Survey of Service Discovery Architectures for Mobile Ad hoc Networks

Chunglae Cho[1] and Duckki Lee[1]

[1] Computer and Information Science and Engineering Department
University of Florida, Gainesville, FL-32611, USA
{ccho, dlee}@cise.ufl.edu

Abstract. Efficient and timely service discovery is a crucial feature for the usability of mobile ad-hoc networks. In recent years a number of service discovery protocols have been developed and they have different view points toward the mobile ad hoc environment. These different view points made them have different service discovery architectures. In this paper we classify the existing service discovery protocols for mobile ad-hoc networks into several discovery architectures and survey the features of the discovery architectures. Depending on whether a directory exists or not, the service discovery architecture can be divided into two categories: *directory-less architecture* and *directory-based architecture*. The directory-based architecture can be also divided into two categories: *centralized directory architecture* and *distributed directory architecture*. The centralized directory architecture relies on one or a few centralized directories but in the distributed directory architecture, directories are further distributed and deployed dynamically. We also subdivide the distributed directory architecture, depending on whether directories reside on mobile ad hoc networks or on infrastructure-based networks, into two categories: *infrastructure-less distributed directory architecture* and *infrastructure-based distributed directory architecture*. In this paper we discuss the difference of the environments on which each of the discovery architectures is based. Moreover, we show the pros and cons of them compared with one another.

1 Introduction

Mobile ad hoc networks are characterized by their highly dynamic, multi-hop, and infrastructure-less nature. In this dynamic environment, different nodes offering different services may enter and leave the network at any time. Efficient and timely service discovery is a prerequisite for good utilization of shared resources on the network.

Discovery of services is a crucial feature for the usability of mobile ad-hoc networks. Service discovery allows devices to automatically locate network services with their attributes and to advertise their own capabilities to the rest of the network.

To enable service discovery within mobile ad hoc networks we should consider the following challenges. The first challenge is enabling resource-constrained, wireless devices to discover services dynamically. The second one is enabling service discovery in large mobile ad hoc networks. The last one is enabling bridging mobile ad hoc networks with infrastructure-based networks since Intranet and/or Internet connectivity remains the primary source of service provisioning.

In this paper we classify the existing service discovery protocols for mobile ad-hoc networks into several discovery architectures and survey the features of the discovery architectures. Depending on whether a *directory* exists or not, the service discovery architecture can be divided into two categories: *directory-less architecture* and *directory-based architecture*. A directory is an entity that stores information about services available in the network so as to enable service discovery and invocation. The directory-based architecture can be divided into two categories: *centralized directory architecture* and *distributed directory architecture*. The centralized directory architecture relies on one or a few centralized directories that store the descriptions of all services available in the network. On the other hand, in the distributed directory architecture, directories are further distributed and deployed dynamically. We can also subdivide the distributed directory architecture, depending on whether directories reside on mobile ad hoc networks or on infrastructure-based networks, into two categories: *infrastructure-less distributed directory architecture* and *infrastructure-based distributed directory architecture*.

In section 2 we present relevant work related to service discovery protocols in mobile ad hoc networks. In section 3 we present the features of the service discovery architectures and compare them with one another. Finally, we conclude this paper in section 4.

2 Related Work

In this section we provide extensive description of the existing service discovery protocols, which are grouped according to the taxonomy defined in the introduction of this paper.

2.1 Directory-less Architecture

DEAPspace. DEAPspace [6] investigates completely decentralized discovery solutions, specifically suited to wireless ad hoc networks. DEAPspace is a pure push solution, in which all devices hold a list of all known services, the so called “world view”. Each device periodically broadcasts its “world view” to its neighbors, which update their “world view” accordingly.

UPnP. The Universal Plug-and-Play (UPnP) [11] is architecture for pervasive peer-to-peer network connectivity of intelligent appliances, wireless devices, and PCs of all form factors. UPnP is more than just a simple extension of the plug and play peripheral model. With UPnP, a device can dynamically join a network, obtain an IP address, convey its capabilities, and learn about the presence and capabilities of other devices—all automatically; truly enabling zero configuration networks.

The UPnP network consists of three basic building blocks: *devices*, *services* and *control points*. A device is a container for services and nested devices. A service is basically the smallest unit of control, which offers a set of actions. A control point in a UPnP network provides capability of discovering and controlling devices by receiving device and service descriptions, invoke service actions, or subscribe to a service's event source in order to be notified each time the service changes its state. Concerning discovery, the Simple Service Discovery Protocol (SSDP) was created as a lightweight discovery protocol for UPnP initiative, and defines a minimal protocol for multicast-based discovery. The SSDP allows for a control point to look for devices and services, and for a device to announce its availability. A UPnP control point sends out an SSDP search request to discover devices and services available on the network. A UPnP device in turn listens to the multicast port.

