Service Advertisement and Discovery in Mobile Ad hoc Networks

Liang Cheng
Department of Computer Science and Engineering, Lehigh University
19 Memorial Drive West
Bethlehem, PA 18015
+1 610 758 5941
cheng@cse.lehigh.edu

ABSTRACT

Service advertisement and discovery is an important component for ad hoc communications and collaboration in ubiquitous computing environments. In this paper, a lightweight service advertisement and discovery protocol for MANET (Mobile Ad hoc NETworks) based on ODMRP (On-Demand Multicast Routing Protocol) is presented. Its service advertisement and discovery information is piggybacked in ODMRP routing control packets. Thus the implementation workload and resource consumption are lightweight as long as mobile devices in MANET support multicast routing functionality. The protocol avoids excessive traffic overhead of periodic advertisements since only updated service advertisements are distributed and the service query/reply mechanism in the pull model is used. Moreover, the pull model deals with mobility in MANET because the advertised and registered service can be obsolete after a certain period of time due to the mobility of either the server or client devices.

Keywords

Service advertisement and discovery, mobile ad hoc networks

INTRODUCTION

A service is an entity that can be used by a person, a software program, or another service. It may be a computation, storage, a communication channel to another user, a software filter, a hardware device, or another user. Service advertisement and discovery is an important component for ad hoc communications and collaboration in ubiquitous computing environments since it enables communication and collaboration entities to provide services to peers and to be aware of and use the available services from peers. In this paper, service discovery is an interchangeable terminology to the service advertisement and discovery.

Related Work

The topic of service advertisement and discovery has been extensively researched in technologies such as Jini [3] and the service location protocol [7].

Jini

The Jini technology infrastructure and programming model are built to enable services to be offered and found in networking environment. These services take advantage of Jini technology to announce their presence to other services and users, to discover each other, and to make calls to each other. In this paper, the focus is the service advertisement and discovery in Jini technology, which includes a trio of protocols called discovery, join, and lookup.

A pair of these protocols, discovery/join, fulfills the service advertisement functionality. It occurs when a device is plugged in. The discovery protocol is used when a service looks for a lookup server with which to register. The join protocol is used when a service has located a lookup server and wishes to join it. The lookup protocol is used when a client or user needs to locate and invoke a service described by its interface type (written in the Java programming language) and possibly, other attributes.

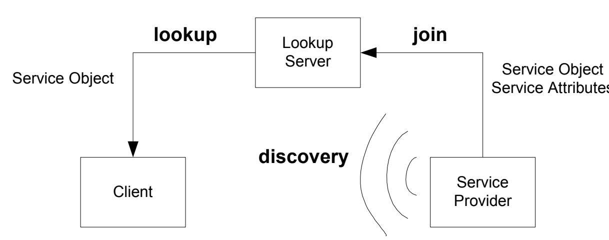

Figure 1. Jini processes of discovery/join/lookup.

Figure 1 outlines the discovery/join/lookup processes in Jini. The discovery/join is the process of adding a service to a Jini system. A service provider, e.g., a device or software, is the originator of a service. First, the service provider locates a lookup server by multicasting a request on the local network for any lookup server to identify itself (discovery in Figure 1). Then, a service object for the service is loaded into the lookup server (join in Figure 1), which contains a Java programming language interface for the service including the methods that users and applications will invoke to execute the service, along with any other descriptive attributes. The service is now ready to be looked up and used.

A client locates an appropriate service by its type, i.e., by its interface written in the Java programming language, along with descriptive attributes that are used in a user interface for the lookup server. Then the service object is loaded into the client to be invoked.

Service Location Protocol

Service Location Protocol (SLP) establishes a framework for service discovery using three types of agents that operate on behalf of network-based software: (*i*) Service Agents (SA) advertise the location and attributes on behalf of services, (*ii*) Directory Agents (DA) aggregate service information, and (*iii*) User Agents (UA) perform service discovery on behalf of client software.

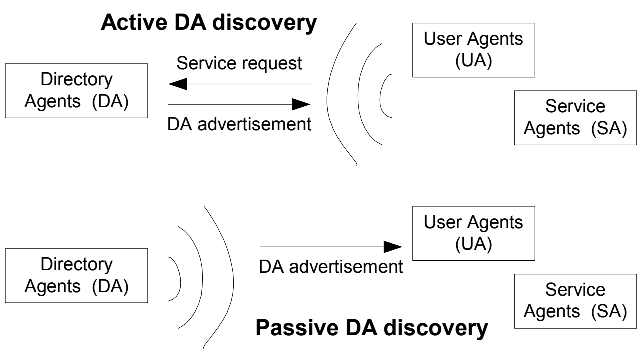

Figure 2. Methods of service discovery in SLP.

Figure 2 illustrates active and passive methods for service discovery in SLP. In active discovery, UAs and SAs multicast SLP requests to the network. In passive discovery, DAs multicast advertisements for their services and continue to do this periodically in case any UAs or SAs have failed to receive the initial advertisement. SLP has two modes of operation: (*i*) when a DA is present, it collects all service information advertised by SAs, and UAs unicast their requests to the DA, and (*ii*) in the absence of a DA, UAs repeatedly multicast the same requests they would have unicast to a DA. SAs listen for these multicast requests and unicast responses to the UAs if it has advertised the requested service.

