A Service Discovery Model for Wireless and Mobile Terminals in IPv6

Bilhanan Silverajan, Jaakko Kalliosalo, and Jarmo Harju

Dept. of Information Technology,
Tampere University of Technology,
P.O. Box 553, FIN-33101 Tampere, Finland
{bilhanan,kalliosa,harju}@cs.tut.fi

Abstract. As the mobility and the number of affordable, powerful, and highly portable devices becoming networked increases, so will the amount of networked services offered, managed and discovered. In this paper, we discuss the need and use of service discovery mechanisms in future fixed and mobile networks. In particular, we choose to focus on future enterprise networks that we anticipate would need to support both fixed and wireless terminals with IPv6 and Mobile IPv6 access. We describe service discovery mechanisms using the Service Location Protocol (SLP) in IPv6, and propose extensions to the protocol for utilisation by visiting mobile devices using Mobile IPv6, that will allow service discovery to be performed without breaking compatibility to standard SLP. The extensions introduce 2 new Agents for mobility detection and overcoming limitations of multicast usage by Mobile Nodes in foreign networks. Finally we describe our SLP implementation for IPv6.

1 Introduction

The much anticipated demand in supporting mobility in modern networks for pervasive and nomadic computing is rapidly being realized, spurred by the availability today of affordable, powerful, and highly portable computers. Next-generation networks are widely expected to support user, terminal and session mobility together with mobile hosts possessing location and context awareness. Closely tied to these expectations are issues regarding service location. As the mobility and the number of these devices becoming networked increases, so will the amount of networked services offered, managed and discovered. Often, manual configuration and service discovery administration for these services may not be possible without increasing the complexity of administration, especially in mobile and ad-hoc environments.

In general, the mechanism of locating a specific resource, object or service can be performed in two ways [1]:

The first is a passive, “Lookup” method which is initiated by a seeker. It requires the existence of some directory or other agent to answer the request. Successful lookup may be achieved by exact name or address, or by some matching criteria. Lookup may be done in a statically configured environment.

M. Conti et al. (Eds.): PWC 2003, LNCS 2775, pp. 385–396, 2003.

The second is an active, more dynamic "Discovery" method, performed without the assistance of an intermediate entity. Entities use discovery mechanisms to spontaneously locate each other without previous configuration or knowledge of other entities. In many cases in the absence of configuration, discovery is often the first step to locate lookup services for subsequent interaction.

Today, "service location" implies substantially more than simply locating network printers or web servers. There is a widespread need for service location methods ranging from basic bootstrapping operations at low-level system operations such as network-based boot-up for diskless computers, to high-level demands for locating objects and architecture-specific services needed by application-level programs in distributed systems. As an example, [2] demonstrates how the CORBA Naming Service [3], a traditionally statically configured lookup service for CORBA [4] applications was enhanced with dynamic service discovery by using the Service Location Protocol (SLP) [5]. The Naming Service itself was implemented with an LDAP [6] directory service that also supported LDAP-based lookups for accessing entries.

When considering mobile terminals in conjunction with wireless and mobile networks, a naturally inherent need arises for service location, because terminal and user mobility would be high. Location awareness for computing needs, as well as context preservation while roaming, would become important factors. All kinds of local network and application services would need to be dynamically discovered, used, or even offered, and the likelihood is small that these mobile terminals possess prior knowledge of foreign visited networks.

In this paper, we choose to focus on future enterprise networks that we anticipate would need to support both fixed and wireless terminals with IPv6 and Mobile IPv6 access. We describe service discovery mechanisms using the Service Location Protocol (SLP) in IPv6, and propose extensions to the protocol for utilisation by visiting mobile devices using Mobile IPv6, that will allow service discovery to be performed without breaking compatibility to standard SLP.

Section 2 quickly introduces SLP in its present form for IPv4 networks. Section 3 then discusses the use of SLP in IPv6 and how we propose to extend its functionality for use over Mobile IPv6. Section 4 describes our IPv6 SLP implementation and Section 5 presents our conclusions.

