Green Service Discovery Protocol in Mobile Ad Hoc Networks

Janine Kniess
Santa Catarina State University (UDESC)
Joinville - SC, Brazil
janine@joinville.udesc.br

Orlando Loques, Célio V. N. Albuquerque
Fluminense Federal University (UFF)
Niterói - RJ, Brazil
loques@ic.uff.br, celio@ic.uff.br

***Abstract*—Energy efficiency is an important aspect of green computing. In order to achieve green computing in service discovery protocols for MANETs, it is essential for researchers to investigate energy efficiency mechanisms able to prevent collisions and interferences, which both represent a waste of channel bandwidth and energy. In this paper, we propose a service discovery protocol that applies a data aggregation scheme in intermediate nodes to reduce messages replies in these networks. The results show that this mechanism makes it possible to maintain the trade off between the discovery success rate and the message reduction, hence, minimizing the energy consumption.**

Keywords-energy; discovery; selection services.

I. INTRODUCTION

A mobile ad hoc network (MANET) is a self-organizing and dynamically reconfigurable wireless network which, operates in the absence of a fixed infrastructure [1]. In MANETs, most mobile devices have limited processing capabilities and operate using limited peak voltage batteries, which imposes a stringent limitation on the instantaneous energy consumption they can suport [2]. In addition, wireless communications are prone to collisions and interferences due to the broadcasting nature of the radio medium. In this context, careful transmission mechanisms prevent collisions and interferences, which both represent a waste of channel bandwidth and energy [3]. Some of the issues identified in MANETs are service discovery, mobility management and energy consumption. Service discovery is an essential component for the usability of such networks on the grounds that service discovery enables devices to use their functions to automatically locate network services.

The extremely dynamic nature of MANETs has motivated the development of their application in different scenarios like MANETs in assisting emergency missions and taxi service. In such applications, the service provider's location is not previously known. As a consequence, a common approach to disseminate requests is to use broadcast mechanisms [4]. This is often the case where multiple providers can offer the specified service. In this case, a data aggregation approach can be used to minimize the reply transmission cost in network.

This paper proposes a location-aware service discovery protocol for MANETs composed of an autonomic, distributed and location-aware service selection mechanism that uses a data aggregation scheme in intermediate nodes to combine the replies coming from different providers, minimizing the number of transmissions, thus improving the network performance and saving energy. Also the protocol provides a service discovery mechanism that adjusts a search area for each individual request in order to save energy.

In applications such as MANETs in assisting emergency missions, the maximum response time to attend one request is essential to guarantees that service discovery is successful. Thus, we identified a number of aspects as being critical to a successful outcome: the geographic location where the service provider is being requested; the maximum response time (we suppose that the provider needs to arrive at the place where the service is required within a maximum time); the speed at which the service provider moves; and the number of providers that must be requested. The proposed protocol takes into account the above mentioned aspects in the service discovery process.

Briefly, our contributions are the proposals of: (1) a service selection mechanism that applies a data aggregation scheme in intermediate nodes to filter the responses aiming to reduce the exceeding replies and save energy; (2) a service discovery mechanism that adjusts a search area for each individual request; and (3) a service invocation mechanism that specifies how the service providers will be accessed and used by the requester node.

This paper is organized as follows: Section 2 presents the motivation scenario and the main related proposals regarding service discovery for MANETs; Section 3 introduces the Location Aware Discovery Protocol (LADP); Section 4 describes the evaluation and the results and, finally, Section 5 presents the final conclusion and suggests further research directions.

II. APPLICATION CONTEXT AND RELATED WORK

Service discovery in MANETs has emerged as a useful application in scenarios where the service providers are mobile and must be efficient and capable to attend a request within a maximum time, for instance, MANETs in assisting emergency missions.

ISBN: 978-1-61208-259-2

A. Application Scenario

In this context, suppose an area that has been struck by natural disaster. In this case, we believe that the distribution of a specialized rescue team (formed by vehicles, robots and humans beings, for example), interconnected by a wireless mobile ad hoc network could contribute greatly to the rescue of survivors.

In the proposed scenario, some of the mobile elements of the rescue team provide a resource, i.e., an ambulance, a robot with the ability to access places that are not reachable by man, such as areas with a contamination risk, or even a human being transporting medication. The nodes that provide services are the providers and those that request the service are the requesters. We assume that each node in the network is aware of its geographic position by means of a localization system, such as a Global Positioning System (GPS), and the time in the devices is synchronized. For instance, is possible that a node identifies the existence of a leak of a given type of gas and, from specialized algorithms, the node can derive the need for a provider, such as a fire engine, which could be at the location in a matter of minutes. After identifying which resource is needed, the node will send a request message to the network searching for the appropriate resource.

