidence in all the results we present is 95%.

In this point, we should also stress out that in almost 83% of times that we have a service selection the two strategies select different providers.

A. Varying the Clients to Service Providers Ratio

Figures 1, 2 and 3 show the results from the simulation of the scenarios, where we randomly placed a varying number of clients (from 10 to 24 clients with a +2 increment) and a constant number of providers (15 providers) on a square terrain size of 400mx400m. Every participating node was moving with Max_Speed=1,5m/s and the shared file’s size were set to 34Kb.

The first thing that we observe from the above Figures is the increment in DP and SSI, as the number of clients increases. This occurs because of the way that we have built our scenarios and the fact that each client sends 1 SREQ per 20 seconds. So when we have more clients, more SREQ will be sent and the DP will increase. With respect to the increase of the SSI, the 400m x 400m network is dense enough not to allow many path breaks between clients and providers.

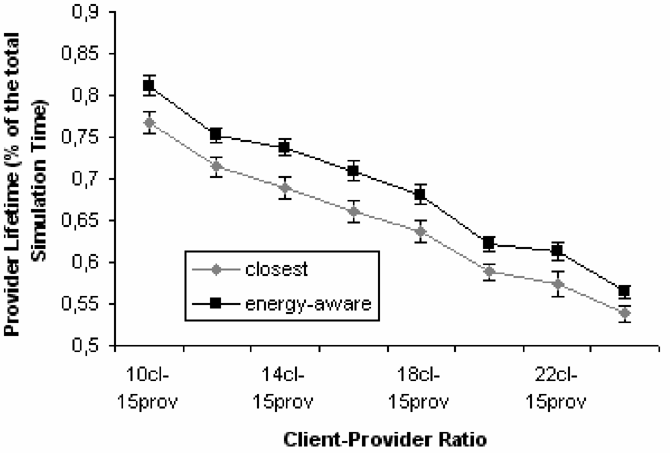

Figure 1: SPL vs. number of clients

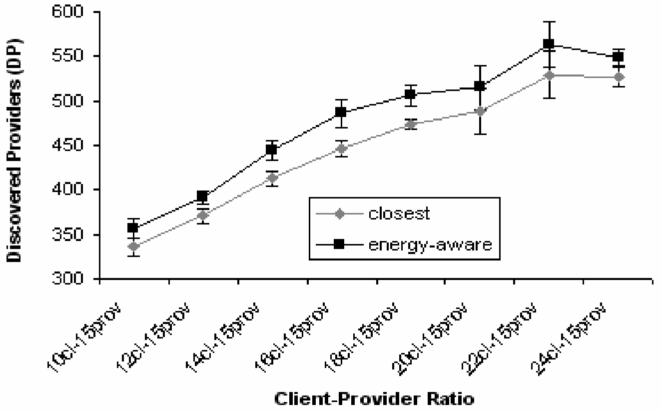

Figure 2: DP vs. number of clients

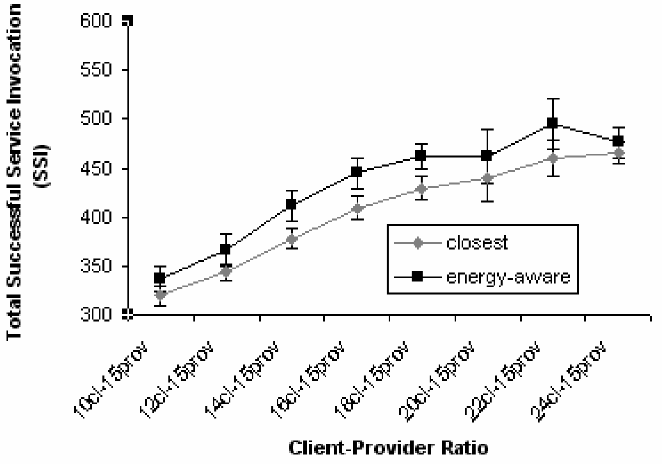

Figure 3: SSI vs. number of clients

Another thing that we must mention is the fact that the SSI/DP ratio remains the same in all the sets of scenarios (approximately 92%). The ratio’s high number can be explained if we take into consideration the fact that the network is very dense and that the shared file’s size is only 34Kb (the downloading of the entire file requires less than 0,3 seconds). In other words the network’s density, the rate of requests and the size of the exchanged file do not saturate the network. This means that there is no congestion and more requests can be satisfied.

