BF-SD-ZRP: A smart integrated scheme for service and route discovery in Mobile ad hoc network

Fatma Outay*†, Florent Kaisser*, Veronique Veque*, Ridha Bouallegue†
*Institute of Fundamental Electronics,
University of Paris-Sud 11
Bat 220, Orsay 91405
name.surname@u-psud.fr
† SUP'COM, 6'Tel Laboratory
name.surname@supcom.rnu.tn

***Abstract*—Mobile ad hoc networks are characterized by their highly dynamic, multi-hop, and infrastructure-less nature. Due to limited computing power, scarce bandwidth, high mobility and the lack of a central coordinating entity, service discovery in these network is a challenging task. The great majority of service discovery protocols developed for MANETs deal with the above issues at the application layer.**

In this paper, we discuss the fact of implementing service discovery at the routing layer instead of the application layer, in order to reduce Service Discovery (SD) overhead and to limit resources consumption. We develop an integrated service discovery protocol, called BF-SD-ZRP, utilizing a combination of different optimization techniques: piggibacking of SD information on the routing messages, compact description of SD using Bloom filter (BF) and service caching. Our simulation results allows to validate our scheme and demonstrate its efficiency in terms of overhead reduction and service discoverability.

***Index Terms*—MANETs, Service discovery, Routing protocols, Service description, Bloom filter, ZRP, OLSR.**

I. Introduction

Recent research in mobile ad hoc networks (MANETs) has focused on providing a basic network support such as medium access control, routing protocols and TCP. There is no doubt that these issues are critical and must be resolved to meet several requirements of ad hoc network environments (ie shared wireless medium, multihop routes and their dynamic changes, processing capabilities and limited resources, etc.). However, other important researches are still in their initial stages. These include among others the power management, network security, and middleware services. Especially, research on middleware for ad hoc networks is important to make easier the development and deployment of MANET applications. Among a variety of services middleware, service discovery played a significant role in MANETs. With the auto-configurable and autonomous characteristic of ad hoc networks, mobile nodes should be able to automatically and efficiently discover available network services, e.g., DNS, Webserver, mail server, print server and TFTP. Several protocols have been proposed for service discovery in wired connection, but unfortunately these protocols can not adapt to the ad hoc environment due to its dynamic nature and very limited resources. These service discovery protocols like Jini, UPnP and SLP, etc., rely primarily on their assumption of the use of a centralized repository, and it is not suitable for ad hoc network scenarios.

In recent years, several protocols have been presented to support the service discovery specifically designed for MANET environments [1], [3]–[5]. These service discovery protocols for ad hoc can be divided into two categories: centralized directory-based protocols and distributed directory-less protocols. In the first category, some agents of the directory (or service brokers) are involved as a logical entity in a process of service discovery for a communication between client and service providers [4], [7]. However, in [1], [3], [5], [6], the distributed directory-less approach without central repository has been proposed where each node is required to provide some form of directory services and locate resources in a Peer to Peer way. The distributed directory-less protocols are deployed with two different models of operation: "push model" and "pull model". With the push model, service announcements are flooded periodically in an active manner by service providers, so clients learn passively services available by simply receiving these advertisements. While in the pull model, clients are more active, requesting explicitely services, by broadcasting service requests to all the servers which will respond eventually.

Although the algorithms proposed for MANET applications differ in the approach used to search for and provide the desired services, they are all oriented middleware solutions and are based on the common assumption that some routing protocols may exist below (separately from the middleware layer) to support the delivery of client service requests to service providers or vice versa. This may make their use inefficient in the ad hoc wireless networks, due to the problem of redundant control packets flooding, which in part can cause an extremely high network overhead.

It is important to note that both service and route discovery protocols are based on the mechanism of broadcasting with different goals of discovery. Motivated by this interesting observation, in this paper, we focus on the combination of these two different problems of discovery (but quite similar in some sense) such that service discovery can be combined with the discovery of a route to those services. As the SD takes place at the application layer and routing is a network issue, we develop a cross-layer approach.

In this paper, we focus on integrated approaches of service and route discovery. In our previous work [2], we have introduced our scheme of combining service and route discovery, using a smart compression scheme "Bloom filter" for service description , and theoretically proved that by exploiting service discovery information provided by the routing layer, the resulting communication overhead is significantly reduced and the service discovery efficiency is proved. In this paper, we develop our integrated approach on a simulator by modifying the ZRP protocol. Our simulation results allows to evaluate our scheme and demonstrate its efficiency in terms of overhead reduction and service discoverability.

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 looking for a service, in addition to discovering the service, it is also informed of the route to the service provider at the same time. We extended the hybrid routing protrocol ZRP [15]

Differing from existing work on cross-layer service discovery, this paper focuses to support low bandwidth environments and investigates an efficient way to describe services using Bloom Filter combining with service caching. The analysis and simulation results show that our optimised approach named (**B**loom **F**ilter based **S**ervice **D**iscovery combined with **ZRP** *BF-SD-ZRP* induces very low overhead compared to other integrated protocols and satisfy several network issues, where each node should discover the maximum number of services.

The reminder of this paper is organised as follows: In section II we briefly present some existing integrated approachs of routing layer supported service discovery and their limitations. Section III presents BF-SD-ZRP protocol in details. Section IV describes our implementation and discusses the simulation results. Finally, in section V we provide our conclusion and introduce our future research directions.

II. Existing Integrated scheme

In an ad hoc network, each message transmission consumes a considerable amount of network bandwidth as well as some processing power and battery at each crossing by a node along its path. The discovery service itself requires the exchange of many messages. The subsequent discovery of a route to the service provider via the underlying routing protocol requires a further exchange of messages. However, when integrating service and route discovery, we are able to discover a service provider and a route to it using the same set of messages. Therefore, a combined approach will significantly reduce the consumption of bandwidth and the total latency in the discovery process of a service and a route to its provider.