2 Overview of SLP

Although SLP has been standardized several years ago, its usefulness is only just becoming apparent. In IPv4, SLP has formed the basis of several modern discovery mechanisms today in AppleTalk, Novell Netware and Sun Microsystem's Jini technology. An open-source implementation of SLP, called OpenSLP [7] is available in addition to Sun Microsystem's own version, shipped with Solaris 8 and 9. Test versions for research purposes also exist [8].

SLP standardization efforts in the Internet Engineering Task Force (IETF) have produced several Requests for Comments (RFCs). The initial protocol version and its components for IPv4 were described in 1997 [9]. Then, a second version of the protocol was introduced in 1999 [5]. Among other things, SLPv2 rectified race conditions that existed in SLPv1, introduced clearer usage for service scopes, and added LDAP compatibility for string encoding of attributes and search filters. Though SLP v2 introduced slight incompatibilities with SLPv1, it provided a migration path away from the original by allowing a limited level of backward compatibility; the general components and overall capabilities, however, have been largely preserved. Today, SLPv2 remains the more prevalent of the two versions in use for service discovery in IPv4 networks. It is this version that all subsequent references to SLP in this paper are based on.

Briefly, using SLP involves introducing 3 types of agents in a network: The Service Agent (SA), User Agent (UA), and Directory Agent (DA). The UA is used by client applications on their behalf to discover the locations of services, by querying the network with SLP request messages containing a "service:" string, which may be formulated as a URI, and returning responses received from the network to its applications.

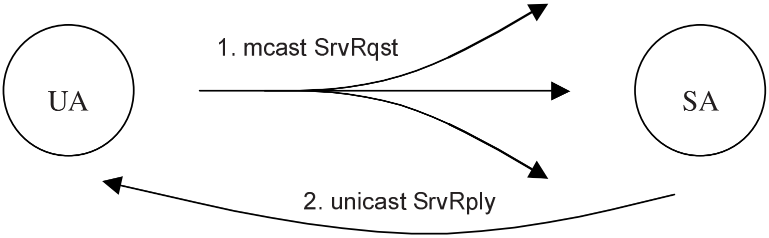

Fig. 1. Service discovery with UA and SA

Generally, SAs tends to reside on the same hosts as server applications, but they could reside elsewhere in the network too. Server applications register and unregister their service with SAs with a Service Type string URL. For example, a print server might register itself with an SA with the URL

"service:printer:http://printerlocation.mycompany.com".

The SA then listens for UAs multicasting or unicasting requests for a printing service, such as "service:printer" or "service:printer:http", and provides unicast replies containing the location of the printer. SAs also unicast SA Advertisement messages on the network if solicited by a UA. Figure 1 depicts a User Agent multicasting a service request, and receiving a unicast response.

The DA is an optional element in SLP. If the DA is present in the network, it periodically announces its presence with DA Advertisement multicast messages. The main role of the DA is to provide a centralized service for all service announcements in the very large network, so that a point of single contact exists for a UA trying to discover various services. Thus, SAs are required to register their services with DAs they discover from DA advertisements. UAs also interact directly with the DA instead of SAs, if one is present in the network. Figure 2 demonstrates how DA Advertisements, SA Service Registrations and UA Service Requests proceed in a network.

In SLP, UAs, SAs, and DAs can be flexibly configured statically (manually) to find each other prior to start-up, or dynamically to use multicast and broadcast discovery mechanisms for Request messages. All multicast messages use the administratively scoped multicast address 239.255.255.253. Responses however, travel directly point-to-point.