As we can see in Figure 1, the service providers' lifetime decreases as the client number increases. This can be easily explained if we consider that when we have more clients, then more SREQ are sent, so each provider of the network has to serve more requests, depleting its battery resources at a faster rate.

It is obvious that the energy-aware service selection strategy is better than the closest service selection strategy for all the performance metrics and that the overall trend is the same for both strategies. Note however, that the energy-aware service selection strategy demands an extra field in the RREP messages to include remaining energy information, while the closest service selection strategy uses the hop count field that is already present in RREP messages (no extensions needed). Furthermore, closest service selection strategy presents lower time delays, because it always selects the closest provider. Taking these into account we could say that employing the "simpler" shortest path based service selection strategy gives very competitive results as compared to the energy aware strategy which is considered optimal (when talking about energy preservation). In this point we should also stress out the role of the expanding ring search (ERS) technique that AODV uses in the process of discovering and selecting a provider in the network. In ERS TTL_START is set to 1, so in dense networks energy-aware service selection tends to act pretty much like closest service selection, whereas it also selects the provider with the maximum remaining energy of the provider.

B. Varying Terrain Sizes

Figures 4, 5 and 6 show the results from the simulation of the scenarios, where we randomly placed 10 clients and 15 providers on a square terrain size varying from 400m x 400m to 1000m x 1000m. Every participating node was moving with Max_Speed=1,5m/s and the shared file's size were set equal to 34Kb.

As we can see from Figure 4, the more the terrain size increases the more the providers live. This happens, because as the terrain size increases the network becomes sparser and each provider expends less energy either for serving clients or for 'overhearing'/and or relaying neighbor-nodes' packets, since in sparser networks the neighbors of a node are less in number.

The most interesting result is depicted in Figures 5 and 6, where it is shown that as the terrain size increases, so do the DP and SSI numbers, until they reach a maximum point at 800m x 800m terrain size and then they start to decrease. What really happens is that in dense networks (i.e. 400m x 400m) the providers have a short life because of the message 'overhearing'. In this case the main reason for the depletion of a provider's battery is actually the "overhearing" process and not the actual "serving" process.

