Multi-Level Zone Based Hybrid Service Discovery Protocol for mobile Ad-Hoc Grid

Askari Parichehreh
Department of Computer Engineering
Islamic Azad University of Qazvin
Qazvin, Iran
apmastermail@gmail.com

Bahman Javadi
Department of Computer Engineering
Islamic Azad University of Qazvin
Qazvin, Iran
javadi@aut.ac.ir

Omid Jafarzadeh
Department of Computer Engineering
Islamic Azad University of Behshahr
Behshahr, Iran
Omid.jkh@gmail.com

*Abstrac*t – **hybrid service discovery is introduced as an efficient service discovery protocol in mobile ad-hoc grid witch has a good compromise between response time and network traffic overhead. In this paper we modify the basic idea of this protocol called Zone. We use multi-level zone in our proposal and called the protocol as multi-level hybrid service discovery protocol. Our algorithm tries to provide the advantage of larger zone with only a little increase of network traffic overhead. We used two levels of zone, inner zone and outer zone which we called internal zone and external zone, respectively. Difference between two zones issued from their advertisement frequency. Simulation results show that our proposed algorithm has better performance in network's traffic and response time than traditional hybrid service discovery protocol.**

Keywords—mobile ad-hoc grid; service discovery; hybrid protocol;

I. INTRODUCTION

Mobile ad-hoc grid is a computational environment that use parallel and distributed technology. Computing nodes in this system establish on mobile ad-hoc network, so this technology combines capability of traditional grid with flexibility of mobile ad-hoc networks. However, the development of ad-hoc grids entails new challenges compared to traditional wired grids. Resource discovery, power consumption, QoS security and etc, are problems that have still to be solved. In this paper we study in-depth the problem of resource discovery in mobile ad-hoc grids. Generally, we can classify service discovery mechanism in three main categories:

1. Advertisement based service discovery
2. Query based service discovery
3. Hybrid service discovery

In the advertisement based mechanism or proactive model, servers advertise periodically, by broadcasting or multicasting location and attributes of resources and services. So that the client can build a local database with all services which available on the grid environment. Examples of P2P Advertisement mechanisms are the Konark passive push protocol [1], and the UPnP discovery service [2].

In query based mechanism or reactive model, client sends a query message to other nodes, by Broadcasting or multicasting and asks about services that match same specific attributes. Any servers which have that service respond to the client's query by sending a description of the service, Examples of P2P query-based systems are the Bluetooth SDP[3], the Konark active pull protocol [1], and the service discovery mechanism proposal for on-demand ad-hoc networks[4,5].

Hybrid mechanism tries to find a good compromise between proactive and reactive methods. The basic idea of hybrid mechanism is to limit the advertisement message within small zone; a zone is defined for any node individually to reduce advertisement overhead and in the other hand enhance query response time. Example of hybrid service discovery protocol can be observed in [6].

Using the larger zone reduces response time but increase network traffic overhead. We found a solution to decrease response time without increasing network traffic.

The remainder of the paper is organized as follows. First of all, we make a survey on service discovery protocol in section II. The proposed algorithm is presented in section III. We analyze our service discovery mechanism overhead in section IV and results of performance evaluation are presented in section V. Finally, section VI concludes this paper.

II. RELATED WORK

A considerable amount of works has been spent into studying and designing ad-hoc grid environments. In this section we want to describe four important service discovery protocols which are implemented in real situations.

A. Service location protocol (SLP)

This protocol was introduced by the IETF SVRLoc Working Group and initiated in 1997. SLP version 2 is the last version of SLP and was standardized in 1999. [13, 14]

SLP is a lightweight, open and scaleable protocol for service discovery in IP networks. It can use several architectures such as centralized or decentralized directory and even directory-less architecture. SLP also use three agents: (i) user agents (US) that perform service discovery on behalf of clients or applications, (ii) service agents (SA), that advertise the service information and position, (iii) directory agents (DA) that service information ore registered on it. The SLP uses the centralized architecture when DA is present, and otherwise uses the directory-less architecture.

