Advances in Networks
2013; 1(1) : 1-9
Published online January 10, 2013 (http://www.sciencepublishinggroup.com/j/net)
doi: 10.11648/j.net.20130101.11

SciencePG
Science Publishing Group

A novel approach of service discovery based on tree directory architecture in Ad-Hoc grid environment

Maysam Hedayati[1], Ali Reza Sahab[2]

[1]Department of computer, Arak Branch, Islamic Azad University, Arak, Iran
[2]Department of engineering, Lahijan Branch, Islamic Azad University, Lahijan, Iran

Email address:
Hedayati_maysam@yahoo.com (M. Hedayati), Sahab@liau.ac.ir (A.R. Sahab)

To cite this article:
Maysam Hedayati, Ali Reza Sahab. A Novel Approach of Service Discovery Based on Tree Directory Architecture in Ad-Hoc Grid Environment. *Advances in Networks.* Vol. 1, No. 1, 2013, pp. 1-9. doi: 10.11648/j.net.20130101.11

Abstract: Grid computing is a new technology that using communication infrastructure facilities of computer networks and distributed systems features are able to connect to heterogeneous computing resources, so that the whole structure seems as an integrated virtual machine. One of the important problems in ad hoc grid environment is service discovery. In this paper, we propose tree directory architecture for service discovery in ad hoc grid environment, So that we used directory technique for implement of service discovery mechanism. We'll show that having a directory in ad hoc gird network isn't useful, so we need to have a set of directories. Therefore, the geographical area of ad hoc network divide into a 2-D logical hierarchical grid and each cell in the grid, select a directory as agent cell. By using tree architecture, the whole directory is designed as a multi-layer tree. Then, the request of requestor can be search for service discovery from bottom to top. We show using the simulation results that tree architecture for directories to raise the discovery success ratio.

Keywords: Grid Computing; Ad-hoc Grid Environment; Service Discovery; Tree Architecture

1. Introduction

The resource sharing potential of Grid computing opens up the possibilities of new applications that involve the integration of Grid technology with new areas such as mobile ad-hoc networks [1][2][3]. A mobile ad-hoc Grid is an infrastructure-less Grid computing environment that consists of mobile devices with limited power supply and transmission range. Mobile ad-hoc Grids are expected to have wide applications in battle fields, eHealthcare in emergency, crisis management and wildfire fighting, etc. Service discovery is one of the key components that mobile ad-hoc Grids inherited from Grid computing. The purpose of service discovery in mobile ad-hoc Grids is to transparently and seamlessly locate available resources/services throughout the Grid environment. In the paradigm of service discovery, service consumers advertise requests containing key words or port numbers representing the services they need. A few service providers who want to share their resources keep listening on a specified port or interface for service requests and reply to those matching the services they hold [4].

Service and resource discovery is becoming more and more important with the growth in the size and the diversity of computer networks. Furthermore, the ubiquity of the ad hoc grid environment is making different discovery services critically important in the future. Considerable amount of work has been done in this field, but mostly the resource and service discovery solutions have been approached as an implementation task to develop new protocols or frameworks, not trying to classify, categorize and to seek out generalities [5].

2. Related Work

Besides the resource discovery mechanisms specifically implemented for grid environments, in the last few years a broad range of protocols for service discovery in wired and wireless networks have been developed and standardized. Service discovery protocols are essential for pervasive computing environments, enabling applications or users to automatically discover the location of software entities/agents that provide access to network resources such as devices, data and services. So resource discovery can be viewed as a specialization of service discovery. Next, we briefly describe the most outstanding protocols for service discovery [6].

Service Location Protocol (SLP). The Service Location

Protocol [7] is a service discovery technology standardized by Internet Engineering Task Force (IETF, RFC 2608). SLP consists of three main components: User Agents (UA), software entities that perform service discovery on behalf of clients or applications; Service Agents (SA), software entities that advertise the attributes and location of services they represent on behalf of these services; Directory Agents (DA), software entities that aggregate service information received from SAs into a repository and respond to service requests from UAs. SLP has two modes of operation: with or without DAs. If DA is present, it collects and aggregates all service information advertised by SAs. UAs must discover the presence of a DA, and then they unicast the service information requests to that DA. In the absence of DAs, UAs multicast service information requests to the network. SAs listen for these multicast requests and unicast the responses to the UA. In the first operation mode (with DAs), there are two different mechanisms for DA discovery: active and passive. In active discovery UAs and SAs multicast service requests to the network. In passive discovery, DAs periodically multicast advertisements for their services. Services are advertised using a Service URL, which contains all the necessary information about the type, location and template of the service. A service template is a document that specifies the attributes of a particular service type, and their default values and interpretation.

