Journal of Universal Computer Science, vol. 22, no. 7 (2016), 896-920
submitted: 11/11/15, accepted: 20/6/16, appeared: 1/7/16 © J.UCS

A Semantically Enhanced Service Discovery for MANET

Unai Aguilera
(DeustoTech-Deusto Foundation, University of Deusto
Av. Universidades, 24, 48007, Bilbao, Spain
unai.aguilera@deusto.es)

Diego López-de-Ipiña
(DeustoTech-Deusto Foundation, University of Deusto
Av. Universidades, 24, 48007, Bilbao, Spain
dipina@deusto.es)

Abstract: Service discovery is a fundamental aspect of many services oriented applications; however, it cannot be directly applied to mobile ad hoc networks (MANET) due to their dynamic nature. The lack of central manager nodes results in the need of specific discovery solutions for this kind of networks. In this paper, we present a solution for the discovery of services in MANETs that is based on the classification of service's parameters according to a shared domain ontology. Our proposed discovery protocol encompasses two main processes: dissemination and search. The search process defines a *pruning* mechanism that enables to detect if search messages must be propagated or not, reducing the number of messages communicated through the network. The dissemination and search processes are integrated within the route management mechanism that enables to reduce the number of propagated messages.

We have implemented and tested our discovery solution using the ns-2 network simulator. Experiments have been performed using rigorously constructed scenarios for testing the characteristics of the proposed solution. The obtained results show how the inclusion of the ontology not only increments the expressiveness of the search and discovery process, but also enables to reduce the number of propagated messages due to the grouping and *pruning* processes.

Key Words: service discovery, semantic, protocol, manet
Category: C.2.0, C.2.1, C.2.2, C.2.4

1 Introduction

Mobile Ad hoc Network - MANET are a type of wireless networks that are based on the propagation of messages via multiple hops. The Latin phrase *ad hoc*, which literally means *for this purpose*, refers to the fact that such networks lack a predefined communication infrastructure. Message transmission in a mobile ad hoc network occurs in an unplanned manner and in response to the device distribution that exist at a particular moment of time. These type of networks do not contain some predefined participants configure to manage the communication tasks and, therefore, the communication is performed as a distributed task amongst all network participants [Murthy and Manoj(2004)].

In this work, we propose a discovery protocol for mobile ad hoc networks whose main feature is to allow locating services based on the type of their input and output parameters, which are categorized according to an ontology of concepts. The proposed solution enables network nodes to determine which input / output parameters are provided by different services offered on the network and, in addition, which are the compatibility relations that can be established among them. In this work, we propose the usage of I/O parameter type information due to the following reasons. First, propagating descriptions about services' parameters is more effective for service discovery when the user is trying to solve other complex tasks that need to establish a functional relationship among services (e.g. service composition) [Aguilera and López-de Ipiña(2014), Novotny et al.(2015)]. Second, when the information about services is disseminated nodes only need to share data about the different used types. Propagating I/O types of the provided services enables to specify multiple descriptions reusing the same data types, which reduces the quantity of disseminated information.

On the other hand, discovery based on service type or identifier requires all nodes to have knowledge about the different services that can exist in the ad hoc network. However, our usage of a concept ontology to categorise I/O parameters of the available services enables to apply some improvements to the data dissemination and service search processes and, consequently, reducing the number of messages required to perform the two processes.

The rest of the paper is organized as follows: Section 2 contains a depth analysis of the current state of the art in service discovery protocols for MANETs. Section 3 introduces the particularities of the discovery algorithm proposed by this paper. Section 4 summarizes the results of the performed evaluation, while Section 5 concludes the paper.

2 Background and related work

The problem of service discovery in mobile ad hoc networks has been the subject of numerous research works, which have tried to solve the problems of adapting the traditional solutions for infrastructured networks to this type of networks.

In the literature, the existing discovery proposals for MANETs can be classified according to the different elements of the discovery process, the services and the applied techniques [Ververidis and Polyzos(2008)]. According to this classification there are directory-based or directory-less solutions. In the first case, directories are specially designated nodes of the network where providers can register their services and customers search for descriptions. In a MANET it is necessary a mechanism to solve the selection of directories that must be capable of dynamically respond to the changes in the network topology [Zhu et al.(2003), Kozat and Tassiulas(2004), Sailhan and Issarny(2005), Seada and Helmy(2004), Pirr et al.(2012)].

Unlike the solutions based on the use of directories, there also exist proposals enabling to locate services without the need to register their description in some special nodes. This approach avoids to centralize the information in some parts of the network [Nidd(2001), Ratsimor et al.(2002), Chakraborty et al.(2006), Gao et al.(2006), Ververidis and Polyzos(2009)].

Regarding the usage of directories there is no clear consensus to decide whether one approach is better than the other is. The main reason is that there are many factors to consider when evaluating a protocol for MANET network: mobility of nodes, density of servers and customers, the frequency of updates, etc. Therefore, in the case of a network with high mobility of nodes it may be preferable an approach without directories to avoid the problems associated with performing a constant update. In addition, it is also desirable to reduce the possibility of containing obsolete information. However, if the number of requests increases on the same network, service directories can be added to start centralizing requests to some nodes and to reduce the number of messages transmitted across the network [Ververidis and Polyzos(2008)]. Despite these issues, the use of an approach without directories can be better than their usage. [Engelstad et al.(2005)].

Another relevant aspect of the existing solutions is the technology used to describe the services, which determines the expressiveness of the search requests and limits the ability to obtain adequate results. The use of universal identifiers is suitable for those mechanisms with a reduced communication capacity because the size of the identifiers may be limited to a few *bytes* [Ververidis and Polyzos(2009), Arias-Torres and García-Macías(2005), Outay et al.(2010)]. On the other hand, an approach that allows greater expressiveness when performing service discovery is the usage of *key-value* pairs to describe the service attributes [Serhani and Gadallah(2010), Park et al.(2013)]. However, the usage of *key-pair* values is very limited; the syntactic search based on string comparison or regular expressions is not appropriate in most cases because it only allows the use of keywords combined with a system of classification. Often, this type of search does not return any result because it is not able to recognize the similarities and differences between, otherwise, semantically related concepts [Paolucci et al.(2002), Aguilera et al.(2007)].

Due to the limitations introduced above, there are solutions that use the possibilities offered by semantic technologies, e.g as OWL [W3C OWL Working Group(2009)], for the description of services and search queries. Ontologies are usually applied to classify the service and match them with the searches. [Ruta et al.(2010), Said and Maho(2008), Islam and Shaikh(2012), Chakraborty et al.(2006), Klein et al.(2003)].

Regarding the technology used for the service description, it has been found that the majority of proposals do not consider this aspect. Only some of them provide specific information about the usage of a UUID based description or the introduction of semantics. However, it is clear that the integration of semantic technology in the service discovery process can provide some benefits. This is not only because the usage of semantics allows a greater expression, but also because using relationships between concepts enables to apply a number of improvements in the discovery process aimed at the reduction of messages sent during the processes of publication and search services, as it will be explained in this paper.