B. Jini

Jini was introduced by Sun Microsystems in 1998. The Jini Community was initially established in January 1999 with the release of the first version of the Jini

978-1-4244-5824-0/$26.00 ©2010 IEEE

Technology Starter Kit from Sun Microsystems. The last version of the Jini Technology Starter Kit has been released in February 2004 [15, 16].

Like SLP, Jini can provide several architectures. Jini uses Lookup Service (LUS) as directory agent which maintains a database for all services in the network. When LUS is found the services will register with it and clients can query it using Remote Method Invocation (RMI). Then LUS matches the query against the registered services information and response to the client. The client may receive several responses from several LUS; therefore the client may run filtering process in order to make a selection over the received messages.

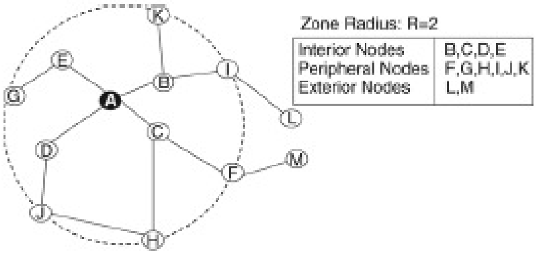
C. Konark

Konark is a peer to peer service discovery mechanism, in which each node that provides services runs a SDP manager, SDP managers in each mobile device used for two basic functions: (i) registering and advertising local services and (ii) service discovery on behalf of the client's application.

To discover the service the Konark uses two different models: Active pull and passive push model. In active pull model, client multicasts a service discovery message and then the servers with the required service respond to the message. In passive push model, servers advertise periodically the services that are locally registered.

D. Hybrid service discovery protocol.

This protocol is similar to idea that introduced in Zone Routing Protocol [10-12]. A discovery zone is defined for any node individually, and is composed of the entire neighbors whose distance to them dose not exceeds a certain number of hops, that we named in R where R is the zone radius.



Zone Radius: R=2

Interior Nodes	B,C,D,E
Peripheral Nodes	F,G,H,I,J,K
Exterior Nodes	L,M

Figure 1. Classification of deferent nodes in a zone with R=2.

According to Fig 1 within the discovery zone of a given node, we can classify nodes into three classes:

1- Interior nodes which its distance to given node is less than R.
2- Peripheral nodes which its distance to given node is equal R.
3- Exterior nodes which its distance to given node is more than R

Service Discovery mechanism in this protocol uses hybrid peer to peer mechanism. For service discovery within the zone we used advertisement mechanism and for service discovery between the zones the query mechanism is used. When the client node needs a special service, it broadcast or multicast a query message to other nodes and asks about services that match the same specific attributes. Any servers having the special service respond to the client's query by sending a description of the service [7]. Client may receive several replies from several servers, in this case client node can filter reply messages or use first arrival message to decrease response time.

III. MULTI-LEVEL HYBRID SERVICE DISCOVERY

The basic idea of our algorithm is an extension of hybrid service discovery protocol that uses a multi-level zone structure which in any level advertisement frequency is different from other levels.

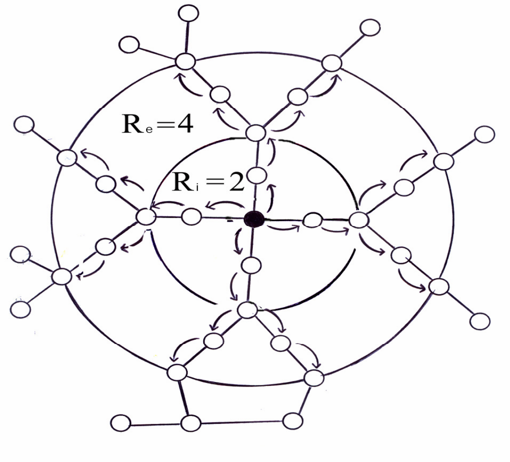


Figure 2. 2-level discovery zone

In this paper we discus the two levels discovery zones for simplification. As is shown in Fig 2, we use two-level discovery zone which the inner level is called internal zone and the outer level is called external zone. Internal radius is equal to 2 and external zone radius is equal to 4(RI = 2, R_E= 4). Difference between the two zones is return to their advertisement messages frequency.

we assume that advertisement frequency in internal zone is similar to traditional service discovery approach and for reduction of advertisement overhead we defined external zone frequency, F_E= $1/F_I$, that F_E addresses the advertisement frequency in internal zone. In addition we use query control mechanism that describe in [8] for this goal.

