# Service discovery in ad hoc networks

Mohammed Haddad and Hamamache Kheddouci

*Laboratoire PRISMa - Université Claude Bernard Lyon 1,*
*Bât. 710, 843, Bd. du 11 nov. 1918, 69622 Villeurbanne Cedex*

**Abstract.**
*Various researches treating of the communication protocols try to define a certain organization over the network to improve its general behavior. The concept of service is frequently used as an abstraction of software and hardware resources. Service discovery offers service advertisement and location mechanisms. So, it represents a major challenge in highly dynamic networks.*

*In this paper, we present a new service discovery protocol for ad hoc network. This protocol, first, constructs and maintains an adapted dynamic virtual topology, then, it achieves service discovery on it. We propose a new construction method for the topology. This method is fully distributed and based on the concepts of independent sets and cliques. Also, we propose the introduction of various mechanisms allowing the improvement of service discovery. In addition, our protocol was simulated on NS-2.*

***Key words:*** *ad hoc networks, service discovery, dynamic virtual topologies.*

## 1. Introduction

A challenging research area in computer networks is Ad hoc networks (or MANETs: Multihop Ad hoc NETworks). These are defined to be spontaneous and totally autonomous networks. Nodes can move randomly and so, network connectivity may change frequently.

Various communications protocols such as the routing, the resource sharing, the service discovery, mobility prediction… *etc.* uses the flooding to broadcast or gather information. For this reason, recent works try to optimize these exchanges of data flows. Indeed, various researches treating of the communications protocols try to define a certain organization over the network. Different protocols of routing, resource allocation and management … *etc.* use the concept of virtual dynamic topology to improve the general behavior of the network. Indeed, if the various exchanges of data flows are done via an adapted virtual topology, the total performance of the network may increase considerably.

In addition, the concept of service is used as abstraction of the resources (software or hardware). The service discovery offering of the mechanisms of publication and localization represents a major problem which must be solved to succeed in implementing very dynamic networks such as the ad hoc networks. The majority of the service discovery protocols are based on the Client/Server paradigm. In this last, the Clients can send their requests in a reactive way (Pull model) or can remain, proactively, listening to the advertisements of the generated services (Push model). In these two models, the adopted approach is decentralized.

Salutation [1] and UPnP [2] use this approach. An alternative scheme would imply a central directory which would have the responsibility of index all the services present in the network. Protocols like Jini [3] and SLP [4] specify the presence of such a server. However, these various protocols cannot be directly applied in dynamic environments such as the ad hoc networks. For that, several protocols were proposed: Allia [5], GSD [6] and Konark [7] offer solutions holding into account the peep to peer nature of the ad hoc networks. The protocol of Kozat *et al.* [8] defines a mesh structure based on the concept of dominant sets. The protocol of Lenders *et al.* [9] uses an "electrostatic" modeling of the services in order to illustrate their availability.

We studied the whole of works being based on dynamic virtual topologies. We classified them in two great families. The first gathers the techniques basing on covering nodes which seek to build a virtual support of communication (virtual backbone). The second gathers the techniques seeking to minimize the number of links used in communication.

In this paper, we present a new protocol of service discovery for the mobile ad hoc networks which is based on a "mixed" topology. This topology aims at combining the advantages offered by the two classes of topologies (to minimize the number of participating nodes in topology as well as the number of used links). The protocol thus proposed builds and maintains, initially, an adapted virtual dynamic topology then achieves the service discovery itself via the latter.

## 2. Our Solution

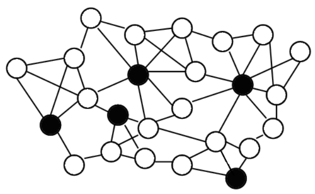
In our approach, we translate the fact of indexing the services to a problem of cover *i.e* to find a subset such as if each one of its nodes knew only the services offered in its neighborhood, the union of collected information would include all the services of the network. From the service discovery point of view, this subset plays the part of a distributed directory on the network. As, we admit as in addition to its property of cover, the induced subgraph of the subset must be connected. Indeed, that is necessary to ensure a lower cost intra-directory communication, to ensure access and exchange of information (gradually between direct neighbors) and to balance the loads in the network.