Traditionally, service discovery is considered as a function of the application layer. However, in this paper, we examine the execution of service discovery at the network layer in order to reduce the communication and processing overhead. Such a modification is reasonable in the case of an ad hoc network. This approach exploits the ability to acquire information service with routing information (the same message) by superimposing the information service with routing messages. In this way, redundant transmissions of service discovery packets at the application layer are avoided and more resources will be saved.

The idea of providing a routing layer support for service discovery was first introduced by Koodli and Perkins [6]. They add extensions to the ad hoc routing protocols to provide support for service discovery and route to these services together. However, no experimental evaluation of their proposal has been published so far.

Another integrated approach was proposed by Jodra in [8], in which, he presents a solution to integrate service discovery with routing protocol OLSR. The various messages in OLSR share a common message header. Using this header, a new message called Service Discovery Message (SDM) was introduced. The SDM packet may contain either an announcement or request for service. However, OLSR is a proactive routing protocol, to maintain updated routing tables, it introduces a considerable overhead. In addition, it should be noted that no comparison with other protocols or integrated application layer has been presented.

Another scheme proposed in [9], is to implement service discovery at the routing layer by piggybacking service information, using UUID descriptor, into ZRP routing protocol control messages, thus enabling the terminals to acquire the service and routing information simultaneously.

As direct comparisons of integrated approaches are difficult due to lack of certain performance data compatible for different protocols, a rough categorization regarding the type of routing protocol (reactive, proactive or hybrid), the most efficient (in terms of Overhead) when including a service discovery protocol is desired.

ZRP seems to be the most appropriate protocol for this type of network using the following criteria:

- It is ideal for environments where local information, either routing or service information, is of particular interest because it can discover (through the concept of zones) in a fast and efficient way in terms of resources consumption.
- It is scalable as it propagates in an intelligent way the information to remote nodes by avoiding floods.

The approach described in [9] seems to be ideal in terms of choice of routing protocol because the cross layer solution used is based on ZRP protocol, however, in terms of service description, using the universal identifiers UUID is not the optimal solution.

A. Limitations of existing service discovery approaches in term of service description

There are different approaches to describe service information in requests and advertisements (ads). Many service discovery protocols use XML to describe the service information. Such a method is adopted in [1]. Another approach is to create service descriptors from ontologies designed for semantic web services using the Web Ontology Language (OWL) as proposed in [10]. Using this approach, the ontology must be shared between nodes before communication. However, XML and OWL descriptions require considerable bandwidth, which is sparse in ad hoc networks. A kind of compression can be used to fill this gap if rich and flexible service descriptors are necessary. Some proposals aim to reduce the consumption of bandwidth, and do not see the benefits of using XML or OWL as necessary. By mapping a set of service descriptors to a predefined set of integers as in [8] or unique identifiers Universal (UUID) as in [11] and [9], the description can be reduced by a few bytes.

These solutions can save bandwidth compared to the transmission of XML files. However, these solutions are neither flexible enough nor scalable, when the maintenance on each network node is required when adding new categories of services. From these studies, we found Bloom filters [12]. Using Bloom filters, any textual description of services can be hashed into an array of defined size without requiring a predefined set of stationary keywords.

III. BF-SD-ZRP DESIGN

A. Overview

In this paper, we propose the piggybacking of service information into routing messages to reduce communication overhead and latency and prove the effeciency of service discovery. We demonstrate the advantages of our approach (i.e., routing layer supporting service discovery) by extending the routing protocol (ZRP), which is an hybrid routing protocol (i.e., proactive in a number of hops around a node called the node zone, and reactive for applications outside this zone), so it is able to encapsulate service information in its messages. We have changed the behaviour of ZRP protocol to support service information. We have added an extra field in NDP packets. Our intention is to evaluate the performance of a combined approach in several scenarios. Our proposal also introduces a cache of discovered services for each node in the network.

Bloom Filters are used to describe service information encapsulated in routing messages. Our service discovery approach is used to distribute the filters using a protocol format that extends the routing protocol ZRP.

In order to add mechanisms for service discovery to the ZRP protocol, we incorporated an additional field in the "hello" messages to store the service identifier. We used the concept of Bloom Filter instead of service descriptions to keep small size of packets for routing messages and minimize the impact on the network (the bigger the messages, the larger the delays and the possibility of transmission errors). Such an approach requires that all nodes use the same number of hash functions and the same MD5 [17] cryptographic algorithm to convert a service description into a Bloom Filter.

ZRP has been extended to include service information in each entry of IARP routing messages. Each node of the ad hoc network uses a cache SvcCache where every provided service will be stored through service announcements gathered from the incoming NDP packets . In this cache the following

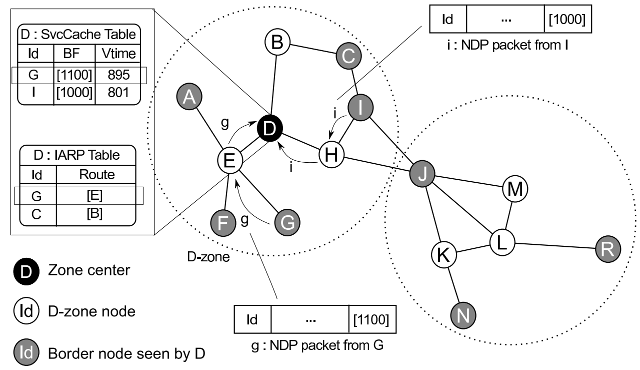


Fig. 1: SvcCache and IARP table update and interaction upon receiving of NDP "hello" messages announcing *video* (1000) service provided by node I and *printer* (1100) service provided by node G. Node D looks for a printer service, which is converted into a BF (1100), it checks first its SvcCache, retrieves the printer's provider which is G and looks for a route to G in its IARP Table.

service informations are stored: provider, service name (as a BF vector) and its lifetime. When a node wants to use a service (Figure 1), checks its cache, if it is found, retrieves its provider address and check for a route to it in the routing table.

