



Service Discovery in ADHOC Networks

¹G.Naga Satish ²Ch.V.Raghavendran ³G.JoseMoses ⁴Prof.P.Suresh Varma ⁵SNVSST Murthy

^{1,2,3} Research Scholar, Department of Computer Science, Adikavi Nannaya University, Rajahmundry. A.P., India

⁴ Professor, Department of Computer Science, Adikavi Nannaya University, Rajahmundry. A.P., India

⁵ Associate Professor, Srinivasa Institute of Engineering & Technology, Amalapuram, A.P., India

Abstract— *Service discovery dilemma creates new confront in adhoc networks. Due to the highly energetic nature of these networks and their bandwidth, energy these networks pose new challenges during service discovery. Traditional resolutions used in the Internet are hence not well suited for adhoc networks. In wireless adhoc networks, the system chattels are limited: the terminals are constrained in energy, bandwidth, storage, and giving out capabilities. As a result of the decentralized and possibly large scale network configuration, service discovery becomes a well-known design issue for wireless adhoc networks. In this paper we present service discovery, architectures, service discovery mechanisms, modes and service state maintenance techniques. We finally conclude for future research in this area with summary and directions.*

Keywords-Adhoc networks, Selection, energy, mechanisms, architecture

I. INTRODUCTION

Adhoc Networks are networks comprised of mobile nodes equipped with wireless interfaces and communicating with each other without relying on any infrastructure. In these networks each mobile node may act as a client, a server and a router. Adhoc Networks have emerged to fulfill the need for communication of mobile users in locations where deploying a network infrastructure is impossible, or too expensive, or simply is not available at that time. Characteristic scenarios for adhoc networks are adversity relief operations, battlefields and locations where infrastructure-based WLAN coverage (also called hotspots) is not provided and wireless WANs are too expensive to use or too slow. Most of the research on adhoc networks has focused on issues dealing with the connectivity between mobile nodes in order to manage with the dynamism of such networks and the arising problems thereof. This dynamism is due to the mobility of nodes, the wireless channel's adverse conditions and the energy limitations of mobile nodes, all of which lead to frequent disconnections and/or node failures. These research efforts have led to the creation of a sound technical basis for dealing with the aforementioned problems regarding node connectivity in adhoc networks (mainly through routing protocols, link layer protocols etc.) However, solving the problems of connectivity alone is not sufficient for the adoption of adhoc networks. Since their basic role is to allow users to exchange data and use each other's services, there is also a need for architectures, mechanisms and protocols for Service Discovery. Service Discovery is defined as a process allowing networked entities to:

- Advertise their services.
- Query about services provided by other entities.
- Select the most appropriately matched services.
- Invoke the services.

Service Discovery has been mainly addressed in the context of wired networks in the past. However, in the context of adhoc networks new challenges arise:

- Node mobility, affecting service availability.
- Frequent disconnections of the server, the client or intermediate nodes breaking or changing the path and the service selection parameters.
- Channel variability, leading to significant communication characteristics variability (data rate, delay etc.)

We believe that a comprehensive overview of open issues for service discovery would be useful. The paper is organized as follows: Section II describes the Service discovery in Adhoc networks and Section III presents service discovery architectures. In section IV service discovery mechanisms, Section V presents about service discovery modes and Section VI describes service state maintenance techniques.

II. SIGHTING IN ADHOC NETWORKS

A. Service Narrative

Service narrative needs a description that exists. This enables the users to identify a specific service [2]. The most used description format is the attribute-value structure. For instance, in Bluetooth Service Discovery Protocol, service information is contained within a service record, which consists of a list of attributes.

B Storage of service information

Information about available services depends on a storage system. In adhoc networks, storage may be non-existent, for example if the network is composed of resource-poor devices, the most adapted storage system is thus an unstructured distributed system.

C. Search Techniques

In unstructured distributed storage systems, nodes have to manage themselves how to find appropriate services. The most organized and distributed information is, less search effort is required, but a complex storage mechanism [2] makes the information consistency difficult to maintain. There are two ways to obtain information about services:

- Passive Discovery (or Push Model) - services or brokers announce or advertise their presence
- Active Discovery (or Pull Model) - an application needs information about services or brokers

D. Preservation

Preservation [2] is regarded as a permanent adjustment of the service information stored in nodes. Each node stores the service information it provides and it is interested in. The Preservation against changes in service description has to be managed by advertising the service with a different description. The nodes which have an older description can then update it and those which do not have it yet are informed of the availability of this service. The safeguarding against topology changes is a real issue because of mobility. Protocols dealing with this problem include Service Rings and Lanes. Service Rings groups together devices that are both physically close to each other and offer similar services. Lanes groups together devices into a lane where each node knows its predecessor and its successor in order to help routing for sending messages.