B. Related Work

Service discovery is the action of finding and locating a service in the network [5] and service discovery protocols can be generally classified into three categories [4]. In chronological order we mention: fixed network protocols such as Jini [6] and Service Location Protocol (SLP) [7]; wireless single−hop networks such a Distributed Embedded Application Platform space (DEAPspace) [8] and Bluetooth [9]; and wireless multi−hop networks. Neither the first category (fixed network protocols), nor the second one (wireless single−hop networks) are feasible for decentralized environments such as the one presented in the scenario above. In these environments, it is possible to create multi-hop networks. The following protocols stand out in this last category: Konark Gossip [10], FTA (Field Theoretic Approach) protocol [11], the P2P Discovery Protocol (P2PDP) [12] and the cross-layer approach of Varshavsky et al. [13].

Although considerable previous research has been done into service discovery for distributed systems, few protocols have considered the development of distributed aggregation responses in the selection phase in order to save energy. Service selection is the phase that comes after service replies are gathered by the service requester.

The approaches that have been proposed by Lenders et al. [11], Gomes et al. [12] and Varshavsky et al. [13] provide a mechanism for autonomic service selection. However, energy consumption issues are not considered in these works.

The FTA approach proposed by Lenders et al. [11] is based on the theory of electrostatic fields. Requests to an instance of a given service type are routed selectively in the direction of the provider that generated the highest field gradient. However, this approach does not scale well when different types of services are available. In Varshavsky et al. [13] the selection service function is integrated into the routing mechanism in a cross-layer approach. Moreover, the selection occurs only when the replies arrive at the requester node. In Gomes et al. [12] it shows a suppression vicinity service selection mechanism that can discard exceeding replies during the reply transmission. However, this mechanism filters response messages through the reverse path, i.e., messages are routed by a reverse path traversed by the request. This concept can easily fail in MANETs due to highly dynamic topologies.

Unlike the mentioned approaches, our work also considers that the providers must move as far as the place where the resource is needed. The other approaches consider that the services can be remotely accessed and used.

III. THE PROPOSED SERVICE DISCOVERY PROTOCOL

In this section, we will firstly describe how the Location Aware Discovery Service (LADS) [14] can save energy in MANETs. Secondly, we will present the Location Aware Service Selection (LASS) scheme to aggregate replies in intermediate nodes, and to select the best providers during the reply transmission. Table 1 summarizes the notation used in this paper.

A. Location Aware Discovery Service (LADS)

This section describes the strategy adopted by the service discovery mechanism, LADS to save energy in service discovery for MANETs.

LADS works as follows. Suppose a node in the network needs information about service providers and sends discovery messages. The LADS mechanism limits the search diameter R_i, on the basis of the maximum speed that a node can reach, v_{max} (each type of resource knows this value), and the maximum response time for one request, Δ_{tmax}. Using R_i, this mechanism prevents unnecessary request and reply transmissions in the network. The diameter R_i is given by the equation:

$$R_i \leftarrow v_{max} \times \Delta_{tmax} \tag{1}$$

The mechanism defines the diameter using v_{max} so that the search area includes the greatest number of apt providers. Given the pair (*i*,*j*), being *i* the requester and *j* the provider, it is assumed that the speed (v_j) of the latter is known.

After sending a message, the requester node starts the timer, $\Delta_{trequester}$. This timer defines the maximum amount of time during which the requester will wait for responses, and this time is proportional to the diameter of the request. If a requester does not receive responses within this time, it will resend the discovery message. $\Delta_{trequester}$ is given by,

Table I. SUMMARY OF NOTATION

Symbol	Definition
$vmax$	maximum node speed
Δ_{tmax}	maximum response time for one request
R_i	search diameter for requester node
$n_{provider}$	reply counter
$max_{provider}$	maximum number of responses specified by the requester
$\Delta_{trequester}$	client node timer
$\Delta_{tintermediate}$	intermediate node timer
β	maximum time for the closest intermediate node to store responses before forwarding them

$$\Delta_{trequester} \leftarrow K \times (R_i/R_t) \tag{2}$$

The value of K is defined as $\alpha + \beta$. The value of α specifies the forward and backward delay for one hop on the network. The value of β specifies the maximum amount of time that the intermediate node closest to the requester can store one response before forwarding it. The range of the antenna is given by R_t, and R_i/R_t represents an estimate of the hops number. The closest intermediate nodes maintain the replies stored for longer time in compared with more distant nodes.