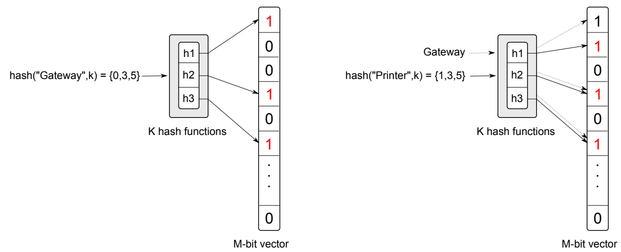
The protocol uses local services caching advertised by neighbour nodes to save network bandwidth and reduce latency of discovery. The process of service discovery and location of provider is described in the algorithm1:

Algorithm 1 Return Px provider of service X

Require: $SvcCache$ not empty
if $\exists LookupSvcCache(X)$ **then**
  $Px \leftarrow LookupSvcCache(X).@$
  if $\exists RoutingTable(Px)$ **then**
    Return Px
  end if
end if

B. Service description using Bloom Filter

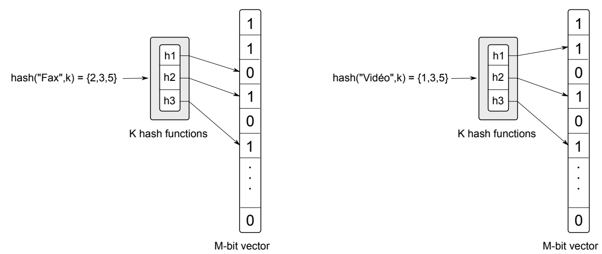
In our context, the intention of Bloom Filter is to represent a set $S = (x1, x2, ..., xn)$ of service descriptors n in an efficient manner. We begin by defining a V Bloom Filter implemented as an array of m bits. All bits $(1, ..., m)$ are initially set to 0. The filter uses k independent hash functions $h1, h2, ..., hk$ With the range $(1, ..., m)$ for hashing each service descriptor x in the filter V. For each service descriptor $x \in S$, the output hash $hi(x)$ represents a position in Table V, $V[hi(x)]$ is set to 1 for all hash functions $i = 1, 2, ..., k$. One position in V can be set to 1 several times, but it is clear that only the first change has an effect. Figures 2(a) and 2(b) illustrate two different services hashed with three hash functions, and then added to the same filter. To check whether a service z is in the Bloom Filter, we must determine whether all $hi(z)$ are set



(a) The first service added to the Bloom Filter

(b) The second service added to the Bloom Filter

Fig. 2: Construct a Bloom Filter



(a) The service is not in the Filter

(b) This request give a false positive

Fig. 3: Check a Bloom Filter

to 1. If so, we assume that the service z is available. If all the $hi(z)$ are not set to 1, the service is not part of the filter, as in the Figure 3(a). However, the Bloom Filter can give a false positive if the filter indicates that the service descriptor $z \in S$, even if it is not (Figure 3(b)). The chances of obtaining a false positive can be estimated using the calculation of probabilities [12].

The heart of Bloom Filter is a hash function. Any hash function can be used as it is able to map an element (for example service descriptor) to a uniform pseudo-random number over the range $1...m$.The results of various k hash functions must be independent. One way to implement a hashing function is to use a serie of modulo functions [18]. Another approach is to use a cryptographic hash function MD5 [17]. MD5 is deterministic and uniform, and has excellent collision resistance for our purpose. MD5 also exists as an open source code for many programming languages, and implementations are relatively fast.

Our Service Discovery design uses MD5 in the following manner: the k hash functions, which constitute the Bloom Filter, are constructed from k groups of each r bits out of the 128 bit hash from the MD5 operation. Any set of sub-bits from an MD5 output can be used as an input to an independent function. Each of these k functions sets one bit in the filter v.

In BF-SD-ZRP, the hash functions are implemented as shown in the algorithm 2:

Algorithm 2 Calculate the Bloom Filter V for a service X

Require: : $x \neq 0$

$a \Leftarrow MD5(x)$

$r \Leftarrow \dfrac{128}{k}$

for $i = 0$ TO k **do**

  $f \leftarrow subbits(r * i, (r * (i + 1)) - 1, a)$

  $V[f \mod m] = 1$

end for

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Link Source Address																															
Link State Seq Num																Zone radius								TTL							
RESERVED																RESERVED								Link Dest Cnt							
<Node Address 1>...<Node Address n> \																															
Service Filter \																															

Fig. 4: IARP packet Format

C. Our Protocol format

The ZRP protocol [15] is an hybrid model between proactive and reactive scheme. ZRP combines several sub protocols, namely, IARP (inside the zone), IERP (outside the zone), and others (NDP, ICMP and BRP... Etc..). The detection of neighbors is done by consulting the lower layers. ZRP simply retrieves neighbors table provided by another neighborhood search mechanism. This table will be used by IARP to construct the network map. Each modification of the neighbors table will lead to an updating of the zone by IARP and updating routing tables by IERP.

Neighbor Discovery Protocol (NDP) is implemented at the link layer. Using NDP, each node periodically broadcasts a "hello" message to indicate its presence. NDP updates IARP table, IERP uses IARP table, IERP then relays the requests to border nodes via BRP. BRP uses IARP table to guide requests. In our implementation, we have built an extra field in the "hello" messages to store the service identifiers. We used the concept of Bloom Filter instead of an ordinary service description to keep the NDP packet length small. The Bloom Filter length used in our implementation is 128 bits.

Our proposed solution is therefore to distribute a summary of offered services in a vector described as a Bloom filter. A Bloom filter is a data structure that allows the representation of a data in a simple and unobtrusive way. The IARP packet(figure 4) carries all the information about neighbors and their services in a Bloom filter to all zone nodes.

