A Secure Service Discovery Protocol for MANET*

Yuan Yuan Ashok Agrawala
Computer Science Technical Report CS-TR-4498, and
UMIACS Technical Report UMIACS-TR-2003-67
Department of Computer Science
University of Maryland
College Park, Maryland 20742

***Abstract*:** Service discovery technologies are exploited to enable services to advertise their existence in a dynamic way, and can be discovered, configured and used by other devices with a minimum of manual efforts. Automatic service discovery will play essential role in future network scenarios. Especially, the development Mobile Ad Hoc Networks (MANETs) to support the proliferation of mobile devices and emergence of pervasive computing gives rise to the challenges of the service discovery techniques, because MANET allows these devices to communicate dynamically without fixed infrastructure and centralized administration. In this paper, we present a dynamic service discovery infrastructure that uses XML to describe services and match using the semantic content of service descriptions for MANET. We believe that the architecture we have designed is a necessary component of any discovery of non-infrastructure services effectively and correctly. We further exploit the secure and performance issues of this infrastructure.

1. Introduction

We refer service as work or resource contributed by one or more entities that can help accomplishing the task of other entities. To make greater utilization of resources in vicinity, it is important for nodes in MANET to be able to discover remote services seamlessly and carry out transactions with the service providers, while security is paramount to the success of the transaction. However all these processes are complicated by the fact that there is no fixed infrastructure and established administration.

In MANET, each node can be a combination of service user, service provider and service directory, which caches the service providers in vicinity. Therefore a decentralized approach is required for maintaining service and information about service objects. Each node needs a local registry to effectively manage, advertise and discover services. Resource discovery and management is a real challenge for MANET. Nevertheless researches in the field of MANET have primarily focused on

* This work funded by MIND Laboratory

routing and several routing protocols like DSR [1], AODV [2], TORA [3], DSDV [4], which have come up in the past few years and primarily deal with the problem of managing the flow of data from a mobile node in the MANET to another node in the same MANET with the help of intermediate mobile nodes. The primary assumption in such routing protocols is that the destination address of the data packets is known to the source nodes beforehand.

Often mobile nodes inside of MANET need to utilize resources or services that are present on other mobile nodes in their neighborhood. Hence, it is important for mobile nodes to be able to seamlessly discover other remote services or resources presented on nearby mobile devices and carry out transactions with the service providers. Obviously standard routing protocols are unsuitable for service discovery since the destination address of the service is still unknown. In this paper, we first present popular service discovery protocols existed nowadays, then discuss the problem when applying these protocols into MANET. We give out our scheme for service discovery and a secure mechanism for evaluating and utilizing discovered services in section III. Section IV concludes this paper and Section V proposes our further work.

2. Service Discovery Protocols

Discovery protocols enable software components to find each other on a network, and to determine if discovered components match their requirements. Further, discovery protocols include techniques to detect changes in component availability, and to maintain, within some time bounds, a consistent view of components in a network. Computer networking community approaches the problem from a various perspective and proposes different architectures and protocols for service discovery, including JINI [5], UPnP [6], Salutation [7], SLP [8], etc. Among these well-known contenders, Universal Plug and Play, Jini and Salutation architectures are prominent, coming primarily from the industry. SLP is a standard developed by IETF (Internet Engineering Task Force) working group. This section briefly presents and evaluates the leading technologies and then discusses the problems applying them into MANET.

2.1 The Service Location Protocol (SLP)

The SLP defined by IETF protocol is a language-independent protocol for automatic service discovery on Internet Protocol (IP) based networks. SLP infrastructure consists of three agents: *User Agent (UA), Service Agent (SA), and Directory Agent (DA)*. UA is a software entity that sends service discovery requests on behalf of a user application, while SA advertises location and available attributes on behalf of a service. As a centralized service information repository, DA caches advertisements from SAs, maintain a list of services in a network, and, afterwards, responds to requests from UAs.

