Service Location Protocol for Mobile Users

Charles E. Perkins*

ABSTRACT

Service Location Protocol has been designed within the Internet Engineering Task Force to simplify or eliminate the configuration needs for users of the Internet and World-Wide Web. This is of increasing importance to mobile users, first because they experience frequently changing network service environments, and secondly because the Internet is itself becoming more service oriented. The basic Service Location Protocol is described, with detailed descriptions of the various protocol entities, messages, and ways of selecting appropriate services.

1 Introduction

Mobile computing introduces a new requirement for time-dependent resolution of many network relationships. Existing approaches for supporting mobility at the network layer [10] can be understood as a way of introducing time-dependency into the the link between a mobile node and its home network; the link is replaced by a time-varying tunnel when the mobile node is no longer resident at its home network. There is widespread agreement that data-intensive applications will have to adapt the flow of data into a network over time when wireless media provide the link between infrastructure elements and mobile computers.

Mobility means that the geographic neighborhood in which the computer user is located changes over time. The set of network-connected devices in the neighborhood is likely to be of interest to the mobile user, and depends on the geographic neighborhood (albeit in a somewhat unpredictable way). Consequently, the collection of network services that can be considered nearby the mobile node changes drastically with time. From the perspective of the mobile user, this has fundamental effects on the productivity, comfort, convenient operation, and even the usefulness of the mobile computer.

Even when the mobile user can find out the right person to contact regarding network connectivity to local printers and other network resources, it is still awkward and time consuming to do so. It would be a lot better if the questions about locating network services could be framed and issued automatically to a local directory service which could return the location information. Today, this is just now becoming possible by use of the Service Location Protocol (SLP) [11, 5]. Part of the timeliness of the development of this protocol hinges on the widespread occurrence of URLs (Universal Resource Locators). Another part is driven by the needs of mobile computer users. Proposals for *zero administration* of enterprise services has given a great deal of impetus to the design of several automatic configuration protocols such as SLP, DHCP [3], and Stateless Address Autoconfiguration for IPv6 [8] Finally, the growth and diversity of the number of services available on the Internet is making it more and more difficult to run manual configuration procedures for all of them.

In this paper, we explore the design issues relevant to SLP, and emphasize the usefulness of SLP to mobile computer users. The next section contains an overview of the protocol, including a description of the protocol entities and messages transacted between them. Section 3 discusses a couple of examples, showing how SLP is used to dynamically resolve an application's need for printing service, and how SLP is used with the Dynamic Host Configuration Protocol (DHCP) and Mobile IP. Section 4 discusses relevant design points that have guided the evolution and standardization of SLP. Lastly, section 5 gives a summary and conclusions, along with some indications about the current status of SLP, and directions for future work.

2 SLP Overview

SLP resolves requests for network services, by providing service handles to applications that ask for them. Applications ask for service by way of a *User Agent* (UA), which is a SLP entity that is capable of issuing the necessary protocol messages. Services, on the other hand, are represented by a SLP entity called a *Service Agent* (SA), which advertises service handles containing the necessary locator information. User agents can obtain the necessary service handles directly from service agents, or alternatively they can query a nearby *Directory Agent* (DA) for the information. These relationships are illustrated in figure 1, where the printer is shown represented by a service agent. DHCP is part of the

*C. Perkins is with Sun Microsystems, Menlo Park, CA. 94025 E-mail: cperkins@eng.sun.com

typical model of operation, because it is so handy for delivering information to UAs and SAs about the network address of nearby DAs.