Each node has a cache in which it stores all the provided services (local and those of neighbors). In this cache, the sotred informations (provider address, service name (Bloom filter) and the lifetime) are retained from received NDP packets. Upon receiving NDP packet, a node updates its neighbor table. IARP listens to informations gathered from NDP messages, updates its IARP table, and periodically broadcasts its table to its neighbors in IARP packets. A node distributing these

IARP update packets sets the TTL field to the radius value of the routing zone, hence packets are dropped when arriving at border nodes. Upon receiving of IARP packets, a node updates its service cache and its routing table. Thus, each node knows the routes to all nodes in its zone as well as services offered by these nodes. In the next section, we demonstrate the performance of BF-SD-ZRP in terms of overhead optimization and service discovery efficiency in different network conditions.

IV. Performance Evaluation of BF-SD-ZRP

A. Simulation setup

The choice of the simulator is an essential step to test and evaluate the performance of any new protocol. In this paper, and in order to evaluate our approach, we will proceed to the simulation using the JIST/SWANS [14] toul (Java in Simulation Time/Scalable Wireless Ad-hoc Network Simulator). This will allow us to conduct a serie of tests to measure the performance of our approach and study the impact of some parameters of MANET namely the mobility degree , the density of nodes and the choice of routing protocol.

In Jist/Swans, the routing entity has only three routing protocols, namely AODV, DSR and ZRP and no proactive routing protocol has been implemented. To study the performance of our integrated discovery approach, we implemented the OLSR (Optimized Link State Routing Protocol)routing protocol [16] at the routing layer of the Jist/Swans protocol stack. The BF-SD-ZRP protocol is compared with OLSR integrated protocol and with standard ZRP routing protcol without service information.

Integrating service discovery protocol with proactive OLSR (Optimized Link State Routing) has been proposed by [8]. A new type of message, called Service Discovery Message (SDM) was added. This message announces (servers) and requests (clients) services. Local services available will be advertised only once, with a specific life time, to avoid increasing the network overhead.

These services will be stored in a service cache which is maintained in each node. When a node requests a service, it will look in the local cache. If the requested service is not yet stored, it will send a query message. In their results, they showed that the introduced packet overhead is insignificant compared to the standard OLSR. For such a mechanism, to be efficient, it is necessary that each node must know the maximum number of services without increasing the overhead of the network.

Our implementation of OLSR combined with service discovery was based on the idea gathered from the approach described in [8]. The SDM message format added to OLSR control messages is shown in Figure 5. Its size is 8 bytes, including the description of the service information as an URL String of size 2 bytes. In our implementation of SD-OLSR, we proceeded like in BF-SD-ZRP, each node maintains a service cache to store services announced in the network. Hence a node in our case does not need to send a service request if the requested service is not in the service cache, it considers that it is not available. We studied the performance of OLSR combined with the service discovery, and compared its behaviour with our extented ZRP approach. Simulation results and performance discussions on performance are presented in the next sections.

0			1		2		3
0 1 2 3 4 5 6 7	8 9		0 1 2 3 4 5	6 7 8 9	0 1 2 3 4 5 6 7 8 9		0 1
Message \							
Message Type		Vtime		Message size			
Originator Address							
TTL		Hop count		Message Sequence Number			
Service Descriptor (URL:String) \							

Fig. 5: OLSR message Format

TABLE I: Default simulation parameters for ZRP and OLSR

Parameters	*Default Values*
Simulation area	$1000X1000m2$
Simulation time	$130s$
Data rate	$10kbit/s$
transmission range	$250m$
Mac Layer	$802.11b$
Default Wifi radio bandwidth	$11Mbit/s$
Zone radius	2 hops
NDP message broadcast intervall in ZRP and BF-SD-ZRP	$10s$
IARP message broadcast intervall in ZRP and BF-SD-ZRP	$2s$
SDM message broadcast intervall in OLSR and SD-OLSR	$10s$
hello message broadcast intervall in OLSR and SD-OLSR	$2s$
Offered services	1 service per node

B. Simulation metrics and scenarios

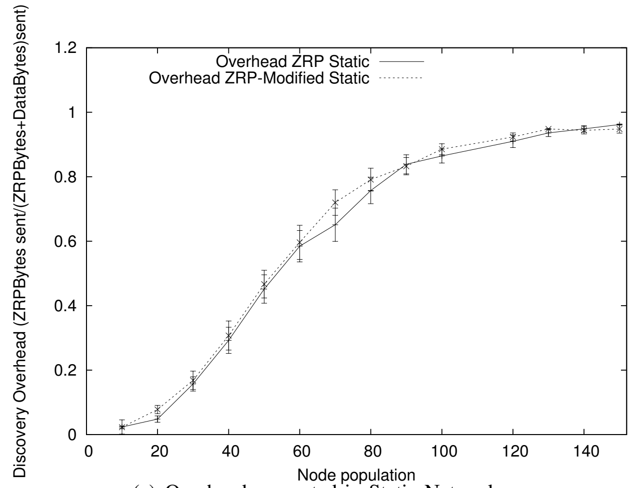
We have implemented our extension of ZRP on Jist/Swans simulator. In this section, we evaluate the performance of our approach. The performance is measured in terms of overhead of control packets after the extension (after adding a service information of size 128 bits in the NDP packet), the packet delivery rate of data packets (PDR) and the impact of mobility, density and zone radius on the efficiency of service discovery. Therefore, the simulations are performed by varying several factors and observing the specific performance metrics. In the simulations, many different scenarios were conducted based on metrics that we wish to assess.

