Titre: Title:	A light-weight service discovery protocol for Ad Hoc networks
Auteurs: Authors:	Ranwa Al Mallah, & Alejandro Quintero
Date:	2009
Type:	Article de revue / Article
Référence: Citation:	Al Mallah, R., & Quintero, A. (2009). A light-weight service discovery protocol for Ad Hoc networks. Journal of Computer Science, 5(4), 330-337. https://doi.org/10.3844/jcssp.2009.330.337

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/66527/
Version:	Version officielle de l'éditeur / Published version Révisé par les pairs / Refereed
Conditions d'utilisation: Terms of Use:	Creative Commons Attribution 4.0 International (CC BY)

Titre de la revue: Journal Title:	Journal of Computer Science (vol. 5, no. 4)
Maison d'édition: Publisher:	Science publications
URL officiel: Official URL:	https://doi.org/10.3844/jcssp.2009.330.337
Mention légale: Legal notice:	© 2009 Ranwa Al Mallah and Alejandro Quintero. This is an open access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Journal of Computer Science 5 (4):330-337, 2009
ISSN 1549-3636

A Light-Weight Service Discovery Protocol for Ad Hoc Networks

Ranwa Al Mallah and Alejandro Quintero
Mobile Computing and Networking Research Laboratory,
Department of Computer Engineering, Polytechnique Montreal,
P.O. Box 6079, Station Centre-Ville, Montréal, Canada H3C3A7

Abstract: Problem statement: In mobile ad hoc networks devices do not rely on a fixed infrastructure and thus have to be self-organizing. This gives rise to various challenges to network applications. Existing service discovery protocols fall short of accommodating the complexities of the ad-hoc environment. However, the performance of distributed service discovery architectures that rely on a virtual backbone for locating and registering available services appeared very promising in terms of average delay but in terms of message overhead, are the most heavy-weight. In this research we propose a very light-weight, robust and reliable model for service discovery in wireless and mobile networks by taking into account the limited resources to which are subjected the mobile units. **Approach:** Three processes are involved in service discovery protocols using virtual dynamic backbone for mobile ad hoc networks: registration, discovery and consistency maintenance. More specifically, the model analytically and realistically differentiates stable from unstable nodes in the network in order to form a subset of nodes constituting a relatively stable virtual Backbone (BB). **Results:** Overall, results acquired were very satisfactory and meet the performance objectives of effectiveness especially in terms of network load. A notable reduction of almost 80% of message signaling was observed in the network. This criterion distinguishes our proposal and corroborate to its light-weight characteristic. On the other hand, results showed reasonable mean time delay to the requests initiated by the clients. **Conclusion:** Extensive simulation results obtained confirm the efficiency and the light-weight characteristic of our approach in significantly reducing the cost of message overhead in addition to having the best delay values when compared with strategies well-known in the literature.

Key words: Service discovery, ad-hoc networks, virtual backbone, stability

INTRODUCTION

With proliferation of mobile communication i.e., mobile phones with higher data rates, UMTS, PDAs that allow mobile network connectivity-mobility and modularity are the current goals of system development. The developers of today's state-of-the-art network architectures are asked to foresee developments like the desire of mobility, host roaming or hosts that can connect spontaneously to a network, an ad-hoc network. These networks are characterized by their lack of required infrastructure and ease of formation; each participating device is mobile and the networks are formed temporarily.

Service discovery and invocation are thus fundamental operations in an ad-hoc network. They enable components to find each other on the network, to join and leave freely and provide them with a consistent view of other components. In a service discovery environment, services advertise themselves and supply details about their characteristics. Clients may locate a service by its type or attributes and make a selection in instances where more than one service was found.

Service discovery has been a topic that has been researched relatively thoroughly in stable wired networks[5,12]. However, fundamental differences in the underlying environment make these techniques unsuitable for pervasive ad-hoc environments. In stable wired infrastructure most of the service discovery protocols are based on centralized registry based architecture. Nodes providing services register with a centralized node and nodes requiring services, query the centralized node to discover them. However this approach is unsuitable for ad-hoc environments because there is no centralized node that can be reached at all times. Thus for pervasive environments a new discovery mechanism is required.