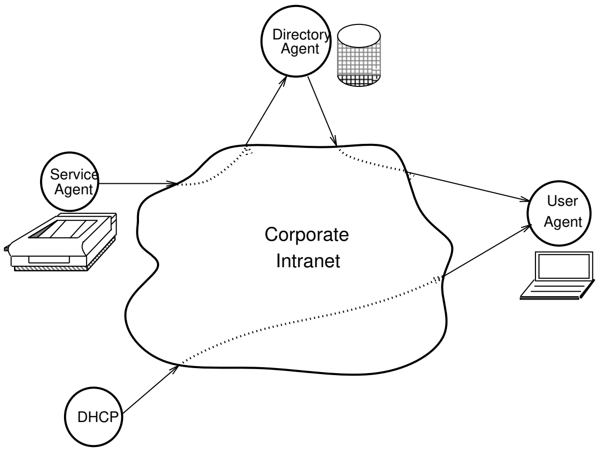

Figure 1. SLP agent model

The messages used by UAs, SAs, and DAs may be broken down into two main types:

- *non-interactive* messages, which are mostly mandatory for all implementations.
- *interactive* messages, which are optional in UAs and SAs, and useful in user interfaces that make network operations visible for user control.

Most (not all) SLP messages fit one category or the other.

2.1 Request and Reply Messages

When a UA needs to connect to a service, it issues a *Service Request*, and expects to receive a *Service Reply*. The Service Request specifies the type of service needed by the application, and contains enough information to precisely select one or more services that fulfill the application's needs, from among possibly dozens or hundreds of services of the same type.

The Service Reply contains the information needed by the application to unambiguously locate the selected services. These locators are formulated as URLs, which conform to a *scheme* [1] definition that is well known between the UA and SA. For example, many schemes are defined by IETF standards documents. SLP introduces a new class of schemes [4] appropriate for the purposes of locating services, called "`service: URLs`".

If the Service Request is unicast to a DA, then the DA will look in its list of *registered* SAs and (if any services can be selected to match the request) it will construct a Service Reply containing the locators. On the other hand, the Service Request may be multicast to the local network or to several nearby networks at a well known multicast address. Every SA and DA is required to listen at the well-known address, and when a Service Request is received by an SA at that address, the SA will determine whether it meets the service requirements in the request. If it does, the SA unicasts a Service Reply back to the requesting UA. SLP makes provisions for a UA to make sure that no Service Replies are lost even when many of them were sent in reply at nearly the same time. The number of networks carrying such multicast is determined by the the *Time To Live*(TTL) of the multicast request, which effectively limits the number of network hops away from the UA.

Service Request messages are mandatory for UAs, and Service Reply messages are mandatory for SAs and DAs.

The relationships of UAs, SAs, and DAs in an enterprise deployment over a larger number of networks is illustrated in figure 2. In the figure, a UA on network 11.0.0 may acquire service information about SAs on that network by multicasting requests. If the UA finds out about the enterprise DA (say, by requesting the information from DHCP), the UA can obtain locator information for any SA in the enterprise that has registered its URL with the DA. Typically, in such a configuration, all SAs would in fact register with the enterprise DA, whose address the SAs would also request from DHCP. For larger deployments, multiple DAs can be used, but the concepts and protocol messages are the same. SLP is engineered to enable precisely this convenient growth model. As the enterprise grows, services can still be conveniently located without encumbering the enterprise networks with excess multicast or broadcast traffic. Thus, SLP avoids some of the scalability problems associated with previous service configuration protocols.

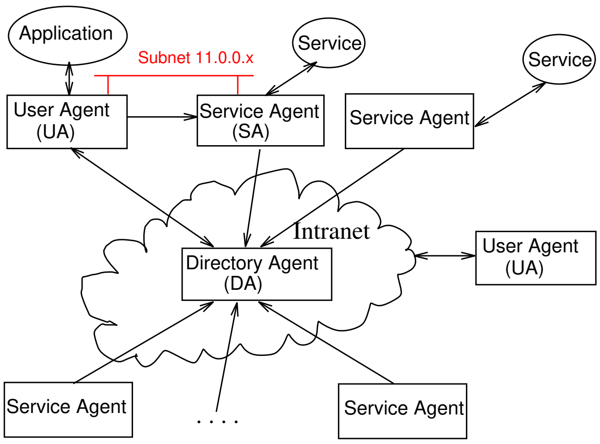

Figure 2. SLP Enterprise Deployment Model

2.2 Registration Messages

In order for a DA to be able to send Service Reply messages to UAs, it has to receive *Service Registrations* from the SAs.

