Impact of Node Mobility and Network Density on Service Availability in MANETs*

Christopher N. Ververidis and George C. Polyzos

Mobile Multimedia Laboratory, Department of Computer Science
Athens University of Economics and Business, Athens 10434, Greece
{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 with zero or minimal administrative overhead. Service discovery can be made more efficient, by piggybacking service information into routing messages. We extended the Zone Routing Protocol in order to encapsulate service information in its routing messages. The extended protocol, E-ZRP, may be seen as a representative of routing layer protocols providing service discovery functionality. We provide a sensitivity analysis for service availability using E-ZRP over various network conditions, measuring the effects of node speed and node density on the duration of discovered services.*

I. Introduction

Much research has been devoted to Service Discovery in fixed networks, applied mostly to the Internet. The emergence of wireless communications and mobile computing devices has created the need for developing service discovery protocols and architectures targeted to mobile environments. Especially, the proliferation of Mobile Ad-Hoc Networks (MANETs) has introduced new requirements for service discovery due to the nature and inherent characteristics of these networks.

MANETs are extremely dynamic due to the mobility of their nodes, the wireless channel's adverse conditions and the energy limitations of small, mobile devices. The great majority of service discovery protocols developed for MANETs deal with the above issues at the application layer. In [12] Koodli and Perkins introduced the idea of extending on demand routing protocols to provide service discovery support. Application layer service discovery protocols implementations keep the abstraction layers of the networking stack intact and thus can be implemented above any routing protocol. On the contrary, cross layer service discovery protocols, frequently impose modifications and/or extensions to the underlying routing protocol in order to provide their functionality, and hence are protocol dependent and protocol specific. However, service discovery can be significantly improved in terms of reducing communication and battery consumption overhead, by exploiting routing layer information. The benefits obtained by such an approach outweigh the disadvantage of breaking the abstraction layers.

In our previous work [15] we proved that by exploiting service discovery information provided by the routing layer, the resulting communication and battery consumption overheads are significantly reduced. Our approach was to implement service discovery in the routing layer by piggybacking the service information into the routing protocol control messages, thus enabling the devices to acquire both service and routing information simultaneously. This way a node requesting a service in addition to discovering the service, it is also informed of the route to the service provider at the same time. Smooth service discovery adaptation to severe network conditions is now possible since service availability is tightly coupled with route availability to serving nodes. Hence when all routes towards a node fail, this is immediately translated to a loss of service availability for the services that this node provides. We extended 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), so that it is capable of encapsulating service information in its messages.

However, a key issue for service discovery protocols for MANETs, besides energy consumption, is the quality of the services discovered. With the term quality we refer to the usability characteristics of a service and not its inherent characteristics (e.g. precision of the provided information). The study of the inherent characteristics of discovered services is beyond the scope of this paper. So, in order to measure the quality of discovered services we define a new metric called SAD (Service Availability Duration), which measures the availability of a discovered service. SAD is defined as the length of time that elapses from the moment the service is discovered until that time when the service is lost, as a result of mobility or interferences. It should be noted that if the path to the original service provider is lost, but there exists another provider for the same service-type in the node's routing table, then the service is still considered 'alive'. Only when all the routes from a node to all the available providers of the service are lost, this particular service is considered not to be available any more to that node. In the literature [16][17], a similar metric, called Path Duration has been widely used to measure the impact of mobility on routing protocols for MANETs. However these studies mainly focus on reactive routing protocols and do not consider service discovery. They also focus on node availability and not service availability, which is a totally different concept. In general a good discovery protocol should be able to adapt to different network

---

* 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."

conditions in order to effectively discover as many long-lived services as possible. The purpose of this paper is to identify the effects of network density and mobility on the ability to discover such services. We have already proved the superiority of routing layer based service discovery protocols in previous work, and in this paper we identify the conditions under which these protocols perform better in terms of SAD.

The remainder of this paper is organized as follows. In section II we briefly present our approach of routing layer supported service discovery, in section III we provide simulation results on the impact of mobility and density on SAD, and finally in section IV we provide our conclusions and discuss our future research directions.