Corresponding Author: Ranwa Al Mallah, Mobile Computing and Networking Research Laboratory,
Department of Computer Engineering, Polytechnique Montreal, P.O. Box 6079, Station Centre-Ville,
Montréal, Canada H3C3A7

Broadly speaking, there are two approaches to service discovery in wired networks. One is the service advertisement based approach and the other is the request broadcast based approach. In the first approach, the node providing a service, broadcasts an advertisement throughout the network which declares that it provides a particular service. The other nodes that receive these advertisements cache them locally in a directory and are then called service broker nodes. Whenever a node needs to invoke a service, it checks its cache to see if the service description is present. If not, it sends a broadcast message throughout the network and the closest service broker answers the request for service. In the other scheme there are no service advertisements and whenever a service is desired the service request is broadcast throughout the network. When this request reaches a node that provides the service it replies to the source. However, these methods do not scale well when the network consists of a large number of nodes or when there are a large number of services in the system. In both schemes, the broadcast storm problem is present and leads to collisions and message loss. We refer to these models as directory and directory-less architecture respectively.

It can be argued that the directory-less architecture is more suited to a MANET (Mobile Ad-hoc Network) because there is no need for any infrastructure[2]. It is also evident that using service broker nodes necessitates assigning this functionality either statically or dynamically to a subset of nodes in the network. Under the given constraints of MANETs, the static assignment is not possible. One of the weaknesses of ad hoc networks is that a route used between a source and a destination is likely to break during communication. Several papers discuss routing in ad hoc networks[6-11].

Service discovery technologies: Currently there are five auspicious protocols, each one eligible in different environments: SLP, Jini, Salutation, UPnP and Bluetooth SDP. The two architectures that come from peer-to-peer approaches on top of traditional networks (TCP/IP) are SLP and UPnP[2]. Since TCP/IP itself was not designed for ad-hoc networks, operation of these two protocols might not be satisfying in sophisticated ad-hoc environments where high node mobility is given. The other three protocols, Jini, salutation and Bluetooth’s Service Discovery Protocol (SDP), were particularly developed for the purpose of ad-hoc networking.

Service description: Service discovery protocols must define a data scheme to represent a service. This is called the service description. It must be unique and contain the service location, i.e., its network address. Also mandatory is the service type, i.e., what kind of service is being described. The description can furthermore contain attributes that characterize the service more exactly, a user interface and/or a program interface. Services are heterogeneous in nature and should be defined in terms of their functionalities and capabilities. The functionality and capability descriptions of these services should be used by the service clients to discover them. The existing service discovery infrastructures lack expressive languages, representations and tools that are good at representing a broad range of service descriptions and are good for reasoning about the functionality and the capabilities of the services. In the existing service discovery infrastructures, it is impossible to find services which require a specific attribute value that can change based on the dynamic content of the environment. Service descriptions and information need to be understood and agreed among various parties. In other words, well-defined common ontology must be present before any effective service discovery process can take place. GSD (Group-based Service Discovery Protocol)[3] exploits the semantic capabilities offered by DARPA Agent Markup Language to effectively describe services present on nodes in the MANET.

Virtual dynamic backbone for mobile ad hoc network: Three processes are involved in service discovery protocols using virtual dynamic backbone for mobile ad hoc networks: Registration, discovery and consistency maintenance:

- Registration: A backbone node holds a set of service descriptions. It must register all service descriptions with all the other backbone nodes discovered. The BB node and the server negotiate a lease time, which is the time after which the BB node will drop the registration if it is not renewed
- Discovery: For services to be able to find each other a discovery process has to take place. The user sends out requests, either a fixed number in fixed intervals or until it has discovered a BB node
- Consistency maintenance: Since we are dealing with distributed systems, new services can be deployed, obsolete ones can be removed, nodes and links may appear, disappear or fail. After login into a network and discovering available services, a user has to ensure that his knowledge about existing services stays consistent with the actual distributed state

To construct a backbone, we start from the fact that we cannot treat all the nodes equally in an ad hoc network. Mechanisms should be established to differentiate the stable and unstable parts of the network. To our knowledge, there exists only one protocol in the literature that suggests a virtual backbone for service discovery in ad hoc networks[3]. The authors use NLFF (Normalized link failure frequency) as a metric in the determination of the stability of a node. This selection criterion represents the number of expired links per unit time normalized by the total number of neighbors. As if mobility is the only attribute of a stable node in an ad hoc network, they construct a backbone satisfying this property.