The registrations contain the URLs allowing UAs to locate the services, along with *service attributes* describing the details of the service offered. The DA also allows services to update the attributes accompanying the URLs. This has the effect of updating the selection criteria used by the DA to figure out which URLs to put into Service Reply messages.

SLP also defines *Service Deregistration* message types to allow an SA to remove its registered URL from a DA. Finally, when a DA receives a registration or deregistration message, it is required to return a *Service Acknowledgement* message to the SA. Registration and acknowledgement messages are required in every DA and SA implementation. Deregistration messages are optional for SAs.

2.3 Service Attributes and Request Predicates

Services formulate the description of their services by tabulating a list of attributes and one or more values available for each attribute, and including the list along with the registration message for the URL locating the service. The UA asks for particular services of a type by including a query in its Service Request, formulated as a boolean predicate of value comparisons for various attributes. For instance, if a printer scheme had attributes "`speed`" and "`q`" for the queue length and the number of pages output per hour, a Service Request might contain the predicate:

```
(&(q<=3)(speed>=1000))
```

This would return one or more printer services with no more than 3 jobs in its print queue and able to print more than 100 pages per hour.

To enable code-reuse, SLP message syntax and predicate syntax is quite regularized. For instance, the predicate syntax just described is actually taken from the *Lightweight Directory Access Protocol* (LDAP) version 3 [6] specification. This has the added benefit of enabling LDAP to supply information to DAs in many cases, by the simple device of inserting a DA as a bridge between other SLP entities and the back-end LDAP database. As another example enabling code reuse and synergistic operation with other protocols and network services, SLP's use of `service: URLs` should make it much easier to integrate dynamic service resolution into today's Web browsers.

2.4 Interactive Messages

Due to length restrictions on this paper, the interactive SLP messages will not be described in full detail. Briefly, SLP defines two such message pairs. One pair (*Service-Type Request* and *Service Type Reply*) enables browsing of locally available service types. Another pair (*Attribute Request* and *Attribute Reply*) enables a user to browse all attributes for a particular service type (or, for a particular service instance), in order to make a selection from the attributes which are actually available for the service type.

These interactive messages are inspired by experience with AppleTalk and Novell protocols, which have been quite successful in the marketplace. In fact, Service Location Protocol was itself inspired by the observation that these other protocols were so much easier to set up and use, because the need for configuration of service connection information was drastically reduced. TCP/IP based networks have been burdened for many years with much higher configuration requirements, and consequently have gotten a largely undeserved reputation for being difficult to install.

2.5 Finding Directory Agents

Directory agents are the key to scalability of SLP. There are several methods defined and suggested by which other SLP entities (UAs and SAs) can discovery a locator for DAs, as follows:

- static configuration
- DHCP option 78 [9]
- Active DA discovery
- Passive DA discovery

Each of these method will be described briefly.

First, a UA or SA may be manually configured with a locator (URL) for one or more DAs. The URL is defined by the service type `directory-agent`, and is of the form

```
service:directory-agent://some.da.fqdn/
```

where the final string delimited by '`/`' characters represents a *Fully-Qualified Domain Name* (FQDN) [7] for some network computer hosting the DA. While it is somewhat incongruous to suggest such static configuration for the operation of an automatic configuration protocol such as SLP, it is still vastly preferable to manually configuring UAs with locators for many individual service instances.

Recently, DHCP option 78 has been defined for use by SLP user agents and service agents to identify a DA for their use. Another option (79), has also been defined for use by SLP entities, allowing them to discover a *service scope*, described later (see Section 4.2). A mobile computer running Mobile IP, for instance, would request option 78 from the DHCP server at the same time that it asked for a care-of address from the DHCP server. Subsequent SLP messages sent to the DA identified by DHCP should use the care-of address as the IP source address.

The last two DA discovery procedures rely on the reception of a *DA Advertisement* message. For active discovery, suppose a UA multicasts a Service Request with service type `directory-agent`. Then, any DA listening at the well-known SLP multicast address will respond with the DA Advertisement message; this is the only time that a Service Request can elicit something other than a Service Reply message, except for error messages such as ICMP messages issued by protocol modules other than SLP.