Konark. Konark [5] is a middleware designed specifically for the discovery and delivery of services in multi-hop ad hoc networks. With regard to the service discovery mechanism, Konark supports both push and pull modes, with a cache in all devices. With regard to service description, Konark defines an XML based service description language similar to WSDL. Konark uses multicast to advertise and discover services and allows for service delivery by running a lightweight HTTP server on every device that hosts services. In Konark, every node maintains a service registry, where it stores information about its own services and also about services that other nodes provide. This registry is actually a tree-structure with a number of levels that represent service classification.

PDP. PDP [1] is a fully-distributed service discovery protocol designed for ad hoc networks. PDP takes into account inherent limitations of embedded devices such as power-constrained batteries and processing capabilities, in such a way that the protocol reduces the number of messages sent through the network.

PDP prioritizes the replies of the less limited devices, allowing the others to abort their answers. PDP also does away with the need for the central server, and it is a fully distributed protocol that merges characteristics of both pull and push solutions. Devices maintain a cache of services previously announced, that is also used for the answers. In PDP all messages are broadcast, and all devices cooperate by coordinating their replies and sharing the information in their caches. PDP takes into account the different applications needs, and this allows to further reducing the power consumption.

2.2 Centralized Directory Architecture

Salutation. Salutation [9] is the major cooperation architecture to solve the problems of service discovery and utilization among a broad set of appliances and equipment and in an environment of widespread connectivity and mobility. The Salutation architecture defines an entity called the Salutation Manager (SLM) that functions as a directory of applications, services and devices, generically called “Networked Entities". The SLM allows networked entities to discover and to use the capabilities of the other networked entities.

Jini. The purpose of a Jini [10] system is to federate users, devices and services into a distributed system providing on the one hand a single, integrated view on it and, on the other hand, a highly dynamic infrastructure for devices and services to join and detach from the network.

The notion of a *service* is the central concept of Jini. A service (represented as Java object) is a software entity providing any kind of computation or controlling a hardware device, e.g. a file converter or a printer service. Services can make use of each other, and one service can be client of another one. Communication between services is based on Java Remote Method Invocation (RMI).

Services are made available to potential clients in the network by putting a corresponding service entry into a directory or lookup service (similar to the Directory Agent in SLP), named the Jini Lookup Service (JLS), which maps interfaces indicating the service's functionality to an object which implements the service. For addition of a service to a lookup service, two steps have to be performed: a *discovery* step to find out available lookup services that a service can register with, and a *join* step to perform the registration.

SLP. The Service Location Protocol (SLP) [3] is a service discovery protocol designed for TCP/IP networks and is scalable up to enterprise environments. SLP defines three "agents": User Agents (UA), that perform service discovery on behalf of client software, Service Agents (SA), that advertise the location and attributes on behalf of services, and Directory Agents (DA), that store information about the services announced in the network. SLP has two different modes of operation: (1) when a DA is present, it collects all service information advertised by SAs, and UAs unicast their requests to the DA, and (2) when there is not a DA, UAs repeatedly multicast the request. SAs listen for these multicast requests and unicast responses to the UA.

SLP provides a flexible and scalable framework for providing hosts with access to information about the existence, location, and configuration of networked services by scope mechanism. SLP uses service scope, which is a collection of services within certain domain, to achieve scalability, e.g. a set of services can be assigned to a given department of an organization, to a certain building, or geographical area for a certain purpose. However, it is still not suitable for large-scale mobile ad-hoc network.

2.3 Infrastructure-less Distributed Directory Architecture

Sailhan2005. F. Sailhan, et al. [8] proposed a service discovery protocol aimed at large-scale mobile ad hoc networks (i.e., comprising at least about 100 nodes). In this discovery architecture, directories are distributed and deployed dynamically for the sake of scalability. The discovery architecture is structured as a virtual network. A virtual network is composed of a subset of nodes of the MANET acting as directories. These directories represent a backbone of nodes responsible for performing service discovery. They are deployed so as at least one directory is reachable in at most a fixed number of hops, *H*. Directories cache the descriptions of services available in their vicinity which is defined by *H*. A client simply sends a query to the directory for local service discovery. If the description of the requested service is not cached by the local directory, the directory selectively forwards the query to other directories so as to perform global discovery. Selection of directories to which service queries are forwarded, is based on the exchange of profiles among directories. The directory profile provides a compact summary of the directory's content and a characterization of the host capacity.