On the other hand, the design purpose of the DSDP protocol proposed by Chakraborty *et al*.[2] is to provide a connected backbone which is maintained well even in highly dynamic network environments.

MATERIALS AND METHODS

Light-weight service discovery protocol: Light-Weight Service Discovery (LWSD) protocol takes into account the weaknesses of ad hoc networks. In fact, limited resources, including bandwidth and energy, necessitate the introduction of mechanisms to reduce the size of exchanged messages. Processing capacity and memory also require definition of methods in order to preserve resources. Frequent and dynamic topological modifications due to mobile hosts changing arbitrarily their point of connectivity cause link failures and thus extend the delay of access to the desired information. To reduce this access latency, the definition of a method to minimize the distance between a client requesting a service and a server appears essential.

To take into account all of the distinct features that characterize wireless ad hoc networks, LWSD consists of three phases. In the first phase, the design goal is to deploy a relatively stable virtual backbone for service discovery and advertisement even under high mobility conditions. Our objective aims to define an appropriate mechanism which will be able to select the key mobile units having the task of gathering relevant data for service discovery. The second phase of the proposition is the maintenance process which guarantees backbone connectivity. The final phase of LWSD is the discovery and registration process implemented to let servers register their services and clients request the services.

Network model: We assume an ad hoc environment with mobile nodes. All the nodes in the network have an omni-directional antenna and have the same transmission power. All links are bi-directional and nodes share the same communication channel to transmit and receive the control packets. Nodes are dispersed on a certain area and deployment of unit is two-dimensional. All mobile nodes should be able to provide information such as current geographical coordinates, average velocity, residual battery power as well as the time period during which the node will remain in the transmission zone of the transmitter. Partitioning in the network is not allowed because each partition may then be treated as an independent network.

We have already pointed out that one of the requirements of the Backbone (BB) nodes is to maintain a list of the services available in the vicinity. In our model, the other nodes of the ad hoc network are free of any responsibility. Unlike the other backbone or clustering algorithms, the rest of the nodes in the network do not exchange messages or maintain cache tables and are thus completely independent of the service discovery architecture deployed. In[3], the authors assume that any node in the network either is a backbone node or it is neighbor of a backbone node.

Phase I: Backbone formation: Judicious designation of stable nodes requires taking into account several parameters such as the average node velocity, residual battery power, total number of neighbors of the node, available processing resources.

Stability constraint: Stability and reliability are two strongly linked aspects. Indeed, we can qualify a node of being stable when we can anticipate a high level of reliability on his behalf. Integrating criteria such as residual battery power, average node velocity, effective degree (total number of neighbors), available processing capacities and the period of residence in the transmitter's range of transmission gives a good estimate of the stability of a node. Based on these parameters we can define eligible mobile units that can be qualified as stable. Before detailing the procedure of designation of stable nodes, we shall introduce some terminologies which will be used for the elaboration of the model.

Battery power: E_j denotes j's residual battery power at a certain time period T:

$$E_j = E_0 - \frac{1}{T}\sum_{t=1}^{T}\left(E_t - E_{t-1}\right) \quad (1)$$

Where:

E_o = The intrinsic energy power of the unit
E_t = corresponds to different energy power values at different times
T = The instant at which this metric is estimated

Average velocity of a node: Every node computes its average velocity. This metric gives an idea of the node's mobility in the network as shown in the equation:

$$M_j = \frac{1}{T}\sum_{t=1}^{T}\sqrt{\left(X_t - X_{t-1}\right)^2 + \left(Y_t - Y_{t-1}\right)^2 + \left(Z_t - Z_{t-1}\right)^2} \qquad (2)$$

Variables X_t, Y_t and Z_t represent the geographical coordinates of the different positions of the mobile unit at different time slots.