Jini. Jini is a platform-independent service discovery mechanism based on Java and developed by Sun Microsystems [8]. The purpose of the Jini architecture is to federate groups of devices and software components into a single, dynamic distributed system. The Jini discovery mechanism is similar to that of SLP with DAs, by way of a Lookup Server that maintains a database for all services in the network. The core of the Jini system consists of three basic protocols called discovery, join, and lookup. Discovery occurs when a service provider locates a lookup server by multicasting a request to the local network. Join takes place when the service provider registers the service with the lookup server. Lookup protocol is used by a client, to locate and invoke a service registered in the lookup server. Services are described by service objects using a Java programming language interface. A service object includes the methods that users and applications will invoke to execute the service along with other descriptive attributes. Universal Plug and Play (UPnP). Universal Plug and Play is a discovery service technology developed by Microsoft [9] that supports device auto-configuration and automatic discovery of devices and services, whereby a device can dynamically join a network, obtain an IP address, announce its name, convey its capabilities upon request, and learn about the presence and capabilities of other devices. UPnP exhibits a decentralized architecture based on control points. A control point is a controller (such as an application) that retrieves device and service descriptions, sends actions to services, queries service state variables and receives events from services. UPnP uses Simple Service Discovery Protocol (SSDP) for service discovery. When a new device is added to the network, it multicasts an advertisement message to announce its services to the control points running on the network. Similarly, when a control point is added to the network, it multicasts a search message to discover which devices and services are available on the network. When a control point discovers a new device, it retrieves the device's description from the URL provided by the device in the discovery message. The UPnP description for a device is expressed in XML format and includes a list of any embedded devices or services, and URLs for service description, control and eventing.

Salutation. Salutation is a service discovery technology developed by Salutation Consortium [10]. The Salutation Architecture defines an entity called the Salutation Manager (SLM) that functions as a service broker for applications, services and devices. The SLM allows networked entities to discover and utilize the capabilities of other entities (e.g. devices or services). A Salutation Manager works with one or more Transport Managers (TM). Transport Managers are introduced to make the Salutation Manager transport independent. The Salutation Manager maintains a registry with information about services available on the network. A client registers or unregisters itself with the local SLM or a near SLM connected to the client. In addition a SLM can discover other remote SLMs and the list of services registered there. The communication protocol among SLMs is based on broadcast RPC. Salutation can operate as a fully decentralized peer-to-peer discovery model if every device on the network has its own SLM.

Secure Service Discovery Service (SDS). The Secure Service Discovery Service, developed at UC Berkeley [11], is similar in many aspects to Service Location Protocol (SLP), but it provides secure and authenticated communications, and a hierarchical structure that improves scalability. SDS system consists of three main components: clients, services and SDS servers. Clients discover the services that are running in the network by soliciting information to SDS servers. These SDS servers are organized into a hierarchical structure: service announcements and client queries are assigned to a particular SDS server; the network extent that is covered by a SDS server is called domain. Each server is committed to sending authenticated messages containing a list of the domains that it is responsible for, using a well-known global SDS multicast channel. Services continuously listen for SDS server announcements in order to determine the appropriate SDS server for its service descriptions. After determining the correct SDS server, the service multicasts its service descriptions to the proper channel, with the proper frequency, as specified in the SDS server's announcement. A client uses ARMI (Authenticated Remote Method Invocation) protocol to connect to the SDS server providing coverage for its domain, and submits a query in the form of an XML template. ARMI protocol is based on certificates to authenticate each of the endpoints.

Intentional Naming System (INS and INS/Twine). Intentional Naming System is a resource discovery and service location technology developed by the MIT Laboratory of Computer Sciences [12], suitable for mobile devices and ad-hoc networks. INS uses a decentralized directory model for discovery services and devices. INS architecture consists of three main components: Intentional Name Resolvers (INR), Domain Space resolvers (DSR), and services. INRs form an application-level overlay network to exchange service descriptions and construct a local cache based on service advertisements. Services register with an INR sending soft-state periodic advertisements. Each INR uses application level multicast to forward this service information to any other INR in the network, so an INR can collect information about every service in the network using a structure called name tree. Clients send service queries to INRs, which makes a resolution and forwarding decision, depending on the service requested by the client. Domain Service Resolvers (DSR) maintains list of active and candidate INRs. INS service discovery technology is more scalable than other solutions, since it avoids the use of network layer multicasting; however name tree construction and lookup can still cause scalability issues. Twine/INS [13] is an extension of the INS system. Twine tries to improve scalability by using hash-based partitioning of resource/service descriptions amongst name resolvers.

Bluetooth Service Discovery Protocol (Bluetooth SDP) . Bluetooth is a short-range wireless technology that allows devices to exchange data and voice in real-time. The Bluetooth SDP [14] provides a means for applications on a device to discover services on peer devices and to determine the characteristics of the discovered services. Each Bluetooth server device maintains a list of service records, which describes the attributes of services associated with the device. A service record contains information about a single service. The attributes of a service include the type or class of service offered, and the mechanism or protocol information needed to utilize the service. A client may retrieve information from a service record maintained by a SDP server by issuing an SDP service inquiry. Bluetooth protocol provides the following types of service inquiries: search for services by service class; search for services by service attributes; and service browsing. The set of SDP servers that are available to an SDP client can change dynamically. When a server becomes available, a potential client can be notified by a means other than SDP so that the client can use SDP to query the server about its services. Similarly, when a server leaves proximity or becomes unavailable for any reason, there is no explicit notification via the service discovery protocol. However, the client may use SDP to poll the server and may infer that the server is not available if it no longer responds to requests.