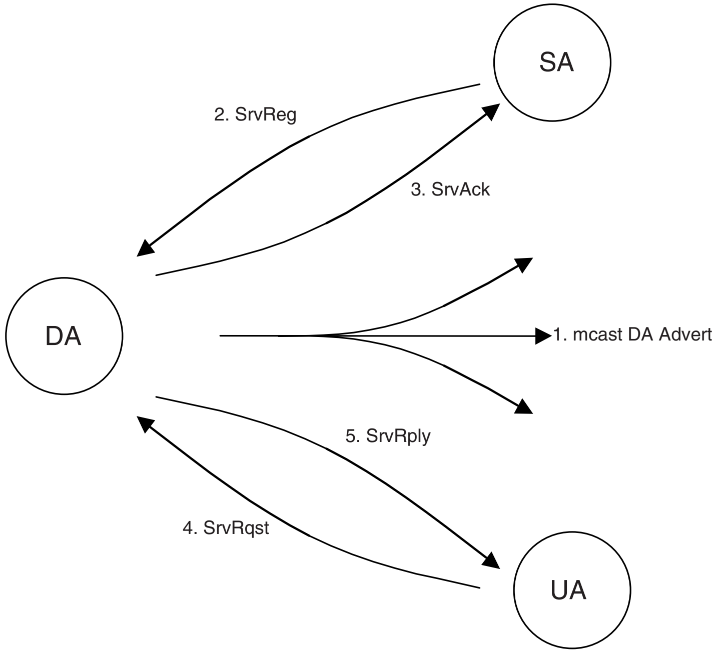

Fig. 2. Service discovery with UA, DA and SA

Apart from simple service request and reply messages, the protocol also defines messages that allow defining various other attributes, service types, and security enhancements through the use of digital signatures for message content verification. Optional extensions such as attribute lists are also specified [10].

3 Using SLP in IPv6 and Mobile IPv6

In 2001, the use of SLP in IPv6 networks was standardized [11], although at the time of this writing, despite its popularity in IPv4 networks, commercial and open-source SLP implementations for IPv6 remain few or non-existent.

Using IPv6 as a network protocol for SLP introduced some major differences from IPv4 in the way the protocol works. Service requests using broadcast methods are replaced with link-local multicast. Also, instead of a fixed multicast address for SLP messages, IPv6 multicast addresses are dynamically calculated using a hash algorithm for the various “service:” types. The addresses range from FF0X:0:0:0:0:0:1:1000 to FF0X:0:0:0:0:0:1:13FF, and DA advertisements use FF0X:0:0:0:0:0:0:123. The value of X can be 1, 2 or 5 and thus defines the scope of the service to be either node-local, link-local or site-local. Lastly, the notion of an SLP zone is introduced. A zone is a connected region of topology of a given scope. For example, the set of links connected by routers within a particular site, and the interfaces attached to those links, comprise a single zone of site-local scope. Agents are not allowed to propagate advertisements or respond to requests across different zones.

In contrast to fixed IPv6, Mobile IPv6 allows an effective way for computers to remain connected to the Internet with one IP address, while roaming across different subnets or organizational networks [12]. Each mobile node (MN) is always identified by its home address, regardless of its current point of attachment. While situated away from its home, a mobile node is also associated with a Care-Of Address (COA).

3.1 MN Usage of SLP in Home Network

A mobile node that is connected to its home link functions in exactly the same way as a fixed IPv6 node. No modification is thus needed in its interaction with other nodes in its home network. Thus, it is able to use all the features of SLP, different classes of scoped multicast, and use and offer services with the UA, SA and DA functionalities as specified in [11].

3.2 MN Usage of SLP in a Foreign Network

When MNs roam across networks using Mobile IPv6 wirelessly or otherwise, we cannot assume that they are always able to rely on static configuration for rapidly locating services in their immediate vicinity. Dynamic discovery is needed, especially in foreign networks the MN has no previous knowledge of. Using SLP would serve this purpose well for locating local site-based services, whenever the MN moves into a new network.

Mobility detection is thus useful in this context for SLP to perform discovery automatically and immediately when the MN moves across into a new network. However, Mobile IPv6 currently shields this from the upper layers, aiming to keep applications unaware of the terminal mobility, which presents a technical difficulty to SLP.