On the other hand, the nodes of the ad hoc network can change their relative positions during the lifetime of the network; this means that the discovered services may not be longer accessible. There exist different approaches addressing the dynamism of the network. One solution is that customers can carry out a regular process that continually sends search requests *(polling)* to check if a discovered service remains usable or it is no longer accesible [Varshavsky et al.(2005)]. Another option is the usage of *notifications*, which are sent by service providers to indicate that a service is no longer available or has changed its state. The use of notifications is less efficient taking into account the number of required messages that the periodic polling but, on the contrary, it enables to obtain a more immediate response to those changes occurring in the network [Dabrowski et al.(2002)].

Finally, the maintenance of all network related data, e.g. directory clustering or other structures (e.g overlay networks), requires a procedure to update the information which continuously changes due to changes in the topology of the network [Klein et al.(2003), Seno et al.(2007), Artail et al.(2007), Singh et al.(2015)].

According to [Mohan et al.], where it is performed a comparison between proactive and reactive approach to the dissemination of information, proactive solutions have better performance when a more customers to service providers and also the number of search requests is not very high. In this case, the cost of maintaining the information about the services offered may result in a decrease in search latency. Our proposal for the discovery protocol services uses a proactive approach for the dissemination of information on services in the MANET.

The main characteristics of our solution, with respect to the state of the art, are the following:

- Directory usage. The proposed solution does not use directories to centralize the service description during the search. On the other hand, those nodes providing services disseminate their information through the network. This avoid the need to maintain the central directories and reduces the points of failure by disseminating the information through the whole network.

- Service description. In our proposal, services are described by the type of its input and output parameters, which are categorized using a concept ontology. The search process uses the generalization/specification relationships provided by the ontology to locate the compatible services. To our knowledge, this is a novel aspect introduced by this protocol that has not been previously applied to the service discovery problem.

– Protocol integration. The discovery and routing processes are integrated in a way that the dissemination and search process trigger the creation of communication routes among the nodes. This effective allows reducing the number of propagated messages. In addition, this reduction is also achieved with the introduction a new pruning mechanism during the search process. [Vara and Campo (2015)].

3 Ontology based discovery protocol for MANET

The proposed service discovery protocol for MANET networks is divided into three main functions: the *Dissemination* layer, the *Service Discovery* and *Route Management* layers. The purpose of the parameter propagation layer is to allow network nodes be aware of the descriptions of the available services' input/output parameters. On the other hand, the task of the *Service Discovery* is to enable the location of those services that meet the specified requirements. Finally, *Route Management* layer provides the mechanism by which, during the service discovery process, routes are established and managed within the network, allowing the communication between clients and service providers.

3.1 Dissemination

The dissemination process begins in those nodes that provide services. By performing successive hops through the ad hoc network, the information about the services is propagated to other nodes of the network. The dissemination using update messages is generated in response to events that occur in the network topology when network changes. The information propagated by a node is broadcasted hop by hop through the MANET to a certain maximum distance that depends on the algorithm configuration. Figure 1 represents the dissemination process.

Each network node maintains a table, called the *Parameter Table*, which contains information about the types of the input/output parameters that have been disseminated by nearby nodes. Using the contents of this table it is possible to know how far, in hops from the current node, a parameter belonging to a given type is located. The table maintains a numeric value for each parameter type that is called the *estimated distance* and is represented as d_e. This value is used during the search process to determine, depending on the distance in hops that the parameter is located from the current node, if a search message should continue its propagation across the network.

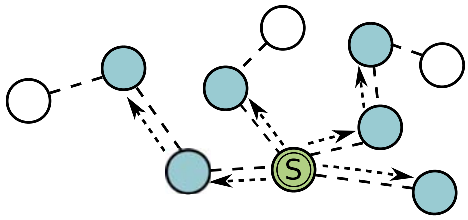

Figure 1: Dissemination of service parameters provided by a node

The higher the indicator value d_e is for a particular parameter the nearer the current node is to a node, or nodes, that provide a service with this type of parameter. The value of the *estimated distance* for a parameter decreases by one unit each time the information about the parameter is propagated one hop across the network. Therefore, those parameters that belong to a service provided by a node have the maximum possible value for its *estimated distance* $d_e = D_d$, in that node's table. This maximum distance is equal to the maximum dissemination distance. Nodes located at successively larger distances will contain a decreasing value of d_e associated with the propagated parameter type.

The dissemination process stops, for a specific parameter type, when the value for its *estimated distance* reaches 1 in a node during the propagation. When this situation occurs that node no longer performs any propagation to its neighbours. Each type of parameter is managed independently during the dissemination process; it is possible, therefore, that some parameters are no longer propagated after reaching the maximum distance, while others will continue the dissemination process.

3.2 Parameter Table

Each node n_i of the network maintains a table P_i containing the *estimated distance* to each one of the parameter types known by the node. The parameter table is defined as a function

$$P_i : O \rightarrow E \tag{1}$$

that relates types of parameters according to an ontology O with a set of *estimated distance* values E. Specifically, each parameter table maintains a set of entries E_p defined as

$$E_p = \{(p, L) \mid p \in O\} \tag{2}$$

where p is the type of the parameter and $L = \{\{e_1, e_2, \ldots, e_k\} \mid e \in E\}$ is a finite set of elements, with each element containing an *estimated distance* of a parameter, which is defined as

$$E = \{(d_e, n) \mid 1 \leq d_e \leq D_d,\ n \in N\} \tag{3}$$

being d_e the *estimated distance* and n the identifier of the neighbour that provided the information about the distance parameter p to the current node n_i.

Because the list L may contain more than one element, we define the *effective value* $d_{eff}(p_i)$ for each entry of parameter type p_i in the table as the item of the list L_i with the largest value for *estimated distance*.

$$d_{eff}(p_i) = e_j \in L_i \mid \forall e_i \in L_i : d_{ej} \geq d_{ei} \tag{4}$$

The effective distance indicator d_{eff} is the value that is actually propagated for each parameter type when the update messages are sent to the neighbour nodes. In a list L there cannot be more than one element e_i with the same value of n. That is, a node does not contain, for a given parameter p_i two different values of *estimated distance* that came from the same neighbour. Therefore, if a node has received an *estimated distance* value from a neighbour and receives a new one from the same node, the new value is considered a replacement for the old one and not a new insertion into the list.

The value n contained in each element of *estimated distance* avoids the back-propagation during dissemination. This problem occurs when a node receives from a neighbour an update message that was initially propagated by itself. Therefore, the update message contains, for each parameter p, the identifier n of the node that initially provide that information.

Finally, each entry of the parameter table stores information about a particular type of parameter. However, the use of an ontology allows the usage of generalization and specification relationships between different concepts. Therefore, when an entry of the parameter table refers to a specific type of the ontology, the entry can represent either a parameter of the exact type or any other type more specific than the one contained in the entry.

3.3 Update messages

Nodes disseminate information about parameter types by sending update messages to their neighbours. The content of these messages depends on the event that has occurred on the MANET network. Therefore, update messages propagated by a node can carry their whole parameter table or only those changes that have occurred since the last update. The use of incremental updates assumes that neighbours succesfully received any previously propagated information about the parameter. The implementation uses ACK messages to assure that neighbours correctly receive the information.