3. Service Discovery

The terms service discovery and resource discovery are often used interchangeably. Although, some subtle differences could be defined on this, we do not to take a firm stand on the issue. [5]

Definition 1 A resource is any source of supply, support, or aid that a component in a networked.

Environment can readily draw upon when needed. Examples are: files, measurements, CPU cycles, memory, printing, control devices, forums, online shops, etc. This definition requires further specification, since different systems support different types of resources.

Definition 2- Resource discovery is the ability to locate resources that comply with a set of requirements given in a query. The process of resource discovery involves three main actors: resource providers, resource users, and the RD service itself.

Definition 3- A resource provider is any networked entity that allows the sharing of its resources.

Definition 4- A resource user is any networked entity that uses shared resources.

Definition 5- A RD service is the service that returns the location of matching resources in response to a query with requirements [15].

3.1. Architecture of Service Discovery

Service discovery architecture is the framework correlating with different domains such as storage of service information, directory design, topology, information flow, routing, etc. There are many approaches to definition of service discovery architectures and classification of SD protocols [16, 17, 18, 19]. The service discovery architectures applicable to ad hoc design can be classified into two general groups: query-based (or directory-less) and directory-based [20].

"Fig. 1" presents the classification of service discovery architectures. Query-based architectures are represented by: traditional client-server (two-party) architectures, based on master-slave mode of operation; unstructured (distributed peer-to-peer) architectures, where all nodes have equal functionalities or by multi-tier architectures, where nodes are layered according to their capabilities into heterogeneity levels. The directory-based architectures can operate with one directory (acting as a service broker or a coordinator, i.e., simple three-party architecture) providing centralized approach. The architectures with more directories can be organized as structured (hierarchical, flat or hybrid) or heterogeneous overlay architectures, designed for heterogeneous environment with more discovery domains. Hierarchical architectures adopt parent-child relations between the nodes, often leading to tree-like structures or implementing some node clustering. Clustering architectures differentiate nodes functionality within the nodes in a cluster and between the nodes in different clusters. They can be combined with the multi-tier approach resulting in hierarchical multi-layer architectures, such as in [21]. Flat architectures are usually distributed DHT based structures (structured peer-to-peer), while hybrid architectures combine the elements of flat and hierarchical approaches as well as of query-based and directory-based elements. Peer-to-peer networks recently gain on popularity. Especially applicable to networks with a large number of nodes, they seem to offer appropriate mechanisms to ad hoc mobile network design, capable of following the nodes dynamism and mobility. They will be addressed in one of the following subsections. Network type and imposed requirements influence the service discovery architecture in ad hoc domain. Absence of any infrastructure, node dynamism and mobility declare the directory-less designs as more appropriate. However,

The large amount of overhead in the client-server approach, distinguishes the distributed principle in both service discovery architectures as the most efficient for ad hoc design (light service discovery protocols. Proposed solutions should cope with mobility and node failures, low caches and resources constraints, scalability.