Significant academic and industrial research has led to the development of a variety of protocols, platforms and architectures for service discovery such as JINI [1], Salutation [2], UPnP [3], UDDI [4], Bluetooths' SDP [5] and SLP [6]. All these approaches, except SDP, are mainly targeted towards the discovery of services in fixed infrastructure networks. They are mostly centralized approaches that assume that reliable communication can be provided by the underlying network. Most of these approaches utilize nodes acting as (central) service directories-repositories, where service providers register the services they offer. Service requestors submit their queries to these 'special nodes' in order to discover services and information about the nodes that actually host these services. It is clear that such assumptions are not consistent with MANETs' inherent features due to their volatile nature.

This has motivated some recent approaches in the field, namely Allia [7], GSD [8], Konark [9], DEAPSpace [10] and SANDMAN [11]. These approaches were developed for pervasive computing environments. However, only the two latter approaches take into account battery consumption and provide related metrics and comparisons, and are briefly presented in the following paragraphs.

DEAPspace employs a periodic broadcast scheme for service advertisements. Each node broadcasts the full list of services that it is aware of in its one-hop vicinity. 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 broadcasts. However, it is targeted for small networks only.

SANDMAN, like DEAPSpace, is another service discovery protocol that implements power savings. This is done by grouping nodes with similar mobility patterns into clusters; in each cluster, one of the nodes (called clusterhead) stays awake permanently and answers discovery requests. The rest of the nodes periodically wake up to provide the actual services and also inform the clusterhead about their presence and services. Simulation results show battery savings of 40% for low numbers of service requests.

A key difference of our approach from those is that we do not expect or allow the nodes to go into sleep mode, since we target environments where continuous communication is necessary. Furthermore, none of the above approaches comments on the quality of the discovered services. In our work, we investigate the performance of our protocol in terms of SAD, under various network conditions. In the next section we present our approach and justify our design decisions.

II. 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 for its own use. Hence, two message-producing processes must coexist: the first one communicates service information among service providers and service requestors; the second one communicates routing information among them. Our approach exploits the capability of acquiring service information along with routing information (from the same message) by piggybacking service information into routing messages. This way, redundant transmissions of service discovery packets at the application layer are avoided and power is saved.

The idea of providing routing layer support for service discovery was first introduced by Koodli and Perkins in [12]. However, as far as we know, no experimental assessment of Koodli's and Perkins' proposal has been published until now.

As stated in the introduction we have extended the Zone Routing Protocol (ZRP) [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. ZRP was selected because: (a) it is ideal for environments where local information-either routing or service information-is of particular interest, as it provides discovery (through the notion of zones described further on) in a fast and energy efficient way and (b) it is scalable, as it intelligently propagates information to distant nodes by avoiding flooding.

ZRP

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

- The Neighbor Discovery Protocol (NDP), through which every node periodically broadcasts a "hello" message to denote its presence.
- The Intra Zone Routing Protocol (IARP), which is responsible for proactively maintaining route records for nodes located inside a node's routing zone (for example records for nodes located up to 2-hops away). This is depicted in fig.1 where nodes B to H are inside the routing zone of node A; hence node A is proactively aware of all the routes to these nodes through IARP.
- The Inter Zone Routing Protocol (IERP), which is responsible for reactively creating route records for nodes located outside a node's routing zone (e.g. records for nodes located further than 2-hops away).

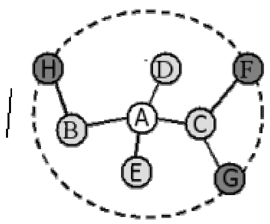

Fig.1: ZRP two-hop zone

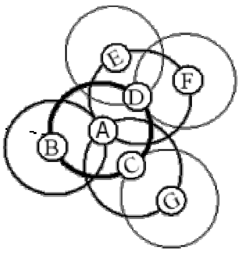

Fig.2: ZRP bordercasting

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 is included in their respective routing zones; if not they also bordercast the request to their own border-nodes. When the requested node is found, a reply is unicasted back to the node that initiated the request. This way, global flooding is avoided and distant resources are discovered in an efficient and scalable manner.