SLP agents can also discover DAs passively, by listening for unsolicited DA Advertisements. DAs are configurable to transmit such advertisements periodically, both to enable such passive discovery, as well as to enable SAs to make a determination about whether their registrations are still available at the DA to which they were sent. If the DA crashes, it increments the timestamp which is included in each advertisement. The timestamp is not incremented otherwise. This makes it easy for an SA to find out when a DA has rebooted since the SA sent its registration. If the DA reboots, it is presumed to have lost all of its service registrations, and the SAs will then reregister.

3 Operational Examples

Two operational examples will be given in this section. The first shows how a mobile node can use SLP, Mobile IP, and DHCP together to reduce or eliminate all such manual configuration requirements. Security is not considered in this paper, although SLP and Mobile IP both have security solutions included in their protocol specifications.

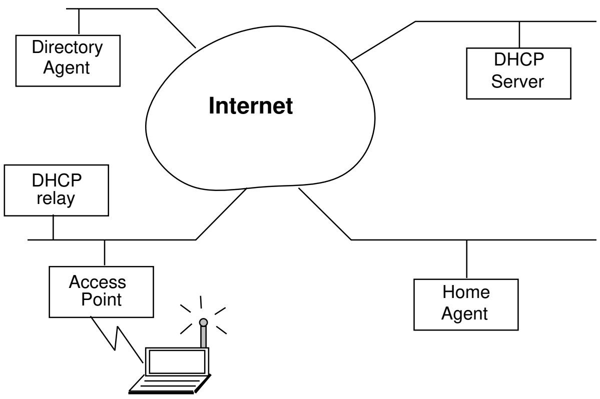

Figure 3. Wireless Mobile Node using DHCP and SLP

As illustrated in figure 3, a wireless mobile node might interact with several network entities upon attachment at a particular *access point*, including a DHCP server, DHCP relay, Mobile IP Home Agent, and a SLP Directory Agent. After establishing the wireless link, the mobile node would broadcast a DHCPDISCOVER message through the local DHCP relay to solicit a DHCPOFFER from the appropriate DHCP Server. The DHCPREQUEST message to the offering server should also include option 78, so that the DHCP server can know to identify the SLP directory agent to the mobile node. Upon receiving a topologically correct IP address from the Server, the mobile node could then use it as a *co-located* care-of address in a Mobile IP registration with its home agent [10]. This care-of address should also be used to mediate any access to local network resources and services, including the SLP DA itself.

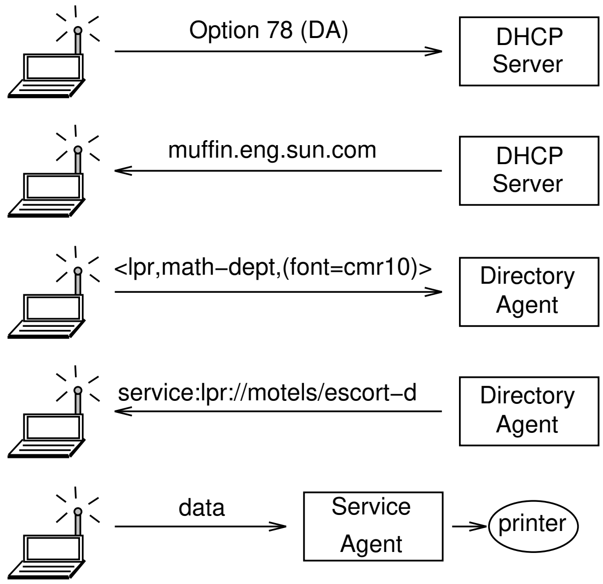

Figure 4. Finding a Printer using SLP

Figure 4 illustrates the general procedure by which a network node (mobile or not) can go about finding a network printer, using SLP and DHCP. In the figure, the first request at the top shows the wireless node requesting option 78 from DHCP, and getting back the information that `muffin.eng.sun.com` should be configured by the UA as the appropriate DA for its use. Next, the UA issues a Service Request to `muffin` with the following information:

- the service type is `lpr`
- the service scope (see section 4.2) is `math-dept`, and
- the `font` attribute must have value `cmr10`

The DA (`muffin`) then selects a `service: URL` (namely, `lpr://motels/escort-d`) matching the request, and sends back a Service Reply to the wireless node containing that URL. The URL indicates that the printer suited to the UAs needs has a print queue called `escort-d` and is controlled at the network node `motels`. Subsequently the UA can use `lpr` to submit print jobs to `escort-d` at print server `motels`.