E. Service selection

The selection of service depends on the user or by an optimization method which is implemented at the client side, this can be done if and only if several servers offer the same service.

F. Service procedure

After discovery, the client can use the selected service by connecting to the server. Commands can be transmitted using Remote Procedure Call (RPC) like Java Remote Method Invocation [8] or Simple Object Access Protocol [9] (SOAP)

G. Security and privacy

Security includes authentication, authorization, trust, confidentiality, integrity, non-repudiation and privacy. In Adhoc Networks security is so important that it has to be a part of the main design strategy.

III. SERVICE DISCOVERY ARCHITECTURES

There are three basic architectures that a service discovery approach may adopt when service information is in distribution.

A. Directory-based Architectures

In this architecture there are three possible roles for a mobile node. A node can be a server, a client or a service directory. Service providers register their services to service directories and service requestors are informed about the available services in the network only through these directory nodes. A directory can be implemented as centralized or can be distributed among several nodes. Centralized approaches were primarily adopted by service discovery protocols in wired networks or in wireless local area networks where one or more fixed hosts take up the role of a directory. A simple centralized directory, however, is not a good solution for an ad hoc network, since no node is always reachable. A centralized directory also represents a single point of failure and is not well suited for such volatile environments. Scalability is also another problem, since in adhoc networks the nodes are resource-poor and a single node acting as a directory would not be able to handle responses for a large number of nodes. Distributed directories are thus more suitable for adhoc networks.

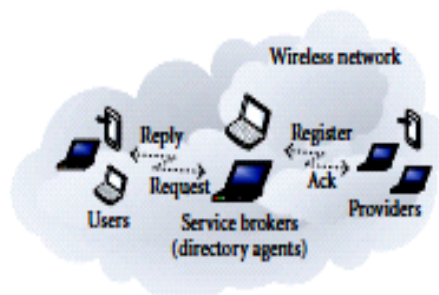


Fig 1: Directory Based Architecture

A basic question is whether global service discovery is to be provided (i.e. to be possible for every node to learn and invoke any service provided in the ad hoc network). One approach is to use full replication for directory nodes in order for every directory to store all services available in the adhoc networks, irrespectively of their location [16]. A classic distributed directory approach is Jini, where a few nodes, named Lookup servers, act as directories. However, there is no communication among Lookup servers and it is at the discretion of service providers to publish their service to more than one directory node and keep them updated. In this case, that automatic replication is not provided, a service may be known only locally, around the directory node that originally hosts it and remote nodes will not be able to easily discover it. Global discovery is hence not supported, since services are advertised only in the area where the Lookup servers reside. In more elaborate distributed directory approaches, nodes acting as directories are in constant communication with each other to disseminate and also replicate service information among them. Such approaches are based on protocols that create and maintain a backbone of directory-enabled nodes [17]. A backbone of directory nodes is formed using a Minimum Dominating Set algorithm. Servers advertise their services to one or more members of the backbone. However, despite the fact that service replication is not inherently provided, global discovery is possible since backbone members disseminate to each other service discovery requests that could not be satisfied locally. This way a service requestor and a service provider connected to the opposite edges of the formed backbone can still discover each other and communicate.

B. Directory-less Architectures

This type of architecture differs from the Directory Based Architectures, in which there are no service directories to mediate communication between service providers and service requestors. It is so easy when compared from directory-based architectures since there is no need for directory selection and maintenance. Service providers broadcast service advertisements and service requestors broadcast service requests. Both processes may take place at the same time in the network. In the early approaches of this type only servers could reply to service requests. Intermediate nodes were also allowed to reply to service requests based on the information they had cached locally. A basic problem in these non directory-based approaches is how to determine the frequency of service advertisements in order to reduce network load and avoid redundant transmissions. Scheduling and prioritization was one of the first techniques proposed to deal with the problem [14] [16].

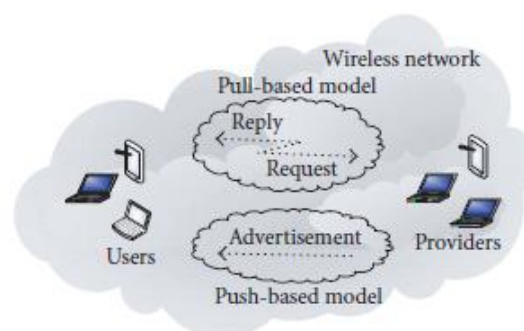


Fig 2: Directory less Architecture

This is why many approaches use various other techniques:

- Advertisement range bounding/scoping
- Selective, probabilistic and intelligent forwarding
- Peer-to-peer (P2P) information caching
- Intermediate node responding to service requests.