1) **Simulation Parameters**: The default settings of the wireless network shown in Table I are used in all scenarios of OLSR and ZRP. We have almost kept the same values to observe the behavior of each approach in almost the same wireless environment.
2) **Metrics used in performance evaluation of BF-SD-ZRP**: In all experiments we have conducted, our hybrid

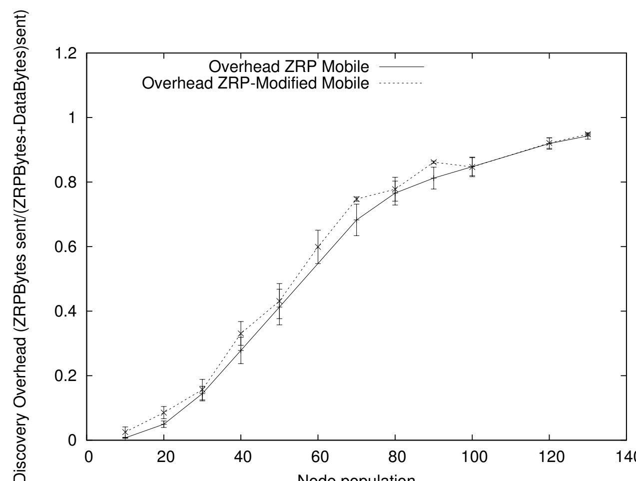
routing protocol ZRP combined with service discovery (BF-SD-ZRP) was compared to a basic routing protocol without the integration of service information, and also to another integrated proactive protocol SD-OLSR. Metrics maintained in the simulations are as follows:

- Overhead: is the rate of control packets sent in the network. The overhead represents the ratio of the size of control packets by the size of data and control packets.
- PDR (Packet Delivery Rate): is the rate of received data packets in the network.
- Efficiency of service discovery: is the percentage of services discovered among all the provided services in the network.

3) **Scenarios and simulation results of BF-SD-ZRP** As we modified the NDP control packet to add a new service information in the form of a Bloom Filter of size 16 bytes, the NDP packet size has increased. Our objective in evaluating the *Overhead* metric, is to prove that we obtain an efficient mechanism for service discovery without introducing a significant overhead in the network.
In this scenario, the overhead, which represents the percentage (size) of control packets sent over total (data and control) is plotted as a function of network density, the result is shown in Figures 6(a) and 6(b) in both mobile and static topologies. The routing overhead introduced by the BF-SD-ZRP protocol is represented by the dotted line. It is clearly observed that the service information add an insignificant overhead in the network in both static and mobile case. Looking at the results below, it is clear that our approach is efficient in terms of overhead. We note that the overhead generated by our integrated protocol is always insignificant compared to the standard protocol. It has increased only by around (0.09%(static) to 0.12%(mobile)) compared to the initial overhead generated by the basic ZRP protocol.
Another set of experiment has been run to measure the *PDR*. We want to acheive a rate of PDR up to 100% . Our goal is to prove that the routing protocol is still efficient and works well even by adding new information to control packets such as service information. The result is shown in Figure 7. In this scenario, we have simulated a network consisting of 10 to 150 nodes, we have varied the network density, the mobility model is the *Random Walk* model with a pause time of 0 seconds. We observe from the figure that the PDR is beyond 92%, which proves the efficiency of our combined approach.
The third metric evaluated in our set of scenarios is the *Efficiency of service discovery*. To demonstrate the efficeincy of our integrated approach in terms of service discovery, we have conducted two sets of experiments. In the first set of experiments, we vary the density of the network (between 10 and 150). The mobility model used in the case of mobile network is the random