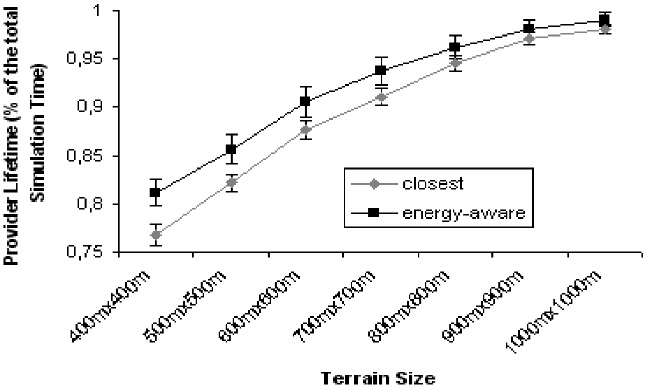

Figure 4: SPL vs. terrain size

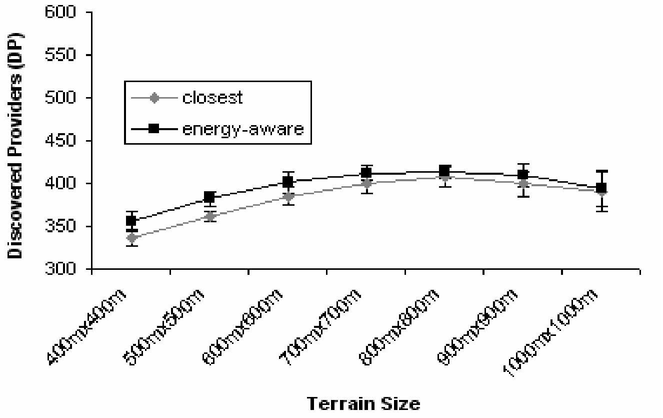

Figure 5: DP vs. terrain size

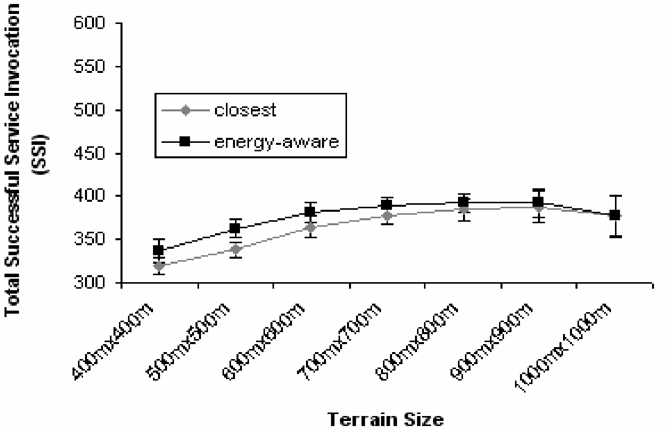

Figure 6: SSI vs. terrain size

As the terrain size increases (the number of participating nodes remains fixed) and the density becomes smaller, servers expend their energy mainly for "serving" purposes and this explains the rise in the DP and SSI number. However, further increasing the terrain size (beyond 800x800) causes DP and SSI to decrease. This happens due to the fact that when the network becomes so sparse, the clients cannot find paths to connect with the providers (because of network segmentations), which also explains the high service providers' lifetime. At around 800m x 800m terrain size is in fact optimal, because the network is neither dense nor sparse, that is why the DP and SSI numbers reach the maximum.

In all the above Figures we can see that the closest service selection strategy closely follows the energy-aware service selection strategy regarding all the performance metrics, but as the terrain size increases the two strategies finally converge. There are two reasons for this: a) In sparse networks the probability to find many providers after a SREQ is small. What actually happens in very sparse networks is that after sending a SREQ, a node will receive SREPs from very few providers (possibly at almost the same distance, due to the expanding ring search operation of AODV) and so choosing the server with the maximum energy among them yields marginally better results than randomly selecting one of them. b) In the rare case that a client discovers many providers (at different distances) in a sparse network, the paths towards the service providers will possibly include many hops and will be very susceptible to path breaks. Following the closest service selection strategy decreases as much as possible the probability of a path break since the closest server is always selected. In this case the closest service selection strategy performs marginally better than the energy-aware selection strategy.

C. Varying Mobility

Figures 7, 8 and 9 show the results from the simulation of the scenarios, where we randomly placed 10 clients and 15 providers on a square terrain size of 400m x 400m. Every participating node moves with varying Max_Speed from 1,5m/s to 18,5m/s and the shared file's size is set equal to 34Kb.

As we can see in Figures 8 and 9, the DP number remains almost constant, while the SSI number shows a slight decrease as the Max_Speed value increases in both service selection strategies. The reason why the DP remains constant is because of the density of the network. The network is dense enough for every requestor to discover at least one provider in each SREQ. Furthermore as the maximum speed increases the SSI number shows a slight (almost unnoticeable) decrease (due to more frequent path breaks). We have conducted additional simulations with file sizes reaching 1Mbyte and the decreasing trend of SSI just becomes more evident.

The service providers' lifetime (Figure 7) remains constant for all Max_Speed values. This happens, mostly because the SSI number and the DP number remain almost constant too (for the reasons we mentioned above). So, the providers of the nerwork serve almost the same amount of SREQs in all speeds and that is why they 'live' for almost the same amount of time.