The current Mobile IPv6 specification prescribes two completely different, but equally acceptable choices for all Mobile IPv6 implementations to support routing multicast packets to and from MNs in foreign networks [12]: In the first method, the MN becomes a direct member to the local multicast router in the foreign network. In the second method, the MN remains a member of its home network and all multicast packets are bi-directionally tunnelled to and from its home agent (HA).

From the perspective of direct site-local multicast by the MN, these two methods are clearly incompatible with each other. The second method would also breach SLP zone enforcement rules for discovering services in a foreign network. There is no way for an MN to clearly determine beforehand which multicast behaviour is in force, since it is implementation dependent. Also, currently none of the present Mobile IPv6 implementations support either of the above 2 methods. Thus proper and consistent multicast usage by the MN when it is not in its home network might very possibly become limited to only the link-local scope, where packets are guaranteed to reach all nodes in the same link, be it in the home network or foreign network.

To overcome the above-mentioned difficulties, we propose extending SLP with 2 new agents. When using Mobile IPv6 and roaming, an SLP Visiting Agent (VA) resides in the MN performing link-local discovery of an SLP Access Agent (AA), sitting on the local link in the foreign network that helps the MN and VA find site-local services.

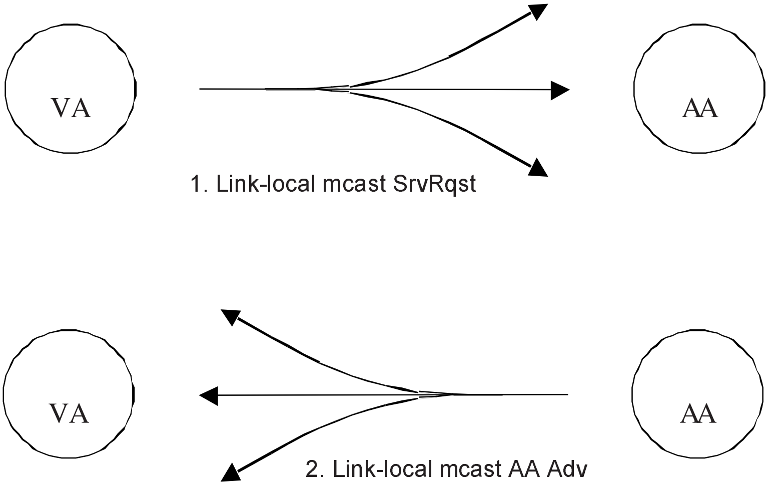

Fig. 3. Active AA Discovery

The VA can either actively solicit, or passively listen for, periodic AA advertisements in the foreign network. This is similar to how UAs and DAs interact in a fixed network, with the exception that multicast always uses link-local scope, and replies also use multicast instead of unicast. This allows other VAs present in other MNs to detect the AA quicker without resorting to active discovery or creating a potential multicast avalanche in the link. Fig 3 shows how active AA discovery is performed when a VA enters into a new network.

AA advertisements contain a "service:access-agent:<address>" URL using the multicast address FF02:0:0:0:0:0:1:1259. This address is calculated using the hash algorithm described in section 4.1 of RFC 3111 [11]. In addition to the source address, AA Advertisements can also piggy-back the address of a local site-wide DA, if one exists in the foreign network.

MN mobility detection is performed by the VA by observing changes in the AA Advertisements. When the source address in the advertisements change, the VA can determine whether the MN has moved or not by checking its Care-Of Address (COA). If the COA has not changed, it implies the MN has not moved, but a new AA has instead become active. In this case, all interaction now proceeds with the new AA. On the other hand, if AA Advertisements become completely absent, the VA confines itself to link-local services only after determining that unicast reachability to the AA and DA (if one is present) has failed.

Because the AA sits on a fixed node, it can perform proper multicast routing for site-local discovery on behalf of the VA. It can therefore relay whatever site-local Service Requests the VA might have, and forward responses it receives from the network back to the VA. The AA also serves to announce all services that the VA may wish to advertise to the foreign network. Viewed from this perspective, the VA and AA can together be seen as a decoupled hybrid User/Service Agent-pair working on behalf of applications in the MN to both utilize and provide services locally in the foreign network.