Effective degree: Effective degree corresponds to the total number of direct neighbors of the node in the vicinity N(v). It is the set of all the nodes that are in the radio range of the mobile unit. This metric is defined as follows:

$$D_j = \left|N(j)\right| = \sum_{j' \in V,\, j \neq j'} \left\{\text{dist}(j, j') \leq R\right\} \qquad (3)$$

Where:

R = The radio range of the mobile unit
V = The set of all the nodes surrounding the unit in question

Period of residence in the transmitter's range of transmission: Period of residence P_{ij} in the transmitter's range of transmission of a transmitter i and a receiver j is given by the following equation[4]:

$$P_{ij} = \frac{-\left(ab + cd\right) + \sqrt{(a^2 + c^2)r^2 - \left(ad - bc\right)^2}}{a^2 + c^2} \qquad (4)$$

Where:

(x_i, y_i) = Coordinates of transmitter i
(x_j, y_j) = Coordinates of receptor j
v_i and v_j = Speed of nodes i and j, respectively
θ_i and θ_j = Orientations of nodes i and j, $(0 \leq \theta_i$ and $\theta_j < 2\pi)$
a = $v_i \cos \theta_i - v_j \cos \theta_j$
b = $x_i - x_j$
c = $v_i \sin \theta_i - v_j \sin \theta_j$
d = $y_i - y_j$

Exceptions: If $\theta_i = \theta_j$ and $v_i = v_j$, then $P_{ij} = \infty$.

Procedure of designation of stable nodes: Having enunciated the metrics allowing a suitable definition of the eligible mobile units that can be qualified as stable: Residual battery power, average node velocity, effective degree (total number of neighbors), available processing capacities and the period of residence in the transmitter's range of transmission, we now propose the link loss metric (LL_i) as being the total number of links lost by the node i for a normalized fixed observation time period. This variable is based on the linear combination of the metrics presented above. The computation is given by the following equation:

$$\begin{aligned} LL_i &= m \bullet E_i + n \bullet M_i + t \bullet D_i + q \bullet P_{ij} \\ & m + n + t + q = 1 \\ & m, q, t \neq 0 \end{aligned} \qquad (5)$$

Variables m, n, t and q are substantial factors and P_{ij} is defined by Eq. 4. A mobile unit is considered stable if it has low mobility, many neighbors, more battery power then the average of the other nodes and finally if it resides longer in the range of transmission of the other nodes. Therefore, LL_i profiles the eligibility of a node.

Stability constraint, NLFF: We finally introduce the normalized link failure frequency metric ($NLFF_i$), also referred to as stability constraint:

$$NLFF_i = \frac{LL_i}{D_i} \qquad (6)$$

The stability constraint is based on two metrics, the link loss LL_i, $0 < LL_i < 1$ and the effective degree D_i. The lower is the first metric the better is the link quality. On the other hand, the higher is the number of direct neighbors the better is the probability of finding a replacing node in the surroundings, in the maintenance process, in case of a link failure with the node. Consequently, $NLFF_i$ reflects the rapidity in which the neighborhood changes in comparison with its degree. For instance, we will prefer a node having many neighbors and a slightly superior LL_i value to another having very few neighbors but a very good link quality.

Network Information Table (NIT): Initially, when first powered on, every node in the network is not a BB node. Before deciding on their role in the network, all nodes must collect Hello messages. These messages bear the following information about the transmitting node: Network identifier ID_i, stability constraint $NLFF_i$ and the network information table NIT.