For that, we propose the adoption of the concept of connected dominating sets (CDS). These nodes will have the responsibility to record and seek the services. Each node of the network will have at least a node of the CDS in its neighborhood. The nodes of the CDS form, between them, a peer to peer network. They maintain local caches which are useful, mainly, to record information concerning the services. The sections which follow present the detailed operation of the protocol.

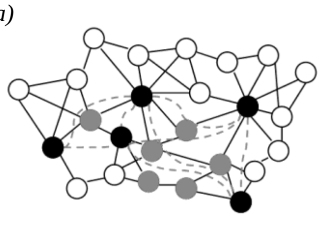
## 2.1. Construction and maintenance of the virtual topology

Dynamic virtual topology consists of a connected dominating set. To build it we choose a rather traditional approach. The latter seeks, initially, to determine a Maximal Independent Set (MIS) then to inter-connect it. However, the principal difficulty in our problem is that this construction must be done by minimizing the number of connectors and in distributed way. Construction is done as follows (see Figure 1):

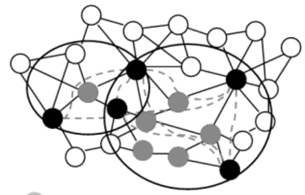
a) Choose a subset which forms a MIS.
b) Form the 3-closure (to add virtual links between any pair of remote nodes within two or three hops one from the other) of the graph modeling the network (but we do this only for the induced subgraph by the MIS nodes).
c) Regroup MIS nodes into cliques in the 3-closure.
d) Find a minimal core of connectors[1] for each clique in the 3-closure.
e) The union of the MIS nodes and the connectors thus selected forms the CDS.



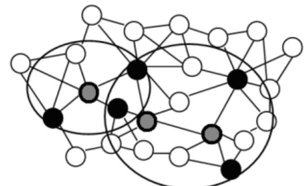
*(a)*



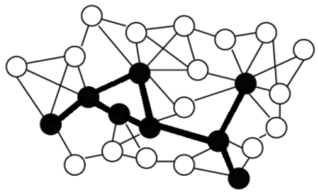
*(b)* Virtual link



*(c)* Intermediate node



*(d)* Elected connector node



*(e)*

**Figure 1: Virtual topology construction.**

To select the MIS, we privilege the nodes which cover the most neighbors. Initially, all the nodes are regarded as potentially independent. A node remains independent only if it has a maximum degree among its independent neighbors (Step a). In the case of equality, we use a rule to decide between the neighbors who are in conflict. The latter can be based on the identifiers of the nodes. In the phase of interconnection of the MIS nodes, we rest on the principle that the distance between two complementary partitions of the MIS is 3 hops at maximum.

In order to reduce the size of the built CDS, we proposed an optimization which reduces the number of connector nodes. Each node not belonging to the MIS is regarded as potentially "connector". We consider the virtual links formed by the 3-closure of the graph modeling the network, in particular those relating to the MIS nodes (Step b). We regroup these nodes into cliques in the 3-closure. It is important to note that any distributed algorithm of maximal cliques determination can be used (Step c). At this stage, it is a question of finding a core of connectors for each clique in the 3-closure (Step d). This last must guarantee the connexity of the nodes of cliques in the 3-closure. Only one MIS node by clique is elected to select the intermediate nodes which will join the CDS.

In the phase of maintenance of topology, we privilege the maintenance of MIS because these nodes form land marks in the topology. The maintenance of connections between these land marks is then performed. The same rule by which we select the MIS

[1] Among the whole of intermediate nodes forming the virtual links (routes) between the MIS nodes, we choose only some which will be enough to avoid, as possible, the presence of redundant routes.

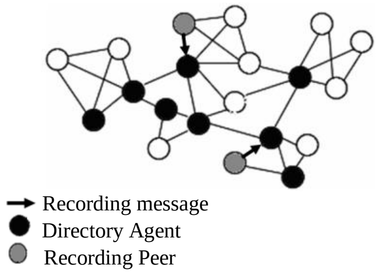
nodes, maintains it in time. Indeed, if two or several independent nodes are found in the same locality, only one of them remains independent and the others lose this privilege. If, on the contrary, a node is found without any neighbor who dominates it then it proposes itself for becoming independent. In addition, a connector not receiving connection messages, lasting certain time duration, leaves the CDS. This mechanism is implemented to offer a certain latency which is useful during the data exchanges of service discovery.