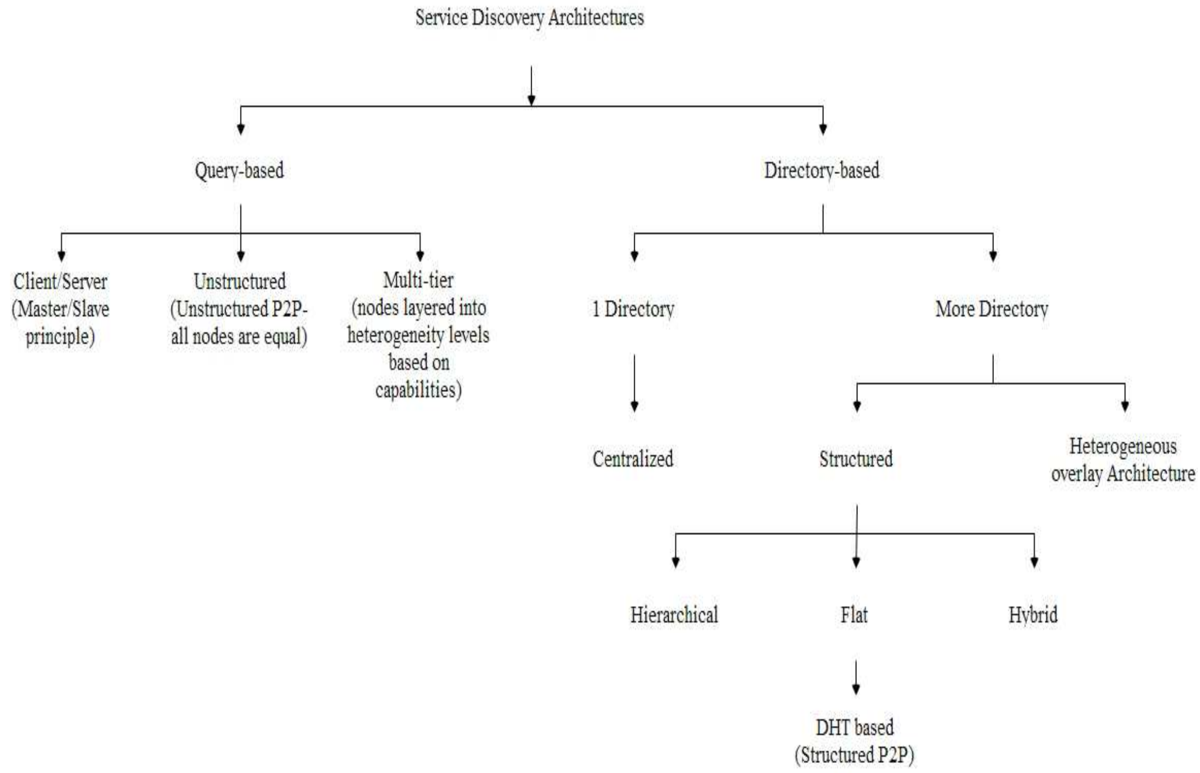

Service Discovery Architectures
Query-based
Directory-based
Client/Server (Master/Slave principle)
Unstructured (Unstructured P2P- all nodes are equal)
Multi-tier (nodes layered into heterogeneity levels based on capabilities)
1 Directory
More Directory
Centralized
Structured
Heterogeneous overlay Architecture
Hierarchical
Flat
Hybrid
DHT based (Structured P2P)

Figure 1. *Classification of service discovery architectures.*

3.1.1. Directory Design

Directories cache service information and answer client's lookup requests. They can be organized in a different way depending on environment according architectures, service information cache strategies and hierarchies. Several classifications, concerning different features, are possible:

- Centralized vs. decentralized directories;
- Number of service information copies (single copy: failure sensitive, multiple copies provided by multiple directories: more reliable, but greater overhead; fully replicated directories: a service search only goes to the directory to which a client is attached);
- Flat (peer-to-peer) vs. hierarchical directory structure (directories have parent/child relationships);
- Service state (soft vs. hard);
- Directory address;
- Number of directory hierarchies.

The directory architecture adopted by different service discovery protocols can be broadly classified as centralized or decentralized (see “Fig. 2”) [22]. In the centralized architectures, the directory information is stored on a central location in the network. Decentralized architectures can be categorized as replicated (the entire directory information is stored at multiple network

Locations — as in INS), distributed (directory information is portioned and parts are stored in different network locations) or hybrid (both replication and distribution methods are used — as in Twine).

The directory information in distributed directory architecture can be stored in dedicated servers, i.e., directory agents (DAs) (as in SLP, Jini and Salutation) or can be cached locally by the service providers in the system (as in UPnP, JXTA, and SLP without DA, pure decentralized and partially-centralized P2P systems).

Centralized directory architecture is not suitable for large systems and is failure sensitive. The major problem in replicated systems is consistency of the replicas. Distributed and hybrid architectures are scalable and provide better level of fault-tolerance. Centralized approach does not fit the dynamic ad hoc nature while decentralized distributed approaches can perform satisfactorily. Since each node acts as client and router at the same time, the network nodes have limited resources to perform the service discovery. So, the small amount of service information cached locally is more convenient to low power ad hoc nodes than directory based approaches, which occupy more resources of directory nodes.