Selection of stable nodes: Relying on NLFF values, we can distinguish the nodes in the network that are the most stable. In other words, we imply that from the set N of all the nodes in the network, each node holds a subset T of N as cached in its NIT. On top of that, from the subset T, nodes with the minimum NLFF form the subset S, with $S \subseteq T \subseteq N \not\subset T \not\subset S$. We point out that this last notation highlights the complete non-inclusiveness of subset S in T and of T in N, thus meaning that only few nodes in the NIT are part of subset S and that the NIT does not hold information about the set N of all the network nodes:

$$S = \{\min{}_i\{T\}\} \qquad (i = 1, \sum_{j \in T} z_j) \tag{7}$$

with $z_j = 1$ si j belong to BB, $z_j = 0$ otherwise.

$\sum_{j \in T} z_j$ is the maximum number of nodes that a unit must choose from his table. We have already anticipated that this number should be bounded. In fact, the number of BB nodes should be small enough to have a significant increase in the efficiency of control message overhead and it should be large enough to provide a certain degree of redundancy.

Optimum number of BB nodes: The limits of the optimum number of stable BB nodes:

$$\begin{aligned} &\eta_1 \le \sum_{j \in T} z_j \le \eta_2 \\ &\eta_2 = \frac{T}{2} \\ &\eta_1 = \sqrt{T} \end{aligned} \tag{8}$$

Variables η_1 and η_2 represent the upper and lower limit of the optimum number of BB nodes. We transposed the idea presented by in[1] to our proposal. The authors estimated the lower limit of the number of stable nodes to be the square root of the total number of mobile units in a cell. We will apply it to our proposal knowing that the performance results will guide us towards a limit more suitable to our model. For the upper limit, we envisaged the worst case scenario where half of the network nodes are considered stable. Here again, we shall observe the simulation results for a conclusive analysis.

Virtual backbone network: Now, we have in the network a subset of the most stable units within the topology. Once a node determines its role as a backbone node (BB = 1), it should assume its responsibilities, such as:

- Periodically sending HEY beacons to every stable node in the subset S. This task is crucial to the maintenance of the stability of the backbone
- Maintain a table, HEY, to record the reception time of HEY packets received by their neighbors in the backbone

Phase II: Maintenance of the virtual backbone: The maintenance process is local in order to rapidly adapt to network topology changes. Our solution consists of a decentralized algorithm. If for some reason a backbone node is disconnected from the network due to lack of energy or excessive mobility, a local maintenance procedure should be invoked. Considering that BB nodes exchange HEY messages periodically in order to maintain their connectivity, when a BB node stops receiving these messages after a certain period of time, T_h, it must wait an additional period of T_R seconds before considering itself isolated and thus out of the BB.

Phase III: Discovery and registration of services: Service discovery is a mapping from a service class and an attribute list to a single IP address or a group of IP addresses. Service registration is the process by which an ad hoc node that has at least one service to offer to the other nodes makes its intention known.

Service registration: Backbone nodes are stable and possess at all times the list of the services offered by the network. In fact, when a server offers a service, it should register the service with any one of the BB nodes. In the registration process, the node receiving the server's publication is responsible to distribute the publication to the other BB nodes. Contrary to other proposals where they need a broadcast or multicast mechanism to do so, in our model the distribution is done in the following fashion: considering that every BB node has a forwarding list of the other nodes constituting the backbone, the node receiving the server's publication sends, via unicast, the publication to the subset of S nodes in his NIT. Every T_E second, servers must renew their registration with any BB node they can reach.. The message flow of a server wanting to register its service is shown in Fig. 1.

Service discovery: As for service registration, when a non-backbone node wants to discover a service, it sends a Service Request with a fixed diameter into the network. The closest BB node responds to the request with a ServiceRep message. Here again, the request was propagated locally with a specific number of hops and does not, by any means, correspond to a flooding procedure. Thus, the traffic in the network is considerably reduced.