Each update message contains the changes that must be applied by each receiving node to its own parameter table. Changes are specified by two lists of actions: *additions* and *removals*. Whenever a node receives an update message, it processes each received list and performs the actions in order to update its own table.

The mobility of the nodes can result in the information contained in the parameter table becoming obsolete, requiring it to be updated. It should be noted that the dissemination protocol is *proactive*, which means that the contents of the parameter tables are updated in response to changes that occur in the network's topology. Therefore, update messages are caused by detected mobility events.

The propagation of update messages is initially produced as a response to the appearance and disappearance of neighbours. However, receiving an update message and the subsequent update of a node's parameter table can result in the generation of new messages. In addition, the entries of the parameter table can also be updated because of changes in the services provided by the node itself. Therefore, the parameter dissemination occurs after the detection of any of following events: *neighbour detection*, *neighbour disappearance*, *new service registration*, *service removal*.

Initially, the algorithm verifies each parameter contained in the update table to check if there are some removals to process. If removals are pending, the entry of the current node's table, which was previously provided by the node that is propagating the received message, is deleted from the receiver's parameter table. If a removal has occurred and the removed *estimated distance* had a value greater than *one* the node needs to propagate such removal to its neighbour. The reason for this notification is that it is necessary to communicate that the removed information is no longer valid and must be eliminated across the network.

Furthermore, if a removal of an entry from the list of elements for a parameter results in a change in the *estimated distance* indicator, being the new value greater than 1, the new value must be propagated to the node's neighbours. This propagation occurs, for example, when a node has received from its neighbours multiple values for the *estimated distance* of the same parameter type and, after removal of the old value, it produces the propagation of one of the alternative values.

Then, the algorithm processes the new *additions* for each parameter contained in the received message. The algorithm only processes additions that were not initially originated by the same node currently processing the message. For this purpose, all *additions* contain the identifier of the node that started the dissemination for each parameter type. If the current node is not the source of the information for the processed parameter, the information is added to the current node's parameter table and the *estimated distance* value is checked to know if it has been modified as result of the update. If the update does not modify the table, the current node stops the propagation of the message. Conversely, if changes occur and propagation to neighbours is needed, only those *additions* whose *estimated distance* after being decremented are greater than *one* are notified to the node's neighbours.

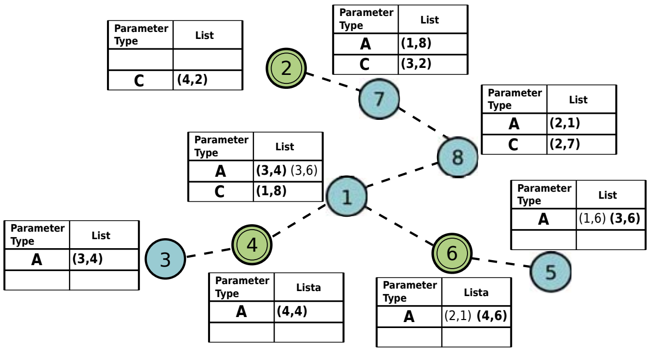

Figure 2: Parameter table status after dissemination

All the *additions* and *removals* obtained from processing an update message are included in a new message that will be sent to the neighbours. The reception of this message by neighbour nodes will result in applying again the previous algorithm. The process, therefore, continues throughout the MANET until the tables converge and no more updates occur.

3.4 Ontology application

By using an ontology it is possible to infer if there exist different relationships between the types of the parameters of two or more services [Paolucci et al.(2002)]. For example, let's suppose we have two parameters P_1 and P_2, according to the information contained in an ontology we can detect the following relationships between concepts:

- **Equality:** this occurs when P_1 and P_2 have the same type, $T(P_1) \equiv T(P_2)$.
- **Subsumption:** it happens when the type of the parameter P_1 is more general than the type of the parameter P_2. That is, P_1 corresponds to a concept that encompasses P_2, represented as $T(P_2) \sqsubseteq T(P_1)$. For example, in an ontology of transport the concept *vehicle* subsumes the concept *car*, since the former is a more general type than the latter.
- **Not related:** occurring when it is not possible to infer any of the previous relationships using the ontology.

The parameter table holds an entry for each known parameter type; however, the entry actually refers to a *parameter group* instead to a single and specific type. Each group represents all those parameters whose types are related by *equal* or *subsumption* relationships, according to the information contained in the used

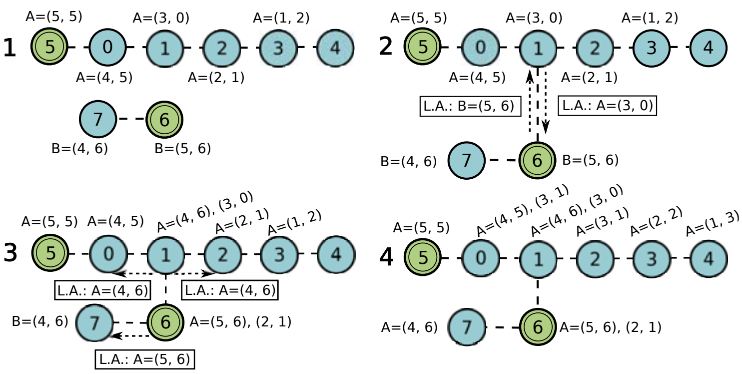

Figure 3: Table status after service parameter dissemination

ontology. Each group is represented by the most general type of those contained in the group, which is called the *group representative type*. The representative type of a group changes every time new parameters of a more general type added to the group. Conversely, removal of a parameter from a group does not change the *group representative type*, which is maintained with is current value until the corresponding entry is permanently deleted from the parameter table. The deletion of an entry occurs when the input becomes empty due to the removal of all the elements contained in the list of associated distances.

Furthermore, a change in the *group representative type* always results in the propagation of the *estimated distance* associated with the group to the neighbours of the current node. The change in the representative type may happen as a result of the reception of an update message from a neighbour or due to the registration of new local services on the node.

Figure 3 shows what happens when dissemination takes into account the information provided by an ontology, which establishes hierarchical relationships among parameter types. Let us suppose that there exists an ontology defining two types of parameters: A and B, being the type of parameter A more general than the type of B.

In state 1, node 5 has disseminated information about the parameters of the services across the ad hoc network, using a maximum distance for dissemination of $D_d = 5$. Then, in state 2, node 6 appears in the neighbourhood of node 1, which causes both nodes to share their full contents of their parameter tables.

In state 3, and as a result of receiving the message propagated by the node 1, node 6 updates the *group representative type* for the parameter. The reason is that this node contained in its table an entry for the parameter B and it has received a message with an entry for the parameter A, whose type is related more general that the registered one.

The change in the representative type causes node 6 to propagate an update message to its neighbours. This propagation results in node 7 also updating the corresponding entry for parameter B in its table, changing the representative type to A. In addition, node 1 will propagate the *estimated distance* to its neighbours because the new received distance from node 6 is higher than the one currently stored. Finally, as shown in network state 4, the reception of the message results in nodes 2, 3 and 4 updating their entries for parameter A.