Discussion

The Jini discovery architecture is similar to that of SLP. Jini agents discover the existence of a Jini Lookup Server, which collects service advertisements in a manner analogous to DAs in SLP. Jini agents then request services on behalf of client software by communicating with the Lookup Server. Unlike SLP, however, where DAs are optional, Jini requires the presence of one or more Lookup Servers.

The above service discovery approaches encounter overhead problem when they are applied to mobile ad hoc networks (MANET) because they were not designed for mobile ad hoc networks by considering that MANET has dynamic network topology. They have to send explicit multicast packets for service discovery frequently in addition to the multicast control packets. This wastes precious bandwidth and battery resources, and causes extra traffic in MANET. In this paper, a lightweight protocol for service discovery in MANET is presented, which is based on On-Demand Multicast Routing Protocol (ODMRP).

Section 2 presents the new service advertisement and discovery protocol in terms of bootstrapping mechanism in the push model and the query/reply mechanism in the pull model, based on the ODMRP, to achieve lightweight service discovery and deal with device mobility. Section 3 discusses the protocol performance and concludes this paper with future work.

SERVICE DISCOVERY IN MANET

In this research, service advertisement and discovery addresses how network devices discover what types of services exist in the network. Multicast is used for service discovery in MANET because of (*i*) the broadcast nature of wireless communication environment, (*ii*) the infrastructureless feature of MANET, and (*iii*) the one-to-many character of service advertisement and discovery. Since multicast protocols are not as well established and supported in infrastructure-less networking environment as in infrastructured networks, multicast protocols for MANET are studied first.

Multicast Protocol for MANET

Several multicast protocols have been proposed to support multicast in mobile ad hoc networks. A performance-comparison study of multicast protocols for MANET in a common realistic simulation environment is reported in [6]. It provides quantitative performance analysis of five protocols with different characteristics: adhoc multicast routing (AMRoute) [2], on-demand multicast routing protocol (ODMRP) [5], ad hoc multicast routing protocol utilizing increasing id-numbers (AMRIS) [8], core-assisted mesh protocol (CAMP) [4], and flooding.

According to the research results, ODMRP is very effective and efficient in most simulation scenarios comparing to other multicasting routing protocols in MANET. Thus, in this research, ODMRP is chosen to support service discovery in MANET. Based on the ODMRP, both push and pull models are implemented for service discovery in MANET considering service mobility and network-resource saving. One approach is the bootstrapping mechanism in the push model. The other is the query/reply mechanism in the pull model. Assume that all protocols are IP based.

Bootstrapping Mechanism in the Push Model

Every mobile device that provides a service, which is called a server, multicasts an advertisement in MANET. Each server and its possible consumers make a multicast group. The advertisement includes the service name, the server's unicast address/port pair, and the protocol type that a client should use to request the service. Table 1 illustrates the advertisement format. Table 2 shows the combination of an ODMRP join query packet header with the service advertisement, where the "Reserved" field specifies whether a service advertisement is attached. The multicast group IP address for the service is the IP address of the multicast group that consists the server and its possible consumers.

Table 1. Format of Service Advertisement

1 byte	1 byte	1 byte	1byte
Type	Option Field	Time to Live	Service Port
Server Address (in unicast form)			
Service Name (ID)			
Protocol Type		Reserved	
Optional Fields			

Type:	0x01 as service advertisement.
Service Port:	The port number of the service point
Option Field:	0, no optional information 1, optional fields at the end
Server Address:	The unicast IP address of the server
Service Name:	The index or the name of the service
Time to Live:	Geographical scope the service covers in terms of the number of hops
Protocol Type:	Protocol type that the server and clients use
Reserved:	0, ignored on reception
Optional Fields:	Service description for multi-level services

Table 2. Revised ODMRP *Join Query* Packet Header

1 byte	1 byte	1 byte	1byte
Type	Reserved	Time to Live	Hop Count
Multicast Group IP Address for the Service			
Other Fields in ODMRP Join Query Packet Header			
Service Advertisement			

Any mobile device with applications interested in this service stores the advertisement in its local service registry, and sends a service awareness reply subject to resource limitations. A join reply packet is used without any attachment because it already provides sufficient information as a service awareness reply. Table 3 highlights the necessary fields in the join reply packet as a service awareness reply, where the "Reserved" field specifies whether a packet is a service awareness reply. The multicast group IP address for the service is the IP address of the multicast group, which consists the server and its possible consumers.

Table 3. Format of ODMRP *Join Reply* Packet as a Service Awareness Reply

1 byte	1 byte	1 byte		1byte
Type	Count	R	F	Reserved
Multicast Group IP Address for the Service				
Other Fields in ODMRP Join Reply Packet				
Sender IP Address (in unicast form)				
Other Fields in ODMRP Join Reply Packet				

Once some clients send back service awareness replies, the server sends its updated service advertisements by piggyback in ODMRP join query packets. Otherwise it will wait for explicit queries from clients and reply back in the pull model, which is illustrated in the following section.