C. Hybrid Architectures

In these architectures service providers register their services with service directories if they locate any in their surrounding area. Service requestors send their queries to the service directories as they are aware of. If they are not aware of any service directory, they broadcast them to the whole network. Service replies may come both from service providers and service directories.

IV. SERVICE/RESOURCE DISCOVERY MECHANISMS

The service/resource discovery mechanisms can be classified in two main categories:

A. Peer-to-Peer Architectures

Peer-to-peer (P2P) architectures use fully distributed mechanisms for resource or service discovery, where networks entities negotiate on-to-one with each other to discover the available services and their attributes, and to find those services that meet the user requirements [15][16]. Two basics mechanisms can be used to service or resource discovery in peer-to-peer systems: query mechanisms and advertising mechanisms.

1). P2P Query-Based Systems (P2P-Query)

In P2P-Query, also called active or pull P2P mechanisms, clients send a discovery message to the network, by broadcasting or multicasting, asking for services or resources that match some specific requirements or attributes. Providers respond to the client query by sending a description of the service or resource attributes. Examples of P2P query-based systems are the Service Discovery Protocol (SDP) used in bluetooth the service discovery mechanism proposal for on-demand ad-hoc networks.

2).P2P Advertisement-Based Systems (P2P-Adv)

In P2P-Adv, also called passive or push P2P mechanisms, providers advertise periodically, by broadcasting or multicasting the location and attributes of resources and services, so that clients can build a local database with all the resources available on the network. Peer-to-peer architectures are useful for very dynamic ad-hoc environments, where network infrastructure is unpredictable, and the presence of permanent dedicated directories can not be guaranteed. However, these mechanisms, which are based on broadcasting (flooding) or multicasting, suffer from huge bandwidth usage and very low scalability, so they only suit well for small networks [15]. Advertising mechanisms use much more bandwidth and scale worst than query mechanisms, since unsolicited information is issued periodically to the network. However, they reduce the lookup time, since every client holds updated information about all the resources and services that are available in the network.

B) Directory-Based Architectures

Discovery architectures based on directory use a centralized or distributed repository, which aggregates and indexes the information about resources and services offered in the network. Providers register their resources and services with this directory, and clients query the directory to obtain information about resources or services. There are three different general schemes of directory-based systems: centralized directory, distributed flat directory, distributed hierarchical directory.

1). Central Directory Architecture: Central Directory architecture is based on a central directory that aggregates information from every provider, and respond to queries from every client. Central directory architecture [15] is a simple solution, easy to administrate, but directory can represent a bottleneck and a single point of failure, which causes the whole system's failure. Therefore, this solution does not scale well and is only suitable for small networks. Some examples of discovery mechanisms based on centralized architecture are the Service Location Protocol (SLP) standardized by the Internet Engineering Task Force the Jini system, which is a platform-independent service discovery mechanism based on Java and developed by Sun Microsystems, and the Agent-Based Service Discovery proposal.

2). Distributed Flat Directory Architecture: In this architecture several directories cooperate in a peer-to-peer fashion, to maintain a distributed repository of information about resources and services. Flat distributed directories can work in two different ways. Directories can exchange information with all other directories, usually by multicasting, so that each directory maintains a complete database about all resources and services in the network. The Intentional Naming Service (INS) [18] and the Salutation protocol [19] are two examples of discovery mechanisms based on this technique. It is obvious that this solution generates high communication traffic level, and hence it is not scalable. The second alternative is to divide the network in clusters or domains, so that each directory maintains a repository with information about services and resources within the cluster or domain. Information exchange between directories in different clusters can be achieved using a peer-to-peer scheme, but using a lower advertising frequency than within the cluster, like for example the INS/Twine system [20], or can be achieved on-demand, like for example the service locating system based on Virtual Backbone [21]. Although clustered solutions are more scalable and suitable for large networks, they must use complex algorithms to manage clusters (cluster formation, selection of directories, addition and removal of nodes to/from the cluster, etc.), and guarantee cluster stability.

3).Distributed Hierarchical Directory Architecture: With this architecture, the network is divided in domains with a hierarchical structure (like DNS) and directories have parent and child relationship. This solution is fully scalable, but it enforces a rigid hierarchical network organization, which does not fit well in ad-hoc environments.