3.5 Search propagation

Whenever node receives a searc message, it performs a comparison between the parameters contained in the message and the entries in the parameter table. Based on the relationships explained in Section 3.4, the service discovery protocol enables to specify two types of searches:

- **Exact:** this search locates those parameters that are exactly of the type contained in the search. The parameters whose type is more general or more specific than the required type are ignored. For example, if the ontology defines that concept A subsumes B and a node starts the search of a parameter of type A only services whose parameter are of type A will match.
- **Generic:** in this case, the search locates parameters that have the same or a more specific type than the required one. Continuing with the previous example, the search process will locate services not only having the exact type A but also others with a more specific type, for example B.

The discovery process is performed by propagating search messages through the ad hoc network. Each time a search message is received by a node it is compared with the type of the parameters provided by its local services. If a match occurs among the service parameters contained in the search message and any of the services offered by the node it is considered to have located a compatible service.

When a service is found, the node providing the service answers with a response message directed to the node that initiated the search. Propagation of search and response messages across the network originates the creation of communication paths. It is possible that there could exist multiple compatible services in different nodes, meaning that the searching node could receive more than one response.

The usage of I/O parameters does not allow completely differentiating two services. In our proposal, the searching node could receive responses from multiple services compatible with a single search. Only with the I/O parameter description it is not possible to infer the internal semantics of the service and,

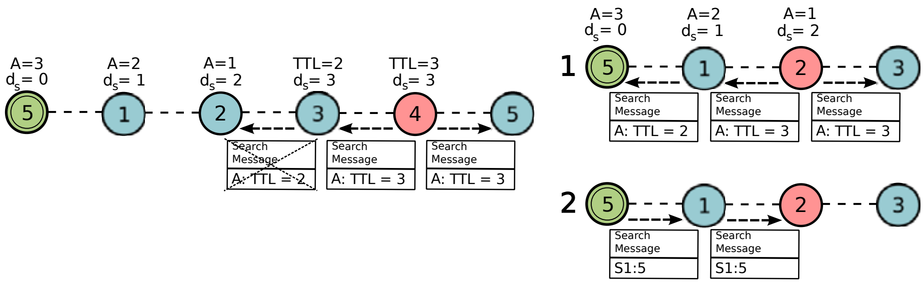

Figure 4: Propagation of search messages and application of the pruning process

therefore, what its real functionality is. However, if a node receives multiple compatible responses, it could apply further filtering using an extended description of the service (name, taxonomy, price, etc.) contained in the node response. This further filtering will could locally performed by the searching node or as a future extension of the proposed search mechanism. However, this paper focuses only on the usage on the I/O parameters of the service.

The inputs and outputs of the discovered service are sent to the node that initiated the search in order to communicate the exact type of the parameters of that service. It should be noted that the searches could contain parameters of a type more general than the one actually provided by the discovered services and, therefore, the exact located type is only known when the node starting the search receives, from the service provider, such information.

During the propagation of search messages, we apply a *pruning* method that aims to reduce the number of sent messages. Search messages are propagated from the node starting the search using broadcast messages. The *pruning* of search messages is intended to prevent the propagation of unnecessary messages on areas where it is known that there are not compatible services. By applying the *pruning* process, nodes only propagate a search message to their neighbours if the TTL associated with a parameter p is equal or greater than the *estimated distance* in hops to that parameter, as reported by the parameter table of the node that is currently processing the message. The TTL for a parameter is decremented at each hop that performs the search message.

In the protocol, searches are maintained active in the network until they are explicitly cancelled. This means that they could locate currently existing compatible services or any other compatible service appearing in the future. Each time a node receives a search message the node stores it in a list of *active searches.* When the network topology changes due to the appearance or disappearance of nodes, active searches are managed to suit the new state of the network.

If the node wants to stop a search process, it must notify this requirement by sending a special message that will perform the cancellation of the active search.

Cancel messages contain which descriptions must be eliminated from an active search. This allows search messages to be cancelled *completely* or *partially*. It should be noted that an active search could group multiple descriptions of services that are propagated together through the network. Therefore, it is possible that the node that initiated the search could only want to cancel a part of the active search. This way, only specific service descriptions will be eliminated from the active search, while the rest of the search message may continue active in the network.

Cancellation messages are processed by the neighbour nodes and re-propagated by those nodes to their neighbours, until they do not produce any change in the node's active searches after being processed.

Moreover, it is possible that nodes that initiated a search could disconnect from their neighbours, either because of their relative motion to other nodes or due to a possible fault in the communication link. When this event occurs, nodes that had received some searches from the disappeared neighbour proceed to completely cancel those searches and to send a notification to their neighbours containing the corresponding cancellation message.

Finally, in order to avoid the erroneous cancellation of searches that may still be correct due to the existence of alternative paths in the network, an active search is only removed if the cancellation message comes from the same neighbour that had previously disseminated the currently removed search. This check avoids to remove messages received from other alternative paths that could still be valid. However, if a node detects that the neighbour that propagated the message has disappeared, it will trigger the removal of the search received from the disappeared neighbour.

3.6 Message routing

Due to the nature of ad hoc networks, there is no prior knowledge of the communication paths existing among nodes. There is a need for a mechanism that allows nodes to discover routes communicating with other nodes [Mekikis et al.(2015)]. The proposed discovery protocol assumes that the communication among nodes is established with the goal of accessing the available services. Therefore, the discovery of communication paths is oriented to sending messages among the node initiating the service search and those nodes providing the located services.

Each network node maintains a *routing table* with information about the available communication paths. This table, which is updated as a result of processing the dissemination and search messages, contains an entry for each discovered path. The routing table R_i of a node is defined as a function

$$R_i : D \rightarrow N_g;\ D,\ \ N_g \subseteq N \tag{5}$$

which relates the set of reachable destinations D with all the neighbours N_g through the message can perform the next hop to reach a particular destination. V is a subset of N, representing the identifiers of all the nodes constituting the network. The routing table contains entries E_e

$$E_e = \{(ID_R, d, n) \mid d,\ n \in N\} \tag{6}$$

where ID_R is the route's unique identifier, d is the destination node and n is the next hop to reach the destination.

3.7 Communication messages

The proposed protocol enables to send two types of messages: *unicast* and *multicast*. Intermediate nodes must contain *multicast* tables in order to send messages following the correct path. When a node has received a message from a searching node, the propagation mechanism also creates a path to communicate with the sending node. However, the node starting a search cannot communicate with the nodes providing services until it receives a response message from them, which creates the inverse communication routes.

Communication messages are transmitted hop by hop only through the neighbours listed in the routing table of each node. Because all the neighbours of a node can receive any message, the *multicast* message contains, at each hop, the list of nodes that must accept it. Propagating neighbours are obtained from the current node's routing table. In addition, during the reception of a message, neighbours that are not contained in the next-hop list will dismiss the message.

The disappearance of a neighbour produces that some previously communication paths cannot be longer used. When a node detects that a neighbour has disappeared it removes, from its routing table, those entries that used the neighbour as the next hop to reach a remote node. After the removal of a route, the node sends to its neighbours a *removed route message* containing a list of identifiers with all the routes that are no longer valid.