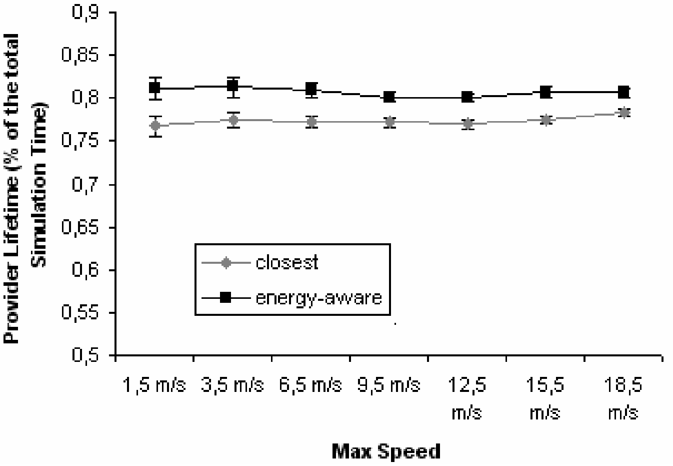

Figure 7: SPL vs. Max_Speed

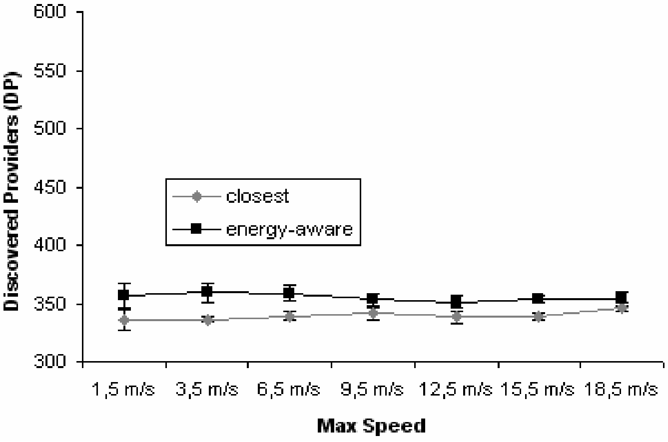

Figure 8: DP vs. Max_Speed

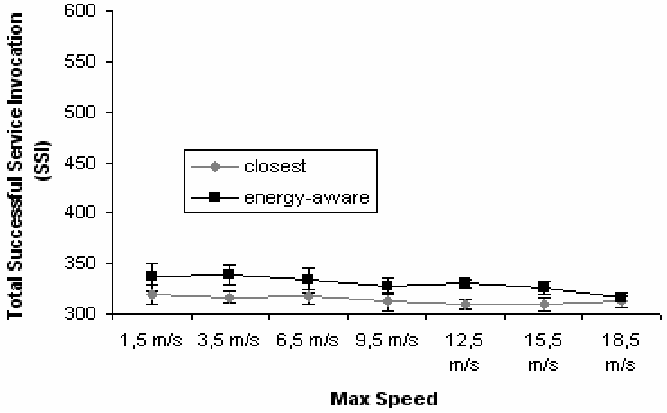

Figure 9: SSI vs. Max_Speed

Finally as we can see in all the above Figures, the proposed energy-aware service selection strategy presents slightly increased service providers' lifetimes (as expected), while it is also marginally better in terms of DP and SSI number. This last result is not intuitive, since one would expect that selecting the closest server especially in highly mobile environments (with frequent path breaks) would lead to more successful service invocations compared to selecting a distant and energy rich server. Looking into simulation traces we discovered that the main reason for this is that the ERS operation of AODV allows the energy aware policy to make a selection only over the closest servers (selecting the one with the maximum remaining energy) and not distant ones (outside of the ERS scope).

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have integrated an energy-aware service selection strategy into an AODV-based service discovery protocol for MANETs. We evaluated the proposed strategy through a simulation study and compared it with the strategy of selecting the closest (shortest hop) provider, which is a popular strategy used in many approaches for service discovery in MANETs.

