Routing Layer Support for 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, 10434 Athens, Greece
Email: {chris, polyzos} @aueb.gr

Abstract

Service discovery in Mobile Ad Hoc Networks is an essential process in order for these networks to be self-configurable. In this paper we argue that Service Discovery can be greatly enhanced in terms of efficiency (regarding service discoverability and energy consumption), by piggybacking service information in routing layer messages. Thus, a node requesting a service in addition to discovering that service, it is simultaneously informed of the route to the service provider .We extend the Zone Routing Protocol in order to encapsulate service information in its routing messages and through extensive simulations we prove the superiority of our routing layer-enhanced service discovery scheme against an application layer-based flooding scheme.

I. Introduction

In the past, much research effort has been devoted on Service Discovery in static networks, like the Internet. The emergence of wireless communications and small 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 inherent characteristics of these networks.

MANETs are extremely dynamic due to the mobility of their comprising nodes, the wireless channel's adverse conditions and the energy limitations of small 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 battery consumption overheads are significantly reduced. Our approach is to implement service discovery at the routing layer by piggybacking the service information into the routing messages, thus enabling the devices to acquire both service and routing information simultaneously.

We propose the piggybacking of service information in routing messages, in order to decrease communication overhead and save battery power. 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. In order to demonstrate the benefits of our approach (i.e. routing layer supported service discovery) versus traditional application based service discovery, we modify the Zone Routing Protocol (ZRP), which 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 order to encapsulate service information in its messages.

The rest of this paper is organized as follows. In section II we provide the essential background on service discovery by presenting the most significant research efforts. In section III we present our approach of routing layer enhanced service discovery, and in section IV we provide simulation results along with their analysis. Finally in section V we conclude and refer to our future research directions.

II. Related Work

Significant academic and industrial research has led to the development of a variety of protocols, platforms and architectures for service discovery like JINI [1], Salutation [2], UPnP [3], UDDI [4], SDP [5] and SLP

* This research is supported by project "Mobile Services" (EP-1221-06), funded by the research program "Pythagoras–Support for Research Groups at the AUEB," which is co-financed by the Ministry of National Education and Religious Affairs of Greece and the European Union, through the program "EPEAEK II."

[6]. All these approaches, except SDP, are mainly targeted in 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 their services. Service requestors submit their queries to these 'special nodes' in order to discover services and information about the nodes that actually host them.

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 [7], GSD [8], DEAPspace [9], Konark [10] and SANDMAN [11]. These approaches were developed with more pervasive environments in mind, and are briefly presented in the next paragraphs. From our point of view, the energy consumption of the discovery approach is of great importance.

Allia is an agent based service discovery protocol, based 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 (does not have a matching 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. It does not address energy consumption, and no such measurements 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 broadcasted but instead forwarded only to those nodes that have already cached information about similar services. However, GSD uses DAML-based service descriptions in its advertisement messages (instead of simple UUIDs) and performs semantic matching, thus increasing battery 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, whose levels 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 poorer), hence allowing semantic matching, leading to increased battery 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 their availability 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 does not find its own services in these messages, it schedules a broadcast sooner than usual, in order to inform 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 periodic broadcasts.

Finally, SANDMAN – like DEAPSpace - is another service discovery protocol that implements power saving. 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. Simulation results show battery 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 two latter approaches take into account battery consumption and provide relative measurements. What differentiates our approach from them is that we do not allow the nodes to go into sleep mode, as we target environments where continuous connectivity is mandatory.

III. Routing Layer Supported Service Discovery

Our motivation for adding routing layer support 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. Hence, two message-producing processes 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 packets. Our approach aims at exploiting the capability of acquiring service information along with routing information by piggybacking service information into routing messages. This way, redundant transmissions of service discovery packets at the application layer are avoided and battery power is saved. The idea of providing routing layer support for service discovery was first introduced by Koodli and Perkins in [12]. However no experimental assessment of Koodli's and Perkins' proposal has been published until now.

As stated in the introduction we have modified the Zone Routing Protocol [13] so that it provides service discovery functionality. To the best of our knowledge this is the first research effort using a hybrid routing protocol for supporting service discovery. In this paper we present experimental results on extensions done on the proactive part of ZRP, while we extensively report on our current and future work on extending its reactive part as well. Next, we describe the basic operation of ZRP and the modifications we have implemented on it, in order to enhance it with service discovery capabilities.

ZRP

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 (e.g. records for nodes located up to 2-hops away). This is depicted in Figure 1 where nodes B to H are inside the routing zone of node A; hence node A knows proactively 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).

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 Figure 2. The border nodes check their IARP tables to find if the requested node belongs to 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, flooding is avoided and distant resources are discovered in an efficient and scalable way.