As a third example, suppose that it is desired to change printers in a department without disrupting operations. Given current practice, one would have to send out numerous e-mail messages to all department members (and the wireless mobile nodes visiting in the area). Then, at some predetermined time, the switch would be made, and everyone would have to reconfigure their computer systems to reflect the change. None of this inconvenience would be necessary if SLP were used by the desktops and wireless nodes to locate the network printer. If one network printer is no longer available, and another one is advertising equivalent service, the newly available printer will be used automatically.

4 Design Principles

In this section, we discuss some of the principles guiding the design of SLP within the IETF.

4.1 Fully Distributed Operations

One of the main features of SLP is that it helps to decentralize the responsibility for administering network services. Using SLP, one can make a printer available for use by an entire department (say) merely by plugging it into the network, giving it an IP address, and specifying its service attributes and possibly a DA address. For printers, service attributes include such things as whether the printer understands PostScript, which fonts are loaded, whether the printer can handle color output, how fast it prints, and so on. If DHCP is available, then the service agent can find its own IP address and a locator for the nearby DA automatically as well.

The distributed nature of SLP will become increasingly important as the number and kinds of network services proliferate throughout the Internet. Already, SLP is being proposed for network management services, filesystem services, web proxy advertisement, and others besides printer services. We expect that fax service agents and calendaring services are not far behind. The current dependence upon centralized management for all network services seems unlikely to survive this explosion of service availability.

4.2 Scalability and Scopes

SLP has been designed with *scalability* as a prime concern. In section 2 some indication was given about how SLP helps a growing enterprise make smooth transitions between operation on one or two networks, into operation over a medium-size enterprise with several departments and buildings, with perhaps dozens of networks. The transition is managed by installing a DA once it is observed that there are too many Service Requests being multicast.

In order to help with scalability, and in order to collect together services that are administered together for the common benefit of a particular class of users, SLP defines the concept of *scope*. A scope is really nothing more than a set of services. Each UA can discover from DHCP (or by manual configuration) which service scopes it should use. Note that in larger installations, which are the ones more interested in using scopes, it is also more likely that DHCP will be deployed. Thus the expected interdependence between DHCP and scope deployment by network administrators is natural and unlikely to pose any procedural barrier.

Each Service Request has to contain a scope identifier; if scopes are not configured, a UA uses the scope identifier `default` in its requests. Likewise, service registrations and interactive request messages are also required to specify a (possibly default) scope string.

There is one special use for scopes. By sharing a (hard to guess) secret value between every service in a scope, it is possible to adjoin a cryptographically strong fingerprint to messages from SAs that is impractical to forge. Only services within the scope could produce the fingerprint; the scope is then called a *protected scope*. If public key techniques are used [2], a user agent possessing a widely-known public key value can verify that only authorized services (i.e., services that belong to the protected scope) could produce (for instance) a particular Service Request.

4.3 DAs and Extensibility to New Service Types

Notice that, in SLP, DAs are not necessarily equipped with any special understanding of the characteristics of a service type. To do the comparisons and evaluations of predicates in Service Requests, the DA uses only lexical indications about how to perform the comparison. This leads to some limitations; for instance, "123" is always an integer and not a character string. Designers of attribute tags for new service attributes should keep this in mind when creating new *service templates* [4].

One benefit of this generic mode of operation in DAs is that new service types can be put into use without any DA reconfiguration at all. The DA does not have to understand anything about the attributes to make selections for Service Replies. The comparisons do not depend on the service type or attribute tag name. A new network service can be configured, deployed, and put into use with no extra administrative overhead other than that needed to assign attribute values. Of course, the service will not be used unless there are user agents that have the necessary code to produce Service Requests conforming to the service type attribute definitions, but that is a natural constraint and localizes the problem to the hosts running the affected UAs.