Let G = (V, E) be the graph modeling the network. We define $\Delta$ as being the maximum degree in the graph and $\Delta_{MIS}$ the maximum degree in the graph of 3-closure induced by the MIS nodes. On the level of each node, the steps (a) and (b) are done out of $O(\Delta^3)$, the step (c) is done out of $O(\Delta_{MIS}^2)$, it is the same for the stage (D).

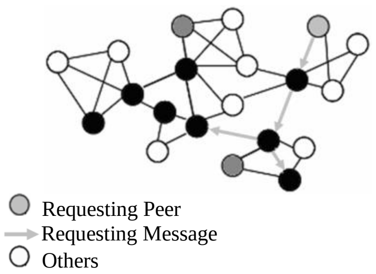
**2.2 Service Discovery**

| Peers: servers/clients | |
|---|---|
| Service discovery protocol | |
| Dynamic topology CDS | |
| MAC layer | LL |
| Physical layer | |

(1)




*(2)*




*(3)*

**Figure 2: Architecture and mechanisms of service discovery.**

The figure 2.1 illustrates the total architecture which we propose for our protocol of service discovery. The link layer is responsible for the hello messages and the neighborhood tables (tables used by the upper layer). The dynamic topology layer offers the access to the distributed directory and ensures that each node is, at most, within one hop of this last. The service discovery layer receives the recordings of the services, the requests and the replies. These messages are conveyed via the topology. With each node of the CDS a directory agent (or DA) is associated. The directory agents form between them a peer to peer network which ensures the correct operation of all the mechanisms of service discovery.

A peer wanting to make available one of its services, must register it. For that, it chooses a directory agent (local or neighbor, information provided by the topology layer) then sends a registering message to it (see Figure 2.2).

By receiving this message, the concerned agent saves the inscription in its services cache. In addition and because of dynamic nature of ad hoc networks, the directory agents do not tolerate a service registration whose lifetime is higher than a certain value TTLmax (maximum Time to live). This value represents a refresh timer (for example, if a service has lifetime equivalent to four times the refresh timer, then the service will have to be registered once and to be refreshed three times).

In order to optimize its services re-registrations, a server calculates a number of hops (radius of the concerned locality). This last relates to the number of service access during the previous refresh timer. In fact, more a service is requested, more it should be diffused on the network.

A peer seeking access to a service chooses a directory agent (local or neighbor) then asks it. The agent thus chosen is regarded as the one in charge for this request. At receiving the latter, the agent consults its cache of services. In the case where desired information is not available locally, the agent diffuses that request on the backbone (links in clear gray in Figure 2.3). So, to optimize this diffusion, the agent calculates a research radius from the cache of requests without replies. Indeed, if another node has previously formulated a similar request, then there would be strong chances that information is present in its locality. Intermediate agents receiving requests having an infinite research radius, can try to calculate a local radius.

However, the agent responsible for the request engages a timer. At the end of this timer, if no reply has been obtained, it generates another type of request message which will not be limited in hops. This last ensures the request diffusion on all the backbone. If after a certain time, no reply arrived, the agent answers its client by no-presence of requested service. The replies are routed according to the opposite direction of routes taken by the requests.

If a node hosting a directory agent leaves the backbone (CDS), it must try to transfer its caches to another CDS node (for this reason, at the phase of topology maintenance, we let spend a certain latency time before a node leaves the CDS). To achieve this treatment, the node privileges its "substitute" neighbors. We define these nodes as being the nodes which recently joined the CDS. The reason of this choice is that these nodes have a great chance to not hold much information. Therefore, each node of the CDS maintains a list of neighbors thus defined.

## 2.3 Protocol Simulation

In this section, we evaluate the performance of our implementation with a network simulator. Two main aspects are evaluated. We look at topology layer performance by varying network density and mobility, then, the behavior of service discovery mechanisms which run over the topology.

