A Routing Based Service Discovery Protocol for Ad hoc Networks

Abdel Obaid, Azeddine Khir, Hafedh Mili, Louise Laforest
LATECE
Université du Québec à Montréal
{obaid.abdel@, khir.azeddine@courrier., hafedh.mili@, laforst.louise@}uqam.ca

Abstract: Ad hoc networks are networks that consist of wireless mobile nodes. They are networks that do not require any pre-established infrastructure. Their ability to quickly and dynamically provide a means of communication makes them an adequate choice for many types of applications both in civil, industrial and military fields. The mobility and the autonomy of ad hoc nodes require a mechanism for service discovery, which allow to provide mobile nodes with information about the services offered in the network. In this paper, we focus on a strategy for services discovery and publication coupled with ad-hoc routing. We propose a new protocol for services discovery called SEDIRAN (Service Discovery and Interaction with Routing protocols in Ad hoc Network), on top of a reactive routing protocol, namely, AODV (Ad hoc On demand Distance Vector). However, we think that our protocol can be easily generalized to other reactive routing protocols.

1 Introduction

Wireless technologies offer new prospects in the field of telecommunications. New advances of wireless communication media and continued improvement of performances of mobile terminals made it possible to take advantages of these technologies in the various fields, and to think of new and innovative applications.

An ad hoc network (also called MANET for Ad hoc Mobile NETwork) is a wireless network wire made up of mobile terminals, connected to each other using wireless links. Each nodes takes part in the constitution of the network and acts as a relay [1] to other terminals. The topology of the network depends on the position of the terminals, their zones of coverage and the signal power. The ad hoc networks provide a mean of communication which can quickly and easily deployed. This makes of them a good choice for applications in various fields: civil (rescue, urgency, organizes, transport, etc.) and military (battlefield communications, etc.).

The transmission range of wireless links in an ad hoc network being limited (from few meters to few hundred meters) (see Figure 1), a node may have to communicate with another nodes using intermediary nodes that act as relay. This requires routings protocols able to route data and maintain links between nodes. Because of the mobility of its nodes, an ad hoc network is subject to topology changes. Therefore, routing protocols in these networks must take into account these topology changes.

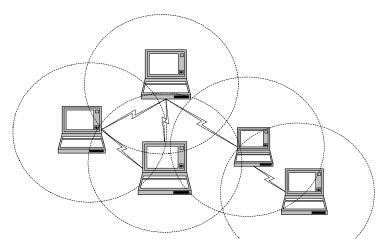

Figure 1: Ad-hoc network example.

The objective of a data network is to offer a means of communication between its terminals in order to allow them to share their data and to use services provided by various servers. In order to allow for an automatic and dynamic integration of services within these networks, several service discovery protocols for the wired networks were proposed such as

Jini[14][15], SLP[12][13], UPnP[22][23][24][25][26] and UDDI[28]. In general, the architectures of these protocols is centralized, where services are stored in a central server directory. This allows to gather information about these services and to organize them using different strategies.

Discovery protocols are important in ad hoc environments. These networks are auto-configurable and consist of mobile terminals which are autonomous and free to move, to leave and join the network to any moment. Service discovery in such an environment requires a decentralized approach, in which a terminal should not depend on another terminal to announce their services. Servers should also be autonomous and able to announce their presence.

The key elements of a discovery protocol are service description, services announcement, services search and service invocation [2]. Service discovery in ad hoc networks is a relatively recent problem which triggered many research activities. There are already some interesting proposals, each one aims at one of the elements of service discovery.

Two basic mechanisms are used to discover services in Ad hoc networks: a) the first one is based on the exchange of request/response messages, in which a user who wants to discover a service, broadcasts a discovery request containing information about the required service. The terminal which offers this service sends the answer, b) In the second mechanism, the service provider periodically sends advertisement messages to announce the availability of its services. A cache memory is used to temporarily record information about the available services. This information is obtained from the advertisement and response messages. A terminal that is interested in the discovery a service first fetches in the cache memory before broadcasting a discovery message. Some protocols use this cache memory to answer discovery requests.

2 Routing and service discovery

2.1 Routing in ad hoc networks

Frequent changes of the topology of ad hoc networks, the low capacity of the wireless links and the battery constraints of the mobiles makes routing protocols designed for wired networks non suitable. It was then necessary to come up with other protocols more adapted to these characteristics while preserving a good performance level.