A SA advertises itself by registering with a DA. This registration message contains the URL for the advertised service, lifetime for the service, and a set of descriptive attributes for the service. There are several mechanisms in SLP to discovery services. In active discovery, UAs and SAs multicast SLP requests or use DHCP to discover DAs and retrieve or register corresponding services, while in passive discovery, DAs multicast advertisements for their known services and repeat these advertisements periodically to facilitate UAs and SAs to initialize or update DAs information. When a DA exists, UAs

and SAs unicast their Attribute Requests and Service Registrations respectively to find and register appropriate services. The presence of one or more DAs can substantially improve performance by reducing the multicast messages and the amount of network bandwidth used. However, SLP also can conduct in the distributed scenario without DAs, where UAs repeatedly multicast Attribute Request to network and receive the unicast replies from SAs directly. This mode provides a relative simple structure for service discovery in small network (such as home LAN), but tends to increase the bandwidth usage.

2.2 Jini™ network technology

A Jini system is a distributed system where the overall goal is to turn the network into a flexible, easily administered tool on which human and computational clients can find resources. The JINI system extends the Java application environment that provides a good computing platform for distributed computing because both code and data can move from machine to machine. The Services communicate with each other by using a service protocol, which is a set of interfaces written in the Java programming language. The Services are found and resolved by a lookup service. The lookup service maps interfaces indicating the functionality provided by a service to sets of objects that implement the service.

A pair of protocols called "discovery" and "join" adds a service to a lookup service. The service locates an appropriate lookup service (by using the "discovery" protocol), and then it joins it (by using the "join" protocol). The communication between services is accomplished using Java Remote Method Invocation (RMI). RMI is a Java programming-language-enabled extension to traditional remote procedure call mechanisms.

This alternative has some drawbacks. The code mobility appears as an efficient solution for supporting information retrieval. The problem is that the code mobility is very complex in an environment which bandwidth is a scarce resource and users' mobility makes continuous communication. Since code mobility gives to the users the access to other machines, security is a concern and the literature hardly addresses the security of MANET.

2.3 The Universal Plug and Play (UPnP)

The UPnP headed by Microsoft, is an industry initiative designed to enable simple and robust connectivity among stand-alone devices and PCs from many different vendors. Unlike Jini, taking the standardization to one extreme, UPnP concentrates on device autonomy we mean the use of proprietary protocols and techniques and the need (market-based or otherwise) to continue using them. Its objective is to define and publish UPnP device and service descriptions, such that the members of the UPnP Forum can easily connect to service devices and simplify the implementation of networks.

UPnP uses Simple Service Discovery Protocol (SSDP) [9] for service discovery. This protocol is used for announcing a device’s presence to others as well as discovering other devices or services. SSDP uses HTTP over multicast and unicast UDP that are referred to as HTTPMU and HTTPU, respectively. Unlike Jini, SSDP can be operated with or without a lookup or directory service in the network. A joining device sends out a advertisement multicast message to advertise its services to control points, which are the potential clients of services embedded into the device. The advertise

message contains service type, service name and location, an URLs that identifies the advertising service and point to an XML file that provides a description of advertising services. The device will send "bye-bye" message to network when it wants to leave the network and cancel its availability of services without leaving any unwanted state behind. In contrast to Jini, there is no central service registry in UPnP. A control point sends HTTPU request to network and any matching devices that hear this multicast should respond to it with unicast messages, which contain URLs to an XML description document describing the services.

2.4 The Salutation protocol

Salutation is a service discovery and session management protocol developed by the Salutation Consortium and drawn more from research on intelligent agents than other frameworks mentioned here. Salutation is an open standard independent of operating systems, communication protocols, and hardware platforms. Salutation was created to solve the problems of service discovery and utilization among a broad set of appliances and equipment in an environment of widespread connectivity and mobility. The architecture also enables application, services and devices to search other applications, services or devices for a particular capability, and to request and establish interoperable sessions with them to utilize their capabilities.