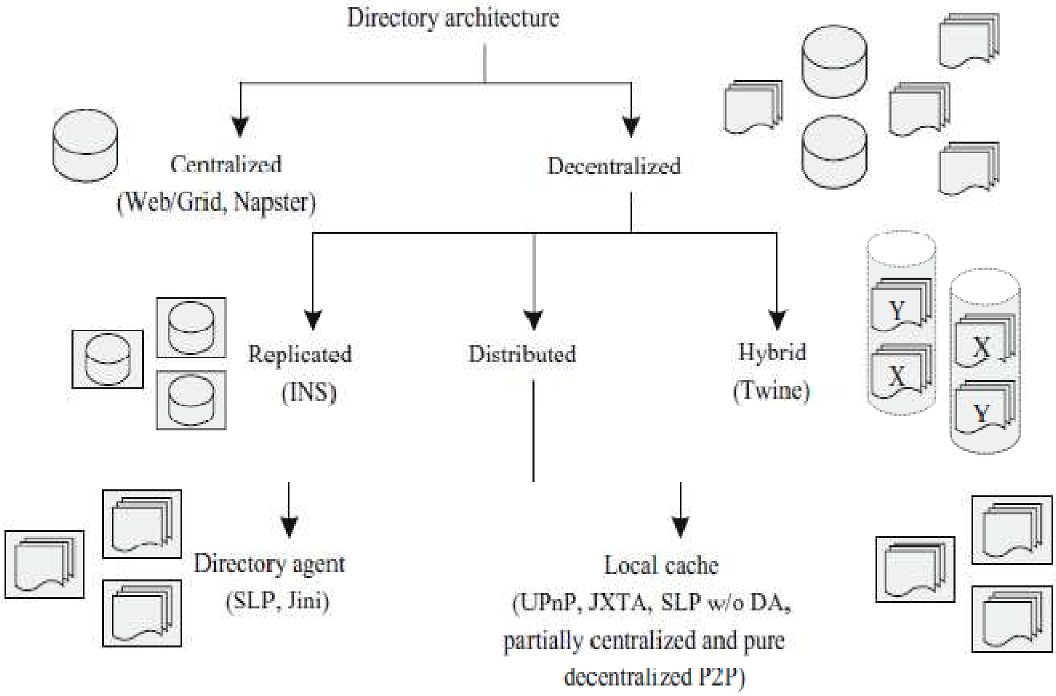


Figure 2. *Directory architectures.*

3.1.2. Peer-to-Peer Architectures

The classical client/server paradigm is hardly applicable in modern networks and is increasingly replaced by peer-to-peer (P2P) approaches, allowing constant changes in network topology and making fixed infrastructure obsolete. P2P networks distinguish from traditional client/server or master/slave networks in sense that there is neither a central point of control nor centralization of data. In recent trends, more and more research efforts are dedicated towards deployment of peer-to-peer networks in the context of mobile ad hoc networks (MANETs). The pervasive computing and emerging sensor networks also contribute to focus research activities on these architectures and their features. P2P technology is recently successfully deployed in several file sharing systems, such as Gnutella [23], Napster [24]. The music file sharing revolution started by them led to considerable interest in ad hoc networking issues. However, they are not scalable. Second generation of P2P systems (Chord, Pastry, Tapestry, CAN) are based on DHT and provide reliable content location (with persistency and availability). They Guarantee a definite answer to a query into a bounded number of hops and are more scalable. Their major characteristic is that all nodes tend to participate and contribute equally to the system. Improvements considering nodes heterogeneity and matching to performance metrics (load balance, system utilization, reliability and trust) are offered in [25].

The basis of both is self-organization, scalability, decentralized information dissemination and discovery, independence from centralized servers and support for constantly changing network topology [26]. Whereas ad-hoc networking refers more to the lower network layers, i.e., radio transmission instead of wired connections, dynamic host address assignment, and special routing mechanisms and so on, the peer-to-peer paradigm refers to the application design and is an antipode to the client server paradigm. Thus, applications in ad hoc networks are very likely to use the peer-to-peer paradigm, but conversely, peer-to-peer applications are not dependent on the network architecture underneath and are currently gaining popularity even in traditional TCP/IP networks [27].

4. Proposed Approach for Service Discovery

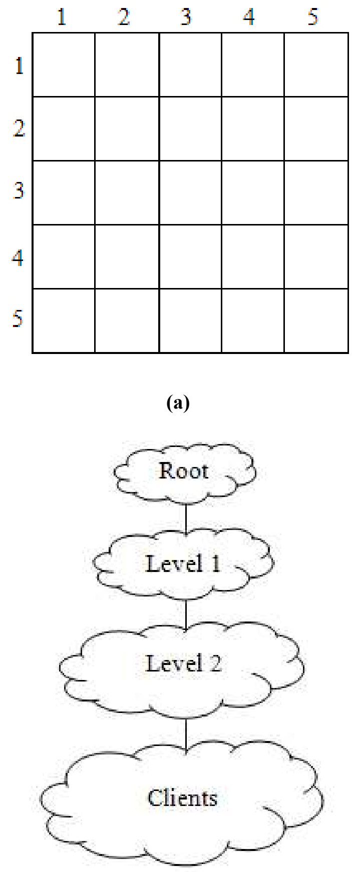
In this paper, we use tree directory architecture for service discovery in ad hoc grid environment. We'll show that having a directory in ad hoc gird network isn't useful, so we need to have a set of directories. Therefore, the geographical area of ad hoc network divide into a 2-D logical hierarchical grid and each cell in the grid, select a directory as agent cell.

The main challenges for service discovery in [28] can indicate to access to low rate of the discovery success ratio. In this section, we propose tree directory architecture for service discovery in ad hoc grid environment, So that we used directory technique for implement service discovery mechanism, that role of this directory is interface between the service provider and service requestor nodes. By using tree architecture, the whole directory is designed as a multi-layer tree. In this architecture, the directory is divided into different layers and levels and so the request of requestor can be search for service discovery from bottom to top. The simulation results show that tree architecture for directories to raise the service discovery success ratio than to SGrid [28].

In this paper, an efficient service discovery algorithm based on tree architecture is proposed, which is called the TSDA. In generally the TSDA Algorithm has n+1 level. The first level is the root that covers the whole area. In general, the root directory is considered as reference of service discovery. When a request is forwarded from the lower levels, if no response services (The lack of services in the directory listing), request to continue to the root level. Levels after root save information based on geographic location and their levels.