Various routing protocols try to solve the problem. In the reactive approach of routing (ex AODV, DSR[16][17][18][19]), route discovery is triggered upon request. This type of protocols reduces the traffic of control information exchanged between nodes. Proactive protocols (ex OLSR [27]) are based on a periodic exchanges of control messages. They provide a routing information immediately when required. However, channels are overloaded by these exchanges due to the updates of topology.

Reactive routing protocols, such as AODV and DSR provide request messages called RREQ (for Route Request) to trigger route discovery process and a route response message called RREP (for Route Reply).

2.2 Service dicovery in ad hoc networks

In order to allow potential users to discover a service, a service provider can announce it. These users can broadcast of service discovery requests of this. The nature of the ad hoc environment requires a strategy of advertisement and services discovery which minimizes the traffic generated by the discovery protocols. In addition, in presence of the frequent changes in the topology of these networks, these protocols must also adapt and continue to function. The Post-Query protocol ([3][4]) uses several strategies to adapt to the constraints of the environment of the network. A Post-Query strategy is a sequence (P1,Q1),…, (Pr,Qr) of Post-Query protocols carried out in R rounds.

Konark [5] is a protocol for service discovery and service invocation which aims at mobile ecommerce applications (.i.e.. ecommerce using wireless mobile terminals). One uses a peer-to-peer mechanism for service discovery. The description of a service is done in XML and the services are invoked using SOAP (Simple Object Access Protocol) [29]. It uses a register organized in a tree-like form. Service classification is generic at the top level (i.e. the root of the tree) and becomes increasingly specific as we go down the tree. The discovery and the publication can be made on any level of the hierarchy.

GSD (Group-based Discovery Service) [6][7] is a protocol based on the concept of peer-to-peer cache memory and of group membership of services. The discovery of services is based on sending publication messages with the possibility of sending discovery requests. Services are described using OWL (Web Ontology Language) [34] and are classified in several groups based on the class hierarchy of OWL. Each service provider periodically announces the list of its services using publication messages. Service discovery request is composed of a description based on ontology of the required service and optionally the group membership of services. In order to convey a response message, an opposite routing mechanism is used (reverse routing).

The process of service discovery and invocation can trigger the routing process, which delays the discovery and the invocation process, and wastes network bandwidth. A protocol service discovery based on routing messages was proposed in [20] as an extension of a reactive routing protocol. A study of performances of the protocol was done in [21]. Other functionalities were added. Reactive routing protocols (e.g. AODV and DSR) provide a route discovery request called RREQ (RoadREQuest) to launch the route discovery process and a corresponding reply message called RREP (RouteREPly). The process of service discovery uses RREQ and RREP messages of the routing protocol with an extension of messages formats.

Another proposal of services discovery and invocation are described in [9] and [10]. It is based on a multicast routing protocol instead of using a broadcast mechanism.

In [11], they propose a distributed architecture for service discovery which is based on a virtual backbone to locate and record the services. A mesh structure is built which contains the nodes acting as service brokers called SBN (Service Nodes Broker) together with a subset of links connecting them. All the nodes in the network are in the SBN set or at a distance of one from it.

A new system based on Web service technology is proposed in order to allow for a dynamic integration of applications [8]. The objective is to make it possible for each terminal to discover and reach in an autonomous way the services of its proximity. The solution is intended for mobile systems based on the WLANs such as Bluetooth [32] and WiFi [33]. The access point to these WLANs acts as a private register to answer requests and record descriptions of all the services which it offers. This new system is uses web service standards like SOAP [29], WSDL [30] and UDDI [28].

3 SEDIRAN

The centralized approach for service discovery is not suitable for ad hoc environments. In a decentralized approach, since there is no central server, services announce their existences by broadcasting advertisement messages in the network. The interested users who want to discover a service broadcast service discovery requests. The response messages are sent in a unicast mode using routing protocol packets. In the case of a reactive routing, the sending of these answer messages will trigger the routing protocol if the terminal does not have a road towards the service requester. A service discovery protocol in ad hoc networks must thus minimize the number of messages exchanged and avoids the activation of the routing protocol.

In service discovery protocols, one must take into account these parameters by coupling them with the routing protocols and by minimizing the use of terminal and network resources.