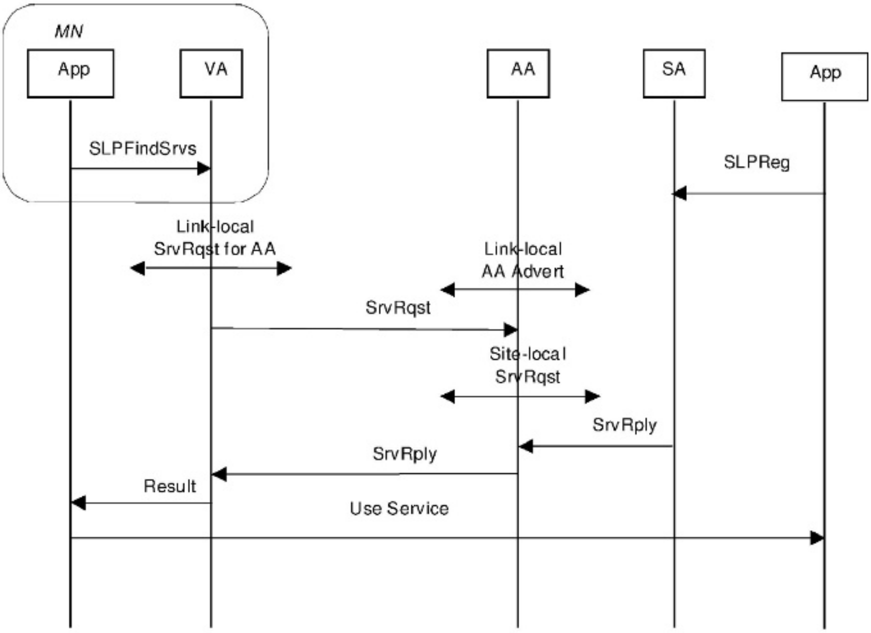

Fig. 4. MN using services in foreign network

Fig 4 illustrates the sequence of events taking place when an MN enters a new network and wants to use site-local services. Here, the VA performs an active AA discovery. Upon receiving an AA Advertisement, the VA can check if there is a DA present in the foreign network. If a DA is present, the VA may optionally choose to communicate directly with it to find site-local services in the event of a communication failure with the AA. In this example, the foreign network has no DA present, so all site-local service discovery proceeds via the AA. The AA behaves as a normal UA, and performs site-local multicast discovery on behalf of the VA to discover services directly from local SAs. The received replies are then forwarded back to the VA.

Fig 5 describes what happens when the MN enters into a new network and wants to also provide site-local services. In this foreign network, an organization DA is present which announces its presence with periodic site-local DA Advertisement multicast messages. This is noted by the AA, which piggybacks the advertisements onto its own periodic AA Advertisements. The VA first performs mobility detection tests to discover whether the MN has moved. Once it ascertains that the new AA Advertisements do indeed originate in a new network, it then proceeds to unregister all previously registered services from the old AA in the previous network (depicted in dotted lines) using unicast messages.

The VA then begins to register services with the current AA. The AA now emulates a normal SA mechanism. If no DA was present, it would perform SA Advertisements and respond to UA multicast service requests. Here, it registers the VA's services with a site-local DA with the service type URL containing the original location as specified by the VA. Service Acknowledgement packets are returned by the AA to the VA once service registration in the foreign network is successful.