Query/Reply Mechanism in the Pull Model

Assume that services have well-known service names, such as "printing" or "address book", and corresponding unique service IDs.

Once a device or an application needs a service, it first gets the service ID and then sends a query to a well-known multicast address, which corresponds to the service query multicast group, asking about the existence of the service. The service query multicast group consists of servers and devices that support the service query/reply mechanism.

The format of the service query message is the same as that of the service advertisement, which is illustrated in Table 1. However, the content is different: (*i*) the "Type" field is set to 0x02 as service query, (*ii*) the "Option" field is 0, and (*iii*) the "Service Name" is the service ID of the queried service. Since it is also piggybacked in an ODMRP join query packet, correspondingly, the "Reserved" field in Table 2 specifies whether a service query is attached. Moreover the Multicast Group IP Address field is set to the well-known multicast address for service queries.

The format of the reply message for a service query is also the same as that of the service advertisement illustrated in Table 1. When a device receives a service query, if it supports the service query/reply mechanism, it waits for a random time to check whether there is already a reply message with latest information that has been issued by a device. If affirmative, it ignores the service query message.

Otherwise, it fills in the blank fields such as Time to Live, Server Address, Service Port, Protocol Type, and/or the Option Field(s), if such information is available locally, and sends the reply message. The server always replies to the service query message with the updated service information. The random timer may suppress multiple reply copies being sent at the same time to save bandwidth.

CONCLUSIONS

Based on the ODMRP (On-Demand Multicast Routing Protocol), a lightweight service advertisement and discovery protocol is designed for ad hoc communications and collaboration in ubiquitous computing environments. Its service discovery information is piggybacked in the ODMRP routing control packets. Therefore the implementation workload and resource consumption are lightweight as long as the mobile devices support multicast routing functionality in mobile ad hoc networking environment.

Moreover this implementation avoids excessive traffic overhead of periodic advertisements since only updated service advertisements are distributed and the service query/reply mechanism in the pull model is used. Furthermore the pull model deals with mobility in the network because the advertised and registered service can be obsolete after a certain period of time due to the mobility of either the server or client devices.

Future Work

It is observed that existing multicast protocols, either with tree configuration or mesh configuration such as ODMRP, have not considered the signal-to-noise ratio (SNR) of the wireless communication link. In fact, in MANET, minimum-hop design without considering SNR information is not always desirable. For example, if the minimum-hop path includes a “weak” link, illustrated in Figure 3, then data transmitted along this path may incur a significant amount of packet loss. Moreover, computing minimum-hop paths when a certain link is marginal (i.e., oscillating between the up and down states) will lead to route oscillation. The subsequent out-of-order arrival of packets of a particular session at the destination node results in inefficiencies in certain higher-layer protocols [1]. In the scenario of this research, the service discovery in MANET, especially when we consider service invocation using mobile objects, would suffer from the "weak link" and the route oscillation. Thus we will study an SNR-based ODMRP (SODMRP) in MANET for ad hoc communications and collaboration in ubiquitous computing environments.

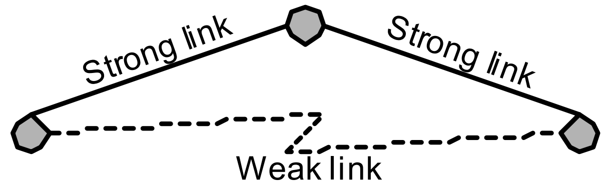

Figure 3. Min-hop path vs. quality-link path

REFERENCES

1. Bellur, B., Lewis, M.G., and Templin, F.L. An ad-hoc network for teams of autonomous vehicles, in *Proc. First Annual Symposium on Autonomous Intelligence Networks and Systems*, Los Angeles, CA, May 2002.
2. Bommaiah, E., Liu, M., McAuley, A., and Talpade, R. AMRoute: ad-hoc multicast routing protocol. *Internet Draft*, work in progress, August 1998.
3. Edwards, W. K. *Core Jini*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2000.
4. Garcia-Luna-Aceves, J., and Madruga, E. The core-assisted mesh protocol. *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 8, pp. 1380–1394, 1999.
5. Lee, S., Gerla, M., and Toh, C. On-demand multicast routing protocol (ODMRP) for ad hoc networks. *Internet Draft*, work in progress, June 1999.
6. Lee, S., Su, W., Hsu, J., Gerla, M., and Bagrodia, R. A performance comparison study of ad hoc wireless multicast protocols, in *Proc. IEEE Infocom'2000*, pp. 565–574, Tel-Aviv, Israel, March 2000.
7. Veizades, J., Guttman, E., Perkins, C., and Kaplan, S. Service location protocol. *Request for Comments* (*RFC*) 2165, Internet Engineering Task Force, June 1997.
8. Wu, C., Tay, Y., and Toh, C. Ad hoc multicast routing protocol utilizing increasing id-numbers (AMRIS) functional specification. *Internet Draft*, work in progress, November 1998.