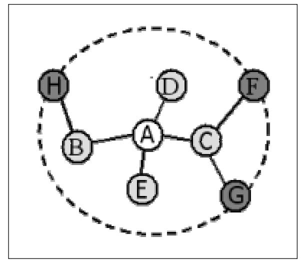

Fig.1.ZRP two-hop zone

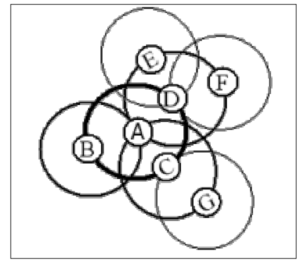

Fig.2.ZRP bordercasting

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 Figure 2. The border nodes check their IARP tables to find if the requested node belongs to 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, flooding is avoided and distant resources are discovered in an efficient and scalable way.

M-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 in order to keep small packet lengths for routing messages, so as not to disrupt the routing process (the bigger the messages the bigger the delays and the possibility of 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 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 service descriptions is of no particular use in these cases, not to mention that these techniques lead to increased battery 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 modified in order to include service information in every routing entry of the IARP routing messages and tables. IARP listens to information gathered from NDP messages, updates its table and then periodically broadcasts its table to its neighbors. 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. This modified version of ZRP, (henceforth called M-ZRP) is capable of providing routing layer support for proactive service discovery. In the following section we present our simulation results from applying M-ZRP in multiple scenarios.

IV. Simulation Results And Analysis

Our simulations were conducted in the QUALNET Simulator [14], which has a ZRP module. For simplicity reasons and in order to facilitate the analysis of the results we assume that each node hosts a unique service, offered to other nodes.

Two sets of experiments were conducted in a purely proactive environment. The first set of experiments demonstrates the savings in battery consumption attained under our approach. The number of hosts does not affect battery consumption as the messages are propagated within the routing zone and not further away. In order to facilitate the analysis of the results, a chain topology where nodes are connected in a row (each one having one neighbor to the left and one to the right) was selected. Moreover, it is assumed that there is no mobility. The settings for M-ZRP, as well as for flooding in these experiments are summarized in Table 1.

Table 1. Simulation Settings

Simulation Duration	1000 sec
Zone Radius	3 hops
Broadcast interval	10 sec
Service deletion interval	40 sec
Mobility	None

As it is depicted in this table, the parameter settings for both approaches are the same, making the comparison fair. Simulation duration is set to 1000 seconds. The IARP Zone Radius is equal to the flooding radius; this implies that restricted area 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 M-ZRP) to neighboring nodes. The same interval is also used in Flooding, with the difference that flooding messages are much shorter containing only the node's own service UUID and no routing information or other nodes' service UUIDs. The Service deletion interval, defines the time after which a node erases records that have not been updated.

Figures 3 and 4 show that by using M-ZRP's proactive routing algorithm (in IARP) we can achieve up to 50% battery savings (compared to the Flooding scheme) for service discovery, irrespectively of the number of static nodes in a 'chain topology'.

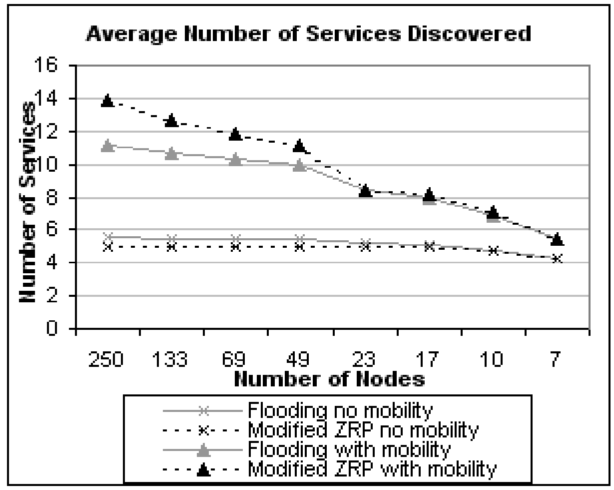

Fig. 3. Average Number of Services Discovered for proactive routing in both mobile and static scenarios (M-ZRP versus Flooding).

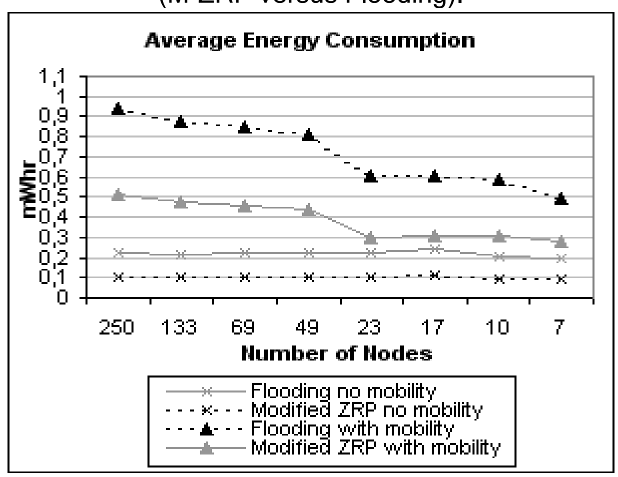

Fig. 4. Average Energy Consumption for proactive routing in both mobile and static scenarios (M-ZRP versus Flooding).