In this work we aim at several objectives: a) to establish a strategy of service discovery and advertisement. This requires to come up with a mechanism which enables us to minimize the traffic generated by the protocol and to adapt to the frequent topology changes, b) to establish a strategy for the use of a cache memory and diminish the affect on the performance of the protocol, c) to establish an interaction mechanism with the routing protocols. This requires that one enhances routing tables during discovery in order to avoid the activation of the routing process.

3.1 Ordinary and special services

Ordinary services are services known to all the nodes of the network. An ordinary service is identified by UUID (a 128-bit Universal Unique Identifier). The search for service of this type is done based on this identifier. An example of such a service could be a game (Chess game for example). Ordinary services are stored in a database. As in the case of the Konark, this database is presented in a tree-like structure of service types. Potential users will be able to browse through this tree (for example using a Web browser) in order to look matches using the UUID of the service type.

Special services are services which are not known in advance to the users and who are represented by a descriptive and meaningful text to the users. These services are not stored in any database. These types of services are generally subject to advertisements. Given that these services are not known in advance, potential users must learn their existence. The discovery on the other hand, is done on ordinary services. Each node will be able to have several ordinary services, since these nodes never announce their services, since it is up to the interested users to a send discovery requests. A cache memory manager for the ordinary services and another for the special services are used in order to store information on the services offered on the network.

3.2 Interaction with routing protocols

The nature of the ad-hoc networks makes discovery protocols important so that the users are aware of the existence of services when they join the network Service discovery protocols provides to the users with the location of available services. To facilitate the contact between service providers and users, we propos an interaction between SEDIRAN and the routing protocol at the discovery time. This solution enables us to avoid making too many calls to the route discovery procedures in the case of a reactive routing like DSR and AODV.

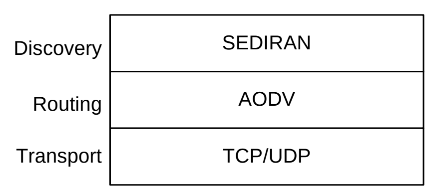

Figure 2: Protocol stack in SEDIRAN**.**

Our protocol extends the routing table (or routing cache) during service discovery. SEDIRAN sits on top of the reactive routing protocol (Figure 2).

3.3 Procédure de publication de services

A server which provides a special service broadcasts a publication message called ADVM (ADVertissement Message) to the nodes of the ad hoc network (Figure 3). This message contains the address of the server which offers the service and its description. Each node traversed by this message stores this information in its cache memory in order to use it later on. The time during which this advertisement is valid is determined by of lease time. This makes it possible to clean up the cache memory when its lease time expires.

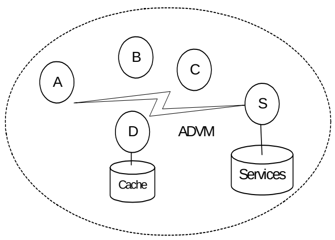

Figure 3: Avertissement message broadcast**.**

3.4 Service discovery procedure

To search for an ordinary service, the user broadcasts a request message called DREQ (DiscoveryREQuest) to all nodes of the network. This message contains a number which indicates the maximum number of hops (Hop-limit) that the message is allowed to do. This number is equal to the number of nodes the message can cross. This mechanism is available in routing protocols such as DSR and AODV. The nodes which offer the of required service type answers the request by sending a message called DREP (Discovery REPly). Figure 4 shows an example of a node (A) that makes a request for the discovery for an ordinary service type and receives a response from several nodes.

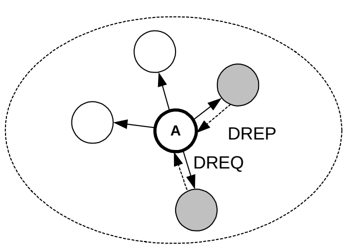

Figure 4: Discovery with a hop limit 1.

The discovery procedure functions in an incremental way. A node which wants to start a discovery, broadcasts a discovery message DREQ with a hop Limit set 1. It receives the answers of the nodes that offer this service. The user will be able to display the result of the discovery. If it is not satisfied or if it wishes to discover other servers which offer the same type of services, another discovery message of DREQ with hop Limit 2 is sent.

Intermediary nodes update their routing table entries towards the requesting node and broadcast the request messages. They store the information contained in the DREP reply message. They also update the routing table entries towards the service provider and add the services discovered in the cache memory of ordinary services. This way, they will not have to perform any service discovery of during for subsequent discovery operations. This process continues until the user decides to stop.