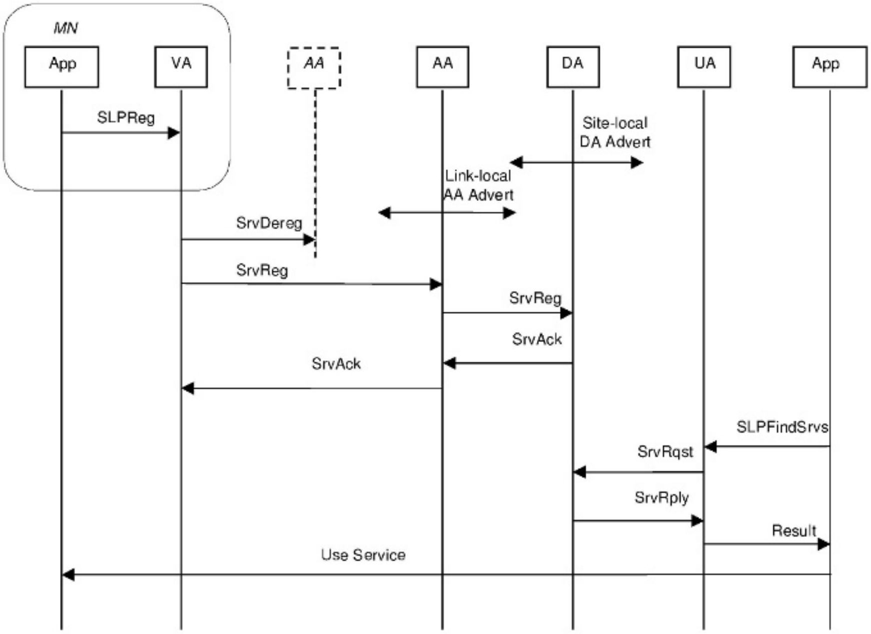

Fig. 5. MN providing services in foreign network

In general, the VA always listens on the link-local scope for service requests, and the AA on the site-local scope. The VA interacts directly with a DA only if an AA is unreachable with both unicast and multicast methods. If there is no local AA, the VA can do a link-local multicast DA discovery. If successful, it registers its services with the DA and can listen for service requests on the link-local scope. It also can use site-local services that it finds from the DA. If no AA or DA is present on the link, the VA only listens for service requests and uses services found with link-local multicast. This is summarized in Table 1.

Table 1. Agent Presence and Service Scope

		AA present	DA present	Absent
VA Service Scope	Service Usage	Site-Local	Site-Local	Link-Local
	Service Provision	Site-Local	Site-Local	Link-Local

4 SLP Implementation for IPv6

This section illustrates our SLP UA, SA and DA implementations for IPv6. These agents were implemented with the use of DOORS [13], a publicly available event-based C++ framework. DOORS facilitates the making of portable and advanced communication software ranging from socket and protocol implementations to object oriented CORBA applications. It abstracts applications into a set of stateful event-driven tasks communicating via asynchronous messages transmitted and received via their ports.

Figure 6 illustrates a simplified UML diagram depicting the classes making up the overall architecture developed for the UA, SA and DA. The SLPPeer class is a common class that is used by all 3 agents, implementing the basic PDU encoding and decoding methods for all the SLP messages.

DAMain and DAMainStateMachine implement the DA functionality. All service registrations are currently stored in memory within a C++ STL vector container, but with a little effort, its functionality can be extended to support the integration of an LDAP-based back end to store service registrations using the schema template defined in RFC 2609 [14].

Applications needing SLP UAs use the UAMain objects. For each service location request received from the application, the UAMain object creates a new UAConn object. The newly created UAConn subsequently sends the request either to the DA residing in the network directly, or to the proper multicast group. The UAConn object is also responsible for resending the request if necessary.

The architecture of the SA is similar to that of the UA. Applications send service registration messages to the SAMain object, which creates new SAReg objects. Each SAReg object is responsible for 1 service registration, and it joins to the correct multicast group to respond to any SLP requests it receives.

Although the UAMain and SAMain tasks are responsible for routing an incoming message to the proper child task for handling, the UAConn and SAReg objects directly use their parents' corresponding socket connections and ports for sending messages to the upper layers, applications or other SLP Agents. As the child tasks are able to communicate with their applications and the network without encumbering their parent tasks, this simplifies implementation and significantly reduces messaging overhead.