In order to implement two-level hybrid service discovery, we need to create some changes in original hybrid service discovery protocol. To do so, we need to modify service advertisement and query mechanism. We explain these modifications in the following.

3.1. Advertisement Mechanism Modification

First of all, we must modify service lookup table and advertisement message structure of any node in ad-hoc grid environment. We add a field ***Zone-type*** to them. We use this field to distinguish type of the server base on its location in internal zone or external zone. Based on the value of this field, information in service lookup table is divided into two types: 1. Information about the services that belongs to nodes in internal zone, 2. information about the services that belongs to nodes in external zone. Furthermore, we use Hop-count value to limit the extension of advertisement message, the value of Hop-count will be set to R_I to cover the internal zone, of course , for covering the external zone , in every F_I times we will set Hop-count value to R_E, that means advertisement messages forwarded out to nodes in external zone.

When a neighbor node receives this message should update its own service lookup table based on that message. The receiver node should specify Zone-type and type of server as an external or internal server. If the Zone-type value is Internal, the node doesn't need to do any extra work, but when the Zone-type value is External, the Hop-count field of the message should be evaluated.

If Hop-count < R_I, it means that distance between the server and the receiver nodes is more than R_I hop, therefore the server is placed in external zone and receiver node sets the Zone-type field of this node by 'External' value in service lookup table. After this, receiver sends the messages to all neighbor nodes.

In the following we focused on the condition that the information in external zone is not always accurate, because advertisement and update frequency in this area are lower than internal zone. To support this condition, the query mechanism of hybrid service discovery protocol must be modified; which is described in the following.

3.2. *Query Mechanism Modification and* Inconsistency *Situation Management*

In this section we describe the query mechanism modification and the condition in query sending that entries in service lookup tables are inconsistent against real situations. For a mobile node like C that receives a query message from clients, there are two conditions in which its service lookup table is not suitably reflected against the real situation. The first situation occurs when the server that has our desirable service is in the external zone, but not found in the service lookup tables of node C. the second occurs when the server that has our desirable service is not in the external zone, but found in the service lookup table of node C.

For first case, since the server is not found in the server lookup table, the mobile node C bordercast the query to its peripheral nodes and each intermediate node that receives query message checks the server is in its internal or external zone. If so, border casting is interrupted and the query message is sent to the server and the reply message is sent to client node C. This condition is shown in Fig 3.

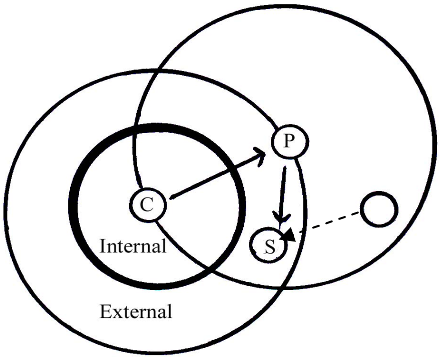


Figure 3. Serveice Discovery with Periphral node P

As it shown in fig 4, if the server node is not detected by any intermediate or peripheral node, the process of service discovery continues to find other servers that provide client’s desirable service, and if the query message after certain time receives time out, a client node C retries to run service discovery process based on the internal zone.

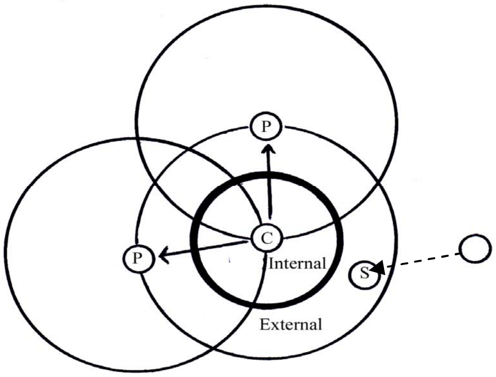


Figure 4. service not detected with periphral nodes.

For second case, since the server is founded in service lookup table of node C, the query message is forwarded to server with routing protocol. Intermediate node that receives query checks the server is in its internal or external zone or not.