It is worth noting that in Flooding we also assume that ZRP is used at the routing layer. 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 at the routing layer for route discovery. This application layer overhead in messages, leads to the observed dramatic difference of battery consumption between the two schemes. At the same time we observe (and that is the most interesting) that service discoverability is almost at the same level as it is for Flooding. On the average, we get only 7% fewer services discovered when using M-ZRP, which is negligible compared to the achieved battery savings of up to 50%.

The second set of experiments demonstrates the battery consumption benefits of our approach in a purely proactive environment, where nodes are mobile and their topology is random. Every node in the simulated scenarios uses the random waypoint model with the following parameters: Node Max Speed 10m/sec, Node Min Speed 0m/sec and Pause Time 30sec. Except from mobility, the other parameters of the simulation are those presented in Table 1. For stability reasons the density is kept fixed when varying the number of nodes by resizing the terrain in which they are allowed to move.

In such an environment it is expected that due to the nodes' mobility, each node will occasionally meet several nodes and hence discover more services as compared to the static scenarios where the neighbors in a node's zone remain fixed until the end of the simulation. This is evident in Figure 3 where it is shown that mobile nodes discover about 50% more services on the average than those in the static scenarios. However, the node density and the speed of the nodes affect this result. We intuitively expect that increased densities and speeds will result in increased number of discovered services. In these mobile scenarios, more services are discovered on the average (up to 30% for the case of 250 nodes) by each node when using M-ZRP instead of Flooding. Moreover, increasing the number of the participating nodes results in faster increase of service discoverability in M-ZRP as compared to Flooding. This is justified by the fact that IARP messages in M-ZRP contain service information about all the neighbors of a node, while Flooding messages contain service information about only that single node. Hence an IARP message is much richer in terms of the information it contains than a Flooding message. The fact that a node has generally more services in its neighborhood than in the static scenarios, also explains the increase in battery consumption for both M-ZRP and Flooding (more neighbors means more messages). However, again M-ZRP outperforms Flooding, giving up to 48% battery savings on the average.

V. Conclusions and Future Work

Most previous research efforts on service discovery do not report on battery consumption. Also, existing application layer service discovery architectures suffer from redundant packet transmissions in order to discover routes towards the services. We presented a new cross-layer architecture that integrates service discovery functionality with an existing routing protocol. Our approach combines route discovery with service discovery, thus allowing nodes to find available services and routes to them simultaneously. This way fewer messages are broadcasted into the network; hence each node saves great amounts of energy. We have experimentally showed that our cross-layer implementation consistently outperforms an application-layer service discovery scheme based on restricted-area flooding in terms of battery consumption. Our proposed protocol (M-ZRP) provides the least amount of battery consumption (up to 50%), but also in certain cases it achieves higher service discoverability (up to 30%).

Our current work focuses on enhancing our approach of providing routing layer support for service discovery by extending the reactive part of ZRP. In our future work we plan to conduct extensive simulations in order to test the reactive service discovery capabilities of M-ZRP by employing the appropriate simulation scenarios. We also plan to further investigate the impact of mobility on cross layer service discovery protocols.

References

[1] Sun Microsystems, JINI Architecture Specification, 1999.
[2] Salutation Consortium, "Salutation Architecture Specification, www.salutation.org/specordr.htm," 1999.
[3] Microsoft Corporation, "Universal Plug and Play: Background," http://www.upnp.org/, 1999.
[4] Universal Description Discovery and Integration Platform. http://www.uddi.org, September 2000.
[5] "Specification of the Bluetooth System," http://www.bluetooth.com, Dec. 1999.
[6] E. Guttman, C. Perkins, J. Veizades, and M. Day. "Service Location Protocol, Version 2," IETF RFC 2608, June 1999.
[7] 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," in ACM Mobile Commerce Workshop, 2002.
[8] D. Chakraborty and A. Joshi, "GSD: A novel group-based service discovery protocol for MANETS," In IEEE Conference on Mobile and Wireless Communications Networks, Stockholm, Sweden, September 2002.
[9] M. Nidd, "Service Discovery in DEAPspace," IEEE Personal Communications, August 2001, pp. 39 - 45.
[10] S. Helal, N. Desai, V. Verma, and C. Lee, "Konark – A Service Discovery and Delivery Protocol for Ad-Hoc Networks," in *Proceedings of the 3rd IEEE Conference on Wireless Communication Networks (WCNC)*, New Orleans, Louisiana, March 2003.
[11] G. Schiele, C. Becker and K. Rothermel, "Energy-Efficient Cluster-based Service Discovery for Ubiquitous Computing," in Proceedings of the 11th ACM SIGOPS European Workshop, Leuven, Belgium, September 2004.
[12] 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.
[13] 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.
[14] Scalable Network Technologies Inc., "Qualnet Simulator," http://www.scalable-networks.com