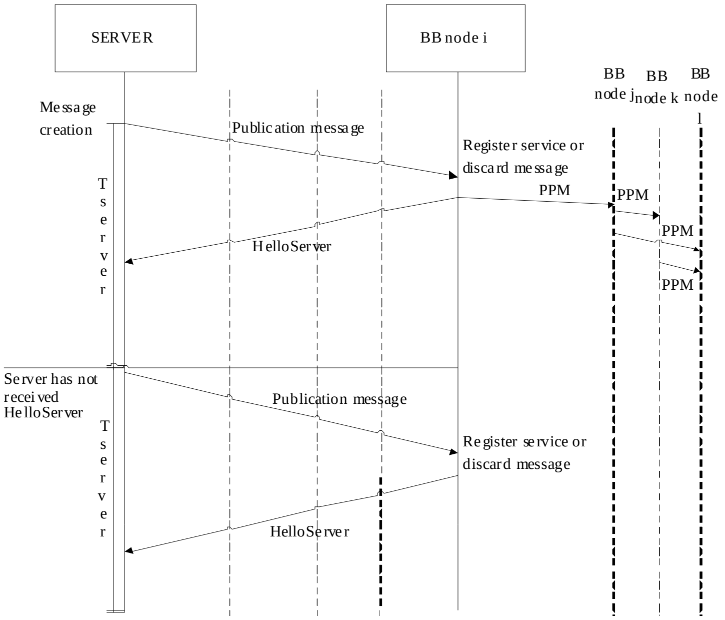

Fig. 1: Sequence diagram for a server

RESULTS

Experimental evaluation: We use the Distributed Coordination Function (DCF) of 802.11 as the underlying MAC protocol with 2 Mbps of nominal bit rate. We implemented Light-Weight Service Discovery protocol (LWSD) under various mobility conditions and different node topologies. Both square (1500*1500-2400*2400 m) and rectangular (1500*300-2400*1500 m) topologies were considered. We compared our model with DSDP[4], the only other service discovery protocol proposed based on a virtual backbone. Simulation environment consisted of node topologies ranging from a topology with 50 nodes to a topology with 150 nodes. We used a random way-point mobility pattern for all the nodes. Mobility is defined by P(S_m, S_{max}), where P = Pause (in seconds) after a node has moved to a new position, S_m = Minimum speed (m sec^{-1}) of movement of the node, S_{max} = Maximum speed (m sec^{-1}). The node moves with a speed within the range (S_m, S_{max}). In our simulations, we have varied node speed ranging from 1-20 m sec^{-1} with a pause time ranging from 0 -900s. Simulation time is set to 900s. Number of clients is selected as 10, 20 and 30, whereas number of servers is varied between 1, 3 and 5. Inter-arrival service request time is set to 8s. Servers periodically register their service every 10s. The routing protocol is DSR[6]. Each point in the simulation plot corresponds to average results on a series of over 30 scenarios.

Performance metrics: The efficiency of a service discovery strategy can be evaluated with the following metrics:

- The first performance metric is the network load in terms of number of packets.
- The second performance metric is the mean hit ratio. Hit ratio is simply the ratio of the total number of successful attempts to the total number of requests.
- The third performance metric is the average time delay between the time any successful request is sent from a client and the time a corresponding reply is received by the same client

Performance metrics were evaluated with respect to: Node mobility, number of servers in the network, number of clients, network topology and total number of BB nodes. This last factor is introduced in the perspective of experimentally studying the limits of the optimum number of BB nodes.

Average time delay: In the first set of experiments, we intend to capture the effect of the factors on the average time delay. We note that for DSDP, average time delay becomes worse as mobility increases where LWSD is found to be insensitive to mobility (Fig. 2). In fact, this result is surprisingly astonishing because with mobility link failures occur more often and thus this can be regarded as counter-intuitive.

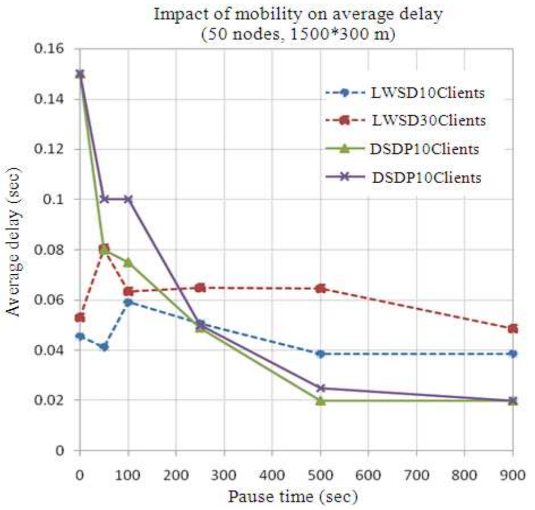

Fig. 2: Comparison of the impact of mobility on average time delay 3 servers/10-30 clients

Because of the way nodes locally handle maintenance and because of the light-weight architecture deployed, the network is not loaded, congestion in the network is less probable, thereby replies to requests are easily delivered to clients.