E-ZRP

In order to add service discovery capabilities to ZRP we embedded an extra field in NDP "hello" messages for storing service IDs. We used the concept of Unique Universal Identifiers (UUIDs) instead of service descriptions, keeping packet lengths small for the routing messages and minimizing the effects on the network (the bigger the messages the larger the delays and the possibility of transmission errors). Such an approach implies that all nodes know a-priori the mappings between services offered in the MANET and UUIDs. This is a common assumption and is justified by the fact that most MANETs are deployed for certain purposes where there is lack of fixed communication infrastructure (e.g. a battlefield or a spot of physical disaster). In such environments, the roles of every participating node are concrete and can be easily classified in types of services. For example, in a battlefield one node may offer radar information to the rest, while another one may offer critical mission update information. In such environments the mapping of services to UUIDs is more than sufficient for service discovery. Semantic matching of rich service descriptions is of no particular use in these cases, not to mention that these techniques lead to increased 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. A node broadcasting this IARP update packets sets the TTL (Time To Live) field in these packets equal to its routing zone diameter, so that they will be dropped at border nodes. This way each node knows the routes to all the nodes in its zone and also the services that these nodes offer; thus adding the service discovery capability to the proactive part of ZRP.

The extended version of ZRP we implemented (henceforth called E-ZRP) is capable of providing routing layer support for proactive service discovery.

In a previous work [15], we have shown through extensive simulations that our cross-layer implementation consistently outperforms an application-layer service discovery scheme based on restricted-area flooding in terms of battery consumption, both in static and mobile environments. Our proposed protocol (E-ZRP) leads to significantly smaller energy consumption (approximately 50% less) and at the same time it manages to discover almost the same (and in many cases a higher) number of services.

In the following paragraphs we demonstrate the performance of E-ZRP in terms of SAD under different network conditions.

III. Mobility and Density Impact on SAD

Our simulations were conducted using the QUALNET Simulator [14], which includes a ZRP module. Of course this module has been extended in order to provide the E-ZRP functionality described in the above paragraphs. A basic assumption in our simulations is that each node may host one out of three possible services, which can be provided to other nodes, and runs E-ZRP as its routing protocol. The selection of any of these 3 services has the same probability for any node, hence at the end of the allocation 1/3 of the node population hosts the first service, another 1/3 hosts the second service and the last 1/3 hosts the third service. At the physical and data-link layer we used the IEEE 802.11b protocol. The bit rate was set to 2Mb/sec and the range was set to 250m.

We simulated a network comprising of 30 nodes uniformly dispersed in a 3000x3000 meters square area. We used a random waypoint mobility model. First, we test the sensitivity of SAD against different speeds. We simulated five different scenarios. In the first scenario each node's speed (in meters/second) was distributed between 0 and 3,5m/s (low mobility), in the second scenario between 0 and 7m/s (medium mobility), in the third scenario between 0 and 9m/s (medium mobility), in the fourth scenario between 0 and 11m/s (high mobility) and in the last scenario between 0 and 14m/s (high mobility). The simulation duration was 2700 seconds in every experiment. Figure 3, depicts the results for these experiments. The X-axis represents the time for which a service remains visible to a node (SAD metric), and the Y-axis represents the sum of services discovered by all nodes (including service re-discoveries). As we can see, E-ZRP tends to discover more short-lived services in highly mobile environments (due to node mobility and service rediscoveries). More long-lived services can be discovered only in low mobility cases. This is explained by the fact that when the nodes are highly mobile, paths are difficult to be maintained and hence far-away services tend to last for a very short amount of time since the probability for a path break is larger when nodes move faster. When nodes move slower these paths tend to be more stable and hence services tend to be available for a longer time.