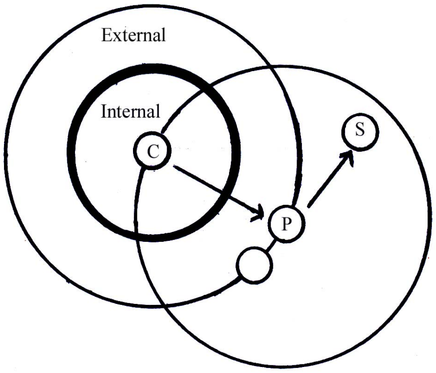


Figure 5. serveice discovery with Periphral node P

As shown in Fig 5, if intermediate or peripheral nodes find that server, send query message to it, otherwise border cast process is run in these nodes, as shown in fig 6.

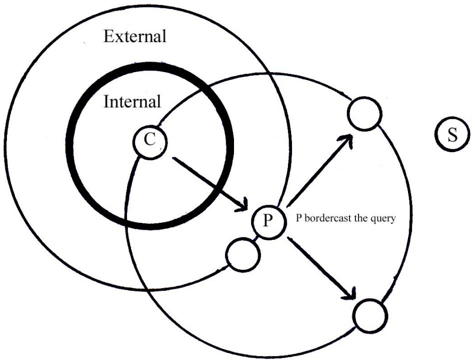


Figure 6. Bordercasting to find special service.

IV. ANALYSIS OF THE PROPOSED PROTOCOL

In this section we analyze our proposed protocol based on multiple parameters that affect on its performance. We divide it in two sections, advertisement overhead and query overhead as follow:

A. *Advertisement overhead analysis:*

Advertisement overhead is triggered based on the changes in connectivity with neighbors. When the neighbor is disconnected form a zone, other nodes in that zone updates their service lookup tables and send it to the other nodes in the zone. When a neighbor is discovered by neighbor discovery process, other nodes in the zone update service lookup tables and send it to other nodes in the zone. Thus, the rate of advertisement overhead in original hybrid service discovery can be expressed as:

$$\text{Adv Messages/Node/S} = \text{V. Update Traffic/Neighbor(R, D)}$$

Where V, R, D, are relative node velocity, zone radius and node density, respectively. But when we modify the hybrid service discovery and use two-level Hybrid discovery, it is observable, that, advertisement overhead increase by 1/Fi factor. Thus the rate of overhead in 2-level hybrid service discovery can be expressed as:

$$\text{Adv Messages/Node/S} =$$

$$\text{V} . ((\text{UpdateTraffic/Neighbor (Ri, D)}) + \text{1/Fi (UpdateTraffic / Neighbor (Re, D)))}$$

This is notable that the advertisement overhead per node is not depend on the overall nodes in grid environment (N); this is due to the local scope of the discovery zone.

B. *Query overhead analysis*

Another criterion is query overhead that can be regarded as the amount of the number of messages that generated per query, multiplied by the number of the queries that initiated. We note that if the network topology does not change during the query propagation, the overhead of query is independent of relative node velocity (V).

The number of query that a node initiates depends on the user's demand for services and the stability of discovered services.

The average query initiated and subsequent use of discovered services is almost application specific and can be expressed by independent parameters: $R_{\text{initial-query}}$ and $R_{\text{service Usage}}$. Therefore the amount of query traffic per node can be expressed as:

$$\text{Query overhead/Node/S} =$$

$$\text{Query messages/ query/ node(R, D). N. (Rinitial - query} + \text{Rservice Usage)}$$

When we use two-level hybrid service discovery model, this is observable that the average of query overhead related to the external zone based query mechanism and after it related to the internal zone based query mechanism. Generally when an applicant node needs a special service, it sends a query message to border nodes of external zone. If the data entries in service lookup tables of these border nodes are consistent, the query mechanism will be continued, but if the data entries in service lookup tables of external nodes are inconsistent, the query mechanism will be restarted again based on internal zone border nodes.