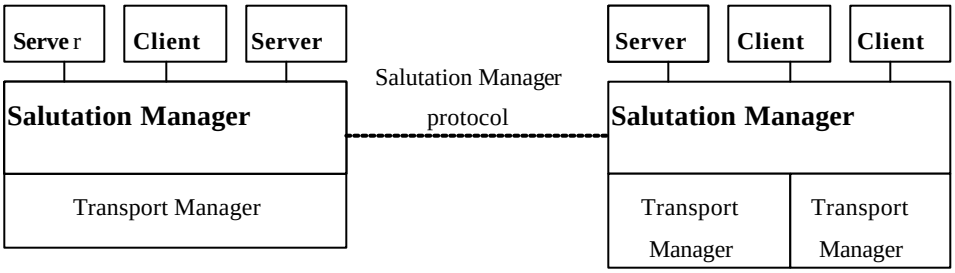

Figure 1. Salutation Architecture

The Salutation protocol aims to integrate different devices into a network by supplying them with a standard communications and API specification for discovering the capabilities of other entities in a network. As shown in Figure 1, this architecture is based on a model called the Salutation Manager (SLM), which is similar to the lookup service in Jini and functions as a service broker for services in the network. The Salutation architecture enables transaction between function units representing essential features of a service (e.g. fax, print, scan etc). Each functional unit is composed of descriptive attribute record. Another important entity called transport manager that isolate SLM from particular transport protocol and provide SLM reliable communication channels independent transport layer. SLM may sit one more than one Transport Manager attaching to different network, and provide a transport-independent interface to Server and Client applications.

3. Our Scheme

3.1 Environment Analysis

An MANET is a temporary network, operating without the aid of any established infrastructure or centralized administration. Such a network can be envisioned as a collection of routers, equipped with wireless transceivers, which are free to move about arbitrarily. The basic assumption in an MANET is that, two nodes willing to communicate may be outside the wireless transmission range of each other but may be able to communicate in multiple hops, if other nodes in the network are willing to forward packets for them. The mobile nodes would be arbitrarily located and would be moving in a dynamic manner. Typical applications of MANET are outdoor special events such as conferences, communications in regions with no wired infrastructure support, in emergencies and natural disasters and in military operations.

Due to the special formulization of MANET, it has several unique characteristics.

- The broadcast nature of wireless communication environment,
- The infrastructureless feature of MANET,
- Highly dynamic topology
- Resource/energy limited nodes

Since MANET comprises of resource limited nodes, these node demand utilization of services and cooperation of other nodes to accomplish a relative complex task. Applications of top of MANETS often times need to utilize resources or services that are present on other mobile nodes in its neighborhood. This leads to greater utilization of resources that are available in our service-rich vicinity. Hence, it is important for applications to be able to seamlessly discover other remote services/resources present on nearby mobile devices and to carry out transactions with other services. And security is paramount to the success of the transaction, but it is complicated by the fact that there is no established key or certificated infrastructure.

The existed service discovery approaches encounter overhead problem when they are applied to MANET because they were not fit for MANET environment 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 traffics in MANET. Also the deployment of Directory Service in some service discovery architecture is not realistic since the infrastructureless feature of MANET.

We exploit new Dynamic Service Discovery Protocol (DSDP) to overcome the problems in MANET environment. The purpose of the SDP is to enable nodes on an infrastructure-less wireless network find services offered by other nodes in the network.

In general, service discovery mechanism can be divided into three components: service registry and lookup, service advertisement and service query. When Designing service discovery protocols for MANET, it gives rise to new challenges because of the dynamic and short energy characteristic of mobile nodes and lacking of per-existing infrastructure in MANET. The new situation makes the service discovery protocols based on central registration out-of-date. The DSDP we proposed here addressed these challenges and target at finding, locating and evaluating services **in vicinity** required by client and fit for high dynamic environment without directory agent or central registry.

This scheme will be suitable for small or medium size MANET. The scalability to larger configuration needs to be proved by further simulation.

3.2 Attribute description

A simple but powerful service description facility and matching mechanism can help service discovery protocol achieve high efficiency. Existing method: XML, or DARPA Agent Markup Language (DAML)[13].