(a) Overhead generated in Static Network



(b) Overhead generated in Mobile Network

Fig. 6: Overhead generated by piggybacking service information in NDP Packets

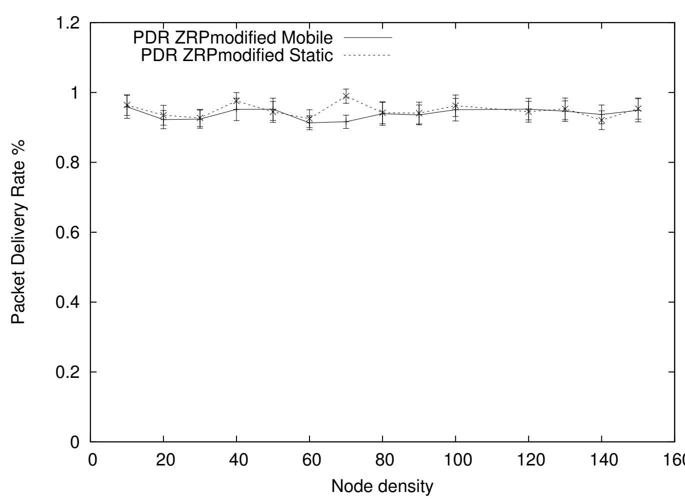


Fig. 7: The resulting PDR after the modification of ZRP

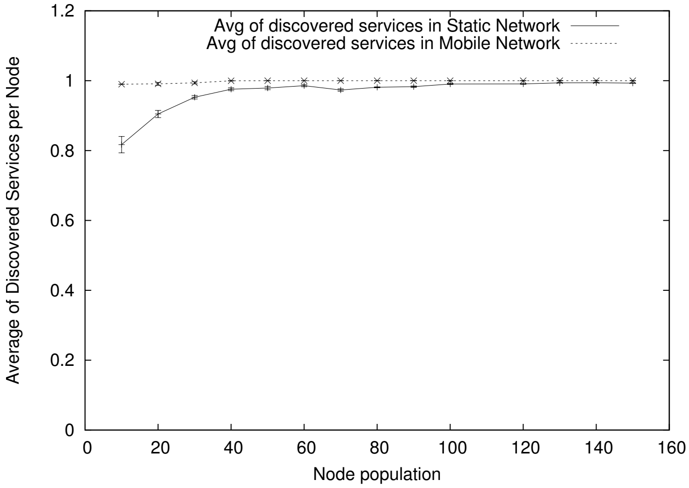
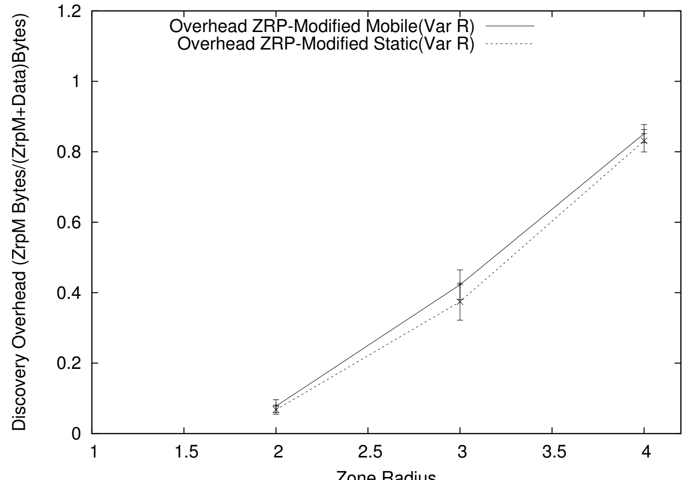
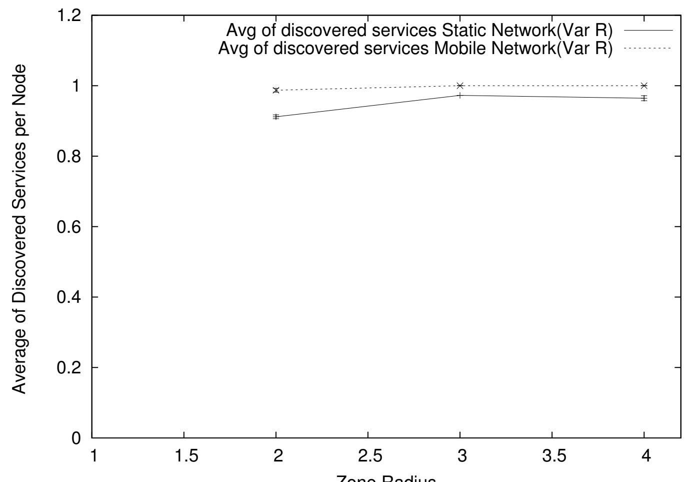


Fig. 8: Service discovery process efficeincy



(a) Overhead is inscreasing with the zone radius



(b) Service discovery rate is almost constant

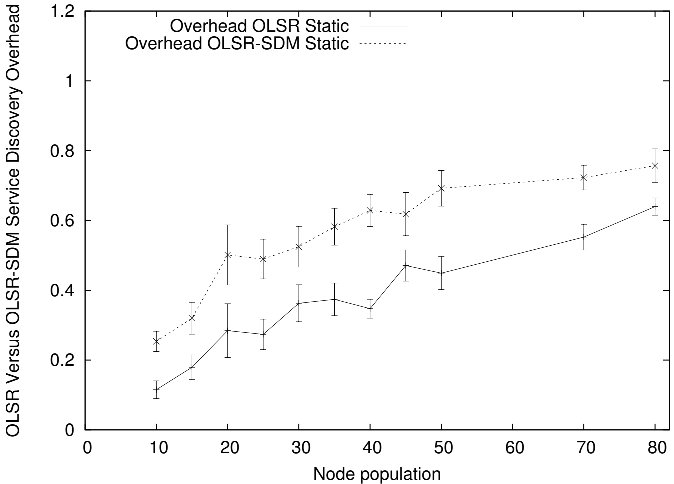
Fig. 9: Compromise between overhead and service discovery depending on the zone radius of ZRP

Random WayPoint model (RWP) at a constant speed of $3.5m/s$ and a pause time of 0 second. In this scenario, we want to analyze the efficeincy of service discovery mechanism. As shown in Figure 8, the average number of discovered services per node was more than 98% for mobile network and almost the same in static network since a density of 40 nodes. For a density of 20 and 30 nodes, as nodes are placed in an area of $1000m2$, and not moving, so they are not within each other range, therefore they can not discover all services in the network for reasons of connectivity. We observe that the protocol gives a better result when nodes are mobile (meaning that each moving node meets several nodes throughout its lifetime).

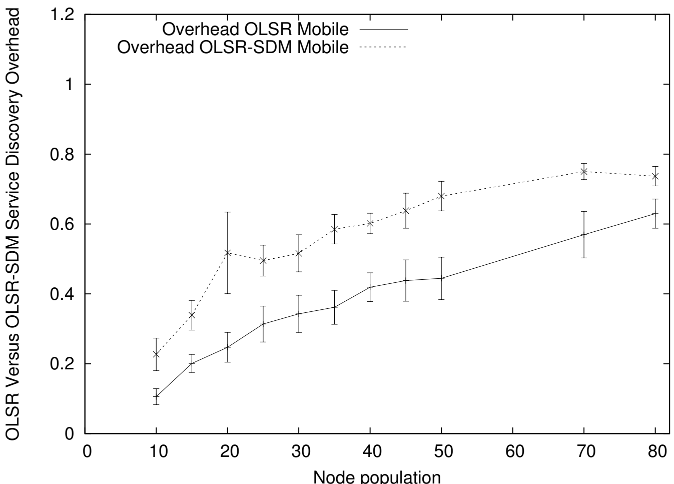
In the second set of experiements, we correlate the overhead generated by our integrated protocol by varying the radius of the area (see Figure 9(a)) with the rate of service discovery (see Figure 9(b)). We note that in our simulation results shown in Figure 9, the overhead increases with the zone radius. This is logical because, when increasing the radius of the zone, the IARP packets dominate the network, therefore, the protocol behave almost like a proactive protocol, thus increasing the overhead in the network because of periodic sending of control packets. Contrary to service discovery, we notice that increasing the zone radius had no impact on the efficiency of service discovery. Service discovery rate in the network is always beyond 90%.