F. Sailhan, et al. proposed dynamic directory deployment that directories are dynamically elected among mobile nodes. Any mobile node can be elected as a directory. The criterion for the election is the capacity of mobile nodes in terms of available resources and its context parameters characterizing the node's environment. The main selection criterion is the directory's coverage. This architecture can be suitable to the mobile ad hoc network which vision is to provide networking capabilities to mobile devices without requiring any infrastructure. It assumes a mobile ad hoc network composed of nodes holding the same network interface, with IP-level connectivity using the underlying routing protocol. However, it also considered that some nodes hold several network interfaces, and act as gateways with other networks, either ad hoc or infrastructure-based. This defines a hybrid network bridging mobile ad hoc networks and infrastructure-based networks, according to the specific networking capabilities of the wireless nodes.

DSDP. U. Kozat, et al. [4] proposed a distributed service discovery protocol (DSDP) architecture that relies on a virtual backbone for locating and registering available services within a dynamic network topology. The architecture consists of two independent parts: backbone management (BBM) phase and distributed service discovery (DSD) phase. The first part, BBM phase, selects a subset of the network nodes to form a relatively stable dominating set, discovers the paths between dominating nodes, and adapts to the topology changes by adding/removing network nodes into/from this dominating set. BBM utilizes only a 1-hop local broadcast control message (Hello Beacon) for forming the backbone, creating virtual links between backbone nodes, and maintaining the backbone. After the first part is successfully carried out, a virtual backbone constitutes a mesh structure with the backbone nodes and the virtual links connecting them. The second part, DSD phase, is used to distribute the request and registration messages from clients and servers to the directories (i.e. backbone nodes). These messages assist in forming multicast trees rooted at client and server nodes on top of the backbone mesh.

2.4 Infrastructure-based Distributed Directory Architecture

VIA. The Verified Information Access (VIA) [2] is an application-level protocol that enables data sharing among discovery domains. Each directory maintains a table of active links to other directories that share related information. A set of linked directories forms a data cluster that can be queried by devices for information. The data cluster is distributed, self-organizing, and responsive to data mobility. Using application-defined data schemas, clusters organize themselves into a hierarchy.

In VIA the master directory service resides on a "*gateway*" that connects a discovery domain to a "*main channel*" allowing communication between gateways. Gateways act as mediators between devices in a discovery domain and resources available in other domains. The main channel acts as a broadcast mechanism between gateways and was modeled as an IP-multicast group. A client sends all queries to the main channel. The gateways can self-organize into a hierarchy based on the types of queries in the main channel as well as the actual data items being shared on the network.

Superstring. Superstring [7] is a service discovery protocol designed for environments where the services are extremely dynamic, and where queries are resolved in the core of the network, which is relatively static and composed of capable servers and high-bandwidth networks. Superstring shifts much of the cost of querying and advertisement to the fixed infrastructure, and focuses on minimizing memory and computation overhead in this infrastructure.

Superstring combines peer-to-peer and hierarchical topologies for directories. It utilizes a flat topology to discover top-level nodes that specialize in a particular kind of service. From this top-level node, a hierarchy is created which reflects the hierarchical structure of service descriptions. Queries are resolved by the specialized hierarchy. To begin with, a single service description is distributed over several nodes. Thus, each directory stores only a small part of the entire description. In Superstring, no single directory processes the entire query. The cost is shared among many directories in a fashion that minimizes memory overhead.

3 Comparison of Service Discovery Architectures

In recent years a number of service discovery protocols have been developed and they have different view points toward the mobile ad hoc environment. These different view points made them have different service discovery architectures. In this section we discuss the difference of the environments on which each of the discovery architectures is based. Moreover, we show the pros and cons of them compared with one another. For comparison, we partly referred to some research efforts that have been focused on the classification of the service discovery protocols in mobile ad hoc environments and the evaluation of those architectures [12, 13].

3.1 Directory-less Architecture

In the directory-less architecture, service providers do not distribute their service descriptions onto other nodes in the network, but leave them stored on their own device. This leads to a reactive service trading: service requests have to be forwarded to all members of the network where they are compared with the stored descriptions. To do that, a device interested in a special service typically sends its search message to all reachable nodes. If one or more of these nodes can satisfy the request, a response is sent back to the requestor. In any case, the search message is forwarded to all reachable nodes except for the previous sender. To reduce duplicate node querying, sophisticated flooding algorithms have been proposed.