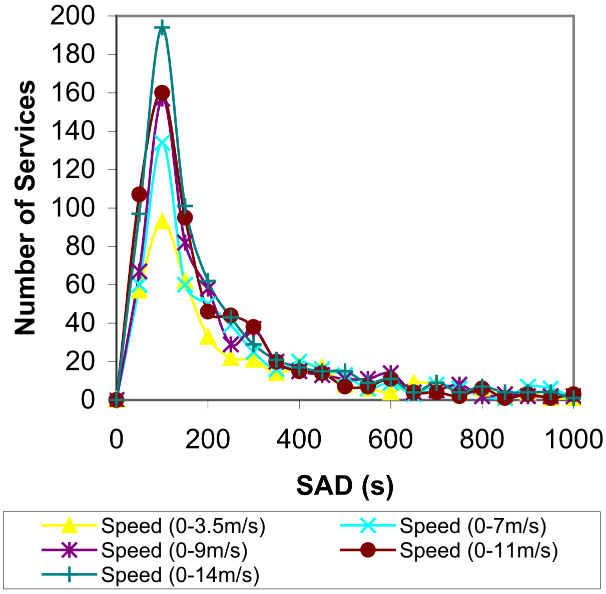

Fig.3.Service Duration Distribution vs. Speed

However, it is not obvious from this figure when we can achieve the maximum average SAD, which is a metric of great importance in analyzing the quality of discovered services. The values of average SAD over low medium and high mobility are presented in figure 4. The lines connecting the 5 spots in the figure do not correspond to speeds other than the five defined above, but are drawn for better viewing. It is evident from this figure that the average SAD actually decreases when speed increases. However, it wouldn't be fair to compare the performance of the protocol under service duration only. The amount of services discovered (including rediscoveries) is also important, since it is much more preferable for a node to discover, throughout its life, for example 30 services with an average SAD of 100 seconds instead of 10 services with an average SAD of 150 seconds (given that a transaction with any service in both cases may last for less than 100 seconds). In figure 5 we show the total amount of services discovered over low, medium and high mobility. As it was expected the high mobility case (maximum speed = 14m/s) outperforms all the other in the total amount of services discovered. So, there is a tradeoff between average SAD and number of discovered services. In order to evaluate when our protocol performs better, we should be aware of the average transaction duration (ATD) between a node and any service. So, for high ATD, the discovery protocol would perform better in a low mobility setting. This is explained by the fact that the additional services discovered in higher mobility settings would be of no use, because their average SAD would be inadequate to complete a transaction. The inverse would be true for low ATD, where a high mobility setting would be ideal for the discovery protocol.