The scheme template specification and format of the Service Request allow further flexibility in the definition of new service types, by allowing local variations upon standard service types. For interoperability across multiple vendors of equipment supporting SLP service agent operation, the IETF standardizes `service: URL` definitions and makes them available to the global Internet. The IETF standard schemes are considered to be interpreted according to the definitions of the IANA naming authority (where IANA means the *Internet Assigned Number Authority*). However, each scheme is allowed to be interpreted according to definitions supported by local (or alternative) *naming authorities*. Naming authorities can embellish the standard scheme definitions, or create entirely new and nonstandard scheme definitions. For details, see the SLP protocol specifications [11, 5].

5 Conclusions and Future Work

We believe that SLP points the way for a new paradigm for providing network services to mobile computer users as well as for enabling zero administration for enterprise network and system administrators. In fact, it is exactly the zero administration features of SLP that makes it appropriate for inclusion into the feature set of mobile computers.

Currently, SLP is a Proposed Standard within the IETF, and implementation experience has indicated the need for several protocol refinements. Version 2 of SLP [5] is now available, and should replace the existing Proposed Standard well before the presentation of this paper. We expect that SLPv2 will go to Draft Standard.

Besides the creation of new Service Types, we expect that work to provide *wide-area* SLP (currently called *wasrv*) will see new impetus as the base protocol deployment spreads across more enterprises, and as demand for wide area services such as Internet Telephony increases. One proposal for *wasrv* involves the collection of wide-area `service: URLs` by *Wide-Area DAs* (WADAs) for access by enterprise DAs and subsequently by enterprise UAs.

Lastly, we should mention the need for a DA-to-DA server protocols. Server-to-server protocols have not been a particularly strong area for IETF standardization, and the case for SLP is not any different. If DAs could learn to rely on each other, then a DA that had no services registered to match a UA's Service Request might be able to help out by sending the unfulfilled Service Request to another cooperating DA instead of just giving up.

6 Final Words

We hope this brief introduction to SLP will engender interest in the deployment of the protocol, and the creation of new service types. The use of SLP across enterprise boundaries is a problem whose solution could further change the landscape of the global Internet. Participation on the SLP mailing list is encouraged; the mailing list can be joined by sending mail to majordomo@www.svrloc.org, including the line 'subscribe svrloc' in the body of the message. One can keep up with general events within the IETF, by selecting the appropriate links on the Web page `http://www.ietf.org`. The author will also gladly answer electronic mail sent to `charles.perkins@eng.sun.com`.

References

[1] T. Berners-Lee, L. Masinter, and M. McCahill. Uniform Resource Locators (URL). RFC 1738, December 1994.

[2] W. Diffie and M. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, 22:644–654, November 1976.

[3] R. Droms. Dynamic Host Configuration Protocol. RFC 2131, March 1997.

[4] E. Guttman, C. Perkins, and J. Kempf. Service Templates and service: Schemes. draft-ietf-svrloc-service-scheme-05.txt, November 1997. (work in progress).

[5] E. Guttman, C. Perkins, J. Veizades, and M. Day. Service Location Protocol version 2. draft-ietf-svrloc-protocol-v2-04.txT, March 1998. (work in progress).

[6] T. Howes. The string representation of LDAP search filters. draft-ietf-asid-ldapv3-filter-03.txt, October 1997. (work in progress).

[7] P. Mockapetris. Domain Names - Concepts and Facilities. RFC 1034, November 1987.

[8] T. Narten, E. Nordmark, and W. Simpson. Neighbor Discovery for IP version 6 (IPv6). RFC 1970, August 1996.

[9] C. Perkins. DHCP Options for Service Location Protocol. draft-ietf-dhc-slp-02.txt, May 1997. (work in progress).

[10] C. Perkins, Editor. IP Mobility Support. RFC 2002, October 1996.

[11] J. Veizades, E. Guttman, C. Perkins, and S. Kaplan. Service Location Protocol. RFC 2165, July 1997.