The UDP6Task object is supplied by the DOORS framework, and provides a uniform and simple message-based interface to send and receive IPv6-based UDP datagrams to and from the network. It is capable of understanding unicast and multicast, and is able to join and listen to multiple multicast groups. The PTask class, also part of the DOORS framework, is a base class containing commonly functionality for protocol development, such as State Machine handling functions and specialised methods which understand communication through User and Provider Service Access Point and Peer Ports. All the five main task classes implementing the three agents of the protocol, are derived from the DOORS PTask class.

5 Conclusions

SLP is a simple and elegant protocol that can be flexibly configured to custom-fit the needs of service location in various types of enterprise networks. Active work will continue with our research and implementation efforts towards extending SLP to support dynamic discovery for mobile terminals within core IPv6 networks.

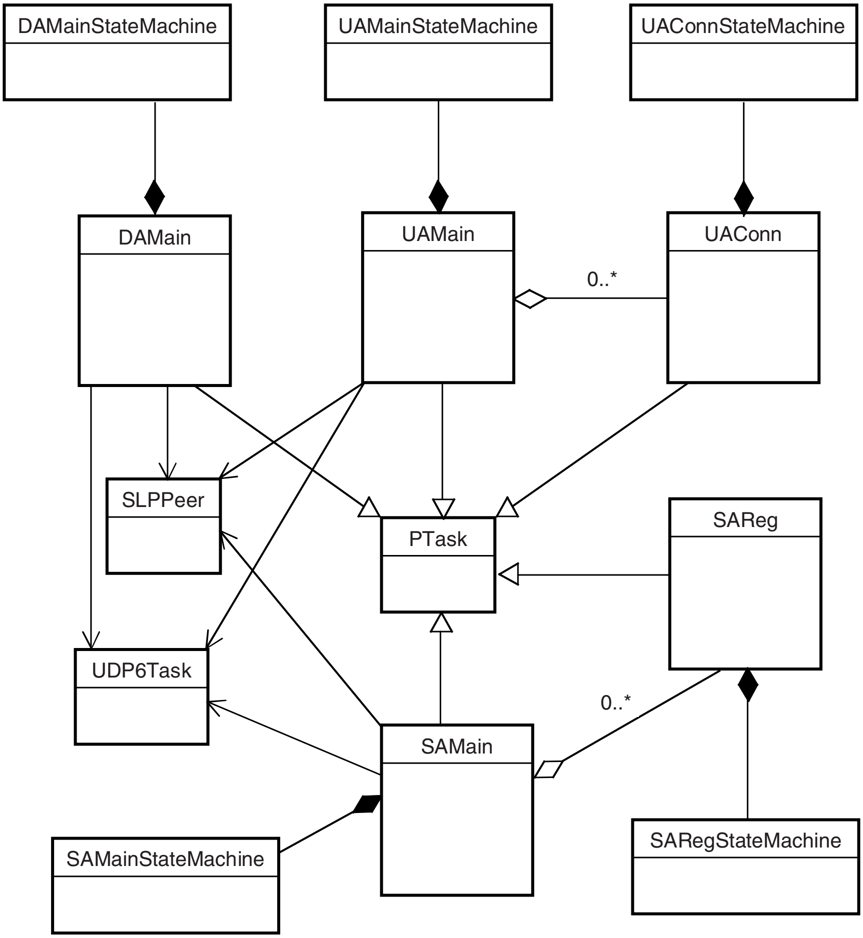

Fig. 6. DOORS SLP design for IPv6

Effort is also being spent on understanding the influence of mobile computing paradigms in IPv6 networks on the evolution of distributed object oriented systems and architectures, particularly with regards to the role of lightweight and flexible service location and discovery methods for both clients and servers. This is especially interesting because, owing to the smaller granularity and size of object-oriented services and servers, there is a high likelihood that they would reside in portable devices, making these server applications just as mobile as client applications.