4.1. Hierarchical Structure

In this paper, we use the hierarchical structure of a distributed service discovery mechanism in Ad-hoc grid. In The n+1th levels are servers and requesting nodes for service discovery that According to given Indexes in their cells that Director determines in high levels. After finding its provider attempt to transmit information at the same level to the provider.



(a)

(b)

Figure 3. Cellular structure and tree structure in TSDA.

For example, the number of cells in 5*5 according to "Fig. 3(a)" and tree levels is "Fig. 3(b)".

The method of Cell mapping into tree structure is first, we calculate the coordinate of the cell and so obtain the cell center. We convert Nodes in these cells to the directory that directory is selected as the root.

For calculating level 1, we consider all cells that one cell away from the root, as level 1. Also, for calculating level 2, we select all cells that two cells away from the root. This process continues to level n ("Fig. 4").

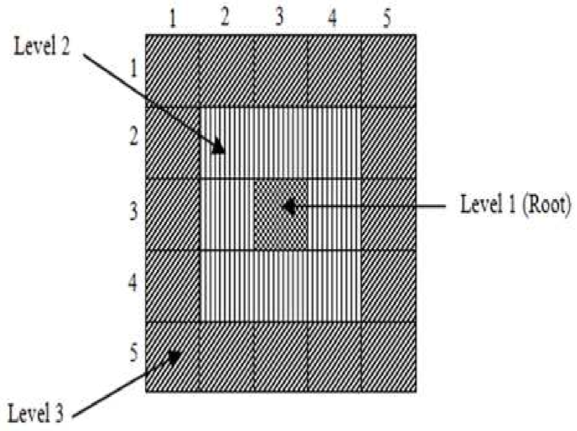


*Figure 4. View of a 5*5 cell.*

4.2. Service Discovery Process

In TSDA algorithm, providers and clients are last level in tree structure and requestors sent request to directory According to location of his cell.

Assume that size of the cells is 5*5 and consider node "A" in level 3, there is a node similar to ("Fig. 4") have service discovery request. In the beginning, node "A" send request to level 3. According to directories are indexed in the deferent levels cellular, therefore the director of Grid (5,4) process the request. According to location "A" cell is in the last level ("Fig. 5"), so the directory is certainly in level n. we are called the third level as access point.

If the service request was discovered by directory, directory sends service provider address with location address to requestor node. Otherwise, service discovery be referred to the current level or root level, While the request will searched by directory in the current level that for increasing network efficiency and performance, maybe available directories on the current level have a respond for considered request. If isn't answered to request in the current level, Then for reducing network traffic, request will transfer directly to the root level.

If node have request service discovery in Grid (4, 4), According to consider cell is in level 2, therefore we can't response to node from access point. So we need a way to aim to level 2. There are cells from level 3 around Grid (4,4) that called Grid (4, 5), Grid (5, 4) and Grid (5, 5). Thus, these three cells are considered as relay nodes ("Fig. 5").

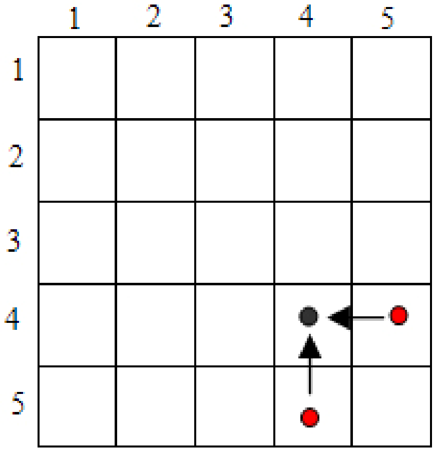


Figure 5. View of access to relay node and service discovery by a node in the tree mode.

5. Simulation Results

We describe simulation results of TSDA and compare with proposed algorithm in [28]. In this simulation, System performance is measured by changing the Grid cell size and speed of packets. Success ratio of service discovery is used for the performance and efficiency of this algorithm. Success ratio shows that service discovery algorithm (TSDA) can answer a response for any requiring of requestor.

5.1. Simulation Model

The used simulator in this paper is MATLAB simulation. In this simulator, we added module ad-hoc and we also simulate environment of ad hoc Grid with capability of MATLAB matrix. In this model, transmission range of the wireless nodes is 250 m and cell size of 5*5. Communication bandwidth between nodes is 2Mb/s. five runs are simulated, with each using the same parameters. The nodes are deployed at random in a square area of 2500*2500 m^2. The number of nodes is varied from 150 to 300. The grid levels *(L)* is set to 1. A requestor does not inquire the same service twice in a simulation run. A requestor can only demand from directory that this directory is as director of itself cell or current level or root directory.

5.2. Evaluating different node number

Performance evaluation is calculated for cell 5*5 and the radio range of 250 m. also the number of nodes is from 150 to 300.

Scenario: In this experiment, we compared the discovery success rate with SGrid algorithm [28] that only 25 nodes are selected as directory. In this experiment, the node speed is varied from 5 to 30 m/s.

“Fig. 6” shows TSDA algorithm evaluation with 300 nodes. “Fig. 7” shows comparing between TSDA and SGrid algorithm with 300 nodes and packet speed from 5 to 30 m/s.