In the discovery message, the requester node sends the following information: a node identification, its coordinates, $coord_X$, $coord_Y$, the maximum response, Δ_{tmax}, the service, s, and the number of desired providers. If node j receives a request from node i, the algorithm verifies the distance (d_{ij}) between both nodes. If $d_{ij} > R_i$, the request is discarded by j because this node is out of the search area. Conversely, if $d_{ij} \leq R_i$, the algorithm verifies the speed of the service provider j. Moreover, the algorithm verifies if this provider offers the searched resource (s) and if the service provider j is available at the moment. If the restriction given by Eq. (3) is satisfied, node j sends a response to node i.

$$d_{ij}/v_j \leq \Delta_{tmax} \tag{3}$$

Assuming that the provider node j has the resource, but v_j is insufficient, j does not send a reply to i, and it only resends the request message. It is assumed that the nodes have maximum speeds defined. The nodes attend to the requests on demand. Each node maintains information about the resource that is offered by it.

B. Location Aware Service Selection (LASS)

As a result of the discovery process, multiple providers can respond to a service request. The LASS mechanism takes into account such aspects as the requester node's geographic location, the maximum response time to attend one request, the speed that the service provider moves, and the number of service providers desired to select and discard answers. The aim is to reduce the number of reply transmissions from the network.

LASS works as follows. Suppose that node k (intermediate node) receives the reply message from one of its neighbors, for instance, node m. Then, node k starts a timer, named $\Delta_{tintermediate}$, with the function of storing replies. $\Delta_{tintermediate}$ is given by Eq. (4).

$$\Delta_{tintermediate} \leftarrow \gamma \times (1/d_{ik}) \leq \beta \tag{4}$$

where γ specifies a proportionality factor with the function to maintain the timer $\Delta_{tintermediate}$ less than β. $1/d_{ik}$ denotes the inverse distance between the requester and the provider node. Since the storage period is inversely proportional to the distance between the intermediate node and the requester, the closest nodes store replies for a longer time. For instance, in Fig. 1, $\Delta_{tintermediate}$ of node *I2* is greater than $\Delta_{tintermediate}$ of node *I1*. This occurs due to the fact that *I2* is closest to the requester node. The LASS data fusion scheme is shown below.

C. LASS Data Fusion Scheme

When an intermediate node receives one response, the intermediate node starts $\Delta_{tintermediate}$. If the maximum number of responses ($max_{provider}$) that meet the requests profile is reached before $\Delta_{tintermediate}$ expires, the intermediate node aggregates these responses and sends only one response to the requester node. After this step, $\Delta_{tintermediate}$ is canceled by the intermediate node. Others responses received are discarded.

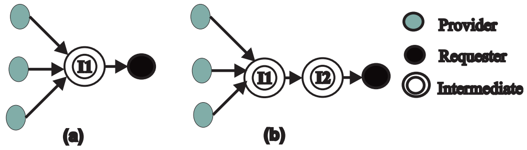

Figure 1: Lass data fusion scheme without aggregation (Figure. 1a) and with aggregation (Figure. 1.b)

In the data fusion scheme, an intermediate node collects responses from different providers, and concatenates two individual parameters in one string, which are $Id_{provider}$ and $ServiceTime_{provider}$. The other parameters(id_{Client}, $id_{Requester}$, $id_{SeqRequester}$, $max_{provider}$) are common for all replies. In the aggregation phase, all parameters are aggregated in one response and transmitted to the requester node. In the response message (*msgReply*()), a provider sends its geographic coordinates, the client identification, the resource identification, the request identification, the sequence number identification, the maximum number of replies, $Id_{provider}$ and the $ServiceTime_{provider}$.

An intermediate node can receive a response directly from a provider or another intermediate node. If the intermediate node receives a response directly from a provider, the response was not aggregated. In this case, while the maximum number of responses is not reached, the intermediate node stores responses. If the intermediate node receives a response from another intermediate node, the LASS algorithm works as summarized in Fig. 1.