4 Evaluation

The discovery protocol has been implemented using the *ns-2* network simulator. Simulation scenarios have been defined using the parameters showed in Table 1.

According to [Kurkowski et al.(2007)], the size of simulation area, the maximum transmission distance and the number of nodes used in our simulations allow the scenario to have the following features: a) ANP (Average Network Partition) $\leq 5\%$, which measures the percentage of nodes in the network that, in average, are disconnected at any given time. Selecting a low partition value for the network guarantees that the protocol is tested in a scenario in which the

Simulation parameter	**Value**
Nodes	100
Simulation area	700 m x 700 m
Mobility model	Random waypoint
Node speed	Uniform distribution [0, 5] m/s
Pause time	50, 100 s
Propagation model	2-Ray Ground
MAC protocol	IEEE 802.11
Transmission range	100 m
Transmission range	11 Mb/s
Max. UDP size	1500 bytes
Beacon period	2000 s
Max. dissemination distance D_d	10 hops
Search TTL	10 hops
Search frequency	0, 2 searches/s
Simultaneous searches	5 searches

Table 1: Parameters used for the simulation of the proposed discovery protocol

network topology enables to locate almost all available services. b) ASP (Average Shortest Path) = 4.15 hops, representing the average shortest distance that exists between two network nodes. This configuration enables to test the proposed discovery protocol in a network where the routing feature is actually used.

As shown in Table 1, the maximum dissemination distance D_D and the TTL for every search have been configure to a value of 10 hops. Considering that both values are much larger than the ASP for the used scenarios, this value makes possible to disseminate search messages covering the whole network, ensuring the possibility of locating almost all services existing in the network and, therefore, testing if the solution behaves appropriately.

In addition, during the experiments we have used a test ontology containing concepts that do not belong to any specific domain. We have only represented generalization and specialization relationship among different concepts. The types of those input and output parameters services used in the experiments are randomly assigned to a concept of the ontology and a variable number of hierarchy relationships among those parameters are also randomly created. This approach enables to test the proposed discovery protocol in a general case without the need of creating a set of services for a particular domain.

Finally, the simulations have been repeated 10 times for each experiment configuration, and then we have calculated the mean values for each configuration in order to show how the discovery protocol behaves in each scenario. We have selected this repetition number because, according to some previous tests, it produced a low variance for the results of each experiment.

4.1 Reduction in the number of dissemination messages

The goal of these experiments was to evaluate the reduction in the number of dissemination messages because of the existence of input/output parameter in the network, which are related according to an ontology. As proposed in this paper, the usage of relations between concepts, either equality or subsumption, enables to group the disseminated information. This grouping must have a direct impact on reducing the number of dissemination messages that are exchanged between nodes in the MANET network.

In the first experiment, we have used parameters, from different services, which are not related through any ontology relations. Therefore, the dissemination protocol will not be able to apply any grouping. Then, these results have been compared to another experiment in which we have distributed the same number of services but, in this case, they are constructed by selecting parameters that are related according to a randomly generated ontology. This way we can compare a dissemination protocol that does not take into account the relationships among parameters with the benefits that the use of semantic relationships introduces.

We have conducted a simulation in which a variable number of services (2 to 20) has been distributed randomly on different nodes in the ad hoc network. Each one of the distributed services has six input/output parameters. During the simulation, we have started different searches of services with parameters randomly selected from all the available ones. Each search is initiated and maintained active for 10 seconds until its cancellation. Five nodes are randomly selected to start their searches simultaneously. This process has been repeated every 5 seconds during a total of 100 seconds of the simulation.

Results of these experiments are shown in Figure 5. Each experiment has been repeated for two different pause times of the mobility of nodes (50 and 100 seconds). The figure shows that the number of messages is significantly lower when relationships are used according to an ontology, thanks to the parameter grouping proposed in this paper.

The figure demonstrates, therefore, that the categorization of the types of parameters not only has the advantage of allowing a greater expressiveness, since it allows a greater richness in searches, but it also reduces, by parameter grouping, the number of messages required to perform the dissemination. When there is no relationship among the disseminated parameters, the number of messages directly increases with the number of services and its parameters.

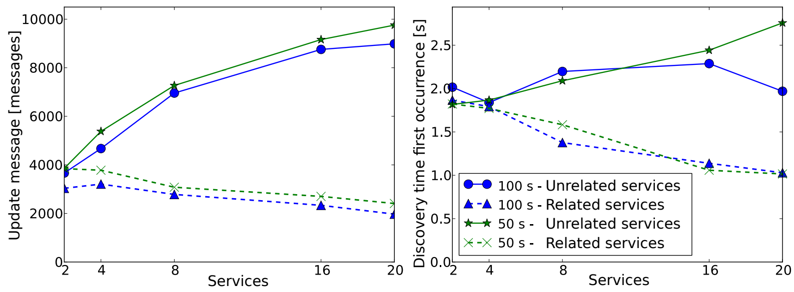

Figure 5: Effect in the number of update messages for the discovery of the first occurrence of a compatible service using a concept ontology

Furthermore, in the same figure we show the average time needed to find an occurrence of a service reduces as the number of services in the network increases. As expected, this time decreases in the case that the number of services that are compatible with a given search increases. The reason is that the number of hierarchically related parameters is greater in each configuration of the experiment; the more compatible services are available the less time required to find a service among multiple compatible ones.

4.2 Search message pruning

The aim of this experiment was to show what effect has the *pruning* process when applied to the search messages. The experiment was performed by randomly selecting 30 nodes in the ad hoc network and deploying services with six input/output parameters each one.

The *pruning* process has been simulated by performing searches aimed at locating services whose parameters does not exist in the network. The percentage of these invalid searches has been increased to see how the application of the pruning rules affects several aspects of the protocol. The experiment was again evaluated for two different pause times of 50 and 100 seconds.

Figure 6 shows the average percentage of discovered services during each configuration. In the figure, invalid searches are expressed as a *ratio* between *0.0* and *1.0*. As shown, the percentage of service discovery is not perfect due to the variations in the connectivity of the MANET. According to the characteristics of the scenario used in our simulation, there is always a *5 %* of nodes that can be disconnected at any time. Therefore, it is possible that either the node initiating the search or the node that provides the required services do not have routes to communicate between them. Furthermore, it is possible that the path between two nodes can be greater than the TTL used in the experiment. This means that some messages will not be able to reach all nodes with compatible services,

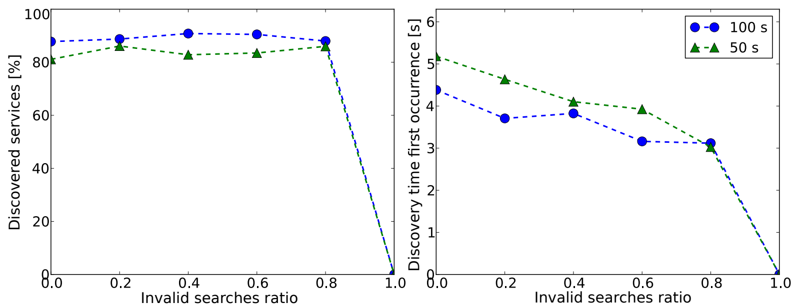

Figure 6: Effect of message pruning on the average discovered services and on the average discovery time