4) **Scenarios and simulation results of SD-OLSR and comparison with BF-SD-ZRP** In this stage of evaluation, we want to compare our approach BF-SD-ZRP with the integrated protocol SD-OLSR in the same ad hoc environment with the same simulation parameters. The purpose of this comparison is to prove that our proposal outperforms SD-OLSR and is more scalable. Our intention is to demonstrate the efficeincy of using an hybrid routing protocol in terms of overhead optimization. We also want to demonstrate the importance of modifying the control packets structure in order to carry the service information, as presented in our approach, instead of adding another type of message circulating in the network carrying service information.

In this set of experiments, we vary the network density (between 10 and 80). To allow a fair comparison with our approach, in the case of a mobile network, we kept the same mobility model to measure the overhead, which is the Random Walk model, with a pause time of 0, and the Random Wayoint model with a pause time of 0 for the service discovery efficeincy. From figures 10(a) and 10(b), we note that the overhead generated by SDM messages is raised to 0.59% in the mobile case and 0.56% in the static case. We remark that, the overhead added by SDM message is almost 50% more than that for basic OLSR routing protocol. From



(a) Overhead of SD-OLSR in static network



(b) Overhead of SD-OLSR in mobile network

Fig. 10: Overhead generated by adding SDM message to OLSR protocol

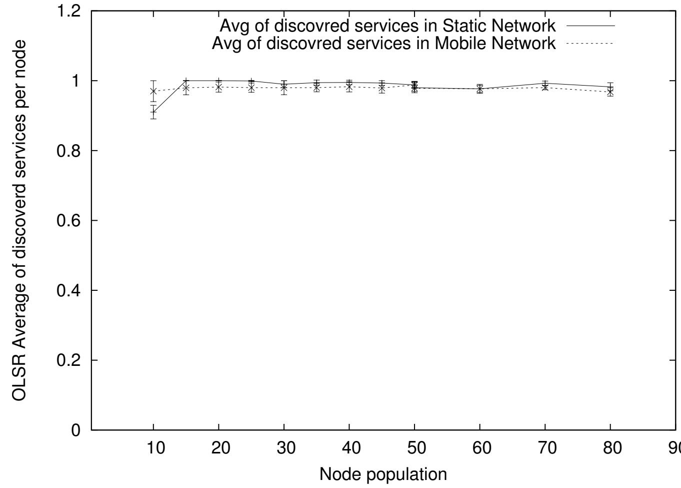


Fig. 11: Service discovery rate using SD-OLSR protocol

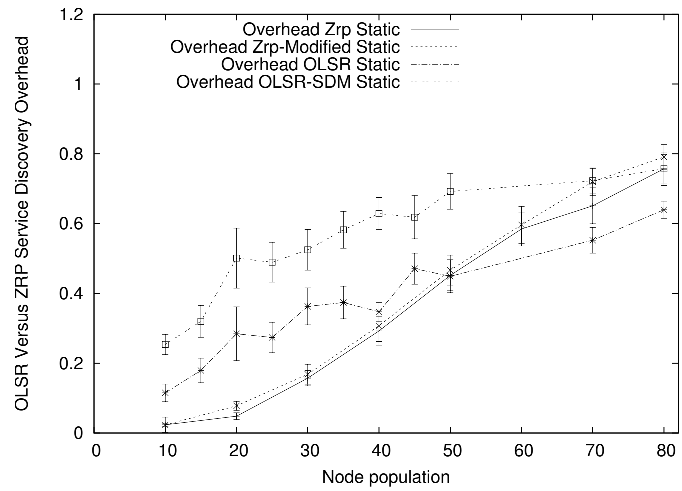


Fig. 12: Overhead Comparaison (BF-SD-ZRP et SD-OLSR) in static network

figure 11, we note that the service discovery using the SD-OLSR approach described in IV-A is also efficeint. We observe that the discovery rate is beyond 85%. As OLSR is a proactive protcol, the latency in the exchange of control packets is too high, depending on network density, which is the reason for which we simulate a network of 80 nodes maximum.

Using the previous simulations, we found that the service discovery is always efficient regardless of the nature of the routing protocol used. Therefore, in terms of performance regarding the efficiency of service discovery, BF-SD-ZRP and SD-OLSR, have almost the same level of performance.

Thus, we would prove, by measuring the amount of overhead generated by piggybacking the service information to both routing protocols, that our approach performs better than SD-OLSR, i.e. that the overhead generated by our approach is negligible as compared to SD-OLSR. To do this, we simulate the two approaches in the same wireless environment with the same simulation parameters. The result of this set of simulations is shown in Figures 12 and 13 in a static and a mobile network.

According to the results shown in Figures 12 and 13, we observe the performance of our approach over the approach SD-OLSR described in IV-A. We note that for a network of 50 nodes, the rate of overhead generated by our approach is much smaller than that generated by SD-OLSR. For a network consisting of over 50 nodes, we note that the dotted curve, which represents the overhead generated by SD-OLSR protocol, is always above all other curves and generates an important overhead. Therefore, we conclude that our approach is always optimal for a large-scale network.