In IPv4, SLP is a mature and popular protocol that has been put to good use, and interaction with many types of applications and services is well understood. IPv6 itself boasts an undeniably rich and flexible feature set for fixed and mobile networks. However, experience has yet to be gained from using SLP in IPv6. The protocol modifications that RFC 3111 [11] introduces for IPv6 changes the fundamental methods of how dynamic discovery and scoping are done, and consequently render it incompatible with its IPv4 counterpart.

In effect, SLP in IPv6 can be regarded as a new protocol whose behaviour and supporting network requirements need to be thoroughly investigated. With the mobility extensions we propose to SLP, MNs will in future easily be able to tap into an SLP deployment in IPv6 networks, by neither breaking any compatibility with fixed IPv6 services nor introducing the need for a completely new service discovery architecture. RFC specification guidelines of not allowing SLP Advertisement and Request/Reply packets to cross SLP zones, are preserved too.

Static configuration of the VA within the MN for service discovery is unnecessary, and this does not also enforce any requirement as to how the agents such as the AA discover other agents in the fixed network, or should be configured.

Measurements have yet to be made as to the optimal duration between AA Advertisements. The need to rely on link-local multicast is required only within the context of the initial AA discovery, since all subsequent communication between the AA and VA use unicast messages. However, in order not to populate every single link in the foreign network with an AA, sane network engineering practices must be applied to deploy AAs only in subnets which are expected to serve MNs and provide services. Also, the widespread use of IEEE 802.1Q Virtual LAN (VLAN) technology in many networks allow much flexibility in segmenting networks logically using VLAN identification numbers (VLAN IDs) without having to be physically on the same link to receive link-local messages; many switches already in common use today support multiple simultaneous VLAN IDs. For example, this could be used to let one AA belong to several VLANS, allowing link-local multicast to reach all of them.

Finally, SLP uses digital signatures for content verification of messages. There is nothing in the proposed model which weakens or strengthens this technique. Agents may verify digital signatures provided in advertisements, but the responsibility of authenticating Mobile Nodes in foreign networks lies with Mobile IPv6.

References

1. Robert E. McGrath: Discovery and Its Discontents: Discovery Protocols for Ubiquitous Computing, Presented at Center for Excellence in Space Data and Information Science, NASA Goddard Space Flight Center, April 5, 2000
2. Bilhanan Silverajan, Joona Hartman, Jani Laaksonen: Investigating Service Discovery, Management and Network Support for Next Generation Object Oriented Services. Proceedings of Smartnet 2002, Saariselkä Finland April 8–10, 2002
3. OMG: Naming Service Specification, February 2001
4. OMG: CORBA v3.0: The Common Object Request Broker: Architecture and Specification, July 2002
5. IETF RFC 2608, “Service Location Protocol, Version 2”, June 1999
6. IETF RFC 2251, "Lightweight Directory Access Protocol (v3)", Dec 1997
7. The OpenSLP Project, http://www.openslp.org
8. Christian Bettstetter and Christoph Renner, “A Comparison of Service Discovery Protocols and Implementation of The Service Location Protocol”, Proceedings of 6th Open European Summer School EUNICE 2000, Enschede, The Netherlands, September 13–15, 2000
9. IETF RFC 2165, “Service Location Protocol”, June 1997
10. IETF RFC 3059, “Attribute List Extension for the Service Location Protocol”, February 2001

11. IETF RFC 3111, "Service Location Protocol Modifications for IPv6", May 2001
12. IETF Mobile IP Working Group: Mobility Support in IPv6, draft-ietf-mobileip-ipv6-22.txt, May 26, 2003
13. Bilhanan Silverajan, Ilkka Karvinen, Joona Hartman, Jani Laaksonen: Enterprise-level Integration and Interoperability in Future Networks with DOORS Middleware. IFIP WG6.7 Workshop and EUNICE 2002 Summer School on Adaptable Networks and Teleservices, Trondheim, Norway September 2–4, 2002
14. IETF RFC 2609, "Service Templates and Service: Schemes", June 1999