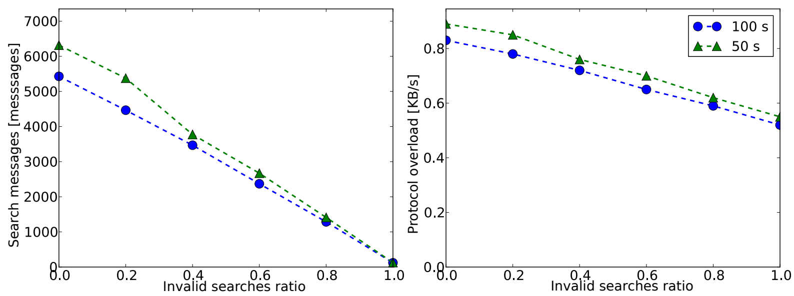

Figure 7: Effect of message pruning on the number of sent message and on the overload introduced by the discovery protocol

therefore, reducing number of located services. In addition, when the ratio of invalid services reaches *1.0*, the percentage of discovered services decays to zero because there are no services compatible with the searches.

Figure 6 also shows the average time required by the service discovery protocol, considering not only the time to locate the service but also the time required to receive the response message from the node that provides the discovered service. As the figure shows, the time is reduced due to the decrease in the number of search messages as a result of the *pruning* process. Although the number of existing services in the network is the same in each case, the search messages are discarded more frequently when *ratio* of invalid searches is increased. Therefore, there is less congestion in the wireless medium of the ad hoc network, which translates into a lower probability of collision between messages. Meanwhile, the decreased number of message collisions results in fewer lost messages and reduces the need to perform retries to send the messages.

Furthermore, the graph shown in Figure 7 demonstrates the reduction in search time due to the decrease in the number of messages explained above. In this figure, we represent the number of messages sent during the experiment and, as can be seen, this number gradually decreases as more non-existent parameters are searched, thanks to the application of *pruning*. Message *pruning* enables to discard certain messages because intermediate nodes have evidence that such messages will never reach compatible services.

Figure 7 also shows the overhead introduced by the dissemination and search messages in the process. This overhead, measured in average *KB/s* transmitted by each node, is obtained by calculating the total size of all messages related to the dissemination and search processes and dividing it by the number of nodes. The introduction of the *pruning* mechanism reduces the overhead due to message traffic by removing unnecessary searches.

Finally, the usage of the proposed discovery protocol introduces an overhead in the information provided by each node. The selection of a proactive mechanism for information dissemination makes nodes to propagate other messages in response to changes in the ad hoc network. This can be seen in Figure 7 when the *ratio* of invalid searches is equal to *1.0*, which is equivalent to a situation when no searches are transmitted across the network. The figure shows that the dissemination process causes about a *75%* of overhead. Approximately, each node transmits *0.6 KB/s* in response to changes in the topology of the ad hoc network.

4.3 Other experiments

In this section, we provide more general experiments oriented to measure the characteristics of the proposed discovery protocol. In this case, we have used a network of 50 nodes as the basic configuration for scenarios, while all the other parameters are maintained as specified in Table 1.

Firstly, we tested how the variation in the number of nodes of the ad hoc network affects the discovery protocol. As can be seen in Figure 8, the number of directly sent messages increases with the number of nodes. Each new node added to the network participates into the dissemination process by increasing the number of *Update Messages*.

Moreover, as the simulation area remains constant for each of the experimental configurations, the greater number of devices produces an increase on the density of nodes in each simulation. Nodes tend to be closer meaning that a path between two pairs of nodes is more likely to occur. Therefore, the percentage of found services initially increases with the number of nodes. Thereafter, the value slightly decreases with the increase in density, because the possibility of disappearance of paths increments with the relative mobility between nodes and collisions between messages.

The right side of the Figure 8 also shows that the overload introduced by the discovery protocol decreases when the number of nodes increments. This

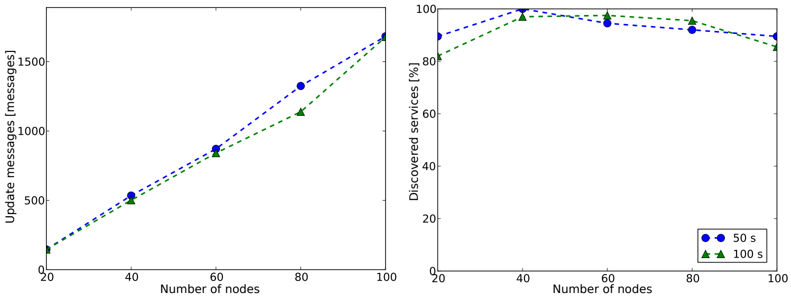

Figure 8: Effect of number of nodes on the number of update messages and on the percentage of discovered services

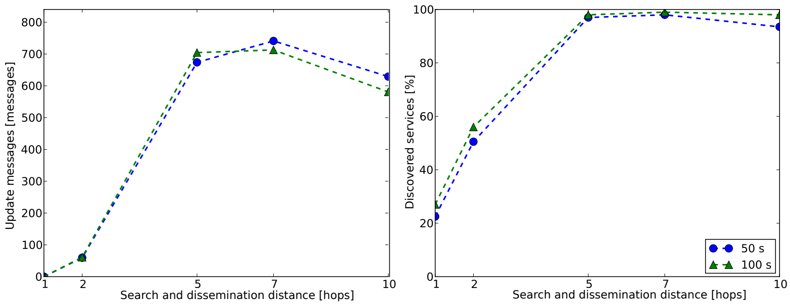

Figure 9: Effect of the dissemination and search distance on the number of sent messages and on the percentage of discovered services

reduction is explained because the nodes are closer to each other, meaning that less paths exist between them and, therefore, a greater clustering of messages occurs during the parameter dissemination.

We have also experimented with the variation in the number of hops during dissemination and search of services. In Figure 9 the number of sent update messages rapidly increases with the dissemination distance. Due to the characteristics of the network, when the distance of dissemination increases to *5-7* hops there is no growth in the number of messages. In addition, it can even be observed a small reduction that is explained because that dissemination is able to completely cover the entire network and the messages are more frequently grouped, reducing the number of total sent messages. Furthermore, the percentage of localized services is directly increased, as expected, with the dissemination and search distance, until almost all available services are located.

On the other hand, Figure 10 shows the effect that the previous experiment has on the service discovery time. As can be seen, the discovery time increases

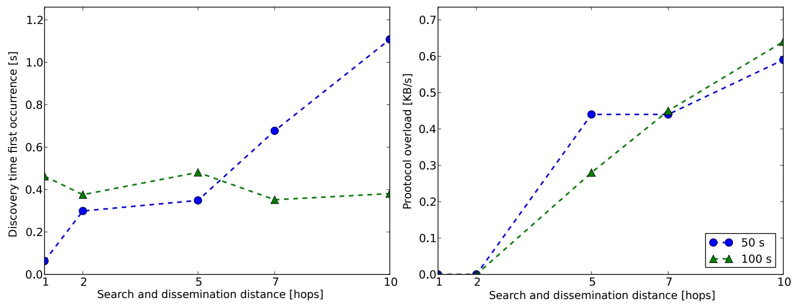

Figure 10: Effect of number the dissemination and search distance on the discovery time and the overload introduced by the discovery protocol

faster when the mobility of nodes is greater (50 s of pause time), because it is necessary to propagate a greater number of update and search messages through the network.