Let *G = (V, E)* be the graph, on order $|V| = N$, modeling the network. Let $A_x(t)$ be the average of the distances separating a node *x* from all the other nodes at the time *t* (*n* is the number of nodes). Let $M_X$ be the average mobility of a node *x* during the simulation (in the following formula $\Delta t$ represents the step of calculation)

$$A_x(t)=\frac{1}{n-1}\sum_{i=1}^{n}Dist(n_x,n_i)$$

$$Mx=\frac{1}{T-\Delta t}\sum_{t=0}^{T-\Delta t}\left|A_x(t+\Delta t)-A_x(t)\right|$$

So, average mobility is:

$$Mob=\frac{1}{n}\sum_{i=1}^{n}M_i$$

For our simulations, we proposed three values of mobility: 1, 3.5 and 7. These lasts correspond respectively to low, medium and high mobility. So to evaluate the impact of the mobility of the nodes on topology, we propose to measure the number of state changes of a node (state of the node = in backbone or out of backbone) according to mobility.

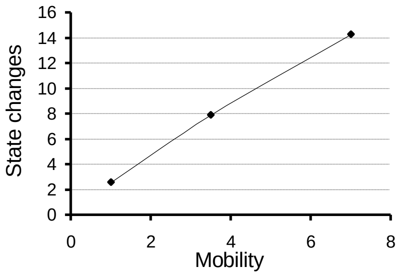



**Figure 3: Mobility impact**

More mobility increases more the number of state changes per node increases. The graph shows that this increase is linear. The principal goal of our dynamic topology is to offer, to each node, an access point to a backbone including all information of service discovery. The graph shows the mobility impact absorbed by the virtual topology. Indeed, theses changes occur transparently to service discovery layer.

Also, in addition to the mobility, we vary the following parameters:

- *Network density: between 20% and 80%*
- *Network nodes number: 20, 40 and 60 nodes*

To calculate the following Metrics:

- *Backbone (CDS) nodes ratio*
- *Backbone stabilization time*
- *The average number of hops carried out per reply*

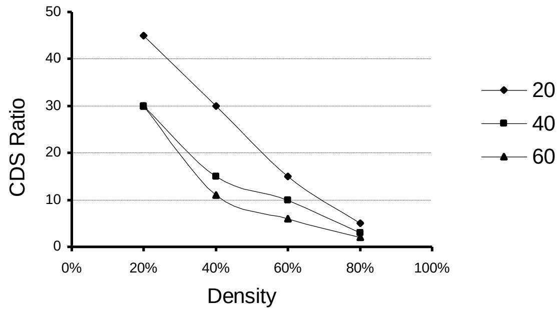



**Figure 4: Backbone nodes ratio**

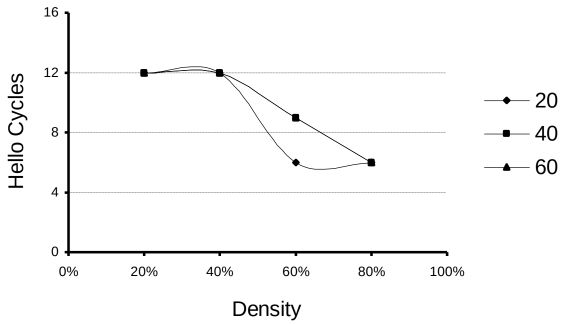



**Figure 5: Stabilization time**

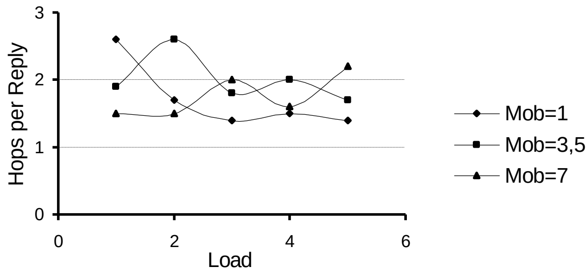



**Figure 6: Average hops per reply**

The groups of Figure 4 curves are decreasing. This shows that the number of nodes belonging to backbone decrease as the density increase. This result is completely normal considering the way in which nodes are chosen. Indeed, node's covering capacity allows it to maximize its chances to join backbone. Increasing density involves reducing the number of necessary nodes to cover the entire network.

From the groups of curves of Figure 5, we can note that the number of nodes composing the network do not influence much over the stabilization times. The total stabilization time is of twelve cycles for 20% and 40% density, and then decreases gradually until reaching the value of six cycles for 80% of density.