We further run a set of simulations to understand how sensitive the average time delay is against the number of BB nodes. In fact, Fig. 3 shows the impact of BB density on a topology of (1500*300 m) with 40, 60, 80, 100 and 150 nodes in the network. We notice that saturation point occurs when then value number of nodes in the BB is above 30. In other words, average time delay values become intolerable for a BB density of more than 30 nodes. The negative drift is also seen for fewer BB nodes. The lower boundary can be determined by the experimentation results of Fig. 3. In fact, results confirm that there is an optimum number of BB nodes that ranges from 15-25 nodes.

Control message overhead: We intuitively knew that, in comparison to the protocols studied, the major contribution of our approach is the reduction of control message overhead. This insight was confirmed by all analyzed factors. In fact a dominant reduction of almost 80% of network load was observed when all metrics were studied. Considering that maintaining a backbone node is costly in terms of message signaling, the fact that our backbone structure is constituted of a specific number of nodes judiciously elected contributed to the decrease of control message overhead in the network. The impact of mobility on the network load is shown in Fig. 4. The results for LWSD are much better than those of DSDP.

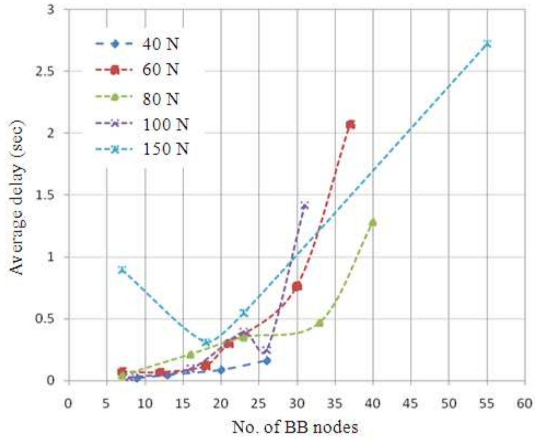

Fig. 3: Impact of BB density on the average time delay

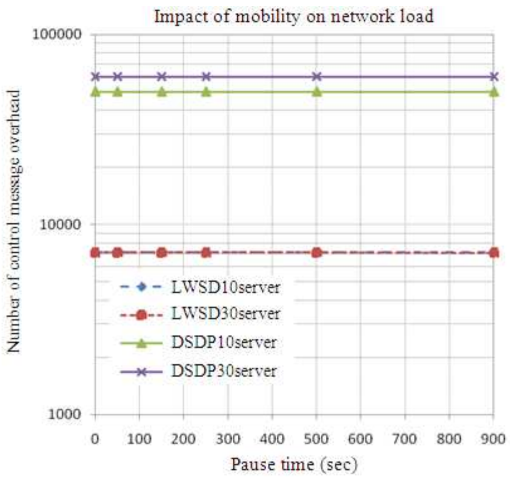

Fig. 4: Impact of mobility on the number of control message overhead in terms of packet

DISCUSSION

Overall, results acquired were very satisfactory and meet the performance objectives of effectiveness especially in terms of network load. A notable reduction of almost 80% of message signaling was observed in the network. This criterion distinguishes our proposal and corroborate to its light-weight characteristic. On the other hand, results showed reasonable mean time delay to the requests initiated by the clients. They ranged well with the other protocols studied. As for the average hit ratio, results showed percentages of over 90% of successful attempts which is considered a very fine average for ad hoc networks. When high mobility of units, wide topologies and large number of network nodes is envisaged, we showed that our proposition scales better than the existing service discovery solutions for ad hoc networks.