3.3 Service Discovery Protocol

Considering the energy and functionality constraints of mobile nodes in the ad hoc environment, this protocol is based on pull model of service discovery. Each node will maintain a small size cache to keep the present valid service descriptions and behave as a delegate of the service to response service request.

- **Service Request**

  Each individual node that requests one or more services initiates Service request. A service request packet consists of the following fields:

  < Packet-Type, Service-Description, Source Address, Route, Request ID, Lifetime, Max-Hops>,

  Max-Hops field is set to limit the number of hops the request can travel. It helps us in regulating the extent to which a request package can reach and this can be used flexibly in different mobility situations. Route field is empty in the service request package.

  Request-ID is a nonce increased monotonically. The pair of Source Address and Request-ID can be used to identify duplicate request. Service-Description field contains the type of service client want to discover in this process.

- **Intermediate nodes**

  Each node will maintain two small caches, a service-list and a cache-service-list. Service list contains the descriptions of services supplied by itself, while cache-service-list keeps track of services provide by other nodes. The entry in cache-service-list contains the following fields.

  <Address, Service-Description, Hops, Route, Lifetime>

  Whenever a node receives a service request, it performs the following steps and we demonstrate the process in pseudo-code as following.

  ```
  If( Max-Hops == 0)
  Discard the packet;

  If ( Duplicated Request)
  Discard the packet;
  Else
  Extract the Service-Description field;

  If(has match in the its own service-list)
  ```

Generate Service-Reply packet (1);
Else
If(has match in cache-service-list)
If(hops > Max-Hops)
Discard the packet;
Else
Generate Service-Reply packet (2);
Else
Forward the request packet(3)

1) **< *Packet-Type, Service-Description, Source Address, Route, Request ID, Lifetime, Hops-Count*>,**
   Since the node is the service provider who matches the requirement of Service Request. The node will append its address to the field Route. Now the Route field contains a full route from Client to Service provider. And this packet will send back along the reverse path of Route field.
2) **< *Packet-Type, Service-Description, Source Address, Route, Request ID, Lifetime, Hops-Count*>,**
   The node will add the path to real service to the Route field. The Packet will send back over the original Route field.
3) **< *Packet-Type, Service-Description, Source Address, Route, Request ID, Lifetime, Max-Hops*>,**

After attach its own address to the end of Route field, and decrease Max-Hops by 1, the node will continue broadcast the service request packet to the neighbors.

- **Caching**

  To facilitate the process of service discovery, each node caches the Service Reply packet into its cache-service-list by extracting the address of the service, Service-Description, route to the service. Since there is TOL for each service in given Service-Description, node will update cache-service-list periodically and eliminate the expired entries.

- **Piggyback**

  The service discovery is very similar the process of service discovery process, we can consider piggyback the packet of service discovery packet over route protocol to improve the performance.

- **Security Issue:**

After the process of service discovery, the node may have several entries in its cache that match its service requirement. Which service should it choose if it want the most reliable server, or the server providing best quality of service (QoS)?

The basic idea of assess the QoS of a service is to make the transaction process verifiable and exploit consulting mechanism. And it requires a service provider maintain a dynamic transaction history list,

which stores the transaction results for the active nodes it served before. The item in the list should be composed of {(Client Node Identity), (HMAC (Receipt, QoS)}, which server generates receipt, HMAC is hash code using the private key of client and QoS field is the description of quality of the service.

The following process is the process of assessing a service and using the service.

Client -> Server: Request for transaction history list
Server -> Client: Transaction history list (if any) :
{Consultant 1, HMAC (receipt A, QoS 1)}
{Consultant 2, HMAC (receipt A, QoS 2)}…
{Consultant n, HMAC (receipt A, QoS n)}

Client -> Consultant 1…m: Request the receipt and QoS description of the Server
Consultant 1…m -> Client: {Receipt I, QoS I}
Client: Verify the {Receipt I, QoS I} using the HMAC {Receipt I, QoS I}
Assess the quality of service using the information
Make decision depending on the access result ,

if (client decide to use the service)
Client -> Server: Serve me ….
Server -> Client: Result of Service + receipt
Client -> Server: {HMAC (Receipt, QoS)} and store (Receipt, QoS) in its own memory
Server: Update its transaction history list

By gathering the assessment of other active node to the service, the node can make reasonable decision by give more value to the assessment of a node which it trust. Also to reuse this process, nodes can use trust node's assessing result.