In Fig. 1, let us suppose the intermediate node *I2* receives one response from the intermediate node *I1*. Firstly, *I2* will verify the number of aggregated responses in this message. After this step, *I2* verifies the number of responses stored by it. If $max_{provider}$ was not reached, *I2* stores the responses.

The number of stored responses plus the number of aggregated responses may not exceed $max_{provider}$. In order to solve this problem, the LASS algorithm updates the stored responses on the condition that the aggregated responses have better quality than the stored responses.

The LASS mechanism is presented in Algorithm 1. It is noteworthy that the repeated answers (already treated by a particular provider) are discarded.

Algorithm 1. Service Selection Algorithm

```
PROCWAITMAXPROVIDER()
node k verifies the number of aggregated replies ($number_{agregate}$) in msgReply();
node k verifies the number of replies already received by it ($number_{provider}$);
$number_{totalresponses} \leftarrow number_{agregate} + number_{provider}$;
if $number_{totalresponses} < max_{provider}$
    $number_{provider} \leftarrow number_{totalresponses}$;
    node k stores the news responses;
else if $number_{totalresponses} = max_{provider}$
    node k aggregates received replies;
    node k sends msgReply();
    node k cancels $\Delta_{intermediate}$;
else
    node k discards msgReply();

PROCENDTIMERWAITMAXPROVIDER()
if $\Delta_{intermediate}$ expires
    node k aggregates received replies;
    node k sends msgReply();
```

D. Service Invocation Mechanism

The service invocation mechanism operates after the selection service phase and establishes rules for access and use of the providers already selected. In the service invocation phase, requesters and providers verify the viability of the attendance and the providers physically move to the place where the service is required.

As we can see in Fig. 2a, after the selection service phase represented by (2), the requester node sends a service invocation message (3) to the selected provider (Fig. 2b). Then, the selected provider sends a service confirmation message (4) in which the provider informs that it is available go to the place where the service is required.

In the service confirmation message, the provider sends the following information: request identification ($id_{Request}$), provider identification ($id_{Provider}$), and requester node identification (id_{Client}). When a provider informs that it is available go to the place where the service is required it begins to move toward the requester. However, the confirmation message may be lost. In order to attend this fault, we considered that if the provider receives a discovery message of the same requester, the provider restarts the process and sends a reply message (*msgReply*) to the requester. Also, in this case, the provider can respond to other requesters. A provider is only allocated to a requester after sending the confirmation message.

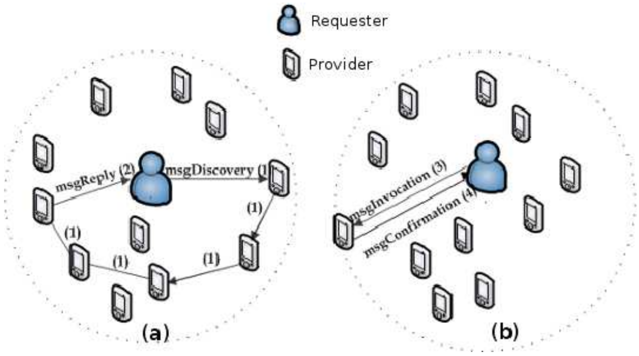

Figure 2: Transition diagram of the messages in the discovery process

IV. SIMULATION RESULTS

In order to evaluate the proposed approach, we used the Network Simulator (NS-2) [15] and the scenario presented in Section II was mapped for different area sizes. Specifically, 4km^2 (2000m x 2000m) and 25km^2 (5000m x 5000m). The number of nodes in the network varies between 100, 200, 300 and 400, and the node speed varies between 0.5 m/s and 15.0 m/s.

Experiments were carried out where resources were attributed to 10%, 15%, 20%, 25% and 30% of the nodes in the network. Each provider offers one type of resource. A random distribution of resources was conducted. In the simulation we used the Optimized Link State Routing (OLSR) protocol, [16] to send the service discovery messages to the neighbor nodes.

All scenarios started with a 3600 second initialization phase. The minimum number of providers that must be delivered could be 1 and 2, and the maximum response time was set at 10.0 min. All nodes in the network are mobile. However, the requester node remains static while waiting for the reply.

We conducted experiments with four mechanisms: LADS (discovery without selection), LASS+Fusion (discovery with selection and aggregation), Flooding-1 (traditional flooding), and Flooding-2. In Flooding-2, we extracted from the responses obtained with Flooding-1, the responses of apt nodes, that is, the providers that can reach in time where the resource is needed. The confidence interval presented in the results is 95%.