Figure 6 shows that the number of hops carried out by a reply is stable enough for various mobility levels. This number varies between 1.5 and 2.5 hops for all load values. This is due to diffusion of the most asked services.

## 3. Conclusion

The first objective of this work was to give a new solution to service discovery problem in ad hoc networks. In fact, in addition to known problems of static networks, mobile ad hoc networks are characterized by mobility, absence of infra-structures and limited capacities in bandwidth and energy... etc. This makes more difficult the design of communication and control protocols. The presented protocol build, initially, a virtual dynamic topology, maintains it in time then completes service discovery.

The basic idea of our protocol of service discovery is based on concept of connected dominating sets (for their covering and connection properties). The nodes of the connected dominating set (CDS) form a network backbone (in our case, a connected sub-network containing all information concerning the services). These nodes will have the responsibility to register and seek the services. Each node will have at least one CDS node in its neighborhood. For any operation concerning the services, a node chooses a CDS node to represent it in the backbone. For the topology construction, we propose an original method. This method is completely distributed and based on the concepts of independent sets and cliques. Also, we proposed the introduction of various mechanisms allowing the improvement of service discovery (to widen the diffusion of the most asked services, to seek the services where they were recently asked... etc.).

The protocol ensures symmetry of the roles played by the different nodes. Indeed, all the nodes are able, at various moments, to join or leave the CDS. Also, all nodes can be servers and/or clients and, therefore, form a peer to peer network. The use of dynamic virtual topology makes service discovery easier. Indeed, it is just enough to reach the backbone. The services declaration and location are done gradually on the backbone. The fact that a node can be covered by several backbone nodes offers some robustness and decreases collision risks. The CDS quality can only improve the service discovery. The suggested approach remains compatible with communication and service piloting techniques used in the directory oriented protocols such as: Jini, SLP... etc.

Simulation showed that our protocol gives well satisfying results. Indeed, from different test series, it releases a rather good adaptation to the various constraints imposed by network load, density and mobility. The number of nodes forming the backbone is rather weak and this for different network densities and sizes. The stabilization times are also reasonable. Number of hops carried out by a reply is relatively weak.

Among the prospects for our work, thorough simulations are considered. It would be interesting to evaluate the protocol in greater scale dynamic environments. Thus, a comparison with the other existing protocols of service discovery for the mobile ad hoc networks, would allow a better evaluation of our protocol. In addition, we estimate that the integration of nodes energy consumption in their dominance and connection factors would allow a better load balancing between nodes.

# References


[1] Salutation Architecture Specification Version 2.1, *the Salutation Consortium, 1999*.

[2] Understanding Universal Plug and Play White Paper, *Microsoft Corporation, June 2000*.

[3] Jini Architecture Specification, *Sun Microsystems, June 2003*.

[4] Erik Guttman, The Service Location Protocol, *IEEE Internet Computing July 1999*.

[5] O. Ratsimor, D. Chakraborty, S. Tolia, D. Kushraj, A. Kunjithapatham, G. Gupta, A. Joshi, T. Finin, Allia: Alliance-based Service Discovery for Ad-Hoc Environments, *Second ACM International Workshop on Mobile Commerce, in conjunction with Mobicom 2002, Sep 28, 2002, Atlanta GA, USA*.

[6] D. Chakraborty, A. Joshi, GSD: A Novel Group-based Service Discovery Protocol for MANETS, *4th IEEE Conference on MWCN. Stockholm. Sweden. September 2002*.

[7] S. Helal, N. Desai, V. Verma and C. Lee, Konark--A Service Discovery and Delivery Protocol for Ad-hoc Networks, *Proceedings of the Third IEEE Conference on WCNC, New Orleans, March 2003*.

[8] U C. Kozat and L Tassiulas, Service discovery in mobile ad hoc networks: an overall perspective on architectural choices and network layer support issues, Ad *Hoc Networks, Volume 2, Issue 1, January 2004, Pages 23-44*.

[9] V. Lenders, M. May., B. Plattner. Service discovery in mobile ad hoc networks: A field theoretic approach, *Pervasive and Mobile Computing 1 (2005) 343–370*.