If a user is interested in the discovery of all the services offered in the network, he can proceed in an a similar way. He sends a request for the discovery of all the services to the nodes of the first level and records in its cache, all the services found together with the route. He then sends a request for services offered by the nodes of the second level. Only the servers which offer services different from those discovered at the preceding level answer by giving these services. The process continues until the last level. This strategy makes it possible for the user to record, for each service, the nearest server which provides it since only the server nearest answers it. Figure 5 illustrates this process. Nodes E, F and H answer by giving services different from those discovered in stage (a).

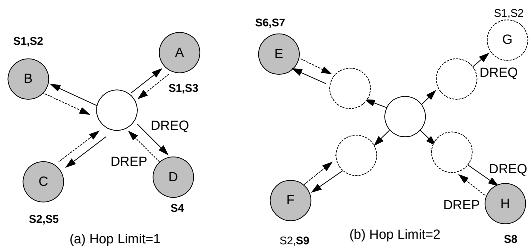

Figure 5: Discovery of available services.

3.5 SEDIRAN packets

SEDIRAN is located at the top of the AODV routing protocol. The discovery message are encapsulated in the routing messages. We also expanded the routing table at the time of services discovery. The sending of the reply messages does not start any route discovery. The recording of a service in the cache is preceded by a recording of the route towards the service supplier in order not to start the routing at the services invocation time.

There are three types of SEDIRAN packets:

- DREQ (Discovery REQuest): contains a request for the discovery of an ordinary service. It specifies the requested service type identifier and the list of services that were already discovered.
- DREP (Discovery REPly) : contains the reply to a discovery request. It also contains the list of identifiers of the services that belong to requested service type together with the provider address.
- ADVM (ADVertisment Message): contains the advertisement of a special service that is provided. Il also contains the description of the service, its identifier and the address of its provider.

3.5.1 A modified RREP packet

We use the RREP message of AODV to encapsulate all the messages of SEDIRAN (DREQ, DREP and ADVM). For have brought some changes to the structure of this packet. Figure 6 shows the original RREP message and figure 7 shows the version used in SEDIRAN

1 … 32

<table>
<tr><td>Type
(8 bits)</td><td>R A
(2 bits)</td><td>...</td><td>Prefix
(5 bits)</td><td>Hop count
(8 bits)</td></tr>
<tr><td colspan="5">Destination IP address</td></tr>
<tr><td colspan="5">Destination sequence number</td></tr>
<tr><td colspan="5">Source IP address</td></tr>
<tr><td colspan="5">Lifetime</td></tr>
</table>

Figure 6: RREP packet in AODV.

1 … 32

<table>
<tr><td>Type
(8 bits)</td><td>R A P S D
(5 bits)</td><td></td><td>Prefix
(5 bits)</td><td>Hop count
(8 bits)</td></tr>
<tr><td colspan="2">Size
(16)</td><td colspan="2">Diameter
(8)</td><td></td></tr>
<tr><td colspan="5">Destination IP address</td></tr>
<tr><td colspan="5">Destination sequence number</td></tr>
<tr><td colspan="5">Source IP address</td></tr>
<tr><td colspan="5">RREP ID</td></tr>
<tr><td colspan="5">Lifetime</td></tr>
<tr><td colspan="5">SEDIRAN Message</td></tr>
</table>

Figure 7: RREP packet in SEDIRAN.

The P field (Publish) indicates that the message is an advertisement (ADVM message). The S field (Search) indicates that the message is a discovery request (DREQ message). The D field (for Discovery) indicates that the message is a response (DREP message) to a discovery request. The size of the message including messages SEDIRAN is indicated in the Size field. Hop Limit is a used in the case of a message DREQ to determine the scope of the discovery. Field RREPID contains an identifier an RREP message. Given that AODV protocol does not broadcast RREP messages, this field is used to broadcast these type of messages. It allows the nodes crossed to know if they already received the message before or not. If they have already received it, they ignore it of it. If not, they re-broadcast it.

3.5.2 Advertisement messages

The ADVM advertisement message (Advertisement Message) is used when the P field of RREP is set to 1. The message is used by a server to announce its special services. It contains the address of the server, its descriptor and the service identifiers and descriptor, and the and the lease period. As indicated previously, this lease makes it possible to delimit the lifespan of the service stored in the cache. The ADVM message is structured as shown in figure 8.