Each node is equipped with the default wireless network module that NS-2 provides, which has a 250m transmission range. We adjusted the transmission power at 42mW and the receiving power at 55mW. The initial energy of all the nodes is set at 1000J. Each node makes requests during the 28800s duration of the simulation.

A. *Results*

Fig. 3 shows the service invocation success rate (SI) according to the node's mobility. The service invocation success determines whether or not a client discovers a service after the service invocation phase. SI is defined as number of responses received after the service invocation phase divided by number of requests sent. We measured the service invocation success rate for mechanisms: LADS, LASS+Fusion, Flooding-1, and Flooding-2. In this experiment, 10% of the participating nodes have the resource and the number of providers that must be delivered is 1.

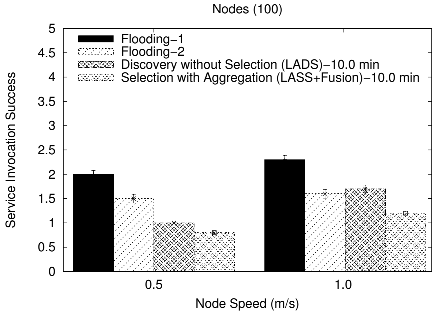

Figure 3: Service invocation success × node speed

The results show that LASS+Fusion performs well for all scenarios. However, we observed that the SI rate obtained with LASS+Fusion when the node speed is 0.5 m/s was less than 1. This result is due to loss of invocation or confirmation messages.

Energy consumption (EC) experiments were performed. In terms of energy consumption, we observed that the LASS+Fusion mechanism outperforms the other evaluated mechanisms without compromising the discovery process. EC is defined as total energy consumption divided by initial energy of the nodes.

In Fig. 4, we analyzed the node energy consumption (EC). For this experiment, we compared the LASS+Fusion mechanism with the LADS mechanism and Flooding-1. The number of nodes in the network is 100. The resource percentage is 10%. It is noteworthy that the Flooding-2 mechanism was not mentioned in this simulation because in Flooding-2, we extracted from the responses obtained with Flooding-1, the responses of apt nodes, that is, the providers that can reach in time where the resource is needed.

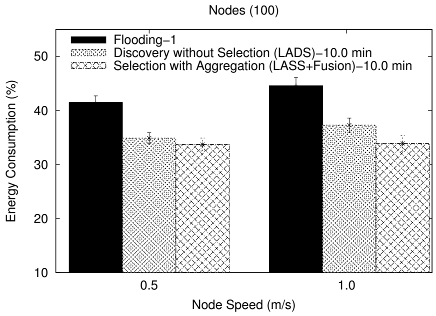

Figure 4: Energy consumption × node speed

From these results, we observed that the LASS+Fusion mechanism saves considerable energy by discarding responses with a longer time limit. For instance, in a scenario where the node speed is 0.5 m/s, the energy consumption was 41.5% with Flooding-1, 34.9% with LADS and 33.7% with LASS+Fusion.

Through this experiment, we concluded that the use of data aggregation for response suppression in service discovery protocols can reduce the energy consumption of network nodes and increase the network lifetime.

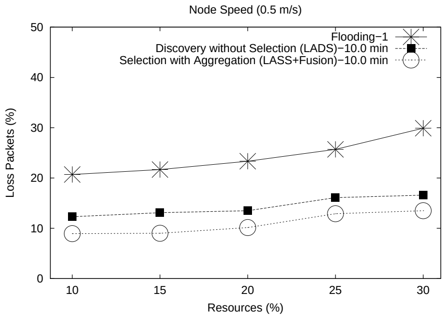

Figure 5: Lost packets × resource percentage

Fig. 5 depicts the loss packets rate (LK) after the service invocation phase. This metric is given by number of lost packets divided by total number of messages generated by LADP. As we can see, LASS+Fusion presented low packet loss in relation to Floding-1. For instance, when 30% of the participating nodes have the resource, the loss rate was 30.90% for Flooding-1, and 13.5% for LASS+Fusion. This result shows that to aggregate replies can also contribute to decrease collisions and avoid packet loss.

In Table II, we present the percentage of success (PE) after the service invocation phase. PE is based on the replies obtained from the Exact Approach.

We implemented the Exact Approach to verify the mobility scenarios, and to show the best provider(s) in each time instant for the requester node. For this experiment, we calculated the PE among the responses obtained with LADS, LASS+Fusion and Flooding-2, with the responses presented by the Exact Approach.