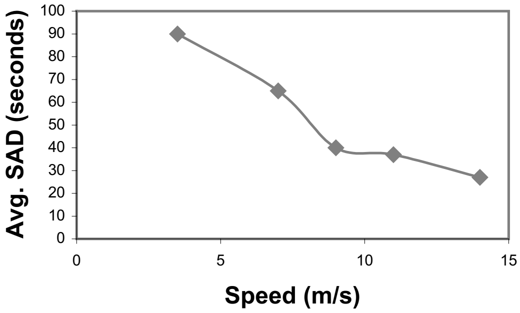

Fig.4.Average SAD vs. Speed

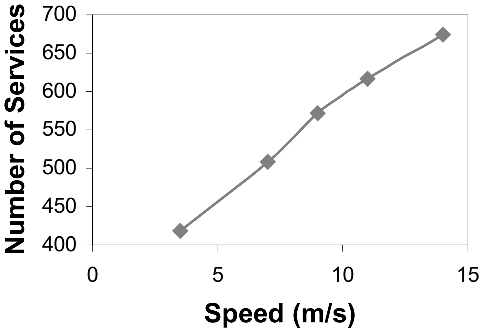

Fig.5.Total number of Services Discovered vs. Speed

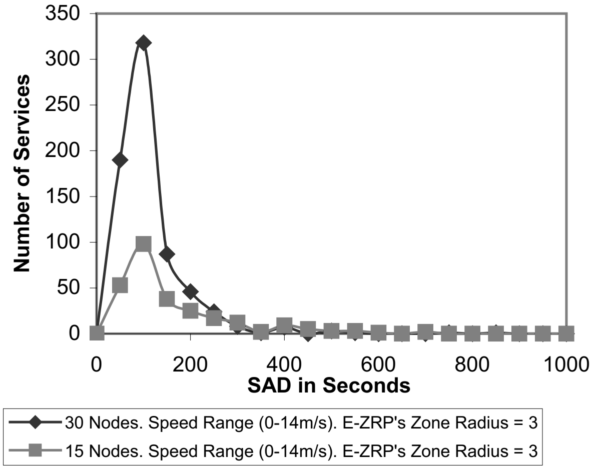

Fig.6.Service Duration Distribution vs. Density

Now related to density, we simulated two scenarios. The first scenario included 30 nodes moving on a terrain of 3000x3000 meters, following the random waypoint model with maximum speed 14m/s (minimum speed is still 0m/s). The second scenario (half density scenario) was identical to the previous one but included only 15 nodes. Both scenarios had duration of 2700 seconds each. The results are shown in figure 6, where it is made obvious that by reducing node density to one half, the service duration distributions follow the same pattern, but the number of services in the half-density case is on the average 1/4 of the services found in the full-density case. This is due to the fact that re-discoveries of services are more frequent in a denser environment. Also the fact that routing messages are marginally increased in size in order to encapsulate service information, presents very good scaling properties when increasing density. The length of routing messages plays a significant role under high-density cases where congestion is present. Hence if messages were altered to include complete service descriptions instead of UUIDs, then an increase in node density would lead to an increase in loss due to congestion and hence a lower number of services would be discovered.

One would expect that in the denser environment services would tend to last longer, since there are more alternative paths to a service provider through which a node can reach a service and also more alternative service providers, hence a failure of one or more paths doesn't necessarily mean that the node cannot access the given service. Simulation results presented in Table 1 show that this is not true. Actually, when density increases, despite the existence of multiple paths and providers, the average service duration is decreased. This is explained by the fact that more nodes create more contention for accessing the channel and transmitting service advertisements. Hence, more packet collisions occur and paths to services are actually broken more frequently, due to the fact that they couldn't be updated timely. The total number of services discovered, however is higher in denser environments (Table 1). This means that high density may increase the number of discovered services but it decreases their quality in terms of availability. Once again, in order to evaluate when our protocol performs better, we should be aware of the average transaction duration between a node and any service.

	Full Density (30 nodes)	Half Density (15 nodes)
Avg. SAD	92 s	136 s
Total # of Services Discovered	687	268

Table1: Average SAD vs. Density

IV. Conclusions and Future Work

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

We have presented a new cross-layer architecture that integrates service discovery functionality with an existing routing protocol. In this paper we examined the implications of network density and node mobility on the availability of services discovered with a representative routing layer based service discovery protocol (namely E-ZRP).

In our current work we extend our approach with an additional mechanism, which allows nodes to predict service availability and hence make near optimal service selections. The mechanism actually uses past service availability information cached on a node, and computes an expected SAD for services in the future considering current network conditions like density and mobility.

References

[1] Sun Microsystems, *JINI Architecture Specification*, Nov. 1999.

[2] Salutation Consortium, "Salutation Architecture Specification," http://www.salutation.org/specordr.htm, 1999.

[3] Microsoft Corporation, "Universal Plug and Play: Background", http://www.upnp.org/resources/UPnPbkgnd.htm, 1999.

[4] Universal Description Discovery and Integration Platform, http://www.uddi.org/pubs/Iru_UDDI_Technical_White_Paper.pdf, Sept. 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 et al., "Allia: Alliance-based Service Discovery for Ad-Hoc Environments," Proc. ACM Mobile Commerce Workshop, Sept. 2002.

[8] D. Chakraborty and A. Joshi, "GSD: A novel group-based service discovery protocol for MANETS," IEEE Conference on Mobile and Wireless Communications Networks, Stockholm, Sweden, Sept. 2002.

[9] S. Helal et al., "Konark-A Service Discovery and Delivery Protocol for Ad-Hoc Networks," Proc. IEEE WCNC, New Orleans, LO, March 2003.

[10] M. Nidd, "Service Discovery in DEAPspace", IEEE Personal Communications, August 2001, pp. 39-45.

[11] G. Schiele, C. Becker and K. Rothermel, "Energy-Efficient Cluster-based Service Discovery for Ubiquitous Computing," Proc. ACM SIGOPS European Workshop, Leuven, Belgium, Sept. 2004.

[12] R. Koodli and C. E. Perkins, "Service discovery in on-demand ad hoc networks," Internet Draft, Oct. 2002.

[13] Z.J. Haas, M.R Pearlman, P. Samar, "The Zone Routing Protocol (ZRP) for Ad Hoc Networks," Internet Draft, July 2002.

[14] Scalable Network Technologies Inc., "Qualnet Simulator," http://www.scalable-networks.com

[15] C. Ververidis and G. C. Polyzos , "Routing Layer Support for Service Discovery in Mobile Ad Hoc Networks", Proc. Wireless Pervasive Networking Workshop, Hawaii, March 2005.

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

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