1 … 32

Type (8)	
Source IP address	
SpecialService.ServiceID	
SpecialService.ServiceDesc	
SpecialService.Lifetime	

Figure 8: ADVM packet.

The Type field is set to *ADVM*. The *SourceIPAddr* contains the IP address of the service provider. For each special service, this message contains the UUID of the service type (field *ServiceID*), the service description as a character string (field *ServiceDesc*) and the lease time (filed *LifeTime*).

3.5.3 Discovery request messages

The DREQ message is used to transport service discovery requests. The structure of this message is given in Figure 9. The *Type* file is set to DREQ. The filed Hop Limit contains the maximum number of hops between the requestor and the nodes targeted by the current request. *SourceIPAddr* contains the source address of the request sender.

1 … 32

Type (8 bits)	Hop Limit (8 bits)	ServiceCount (8 bits)	…
Source IP			
ServiceTypeID			
ServiceID[1]			
…			
ServiceID[ServiceCount]			

Figure 9: DREQ packet.

3.5.4 Discovery response messages

This message is used to respond to a service discovery request. It structure is shown in Figure 10. The *Type* field is set to DERP. *ServiceCount* holds the number of ordinary services that were already discovered. *ResponderIPAddr* is the IP address of the node that sent this response (i.e. the service provider). OrigonatorIPAddress contains the IP address of the requesting node (i.e. the destination address of this packet). These fields are followed by the descriptions of the ordinary services that were discovered. Each description consists of two fields: *ServiceID* that holds the service ID and *ServiceTypeID* contains the service type identifier. *LifeTime* holds the maximum time the service can stay in the cache.

1 … 32

Type (8 bits)	…	ServiceCount (8 bits)
Responder IP address		
Source IP address		
ServiceID[1]		
ServiceType[1]		
LifeTime[1]		
…		
ServiceID[ServiceCount]		
ServiceType[ServiceCount]		
LifeTime[ServiceCount]		

Figure 10: DREP packet.

1		32
Type (8 bits)	…	ServiceCount (8 bits)
Responder IP address		
Source IP address		
ServiceID[1]		
ServiceType[1]		
LifeTime[1]		
…		
ServiceID[ServiceCount]		
ServiceType[ServiceCount]		
LifeTime[ServiceCount]		

Figure 10: DREP packet.

3.6 SEDIRAN procedures

A supplier of special services periodically broadcasts its ADVM messages to the network nodes in order to inform them about the existence of his services. This message is passed to the routing protocol in order to encapsulate it within a RREP packet and to broadcast it on the network. When a nodes receives this message, AODV checks if the message was already received using RREP ID field. If it is not the case, it updates the route towards the service provider and re-broadcasts this message. It then passes an ADVM message to SEDIRAN in order to update the cache for special services. This way, the node has the service and the route towards the provider. Later invocations of this service will not require sending any route discovery message. Figure 11 illustrates this exchange.

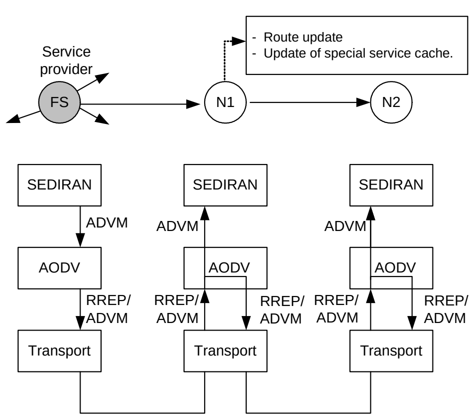

Figure 11: Propagation of ADVM messages.

To discover a service in the network, SEDIRAN generates a DREQ request message. This massage is encapsulated within the RREP message by the routing protocol and broadcast in the network. DREQ Request targets the services offered by the nodes located in a given diameter. Once the request is received by an intermediate node, the routing protocol updates the route towards the requestor in order to prepare the route back of DREP response message. When the request is received by a node belonging to the chosen diameter, the routing protocol updates the route towards the requestor and passes the DREQ message to SEDIRAN (Figure 12), which then looks up the table of local services for the service that corresponds to the request. If it finds some, it generates a DREP reply message.