In this experiment, 10% and 30% of the participating nodes have the resource. The minimum number of providers that must be delivered is 1 and the node speed is 0.5 m/s.

TABLE II. PERCENTAGE OF SUCCESS(%) × NODE SPEED

Resource (%)	Flooding-2	LADS	LASS+Fusion
10%	75.0	70.0	70.0
30%	69.0	75.5	80.52

The results show that the PE with Flooding-2 was 69%, and 80.5% for LASS+Fusion when 30% of the participating nodes have the resource. The PE of the Flooding-2 was lower for scenarios where nodes have more resources. This occurred due to collisions that increased dropped response messages. This exemplifies that reply aggregation can greatly benefit the percentage of success by reducing the number of reply messages and saving energy.

Table III shows the energy consumption related to the number of nodes. Checking Table III, the economy of energy was greater with LASS+Fusion. In a scenario with 400 nodes, the energy consumption of Flooding-1 was 86.92% and with LASS+Fusion was 53.98%.

We conclude that the LASS+Fusion mechanism outperforms the other mechanisms with respect to node energy consumption through the discards of additional responses without compromising the discovery process. In this experiment, 30% of the participating nodes have the resource and the node speed is 0.5 m/s.

TABLE III. ENERGY CONSUMPTION(%) × NODE

Node	Flooding-1	LADS	LASS+Fusion
300	86.87	69.97	53.84
400	86.92	70.10	53.98

Also, were performed experiments with larger areas in order to evaluate the scalability of the protocol. In the experiments presented below, the scenario presented in Section II was mapped for an area of 25 km^2. The number of nodes in the network varies between 50, 100, 150, 200 and 250. The node speed varies from 0.5 m/s to 15.0 m/s and the maximum time limit to service the request is 1.5 min and 15 min.

Fig. 6 shows the service invocation success rate (SI) according to the node's mobility. We measured the service invocation success rate for mechanisms: LADS, LASS+Fusion, Flooding-1, and Flooding-2. In this experiment, 10% of the participating nodes have the resource, the number of providers that must be delivered is 1, the network has 200 nodes and the maximum time limit to service the request is 1.5 min.

The LASS+Fusion mechanism has enabled a greater response suppression without causing a negative impact on the discovery process, i.e., it has reached the goal of 1 (one) response at all speeds, except for the scenario where nodes move at 2.0 m/s. In this case, the service invocation success rate was 0.79. This behavior is due to the low number of providers within the radius. Besides that, there was not a substantial change in the topology in order for additional providers to be found.

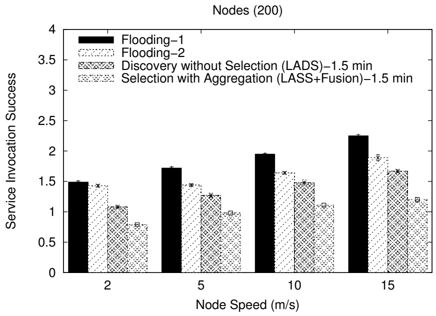

Figure 6: Service invocation success (1.5 min) × node speed

In Fig. 7, we measured the service invocation success rate (SI) according to the node's mobility for the same scenario shown in Fig. 6. However, unlike previously presented scenario, the maximum time limit to attend the request is increased to 15 min and the node speed varies between 0.5 m/s, 1.0 m/s and 1.5 m/s. Therefore, in this scenario, the radius R_i remains the same as in Fig. 6.

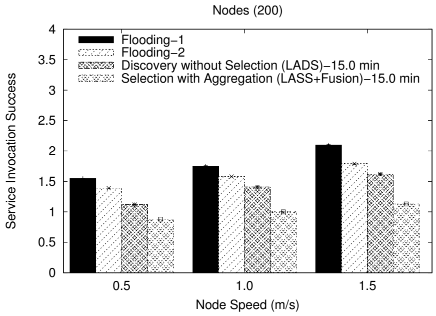

Figure 7: Service invocation success (15 min) × node speed

The results obtained were similar to those presented in Fig. 6. These results indicate that the performance is mainly a function of the radius R_i and that a longer attending time limit can be reached if nodes move at proportionally lower speeds.

V. CONCLUSION AND FUTURE WORK

This paper presented the Location Aware Service Discovery Protocol for MANETs and its mechanisms for service discovery, LADS, and a mechanism for service selection, LASS. The service selection mechanism presented a data aggregation scheme in intermediate nodes.