V. SERVICE DISCOVERY MODES

There are three possible ways of operation for a service request to acquire service information:

A. Reactive

In this mode a service request issues a query is on demand basis to directory nodes or directly to service providers. There are many variations for this mode, some of which have been discussed in the service discovery architectures section of this paper. To name a few options for service requestors, they may choose to set a limited so that they do not flood the whole network when there are no directories. They may expand their search step-by-step by gradually increasing the hops that a service request is allowed to travel. They may utilize mechanisms to selectively forward their requests to specific neighbors only, instead of sending them to every neighbor. They also may unicast, multicast or broadcast a query to one or more directories or to one or more servers.

B. Proactive

In this mode service providers advertise their services on discrete time intervals. The same holds for advertisements originating from directory nodes. Servers and directories have also the option to use ranges for the advertisements instead of flooding the whole network. A basic tunable parameter is how frequently those advertisements should be sent, since it greatly depends on the level of dynamism.

C. Hybrid

In this mode both proactive and reactive communication between service request, service providers and service directories is possible. For example servers may proactively advertise their services to service directories, but clients may issue requests to service directories only

VI. SERVICE STATE MAINTENANCE TECHNIQUES

A basic feature for service discovery approaches, which has been many times underestimated, is service selection. Service selection is the phase that comes after service replies have been gathered by the service requestor. Service selection can be categorized into automatic and user assisted.

The issue of service state maintenance certainly cannot be neglected. It is a challenge to maintain accurate and valid service information and service state especially in Adhoc Networks where the inherent dynamism leads to frequent changes in service availability. One approach is to maintain a hard state of services where a provider must de-register its service/s before leaving the ad hoc network [12]. However, in adhoc networks where unpredictable disconnections occur (due to mobility, path loss, congestion and node failures). The opposite approach to hard state is to maintain a soft state of services. In this case each service record is associated with a counter, upon the expiry of which the service record is automatically deleted. It is the job of the service provider to periodically refresh that counter by re-advertising the service in order for the service to stay ‘alive’ in the caches of nodes and directories. These approaches mainly relate to the availability of a service and not its state.

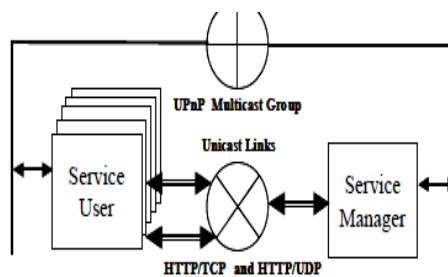


Fig 3. Two Party Service Discovery Architecture

For state maintenance one can use two mechanisms. The first mechanism is polling. Polling is used by service requestors to ask a server about the current state of a service. Polling can be performed proactively or on-demand. The second mechanism, namely notifications, is the inverse mechanism of polling. Notifications are sent by service providers (or directories) to inform clients about service state changes. In this mechanism clients have to register to the server (or to the directory) their interest in receiving service state updates.

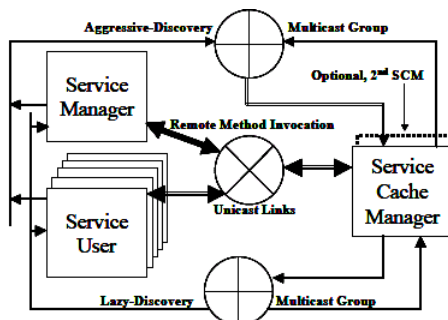


Fig 4. Three Party Service Discovery Architecture

A. Polling

In polling, a Service User periodically sends queries to obtain up-to-date information about a Service Description that was previously discovered, retrieved, and cached locally. In a two-party architecture, the Service User issues the query directly to the Service Manager from which the Service Description was obtained. Polling in three-party architecture consists of two independent processes. In one process, a Service Manager sends a Change Service request to propagate

an updated Service Description to each Service Cache Manager where the Service Description was originally cached. In the second process, each Service User polls relevant Service Cache Managers by periodically issuing a Find Service request, effectively a query with a set of desired Service Description requirements. The Service Cache Manager replies with a Match Found that contains the relevant information for any matching Service Descriptions.

B. Notification

In notification, immediately after an update occurs, a Service Manager sends events that announce a Service Description has changed. To receive events about a Service Description of interest, a Service User must first register for this purpose. In the two-party architecture, the Service User registers directly with a Service Manager. In three-party architecture, a Service User registers with a Service Cache Manager to receive events using a procedure analogous to that used by a Service Manager to propagate a Service Description. As with Service Description propagation, the Service Cache Manager grants event registrations which may be refreshed. When a Service Description update occurs, the Service Manager first issues a Change Service request to all Service Cache Managers to which it originally propagated the Service Description. The Service Cache Manager then issues a Match Found to propagate the event to all Service Users that have registered to receive events about the Service Description.