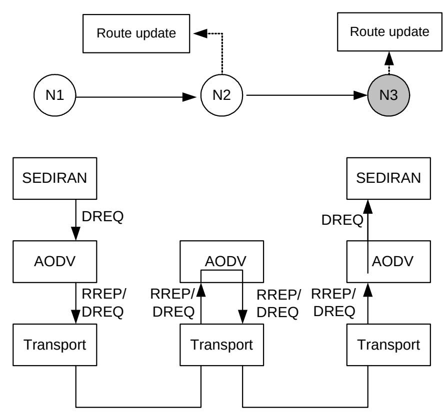

Figure 12: Propagation of DREQ messages.

Once the DREP message is generated by SEDIRAN, it is passed to the routing protocol which encapsulates it in RREP message and sends it to the requestor. When an intermediate node receives this message, AODV updates its route towards the service provider and pass the message to the DREP module in order to update the ordinary service cache. This way, intermediate nodes will know the service and the route towards its provider. Any subsequent invocation of the same service will not require any new route discovery. Once the service requestor receives the message, AODV update its route towards the provider and the pass the DREP message to SEDIRAN which will update its ordinary service cache.

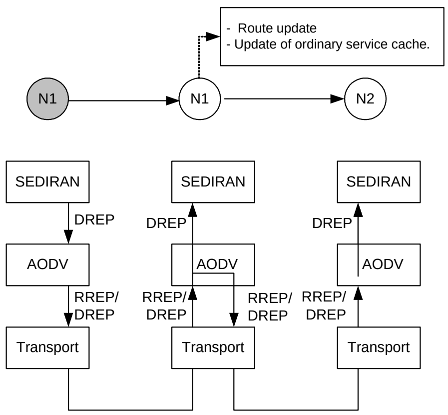

Figure 13: Propagation of DREP messages.

4 Conclusions

We have proposed a new service discovery protocol for ad-hoc networks. This protocol has the following characteristics:

- It is based on a decentralized architecture: Each node in the networks manages its own services and do not depend on any central provider.
- A new strategy for service discovery and advertisement: we distinguish between ordinary services and special services. Only special services use advertisement massages to announce their existence, which minimises the traffic generated on the network.
- Cooperation wiith the routing protocol: Although our proposal users the routing, our protocol does an interaction with the routing protocol, we could create some degree of independence between the discovery task and the routing task using a specific protocol stack.
- Use of a memory cache mechanism: any node in the network can use its cache to fetch services in order to respond to service discovery requests. Only the providers of the requested service respond to requests.

5 Bibliography

[1]. S. Corson, J. Macker, Mobile Ad hoc networking (MANET) : Routing Protocol Performance Issues and Evaluation Considerations. RFC2501, Janvier 1999.

[2]. Rohan Sen, Radu Handorean, Gruia-Catalin Roman et Christopher Gill. Service Oriented Computing Imperatives in Ad Hoc Wireless Settings. Technical Report WU-CSE-2004-05, Washington University, Department of Computer Science, St. Louis, Missouri.

[3]. M. Barbeau, E. Kranakis, Modeling and Performance Analysis of Service Discovery Strategies in Ad Hoc Networks, Proceedings of International Conference on Wireless Networks (ICWN), Las Vegas, Nevada,2003.

[4]. H. Luo and M. Barbeau, Performance Evaluation of Service Discovery Strategies in Ad Hoc Networks, 2nd Annual Communication Networks and Services Research (CNSR) Conference, Fredericton, N.B., Canada, May 2004, pp. 61-68

[5]. Sumi Helal, Nitin Desai, Varun Verma and Choonhwa Lee, Konark – A Service Discovery and Delivery Protocol for Ad-Hoc Networks, In Proceedings of the Third IEEE Conference on Wireless Communication Networks (WCNC), New Orleans, USA, March 2003.

[6]. Dipanjan Chakraborty, Anupam Joshi, Yelena Yesha et Tim Finin. Towards Distributed Service Discovery in Pervasive Computing Environments. IEEE Transactions on Mobile Computing, July 15, 2004

[7]. Dipanjan Chakraborty, Anupam Joshi, Yelen Yesha et Tim Finin.GSD: A Novel Group-based Service Discovery Protocol for MANETS. 4th IEEE Conference on Mobile and Wireless Communications Networks (MWCN), September 09, 2002.