The results showed that the LASS+Fusion mechanism saves considerable energy through the additional response discards without compromising the discovery process. The proposed mechanisms help to maintain the battery of the mobile devices longer.

Another contribution is a service invocation mechanism. This phase is generally not considered in service discovery protocols, however, it is crucial to ensure the attendance success.

We observed that the use of the fault-tolerant service discovery mechanism by means of the $\Delta_{trequester}$ timer contributed to keep the service invocation rate within the expected average despite of the existence of faults.

The results of simulations on the Location Aware Service Discovery Protocol showed that the LASS+Fusion mechanism can provide a good scalability and searching performance in different scenarios.

As future works we would like to extend our store and forward strategy (e.g., understand the trade-off between network scalability and the period of time to keep the message stored in a node).

REFERENCES

[1] R. Balaji, H. Jarmo, and S. Bilhanan, “Service discovery framework for manets using cross-layered design,” in Proc. 6th international conference on wired/wireless internet communications (WWIC’08), Tampere, Finland, 2008, pp. 152–163.

[2] Q. Li and M. Zhou, “The survey and future evolution of green computing,” in Proceedings of the 2011 IEEE/ACM International Conference on Green Computing and Communications (GREENCOM ’11). Washington, DC, USA: IEEE Computer Society, 2011, pp. 230–233.

[3] A. Bianzino, C. Chaudet, D. Rossi, and J. Rougier, “A survey of green networking research,” Communications Surveys Tutorials, IEEE, vol. 14, no. 1, pp. 3–20, 2012.

[4] A. N. Mian, R. Baldoni, and R. Beraldi, “A survey of service discovery protocols in multihop mobile ad hoc networks,” IEEE Pervasive Computing, vol. 8, no. 1, pp. 66–74, 2009.

[5] R. S. Marin-Perianu, “Wireless sensor networks in motion: clustering algorithms for service discovery and provisioning,” Ph.D. dissertation, University of Twente, Enschede, 2008. [Online]. Available: http://doc.utwente.nl/60028, [retrieved: 08, 2012].

[6] K. Arnold, The Jini specification, ser. Jini technology series. Addison Wesley, 1999.

[7] E. Guttman, “Service location protocol: Automatic discovery of ip network services,” IEEE Internet Computing, vol. 3, no. 4, pp. 71–80, 1999.

[8] M. Nidd, “Service discovery in deapspace,” Personal Communications, IEEE, vol. 8, no. 4, pp. 39–45, 2001.

[9] B. Miller and C. Bisdikian, Bluetooth revealed. Prentice Hall PTR, 2001.

[10] S. Helal, N. Desai, V. Verma, and C. Lee, “Konark - a service discovery and delivery protocol for ad-hoc networks,” in IEEE Wireless Communications and Networking Conference (WCNC’03), 2003, pp. 2107–2113.

[11] V. Lenders, M. May, and B. Plattner, “Service discovery in mobile ad hoc networks: A field theoretic approach,” Pervasive and Mobile Computing, vol. 1, no. 3, pp. 343–370, 2005.

[12] A. T. A. Gomes, A. Ziviani, L. dos S. Lima, M. Endler, and G. Chelius, “Mitigating reply implosions in query-based service discovery protocols for mobile wireless ad hoc networks,” in The 11th International Conference on Ad Hoc Networks and Wireless (ADHOC-NOW), vol. 5198. Springer, 2008, pp. 29–42.

[13] A. Varshavsky, B. Reid, and E. de Lara, “A cross-layer approach to service discovery and selection in manets,” in The 2nd IEEE International Conference on Mobile Adhoc and Sensor Systems Conference, 2005, pp. 459–466.

[14] J. Kniess, O. Loques, and C. V. N. Albuquerque, “Location aware discovery service and selection protocol in cooperative mobile wireless ad hoc networks,” in Proceedings of the 28th IEEE international conference on Computer Communications Workshops (INFOCOM’09). Piscataway, NJ, USA: IEEE Press, 2009, pp. 367–368.

[15] T. Issariyakul and E. Hossain, An Introduction to Network Simulator NS2. Springer, 2009. [Online]. Available: http://books.google.com.br/books?id=cD69He_oU60C, [retrieved: 12, 2012].

[16] T. Clausen and P. Jacquet, “Rfc3626: Optimized link state routing protocol (olsr),” RFC Editor United States, 2003.