The propagation of a greater number of messages has, as shown by Figure 10, a direct impact on the overhead introduced by the protocol. As most the dissemination process causes most of the overhead, increasing the distance, causes a greater number of sent messages and, therefore, a higher amount of information to propagate between nodes. Selecting a suitable dissemination distance is an aspect that depends on the size of the expected ad hoc network. Therefore, when using small values, relative to the average size of the network, messages will only cover a small part of the topology, reducing the number of located services. On the other hand, a larger number in the dissemination and search radius can result in not applying the *pruning* processes and sending, as a result, a number of messages greater than those actually required to maximize the service discovery.

We performed a final experiment that involved the variation of the search frequency. Figure 11 shows that, as expected, the variation of the search frequency does not affect the number of update messages, as it only changes the number of search messages sent by the nodes. However, the second part of the figure shows that an increase of the search frequency reduces the number of found parameters, due to the increase in the number of messages required to be sent and processed by the nodes. This increment in the number of messages will result in a greater network congestion.

Finally, Figure 12 shows that the time needed to discover services slowly grows due to increment in search messages. This increment is explained because there is not any modification in the distribution of information disseminated in each case. The variation observed in the figure is caused by the larger number of messages sent in the network due to increased searches. Finally, this graph does not show an appreciable change in the protocol overload, as the information

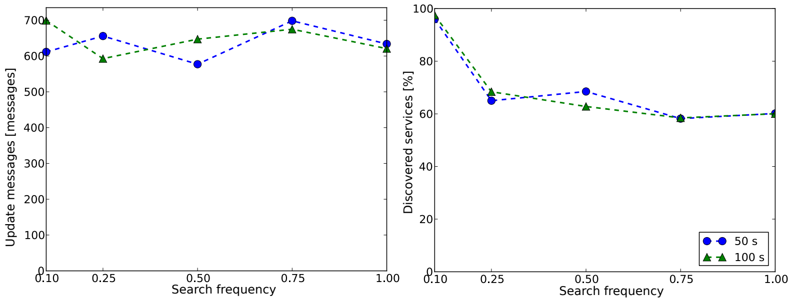

Figure 11: Effect of search frequency on the number of sent messages and on the percentage of discovered services

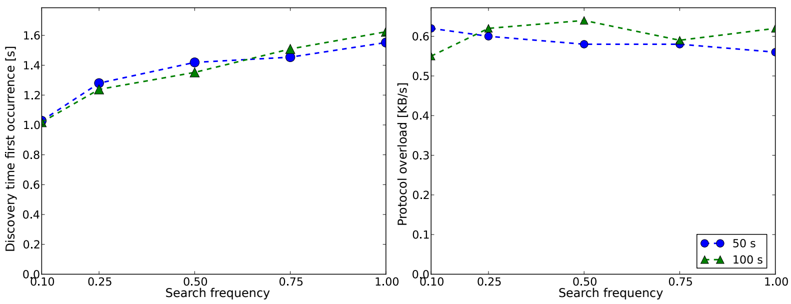

Figure 12: Effect of the search frequency on the discovery time and on the overload introduced by the discovery protocol

transmitted by the nodes is due to the parameter dissemination messages.

5 Conclusions and future work

In this paper, we have proposed a protocol for the discovery of services in mobile ad hoc networks that is semantically enhanced with the usage of the types of the input/output parameters of the services. The protocol is based on the dissemination of information about parameters through the MANET network, using an ontology that categorizes the different types of parameters. The usage of the ontology has two main advantages: firstly, it enables to improve the service discovery process by introducing the possibility of performing accurate and more flexible search parameters; secondly, we have demonstrated that it reduces the number of messages sent during the dissemination process and the subsequent search. The reduction in the number of messages is achieved thanks to the grouping of parameters, according to their type and using the generalization and specification relationships that the ontology provides.

Furthermore, the usage of the disseminated type information enables the application of a *pruning* process to the search messages propagated in the ad hoc network. This process decides if search messages must continue to be propagated through the network, using the contents of the tables maintained by the nodes.

As future work, we will improve our solution to include a mechanism to enable the initial dissemination of the ontology used by the nodes. Currently, the shared ontology must be known *a priori* by the all the nodes participating in the ad hoc network, which may complicate the process of node configuration. Furthermore, it can also be interesting to include a selection mechanism for those situations when multiple compatible services exist for the same search, allowing nodes to discriminate services using other aspects different to the parameter compatibility covered by this research.

Finally, we are also working on the generalization of the solution, which we think that could be applied not only for the dissemination and discovery of services, using their input/output parameters, but also for any resource provided in the MANET. We think that other aspects of the service description, whose information could be categorized according to some ontology, could be also benefit from the dissemination and search mechanisms proposed in this paper.

References

[Aguilera et al.(2007)] Aguilera, U., Abaitua, J., Diaz, J., Bujan, D., Lopez de-Ipiña, D.: "A semantic matching algorithm for discovery in UDDI"; International Conference on Semantic Computing (ICSC 2007); volume 1; 751 – 758; IEEE Computer Society, Irvine, CA, USA, 2007.

[Aguilera and López-de Ipiña(2014)] Aguilera, U., López-de Ipiña, D.: "An architecture for automatic service composition in {MANET} using a distributed service graph"; Future Generation Computer Systems; 34 (2014), 0, 176 – 189; special Section: Distributed Solutions for Ubiquitous Computing and Ambient Intelligence.

[Arias-Torres and García-Macías(2005)] Arias-Torres, D., García-Macías, J.: "Performance analysis of two approaches to service discovery in mobile ad hoc networks"; Advanced Distributed Systems; volume 3563 of Lecture Notes in Computer Science; 368–376; Springer Berlin / Heidelberg, 2005.

[Artail et al.(2007)] Artail, H., Safa, H., Hamze, H., Mershad, K.: "A cluster based service discovery model for mobile ad hoc networks"; Wireless and Mobile Computing, Networking and Communications, 2007. WiMOB 2007. Third IEEE International Conference on; 57–57; IEEE, 2007.

[Chakraborty et al.(2006)] Chakraborty, D., Joshi, A., Yesha, Y., Finin, T.: "Toward distributed service discovery in pervasive computing environments"; Mobile Computing, IEEE Transactions on; 5 (2006), 2, 97–112.

[Dabrowski et al.(2002)] Dabrowski, C., Mills, K., Elder, J.: "Understanding consistency maintenance in service discovery architectures during communication failure"; Proceedings of the 3rd international workshop on Software and performance; WOSP '02; 168–178; 2002.

[Engelstad et al.(2005)] Engelstad, P. E. and Zheng, Y.: "Evaluation of Service Discovery Architectures for Mobile Ad Hoc Networks"; Proceedings of the Second Annual Conference on Wireless On-demand Network Systems and Services (WONS05); 2–15; 2005.