[8]. Robert Steele. A Web Services-based System for Ad Hoc Mobile Application Integration. International Conference on Information Technology: Computers and Communications, Las Vegas, Nevada, April 28 - 30, 2003.

[9]. Liang Cheng, Service Advertisement and Discovery in Mobile Ad hoc Networks. Workshop on Ad hoc Communications and Collaboration in Ubiquitous Computing Environments, in conjunction with the ACM 2002 Conference on Computer Supported Cooperative Work, New Orleans, Louisiana, USA, November 16-20, 2002.

[10]. Liang Cheng et Ivan Marsic, Service Discovery and Invocation for Mobile Ad Hoc Networked Appliances. in Proceedings of the 2nd International Workshop on Networked Appliances (IWNA 2000), New Jersey, USA, November 30 - December 1, 2000.

[11]. Ulas C.Kozat et Leandros Tassiulas. Network Layer Support for Service Discovery in Mobile Ad Hoc Networks, IEEE INFOCOM 2003 - The Conference on Computer Communications, vol. 22, no. 1, Mar 2003

[12]. RFC 2608 :SLP Service Location Protocol, Version 2. http://www.openslp.org/doc/rfc/rfc2608.txt

[13]. RFC 2609 : Service Templates and Service: Schemes http://www.openslp.org/doc/rfc/rfc2609.txt

[14]. Jini™ Architecture Specification. http://wwws.sun.com/software/jini/specs/jini2_0.pdf

[15]. Jini™Technology Core Platform Specification. http://wwws.sun.com/software/jini/specs/core2_0.pdf

[16]. David B. Johnson, David A. Maltz, Josh Broch, DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks. Internet Engineering Task Force (2002).

[17]. C. Perkins, E.Belding-Royer, S.Das, Ad hoc On-Demand Distance Vector (AODV) Routing. RFC3561(Experimental) Juillet 2003.

[18]. Charles E. Perkins, Elizabeth M. Royer, Samir R.Das, Mahesh K. Marina, Performance Comparison of Two On-Demand Routing Protocols for Ad hoc Networks, IEEE Personal Communications, Feb 2001.

[19]. Elizabeth M. Royer, Charles E. Perkins, Multicast Ad hoc On-demand Distance Vector (MAODV) Routing, Internet draft, Juillet 2000.

[20]. C. Perkins and R. Koodli. Service discovery in on-demand ad hoc networks. IETF Internet Draft, Octobre 2002.

[21]. F. Christian and K. Holger. Consistency Challenges of Service Discovery in Mobile Ad Hoc Networks. Octobre 2004.

[22]. Understanding Universal Plug and Play .http://www.upnp.org/resources/whitepapers.asp, White Paper, join 2000.

[23]. UPnP Device Architecture 1.0, Décembre 2003. http://www.upnp.org/resources/documents/CleanUPnPDA101-20031202s.pdf

[24]. Multicast and Unicast UDP HTTP Messages, draft, octobre 2000. http://www.upnp.org/download/draft-goland-http-udp-04.txt.

[25]. Simple Service Discovery Protocol/1.0, draft, octobre 1999. http://www.upnp.org/download/draft_cai_ssdp_v1_03.txt.

[26]. GENA General Event Notification Architecture Base, draft, Septembre 2000. http://www.upnp.org/download/draft-cohen-gena-client-01.txt

[27]. T. Clausen, P. Jacquet, OLSR (Optimized Link State Routing), RFC 3626, octobre 2003.

[28]. L. Clement, A. Hately, C. Riegen et T. Rogers UDDI (Universal Description Discovery & Integration), UDDI Version 3.0.2, UDDI Spec Technical Committee Draft, octobre 2004, http://uddi.org/pubs/uddi_v3.htm.

[29]. SOAP Version 1.2 specification. Juin 2003 http://www.w3.org/TR/soap12

[30]. Web Services Description Language (WSDL) 1.1, W3C, Mars 2001 http://www.w3.org/TR/wsdl

[31]. Extensible Markup Language (XML) 1.1, W3C Recommendation , Février 2004, http://www.w3.org/TR/xml11.

[32]. Bluetooth, Specification Documents https://www.bluetooth.org/spec/

[33]. Wi-Fi http://www.wi-fi.org

[34]. OWL (Web Ontology Language) W3C Recommendation, Février 2004. http://www.w3.org/TR/owl-features/