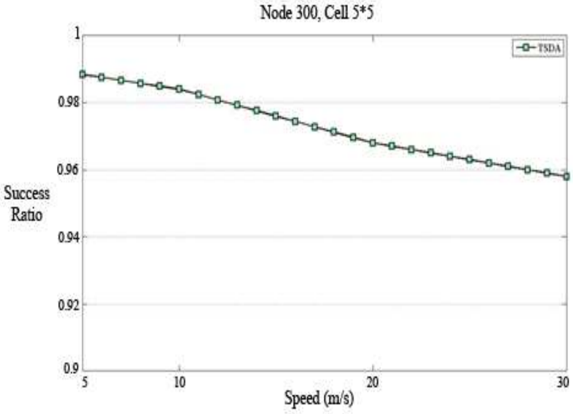


Figure 6. Discovery success ratio in TSDA with 300 nodes.

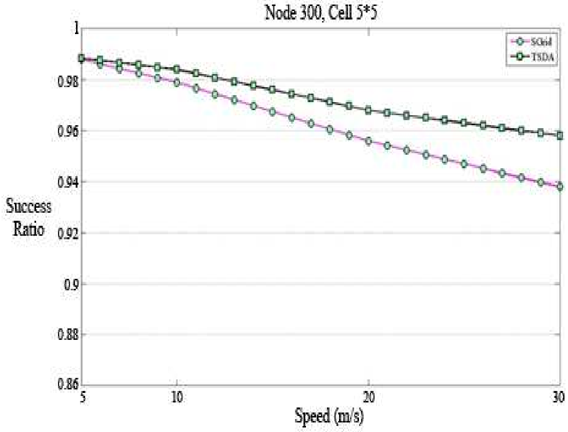


Figure 7. Comparing between SGrid and TSDA with 300 nodes.

We repeat again top experiments with the same size but with 150 nodes in the ad hoc grid. “Fig. 8” shows TSDA algorithm evaluation with 150 nodes. “Fig. 9” shows comparing between TSDA and SGrid algorithm [28] with 150 nodes and packet speed from 5 to 30 m/s.

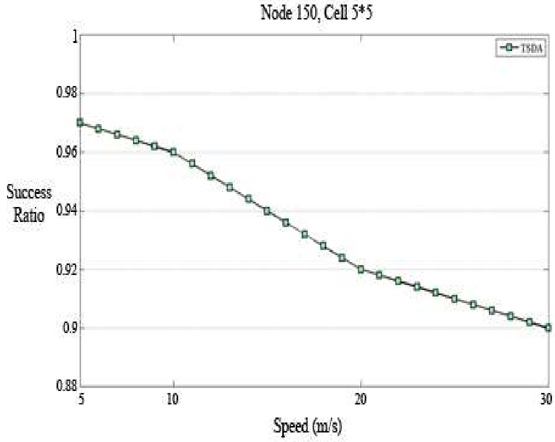


Figure 8. Discovery success ratio in TSDA with 150 nodes.

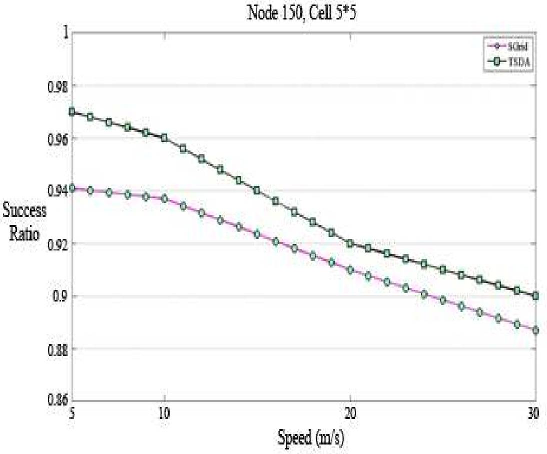


Figure 9. *Comparing between SGrid and TSDA with 150 nodes.*

5. Conclusion

The study on the methods of service discovery in the ad hoc grid, we get the result that all these methods are directory-based or peer-to-peer based algorithms. In this paper is used directory architecture for service discovery. We have shown by simulation results that having one directory in ad hoc gird network isn't useful. Therefore, the geographical area of ad hoc network divide into a 2-D logical hierarchical grid and each cell in the grid, select a directory as agent cell. So we have a set of directories. When a requestor wants to access a service, it discovers the service toward the maximum grid level. The requestor can obtain the reply from the directories.

We have proposed a service discovery method for ad hoc grid that called TSDA. We have measured the performance of protocols by analysis and simulation. We showed that TSDA raises the discovery success ratio.

References

[1] I. Gruber, R. Schollmeier, and W. Kellerer, "Performance Evaluation of the Mobile Peer-to-Peer Protocol," in Proceedings Of Fourth International Workshop on Global and Peer-to-Peer Computing, P.P. 363-371, Washington DC, USA , April 2004.