We proposed a light-weight strategy for service discovery in ad hoc network in order to minimize network load, perfect the hit ratio of service request initiated and decrease the mean time delay of requests. The approach was based on the deployment of judiciously elected nodes in the environment for the service discovery process to take place. The nodes were elected based on their stability constraint. Integrating criteria such as residual battery power, average node velocity, effective degree (total number of neighbors), available processing capacities and the period of residence in the transmitter's range of transmission gives a good estimate of the stability of a node. Based on these parameters we mathematically defined a procedure for a designation of eligible mobile units that can take part of the virtual backbone. Only eligible and elected mobile units maintain the list of services available in the network. They are solicited whenever a node requests or registers a service. The introduction of the diameter in the TTL field helped scope the messages sent to a local area, thus reducing the signaling in the network.

CONCLUSION

In conclusion, we met the purpose aimed in proposing a light-weight service discovery protocol by reducing network load, achieving reasonable mean time delay to the requests initiated by the clients was reasonable and having great average hit ratio, of successful attempts. Therefore, we showed that the deployment of a virtual backbone for the discovery of services, until now almost unexplored, is a promising strategy to be considered in these types of networks.

REFERENCES

1. Ulas, K., R. Ondylis and K. Marwma, 2001. Virtual dynamic backbone for mobile ad hoc networks. Proceedings of IEEE International Conference on Communications, June 11-14, IEEE Xplore Press, USA., pp: 250-255. DOI: 10.1109/ICC.2001.936312
2. Chakraborty, D., A. Joshi and Y. Yesha, 2006. Integrating service discovery with routing and service management for ad-hoc networks. Ad Hoc Networks J., 4: 204-224. DOI: 10.1016/j.adhoc.2004.03.016
3. Chakraborty, D. and A. Joshi, 2002. GSD: A novel group-based service discovery protocol for MANETS. Proceedings of the 4th IEEE Conference on Mobile and Wireless Communications Networks, July 2002, IEEE Xplore Press, USA., pp: 140-144. DOI: 10.1109/MWCN.2002.1045711
4. Zhang, Y. and L. Cheng, 2004. Protocol for location and coordinates estimation-a wireless sensor network approach. Comput. Networks, 46: 670-693. http://cat.inist.fr/?aModele=afficheN&cpsidt=16230656
5. Zegura, E.W. and M.H. Ammar *et al*., 2000. Application-layer anycasting: A server selection architecture and use in a replicatedWeb service. IEEE/ACM Trans. Network., 30: 455-466. DOI: 10.1109/90.865074
6. Johnson, D.B. and D. Maltz, 1996. Dynamic source routing in ad hoc wireless network. Mobile Comput., 5: 153-181. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.37.1126
7. Perkins, C., E. Belding Royer and S. Das, 2003. RFC 3561-Ad hoc On-Demand Distance Vector (AODV) routing. http://www.faqs.org/rfcs/rfc3561.html
8. Chien-Chung, S., S. Thapornphat, R. Liu and Z. Huang, 2004. CLTC: A cluster-based topology control for ad hoc networks. Trans. Mobile Comput., 3: 18-32. DOI: 10.1109/TMC.2004.1261814
9. Meyer, D., 1998. Administratively scoped IP multicast. IETF RFC 2365. http://www.faqs.org/rfcs/rfc2365.html
10. Chakraborty, K., S.S. Iyengar, H. Qi and E. Cho, 2002. Grid coverage for surveillance and target location in distributed sensor networks. IEEE Trans. Comput., 51: 1448-1453. DOI: 10.1109/TC.2002.1146711
11. Sivakumar, R., P. Sinha and V. Bharghavan, 1999. A core-extraction distributed ad hoc routing algorithm. IEEE J. Select. Areas Commun., 17: 1454-1465.
12. Dabrowski, C. and K. Mills, 2002. Understanding self-healing in service-discovery systems. Proceeding of the ACM Workshop on Self-Healing Systems, Nov. 18-19, Charleston SC., USA., pp: 15-20.