[Gao et al.(2006)] Gao, Z., Wang, L., Yang, M., Yang, X.: "CNPGSDP: an efficient group-based service discovery protocol for MANETs"; Computer Networks; 50 (2006), 16, 3165–3182.

[Islam and Shaikh(2012)] Islam, N., Shaikh, Z. A.: "Towards a robust and scalable semantic service discovery scheme for mobile ad hoc network"; Pak. J. Engg. & Appl. Sci. Vol; 10 (2012), 68–88.

[Klein et al.(2003)] Klein, M., Konig-Ries, B., Obreiter, P.: "Service rings - a semantic overlay for service discovery in ad hoc networks"; Database and Expert Systems Applications, 2003. Proceedings. 14th International Workshop on; 180–185; IEEE, 2003.

[Kozat and Tassiulas(2004)] Kozat, U. C., Tassiulas, L.: "Service discovery in mobile ad hoc networks: an overall perspective on architectural choices and network layer support issues"; Ad Hoc Networks; 2 (2004), 1, 23 – 44.

[Kurkowski et al.(2007)] Kurkowski, S., Navidi, W., Camp, T.: "Constructing MANET simulation scenarios that meet standards"; Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE Internatonal Conference on; 1–9; 2007.

[Mekikis et al.(2015)] Mekikis, P.-V.; Kartsakli, E.; Lalos, A.; Antonopoulos, A.; Alonso, L.; Verikoukis, C. Connectivity of Large-Scale WSNs in Fading Environments under Different Routing Mechanisms. Proceedings of the IEEE ICC 2015, 812; (2015).

[Murthy and Manoj(2004)] Murthy, C. S. R., Manoj, B. S.: Ad Hoc Wireless Networks: Architectures and Protocols; Prentice Hall PTR, 2004.

[Mohan et al.] Mohan, U., Almeroth, K., Belding-Royer, D.: "Scalable Service Discovery in Mobile Ad Hoc Networks"; NETWORKING 2004. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications; 34 (2011), 0, 137–149;

[Nidd(2001)] Nidd, M.: "Service discovery in DEAPspace"; Personal Communications, IEEE; 8 (2001), 4, 39–45.

[Novotny et al.(2015)] Novotny P., Ko B. J., Wolf A. L.: "On-Demand Discovery of Software Service Dependencies in MANETs"; IEEE Transactions on Network and Service Management; vol. 12, no. 2, pp. 278-292, (2015).

[Outay et al.(2010)] Outay, F., Kaisser, F., Veque, V., Bouallegue, R.: "BF-SD-ZRP: a smart integrated scheme for service and route discovery in mobile ad hoc network"; High Performance Computing and Communications (HPCC), 2010 12th IEEE International Conference on; 404–412; IEEE, 2010.

[Paolucci et al.(2002)] Paolucci, M., Kawamura, T., Payne, T., Sycara, K.: "Semantic matching of web services capabilities"; The Semantic Web ISWC 2002; volume 2342 of Lecture Notes in Computer Science; 333–347; Springer Berlin / Heidelberg, 2002.

[Park et al.(2013)] Park J. C., Choi M. S.,Lee B. J.,Kim K. T., Youn H. Y.: "Distributed Semantic Service Discovery for MANET"; Ubiquitous Intelligence and Computing, 2013 IEEE 10th International Conference on and 10th International Conference on Autonomic and Trusted Computing (UIC/ATC); 515-520; 2013.

[Pirr et al.(2012)] Pirr, G., Talia, D., Trunfio, P.: "A DHT-based semantic overlay network for service discovery"; Future Generation Computer Systems; 28 (2012), 4, 689–707.

[Ratsimor et al.(2002)] Ratsimor, O., Chakraborty, D., Joshi, A., Finin, T.: "Allia: Alliance-based service discovery for ad-hoc environments"; Proceedings of the 2nd international workshop on Mobile commerce; 1–9; ACM, 2002.

[Ruta et al.(2010)] Ruta, M., Zacheo, G., Grieco, L. A., Di Noia, T., Boggia, G., Tinelli, E., Camarda, P., Di Sciascio, E.: "Semantic-based resource discovery, composition and substitution in IEEE 802.11 mobile ad hoc networks"; Wireless Networks; 16 (2010), 5, 1223–1251.

[Said and Maho(2008)] Said, R., Maho, Y.: "Toward a platform for service discovery and invocation in disconnected mobile ad hoc networks"; Embedded and Ubiquitous Computing, 2008. EUC'08. IEEE/IFIP International Conference on; 238–244; IEEE, 2008.

[Sailhan and Issarny(2005)] Sailhan, F., Issarny, V.: "Scalable service discovery for MANET"; Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications; 235–244; IEEE Computer Society, 2005.

[Seada and Helmy(2004)] Seada, K., Helmy, A.: "Rendezvous regions: A scalable architecture for service location and data-centric storage in large-scale wireless networks"; Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International; 218; 2004.

[Seno et al.(2007)] Seno, S. A. H., Budiarto, R., Wan, T. C.: "Survey and new approach in service discovery and advertisement for mobile ad hoc networks."; International Journal of Computer Science and Network Security (IJCSNS); 7 (2007), 2, 275–284.

[Serhani and Gadallah(2010)] Serhani, M. A., Gadallah, Y.: "A service discovery protocol for emergency response operations using mobile ad hoc networks"; Telecommunications (AICT), 2010 Sixth Advanced International Conference on; 280–285; IEEE, 2010.

[Singh et al.(2015)] Singh N., Mohan Singh H., Pratap Pal S.: "An Enhanced Service Ring Protocol for Service Discovery in MANET"; International Journal of Computer Applications; vol. 120, issue 24, pp. 31-37 (2015).

[Vara and Campo (2015)] Vara M. I., Campo C.: "Cross-Layer Service Discovery Mechanism for OLSRv2 Mobile Ad Hoc Networks"; Sensors 2015, 15(7), 17621-17648 (2015).

[Varshavsky et al.(2005)] Varshavsky, A., Reid, B., de Lara, E.: "A cross-layer approach to service discovery and selection in MANETs"; Mobile Adhoc and Sensor Systems Conference, 2005. IEEE International Conference on; 8 pp.–466; 2005.

[Ververidis and Polyzos(2008)] Ververidis, C. N., Polyzos, G. C.: "Service discovery for mobile ad hoc networks: A survey of issues and techniques"; IEEE Communications Surveys & Tutorials; 10 (2008), 3, 30–45.

[Ververidis and Polyzos(2009)] Ververidis, C. N., Polyzos, G. C.: "A routing layer based approach for energy efficient service discovery in mobile ad hoc networks"; Wireless Communications and Mobile Computing; 9 (2009), 5, 655–672.

[W3C OWL Working Group(2009)] W3C OWL Working Group: "OWL 2 web ontology language document overview"; (2009).

[Zhu et al.(2003)] Zhu, F., Mutka, M., Ni, L.: "Splendor: A secure, private, and location-aware service discovery protocol supporting mobile services"; Proceedings of the First IEEE International Conference on Pervasive Computing and Communications. (PerCom 2003); 235–242; 2003.