The results show that the energy unaware service selection strategy (shortest path) is capable of achieving similar performance to the energy aware service selection strategy. In our future work we plan to investigate the impact of those parameters in AODV that affect the expanding ring search operation, like the values of TTL_START and TTL_INCREMENT, since in this study we have used the default values, as specified in the protocol's RFC [12]. Furthermore, we are designing a 'hybrid' service selection algorithm, which also takes into account the entire path's (towards the provider) energy, so that the selection will not only be based on the provider's remaining energy, but on the path's energy as well.

To conclude, it is our belief that optimizing the service selection mechanisms is an important step towards energy consumption minimization for performing service discovery in MANETs.

REFERENCES

[1] J. Tyan and Q. H. Mahmoud, "A network layer based architecture for service discovery in mobile ad hoc networks," Proc. IEEE CCECE 2004, Niagara Falls, May 2004.

[2] A. Varshavsky, B. Reid and Eyal de Lara, "The Need for Cross-Layer Service Discovery in MANETs," Technical Report CSRG-492, UofT Computer Science, 2004.

[3] C. Frank and H. Karl, "Consistency Challenges of Service Discovery in Mobile Ad Hoc Networks," Proc. 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), pp. 105-114, Venice, Italy, October 2004.

[4] M. Barbeau, "Service discovery protocols for ad hoc networking," CASCON 2000, Workshop on Ad Hoc Communications, Mississauga, Ontario, Canada, 2000.

[5] V. Lenders, M. May and B. Plattner, "Service Discovery in Mobile Ad Hoc Networks: A Field Theoretic Approach," Elsevier Journal on Pervasive and Mobile Computing (PMC), Volume 1, Issue 3, pp. 343-370, September 2005.

[6] Karypidis, A., Lalis, S., "Exploiting co-location history for efficient service selection in ubiquitous computing systems," Proc. 2nd Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, (MobiQuitous 2005), pp. 202 – 209, July 2005.

[7] J. Wu and M. Zitterbart, "Service Awareness and Its Challenges in Mobile Ad Hoc Networks", Wksp. Comp.Science 2001: Mobile Commun. over Wireless LAN: Research and Apps., pp. 551–57, Sept. 2001.

[8] G.P. Halkes, A. Baggio, K.G. Langendoen, "A Simulation Study of Integrated Service Discovery", Proceedings of the 1st European Conference on Smart Sensing and Context (EuroSCC 2006), pp. 39-53.

[9] 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. 2nd Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous'05), pp. 114-123, San Diego, California, July 2005.

[10] J.A. Garcia-Macias, D.A. Torres, "Service discovery in mobile ad-hoc networks: better at the network layer?", Proceedings of the 2005 International Conference on Parallel Processing Workshops (ICPPW'05), pp. 452-457, 14-17 June 2005.

[11] P.E. Engelstad, Y. Zheng, R. Koodli,, C.E. Perkins, "Service Discovery Architectures for On-Demand Ad Hoc Networks", International Journal of Ad Hoc and Sensor Wireless Networks, Old City Publishing (OCP Science), Vol. 2. Number 1, pp. 27-58, March 2006.

[12] C. Perkins, E. Belding-Royer, S. Das, Ad hoc On-Demand Distance Vector (AODV) Routing, RFC 3561, July 2003.

[13] Rajeev Koodli and Charles E. Perkins, service discovery in on-demand ad hoc networks, IETF Draft, draft-koodli-manet-servicediscovery-00.txt, work in progress, October 2002.

[14] E. Guttman, C. Perkins, J. Veizades, and M. Day. Service Location Protocol, Version 2. RFC 2608, Internet Engineering Task Force, June 1999.

[15] The J-Sim simulator, http://www.j-sim.org

[16] The Tcl/Java Project website, tcljava.sourceforge.net/

[17] J-Sim's wireless package, http://www.j-sim.org/v1.3/wireless/wireless_tutorial.htm