Thus the amount of query overhead which depends on hit ratio of consistent information in external border nodes. Thus the rate of query overhead in two-level hybrid service discovery can be expressed as:

$$\text{Query overhead/Node/S} =$$

$$\text{Query messages/query/node(Re , D) . N . (Rinitial - query} + \text{Rservice Usage). H}$$

$$+ \text{ Query messages /query /node(RI , D) . N . (Rinitial - query} + \text{Rservice Usage). (1 - H)}$$

Where H is the hit ratio of information consistency in service lookup tables of external zone’s border nodes.

V. PERFORMANCE EVALUATIONS

We evaluate our multi-level hybrid service discovery protocol in Glomosim simulator [17]. For more accuracy, we choose number of nodes as a variable parameter between 50 to 100, and we used IEEE 802.11 b for MAC protocol that operates at a 2 Mbite/s data rate. The speed of node is set from 0 to 10 m/s similar to human walking rate. We defined 50 deferent services that any node can provide 10 service in built-in form, and when it need other services it should send a query message to other nodes. We use several parameters in our simulation that show in table1:

TABLE 1 SIMULATION PARAMETERS

parameter	**symbol**	**value**
Number of nodes	N	50-100
Network coverage area	A	$490000 m^2$ – $2250000 m^2$
Node speed	V	10 m/s
Node density	D	3-9 node
Internal Routing zone Radius	R_I	2 hop
External Routing Zone Radius	R_E	4 hop

In order to maintain a constant node density, mobile nodes randomly have distributed in an area growing from 700m × 700m to 1500m × 1500m. In our simulations any node moves according to the random waypoint mobility model.

We consider three criteria for our assessment, which includes: 1. advertisement overhead, 2. query overhead and 3. hit ratio of the success service discovery.

It is notable that the neighbor discovery (HELLO) beacons could be considered in our performance criteria, but this overhead is independent and not exclusive component of hybrid service discovery protocol. For this reason neighbor discovery traffic is not in our assessment.

Our experimental result of our simulation shows the correctness of the equation that we present in section 4 about advertisement overhead. It can be seem in Fig. 7 that the advertisement overhead of our algorithm with R_I=2 and R_E=4 is lower than original algorithm with R=4 and a slightly more than original algorithm with R=2, Because we use reduction factor to control advertisement overhead in our algorithm, of course, inconsistency in service lookup tables of nodes in external zone is the cost of this improvement.

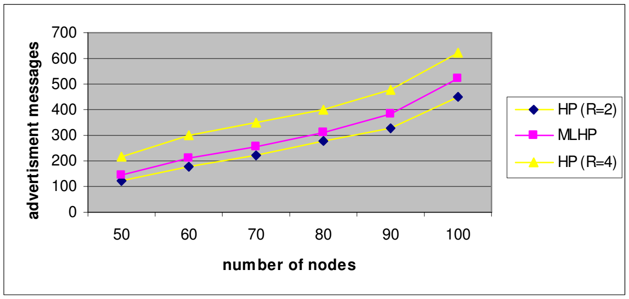


Figure 7. Advertisement overhead

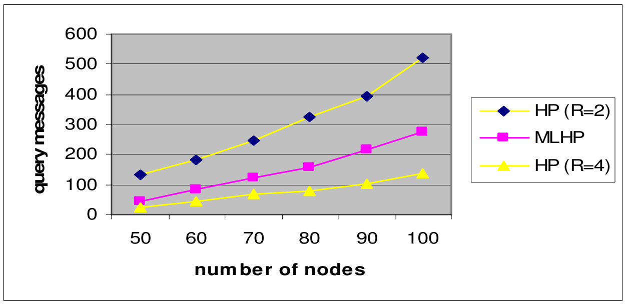


Figure 8. Query Overhead

As it shown in Fig 8 the two-level service discovery with $R_I = 2$ and R_E=4 has lower query overhead than traditional hybrid service discovery with R=2, because using of external zone decreased the number of bordercast in our algorithm. But its query overhead is a little bit more than traditional model because of the inconsistency in service lookup tables of external border nodes yielded to resuming of query mechanism based on internal zone border nodes.

The performance of our algorithm, in addition, is related to hit ratio of success service discovery that is related to the consistency of service lookup table information with real situation. Of course this parameter is changeable based on velocity of nodes and dynamicity of grid topology. Simulation results in Fig 9 show that our algorithm has a Hit ratio more than 95.5 %, which is acceptable for performance of our algorithm.