[2] S. Kurkovsky, S. Bhagyavati, and A. Ray, "Modelling a Grid-based Problem-solving Evironment for Mobile Devices," Journal of Digital Information Management, Vol 2, No 2, P.P. 135 – 136, April 2004.

[3] L. W. McKnight, J. Howison and S. Bradner, "Wireless Grids: Distributed Resource Sharing by Mobile, Nomadic and Fixed Devices," IEEE Internet Computing, Vol 8, No 4, July/August 2004.

[4] L. Zhuoqun, S. Lingfen and C.Emmanuel, "Track-Based Service Discovery in Mobile Ad-hoc Grids," in Proceedings of Second International Conference on Computational Intelligence in Medicine and Healthcare, Vol. 2, No. 2, P.P. 1274 – 1278, 2005.

[5] E. Meshkova, J. Riihijrvi, M. Petrova, P. Mhnen, "A survey on Resource Discovery Mechanisms, Peer-to-Peer and Service Discovery Frameworks," Computer Networks, Vol. 52 No. 11, P.P. 2097–2128, 2008.

[6] R. Moreno, "A hybrid Mechanism for Resource/Service Discovery in Ad-hoc Grids," Future Generation Computer Systems, Vol. 25, No. 7, P.P. 717–727, July 2009.

[7] E. Guttman," Service Location Protocol: Automatic Discovery of IP Network Services," IEEE Internet Computing, Vol. 3, No. 4, P.P. 71-80, July 1999.

[8] Sun Microsystems. Jini. "Architecture Specification", Technical Report, version 1.2, 2001.

[9] B.A. Miller, T. Nixon, C. Tai, M.D. Wood, "Home Networking With Universal Plug and Pay," IEEE Communications Magazine, Vol. 39, No. 12, P.P. 104–109, 2001.

[10] The salutation consortium, "Salutation Architecture Specification", Technical Report, version 2.0c, 1999.

[11] S. Czerwinski, B. Zhao, T. Hodes, A. Joseph, R. Katz, "An Architecture for a Secure Service Discovery Service," in Proceedings of the ACM/IEEE MOBICOM, New York, USA, P.P. 24–35, 1999.

[12] W. Adjie-Winoto, E. Schwartz, H. Balakrshnan, J. Lilley, "The Design and Implementation of an Intentional Naming System," in Proceedings of the 17th ACM Symposium on Operating Systems Principles, Vol. 34, No. 5, P.P. 186–201, 1999.

[13] M. Balazinska, H. Balakrishnan, D. Karger, INS/Twine: "A Scalable peer-to- peer Architecture for Intentional Resource Discovery," in Proceedings of International Conference on Pervasive Computing, Zurich, Switzerland, 2002.

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

[15] K. Vanthournout, G. Deconinck and R. Belmans, "A Taxonomy for Resource Discovery," Perss Ubiquit Computer, Vol. 9, No. 2, P.P. 81-89, March 2005.

[16] F. Zhu, M. Mutka, and L. Ni, "Classification of Service Discovery in Pervasive Computing Environments," MSU-CSE-02–24, Michigan State University, East Lansing, 2002.

[17] R. Marin-Perianu, P. Hartel, and H. Scholten, "A Classification of Service Discovery Protocols," Technical report TR-CTIT-05-25, Centre for Telematics and Information Technology, University of Twente, The Netherlands, June 2005.

[18] 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 Journal, Vol. 2, No. 1, P.P. 23–44, June 2003.

[19] M. Jacobsson, et.al, "A Network Architecture for Personal Networks," Dresden, Germany, 2005.

[20] P.E. Engelstad, et.al, "Service Discovery Architecture for On-Demand Ad Hoc Networks," International Journal of Ad hoc and Sensor Networks, Vol. 1, No. 3, P.P. 27-85, March 2005.

[21] My personal Adaptive Global NET, MAGNET (IST-507102), http://www.ist-magnet.org.

[22] R. Ahmed, et.al, "Service Discovery Protocols: A Comparative Study," IFIP/IEEE International Symposium on Integrated Network Management (IM'2005) Application Sessions, Nice, France, 2005.

[23] Gnutella. The gnutella home page: http://gnutella.wego.com/, 2002.

[24] Napster. Napster home page. http://www.napster.com/, 2001.

[25] B. Gedik and L. Liu, "PeerCQ: A Decentralized and Self-Configuring Peer-to-Peer Information Monitoring System," 23rd International Conference on Distributed Computing Systems, Providence, Rhode Island, USA, 2003.

[26] M. Dyrna, "Peer- to-peer Network Service Discovery for Ad hoc Networks," Seminar on Ad Hoc Networking, Technical University of Munich, 2003/2004.

[27] L. Gavrilovska, R. Prasad, "Ad-Hoc Networking Towards Seamless Communications", Signals and Communication Technology, 2006, XXVII, 284 p.

[28] T. Hua-Wen, C. Tzung-Shin, C. Chih-Ping, "Service Discovery in Mobile Ad Hoc Networks Based on Grid," Vol. 58, No. 3, P.P. 1528 – 1545, March 2010.