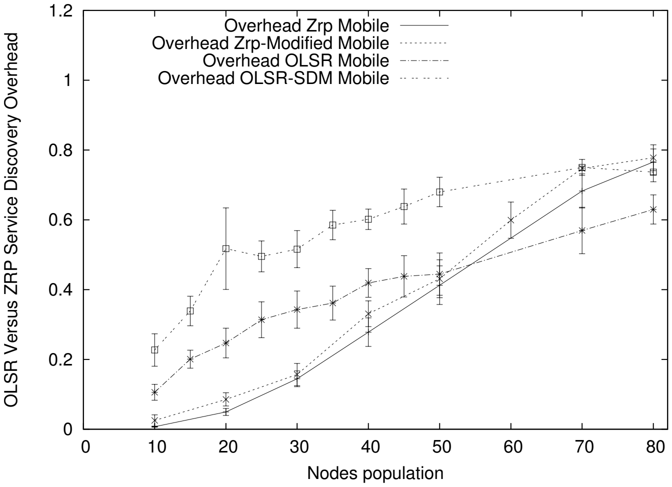


Fig. 13: Overhead Comparaison (BF-SD-ZRP et SD-OLSR) in mobile network

V. Conclusion

Existing Architectures for service discovery at the application layer suffer from redundant control messages transmission of both route and service discovery (in the sense that control messages for information discovery are necessary at the network layer and the application layer). In the literature, it was suggested that the service discovery can be integrated with routing, enabling nodes to find available services and routes at the same time. Therefore, fewer messages are broadcasted in the network. However, no evaluation comparing cross-layer approaches of routing combined with the service discovery has been presented so far. In this paper, we presented a new architecture of routing layer that integrates service discovery functionality with existing routing protocol. Our main objective was to demonstrate the impact of such an approach on the efficeincy of service discovery in ad hoc environment.

We have presented a method for discovering services using a combination of Bloom filters and the extensibility feature of ZRP. Our protocol uses Bloom filters as an efficeint way to describe the information of an arbitrary service. The protocol uses the efficient dissemination of service descriptor using the bordercasting ZRP protocol. In addition, the protocol uses local caching to reduce discovery latency.

We have shown that service descriptor compression obtained from the Bloom Filter scheme (compared to transmission of service descriptors in a text format [8]), and the subsequent piggybacking of service information in ZRP routing packet, does not much increase network overhead. With these optimizations, it is expected that BF-SD-ZRP is better than other cross-layer proposals [8], [9]. The service cache is used to conserve network bandwidth.

We have shown through simulations that our approach outperforms other integrated approaches, for both fixed and mobile environments. We find that a combination of optimization techniques as presented by BF-SD-ZRP is inevitable in order to support efficeint service discovery in environments with limited bandwidth. Nevertheless, the energy in this type of network, is a scarce resource. We believe that our integrated approach with other optimization techniques must target energy consumption optimization in ad hoc network.

In our next study, and with the same objective of optimizing scarce and critical resources of ad hoc network, we will introduce other optimization techniques to reduce energy consumption.

References

[1] Varun Verma, Sumi Helal, Nitin Desai and Choonhwa Lee. Konark : A service discovery and delivery protocol for ad-hoc networks. In WCNC : In Proceedings of the Third IEEE Conference on Wireless Communication Networks , New Orleans, March 2003.

[2] Outay, F. and Veque, V. and Bouallegue, R. Bloom-Filter Based Combined Service and Route Discovery for Mobile Ad Hoc Networks. Proceedings of the The 2007 International Conference on Intelligent Pervasive Computing, pages 188–193, 2007.

[3] Ratsimor, O. and Chakraborty, D. and Joshi, A. and Finin, T. Allia: Alliance-based service discovery for ad-hoc environments. Proceedings of the 2nd international workshop on Mobile commerce, pages 9, 2002.

[4] G. Blair M. Storey and A. Friday. Mare . Resource discovery and configuration in ad hoc networks. in Mobile Networks and Applications Journal, page 377387, 2002.

[5] Liu, J. and Sohraby, K. and Zhang, Q. and Li, B. and Zhu, W. Resource discovery in mobile ad hoc networks. The Handbook of Ad Hoc Wireless Networks, CRC Press, New York, 2003.

[6] Rajeev Koodli and Charles E. Perkins. Service discovery in on-demand ad hoc networks. Internet Engineering Task Force (IETF) Internet-Draft, October, 2002.

[7] Kozat, U.C. and Tassiulas, L. Network layer support for service discovery in mobile ad hoc networks. IEEE INFOCOM,volume 3, pages 1965–1975, 2003

[8] Jodra, JL and Vara, M. and Cabero, JM and Bagazgoitia, J. Service discovery mechanism over OLSR for mobile ad-hoc networks. Advanced Information Networking and Applications, 2006. AINA 2006. 20th International Conference on, volume 2, 2006.

[9] Ververidis, CN and Polyzos, GC. Extended ZRP: a routing layer based service discovery protocol for mobile ad hoc networks. Mobile and Ubiquitous Systems: Networking and Services, 2005. MobiQuitous 2005. The Second Annual International Conference on, pages 65–72, 2005.

[10] Kopena, J. and Sultanik, E. and Naik, G. and Howley, I. and Peysakhov, M. and Cicirello, VA and Kam, M. and Regli, W. Service-based computing on MANETs: enabling dynamic interoperability of first responders. IEEE Intelligent Systems, pages 17–25, 20, 2005.

[11] Obaid, A. and Khir, A. and Mili, H. and Laforest, L. A Routing Based Service Discovery Protocol for Ad hoc Networks. ICNS07: Proceedings of the Third International Conference on Networking and Services, 2007.

[12] Bloom, B.H. Space/time trade-offs in hash coding with allowable errors, Communications of the ACM, 13, pages 422–426, 1970

[13] Sailhan, F. and Issarny, V. Scalable service discovery for MANET. Third IEEE International Conference on Pervasive Computing and Communications, 2005. PerCom 2005, pages 235–244, 2005.

[14] Barr, R. and Haas, ZJ JiST/SWANS, Wireless Networks Laboratory, Cornell University. http://jist. ece. cornell. edu, March 2005.

[15] M. R. Pearlman Z. J. Haas and P. Samar. The Zone Routing Protocol (zrp) for ad hoc networks. Internet draft 4, 2002. Cornell University New York, USA, 2002.

[16] Clausen, T. and Jacquet, P. RFC3626: Optimized Link State Routing Protocol (OLSR). RFC Editor United States, 2003.

[17] Rivest, R. RFC1321: The MD5 message-digest algorithm, RFC Editor United States, 1992.

[18] Flathagen, J. and Øvsthus, K. Service Discovery using OLSR and Bloom Filters. 4th OLSR Interop & Workshop, Ottawa, Canada, 2008.