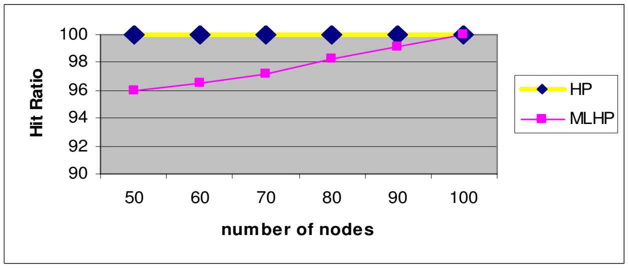


Figure 9. Hit Ratio

VI. CONCLUSION

In this paper we focus on hybrid service discovery protocol and improve its performance using large zone and controlling advertisement message that used for updating service lookup tables, we use multi level zone and specially two-level zones for simplification, for avoiding traffic increasing we reduce advertisement frequency in outer level. This frequency reductions yielded inconsistency in service lookup tables in nodes that located in external zones and for its management we change the bordercast protocol in our algorithm. Results show that our algorithm operates better than traditional hybrid service discovery protocol and in spite of service information inconsistency achieve acceptable Hit ratio of successful service delivery.

REFERENCES

[1] S. Helal, N. Desai, V. Verma, C. Lee, "Konark — A Service Discovery and Delivery Protocol for Ad-hoc Networks," in: Proc.of the Third IEEE Conference on Wireless Communication Networks (WCNC), New Orleans, March 2003.

[2] B.A. Miller, T. Nixon, C. Tai, M.D. Wood," Home networking with universal plug and play," IEEE Communications Magazine 39 (12) (2001) 104–109.

[3] Bluetooth. Service discovery protocol, Bluetooth Specification Version 1.1, Part E, Feb. 2001.

[4] R. Koodli, C. Perkins, "Service discovery in on-demand ad hoc networks," IETF Internet Draft (draft-koodli-manet servicediscovery- 00.txt), October 2002.

[5] C. Frank, H. Karl, "Consistency challenges of service discovery in mobile ad hoc networks," in: Proc. of the 7th ACM Int. symposium on modelling, analysis and simulation of wireless and mobile systems (MSWiM), 2004, pp. 105–114.

[6] R. Moreno, "A hybrid mechanism for resource/service discovery in ad-hoc grids," Dept. de Arquitectura de Computadores y Autom´atica, Universidad Complutense de Madrid, 28040 - Madrid, Spain. 2008

[7] T. Phan, L. Huang, C. Dulan. "Challenge: Integrating Mobile Wireless Devices into the Computational Grid,".2003, pp. 72–8

[8] A. Parichehreh, B. Javadi, A. T. Haghighat, “A Modify on Hybrid Service Discovery Protocol in Mobile Ad-Hoc Grid,” ICFN 2010

[9] Z. Li,"Challenges Of Mobile Ad-Hoc Grids and Their Applications in E-Healthcare," 2005.

[10] Z. J. Hass, “A new routing protocol for the reconfigurable wireless networks,” Proceedings, IEEE 6th International Conference on Universal Personal Communications, 1997, pp. 562-566.

[11] Z. J. Hass and M. R. Pearlman, “The performance of a new routing protocol for the reconfigurable wireless networks,” Proceedings, IEEE International Conference on Communications (ICC), 1998, pp. 156-160.

[12] M. R. Pearlman and Z. J. Hass, “Determining the optimal configuration for the zone routing protocol,” IEEE Journal on Selected Areas in Communications, vol. 17, no. 8, pp. 1395-1414, Aug. 1999.

[13] E. Guttman, C. Perkins, J. Veizades, and M. Day, “RFC 2608: Service location protocol, version 2,” 1999, status: PROPOSED STANDARD.

[14] SRVLOC, “Service Location Protocol (svrloc) Working Group,” 1997.

[15] Sun Microsystems, “Device Architecture Specification,” Sun Microsystem, Inc, Tech. Rep., June 2003.

[16] Sun MicroSystems, “The Davis Project,” 2003. [Online]. Available: http://davis.j

[17] Glomosim simulator homepage, http:// pl.cs.ucla.edu/projects /glomosim