4. Conclusion

Future software services and automated devices will exist in large numbers and will operate in a networked world where they can never be quite sure about the connectivity available, about the other services and devices nearby, or about the state of the network neighborhood a few minutes in the future. In such uncertain environments, individual components will need to discover and maintain awareness of their surroundings and to configure and adapt themselves in response to changing situations. Ad hoc networks were originally meant to be an easy and an efficient mechanism for setting networks infrastructure with no support. The last part of this section presents a novel proposal to the auto configuration problem that would be compatible with the different alternatives. Also by gather information from intermediate nodes, we define a relative secure way to locate the service in proximity.

5. Future Work

In this paper, we gave out the scheme to solve the service discovery problem in Adhoc environment and addressed the security issue concerning the new scheme. Thus leave the simulation work to evaluate the performance of the service discovery to future work. The proof and demonstration of security design also need further consideration. In all, our goal is to design a new Service Discovery Protocol, which are an efficient protocol to find, locate and evaluation the service in vicinity required by client and fit for high dynamic environment without directory agent

Reference

[1] C. E. Perkins and E.M. Royer, "Ad-hoc On Demand Distance Vector Routing", Second IEEE Workshop onMobile Computing Systems and Applications, pp. 90-100, February 1999.

[2] D.b. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks", in Mobile Computing, edited by T. Imielinski and H. Korth, chapter 5, pp. 153-181, Kluwer Academic Publishers, 1996.

[3] V. Park and M.S. Corson, IETF MANET Internet Draft "draft-ietf-MANET-tora-spe03.txt", Novemmer 2000.

[4] M.R. Pearlman and Z.J. Haas, "Determining the Optimal Configuration for the Zone Routing Protocol", IEEE Journal on Selected Areas in Communications, Special Issue Wireless Ad Hoc Networks, pp. 1395-1414, Aug 1999.

[5] Sun Microsystems,. Jini Community Resources: Jini Technology Architectural Overview. January 1999. http://www.sun.com/jini /whitepapers/architecture.html

[6] Microsoft Corporation,. Universal Plug and Play: Background,. http://www.upnp.org/resources/UpnPbkgnd.htm

[7] Salutation Consortium,. Salutation Architecture Specification Version 2.0c. Part 1,. The Salutation Consortium, June 1, 1999. http://www.salutation.org

[8] E. Guttman, C. Perkins, J. Veizades, M. Day," Service Location Protocol, Version 2", RFC 2608, IETF, Jun 1999

[9] Yaron Y. Goland, Ting Cai, Paul Leach, Ye Gu, and Shivaun Albright, .Simple Service Discovery Protocol,. IETF Draft draft-cai-ssdp-v1-03.txt, October 28, 1999. http://www.ietf.org/internet-drafts/draft-cai-ssdp-v1-03.txt

[10] Bob Pascoe, .Salutation-Lite: Find-and-Bind Technologies for Mobile Devices,. Salutation Consortium, June 6, 1999. http://www.salutation.org/whitepaper/Sal-Lite.PDF Choonhwa Lee and Sumi Helal 12

[11] Brent Miller, .Mapping Salutation Architecture APIs to Bluetooth Service Discovery Layer,.Bluetooth Consortium 1.C.118/1.0, July 1, 1999.

[12] Bob Pascoe, .Salutation Architectures and the newly defined service discovery protocols from Microsoft and Sun,. Salutation Consortium, June 6, 1999. http://www.salutation.org/whitepaper/Jini-UPnP.PDF

[13] DARPA Agent Markup Language and Ontology Inference Layer. http: //www.daml.org/2001/03/daml+oil.daml.