Generally, these broadcasting mechanisms are not suited for mobile ad hoc networks due to their heavy consumption of bandwidth and energy, which are not unlimitedly available on mobile devices. Therefore, the network size supported by the directory-less architecture is very limited. Nevertheless, in regions with extremely high dynamics, broadcasting could be the only possible technique.

Comparison. DEAPspace envisions dynamic ad hoc network, and pursues decentralized solution as an alternative to the centralized approach. However, DEAPspace employs a periodic broadcast scheme, which could cause multicast storm. Unlike to DEAPspace, Konark uses Konark service gossip algorithm. In this algorithm, unnecessary repeated delivery is suppressed by caching and the multicast is scheduled to wait a random time, so that the network can avoid the problem. The unique feature in UPnP is self-configuration; truly enabling zero configuration networks, but it doesn't support caching service information. Konark and PDP maintain a cache of services, so it can reduce unnecessary service discovery if it is previously announced. PDP is proposed to solve inherent limitations of embedded devices such as power-constrained batteries and processing capabilities, reduces the number of messages sent through network by merging characteristics of both pull and push solutions.

3.2 Centralized Directory Architecture

In contrast to the directory-less architecture, the centralized directory architecture rely on a central directory that stores the descriptions of all services available in the network so as to enable service discovery and invocation. Discovery of the directory by clients and service providers is in general based on multicasting. Then, service providers advertise their services to the central directory using a unicast message. And, to access a service, a client first contacts the central directory to obtain the service description, which is then used to interact with the service provider.

Centralized resource discovery is much suited to wireless infrastructure-based networks. However, this architecture makes the service discovery process dependent upon the availability of the central directory, which further constitutes a bottleneck. In addition, a centralized directory limits its scope to devices within a local service discovery domain. The boundaries of a service discovery domain can be administratively defined such as an IP subnet, or they can be the result of a physical property such as the range of a wireless network.

Comparison. Service discovery in Salutation is defined on a higher layer, and the transport layer is not specified. Thus, Salutation is independent on the network technology and may run over multiple infrastructures. However, Salutation does not address features like remote event notification, which are no doubt useful in distributed environment. One special feature of Jini is the mobile Java codes, which may be moved among clients, services, and directories. The advantage of Jini is its platform independency, but the disadvantage is that all the clients, services, and directories depend on Java runtime environments directly or indirectly. SLP is more flexible and scalable than other protocols belonging to the centralized directory architecture. Since SLP is able to operate with or without a DA, it is suitable for networks of different sizes, ranging from very small ad hoc connectivity to middle-sized networks.

3.3 Distributed Directory Architecture

The motivation that support the use of the distributed directory architecture for service discovery is that scalability can be achieved when the network size becomes larger. The network size that the directory-less architecture and the centralized directory architecture can support is very limited. The view point on the network environment to which the distributed architecture is applied is different from that of the directory-less architecture and centralized directory architecture. It is assumed that there are a large number of devices in a service discovery domain.

Infrastructure-less Distributed Directory Architecture. In case of the infrastructure-less distributed directory architecture, it is assumed that large number of nodes can exist in a mobile ad hoc network. The architecture assumes that hundreds of devices could participate in a service discovery domain.

This architecture is quite suited to the mobile ad hoc network scenario. Directories are dynamically selected among mobile nodes which have suitable capability (e.g. battery power, memory, processing power, node coverage, etc). It does not require pre-defined infrastructure. Mobile ad hoc networks are characterized by their highly dynamic, multi-hop, and infrastructure-less nature. If any fixed infrastructure is assumed in the protocol architecture, it would be against the nature of the ad hoc networking.

Dynamic assignment presents an extra load to the network in terms of selecting the directories adaptive to the topology changes as well as informing the rest of the network about the identities of the directories. However, compared to the directory-less architecture, there are the major advantages inherent to using directories. First of all, scalability is achieved when network size becomes larger. Secondly, Response time for locating services reduces. Lastly, Directories can apply load balancing techniques that can reduce the load on individual directories and enhance the service performance.

Even though this architecture is primarily designed for the pure mobile ad hoc network, it can also be used in the hybrid network using the concept of a "*gateway*". The gateway resides on the edge of a mobile ad hoc network and a fixed network (e.g., wired network) and act as a directory.