VII. CONCLUSION

Adhoc networks have attracted extraordinary attention from the research community in recent years. Due to new challenges, efficient service discovery is one of the key issues that need to be resolved. We discussed the fundamental architectures for service discovery, classified the basic discovery modes into three categories, based on the way a client acquires service information. We also presented the most important one the state maintenance which is basic feature of service discovery. The state maintenance in service discovery ensures that users eventually obtain the correct view of the service and it is a challenge maintains accurate and valid service information. We presented the two state maintenance mechanisms in which one provides information about a service and another about state changes. Finally we have identified issues that require further research, namely, service discovery, state maintenance.

REFERENCES

- [1] R. Koodli and C. E. Perkins, "Service discovery in on-demand ad hoc networks," IETF Internet Draft, draft-koodli-manet-servicediscovery-00.txt, October 2002.
- [2] "Device And Service Discovery in Mobile Ad-hoc Networks" Jeremie Albert, Serge Chaumette January 16, 2007
- [3] S. Motegi, K. Yoshihara and H. Horiuchi, "Service discovery for wireless ad hoc networks," Proc. 5th International Symposium on Wireless Personal Multimedia Communications, vol.1, 2002, pp.232-236.
- [4] Engelstad, P.E., Zheng, Y., Koodli, R., Perkins, C.E., "Service Discovery Architectures for On-Demand Ad Hoc Networks", International Journal of Ad Hoc and Sensor Wireless Networks, Old City Publishing (OCP Science), Vol. 2. Number 1, March 2006, pp. 27-58
- [5] C. Frank and H. Karl, "Consistency Challenges of Service Discovery in Mobile Ad Hoc Networks," Proc. 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), Venice, Italy, October 2004, pp. 105-114
- [6] S. Czerwinski, B. Y. Zhao, T. Hodes, A. Joseph, and R. Katz, "An Architecture for a Secure Service Discovery Service," Proc. 5th Annual International Conference on Mobile Computing and Networks (MobiCom '99), Seattle, WA, 1999
- [7] H. Hassanein, Y. Yang and A. Mawji, "A new approach to service discovery in wireless mobile adhoc networks," International Journal of Sensor Networks, Volume 2, Number 1-2 / 2007, pp.135-145.
- [8] Java Remote Method Invocation (Java RMI), 2007. available online at <http://java.sun.com/>.
- [9] Simple Object Access Protocol (SOAP), 2007. available online at <http://www.w3.org/TR/soap/>
- [10] Helal, S., "Standards for service discovery and delivery," IEEE Pervasive Computing, Volume 1, Issue 3, 2002, pp.95-100.
- [11] G.G. Richard, "Service advertisement and discovery: enabling universal device cooperation," IEEE Internet Computing, Volume 4, Issue 5, Sep/Oct 2000, pp. 18-26.
- [12] Dabrowski, C. Mills, K., and Elder, J. "Understanding Consistency Maintenance in Service Discovery Architectures During Communications Failure," Proceedings of the 3rd International Workshop on Software Performance, ACM, July 2002, pp. 168-178.
- [13] L. Cheng and I. Marsic, "Service Discovery and Invocation for Mobile AdHoc Networked Appliances," Proc. 2nd Int'l Workshop Networked Appliances (IWNA 00), Dec. 2000;

- [14] S. Helal, N. Desai, V. Verma, and C. Lee, "Konark -A Service Discovery and Delivery Protocol for Adhoc Networks," Proc. 3rd IEEE Conference on Wireless Communication Networks (WCNC), New Orleans, Louisiana, March 2003.
- [15] Rafael Moreno-Vozmediano "Resource Discovery in Ad-Hoc Grids"
- [16] Donggeon Noh and Heonshik Shin "An Effective Service Discovery Protocol for Mobile Ad Hoc Networks"
- [17] 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 2(1), 2004, pp. 23-44.
- [18] W. Adjie-Winoto, E. Schwartz, H. Balakrishnan, J. Lilley, "The design and implementation of an intentional naming system, Proc. of the 17th ACM Symposium
- [19] The Salutation Consortium, Salutation Architecture Specification, Technical Report, version 2.0c, 1999.
- [20] M. Balazinska, H. Balakrishnan, D. Karger, INS/Twine: A Scalable Peer-to-Peer architecture for Intentional Resource Discovery, Int. Conf. on Pervasive Computing, Zurich, Switzerland, 2002.
- [21] J. Liu, Q. Zhang, W. Zhu, B. Li, Service Locating for Large-Scale Mobile Ad Hoc Network, International Journal of Wireless Information Networks, Vol. 10, No. 1(2003), pp. 33-40.