Infrastructure-based Distributed Directory Architecture. In case of the infrastructure-based distributed directory architecture, it is assumed that there are many separated mobile ad hoc networks and each mobile ad hoc network is connected to the fixed network that already has infrastructure for maintaining directories. It is argued that most of the ad hoc networks deployed today are hybrid networks instead of pure ad hoc networks. The architecture assumes the environment that more than a million of nodes could participate in a service discovery domain. Several mobile ad hoc network islands are connected through the fixed network. A client in a mobile ad hoc network needs to find a service that resides on another mobile ad hoc network that is indirectly connected through a fixed network. In this case, directories are on the fixed network that has infrastructure.

While the cost of communication is of primary concern in small groups of devices communicating over wireless links, this cost is of less concern in the wide-area, where much of the cost is absorbed by highly capable servers, connected to high bandwidth networks. In these wide-area environments, computation costs and memory overhead are often as important, or more so, than communication costs. Therefore, the discovery protocols based on the infrastructure-based distributed directory architecture are designed to reduce the computation costs and memory overhead.

In VIA, the top level directory must receive all queries, but it only passes relevant queries to its children. Therefore, a subset of the directories in the system is involved in query resolution for all queries. In Superstring, a single service description is distributed over several directories. Thus, each directory stores only a small part of the entire description so that no directory processes the entire query.

4 Conclusion

In this paper, we explored several architectural choices for service discovery in mobile ad hoc networks. We classified the existing service discovery protocols according to the taxonomy based on service discovery architecture and compared those architectures with one another.

The directory-less architecture is in accord with the nature of mobile ad hoc network that is infrastructure-less but its communication overhead limits its scalability. The centralized directory architecture is suited to wireless infrastructure-based networks. However, the service discovery process is dependent upon the availability of the central directory, which further constitutes a bottleneck. As a result, its scalability is also limited. To overcome these problems, the infrastructure-less distributed directory architecture is proposed to be applied to the large mobile ad hoc networks. On the other hand, the infrastructure-based distributed directory architecture can be used for the environment that many mobile ad hoc networks are connected through an infrastructure-based network.

References

1. C. Campo, M. Munoz, J. C. Perea, A. Marın, C. Garcıa-Rubio: PDP and GSDL: a new service discovery middleware to support spontaneous interactions in pervasive systems, Proceedings of the 3rd Int'l Conf. on Pervasive Computing and Communications Workshops (2005)
2. P. Castro, B. Greenstein, R. R. Mum, P. Kermani, C. Bisdikian, M. Papadopouli: Locating Application Data Across Service Discovery domains, Mobile Computing and Networking (2001)

3. E. Guttman, C. Perkins, J. Veizades, M. Day: Service Location protocol, Version 2, IETF, RFC2608 (1999)
4. U. C. Kozat, L. Tassiulas: Service discovery in mobile ad hoc networks: an overall perspective on architectural choices and network layer support issues, Ad Hoc Networks, 2, (2004)
5. S. Helal, N. Desai, V. Verma, C. Lee.: Konark - A Service Discovery and Delivery Protocol for Ad-hoc Networks, Proceedings of the Third IEEE Conference on Wireless Communication Networks (WCNC), New Orleans (2003)
6. M. Nidd: Service Discovery in DEAPspace. IEEE Personal Communications, (2001) 39-45
7. R. Robinson, J. Indulska: Superstring: A Scalable Service Discovery Protocol for the Wide-Area Pervasive Environment, The 11th IEEE International Conference on Networks (2003) 699–704
8. F. Sailhan, V. Issarny: Scalable Service Discovery for MANET, Proceedings of the 3rd IEEE Int'l Conf.on Pervasive Computing and Communications (2005)
9. Salutation Consortium: Salutation Architecture Specification. Version 2.0c (1999)
10. Sun Microsystems: Jini Technology Core Platform Specification. Version 2.0 (2003)
11. UPnP Forum: Understanding Universal Plug and Play White Paper, http://www.upnp.org (2000)
12. F. Zhu, M. Mutka, L. Ni: Classification of Service Discovery in Pervasive Computing Environments, MSU-CSE-02-24, Michigan State University, East Lansing (2002)
13. P. E. Engelstad, Y. Zheng: Evaluation of Service Discovery Architectures for Mobile Ad Hoc Networks, the Second Annual Conference on Wireless On-demand Network Systems and Services (WONS’05) (2005)