Service Location Protocol

The **Service Location Protocol** (**SLP**, **srvloc**) is a [service discovery](https://en.wikipedia.org/wiki/Service_discovery) [protocol](https://en.wikipedia.org/wiki/Protocol_(computing)) that allows computers and other devices to find services in a [local area network](https://en.wikipedia.org/wiki/Local_area_network) without prior configuration. SLP has been designed to scale from small, unmanaged networks to large enterprise networks. It has been defined in [RFC 2608](https://tools.ietf.org/html/rfc2608) and [RFC 3224](https://tools.ietf.org/html/rfc3224) as [standards](https://en.wikipedia.org/wiki/Internet_standard) track document.

* [1Overview](https://en.wikipedia.org/wiki/Service_Location_Protocol#Overview)
* [2Roles](https://en.wikipedia.org/wiki/Service_Location_Protocol#Roles)
* [3Network protocol](https://en.wikipedia.org/wiki/Service_Location_Protocol#Network_protocol)
* [4Security](https://en.wikipedia.org/wiki/Service_Location_Protocol#Security)
* [5Adoption](https://en.wikipedia.org/wiki/Service_Location_Protocol#Adoption)
* [6See also](https://en.wikipedia.org/wiki/Service_Location_Protocol#See_also)
* [7References](https://en.wikipedia.org/wiki/Service_Location_Protocol#References)
* [8External links](https://en.wikipedia.org/wiki/Service_Location_Protocol#External_links)

Overview

SLP is used by devices to announce *services* on a local network. Each service must have a [URL](https://en.wikipedia.org/wiki/Uniform_Resource_Locator) that is used to locate the service. Additionally it may have an unlimited number of name/value pairs, called *attributes*. Each device must always be in one or more *scopes*. Scopes are simple strings and are used to group services, comparable to the *network neighborhood* in other systems. A device cannot see services that are in different scopes.

The URL of a printer could look like:

service:printer:lpr://myprinter/myqueue

This URL describes a queue called "myqueue" on a printer with the host name "myprinter". The protocol used by the printer is [LPR](https://en.wikipedia.org/wiki/Line_Printer_Daemon_protocol). Note that a special URL scheme "service:" is used by the printer. "service:" URLs are not required: any URL scheme can be used, but they allow you to search for all services of the same type (e.g. all printers) regardless of the protocol that they use. The first three components of the "service:" URL type ("service:printer:lpr") are also called *service type*. The first two components ("service:printer") are called *abstract service type*. In a non-"service:" URL the schema name is the service type (for instance "http" in "[http://www.wikipedia.org](http://www.wikipedia.org/)").

The attributes of the printer could look like:

(printer-name=Hugo),

(printer-natural-language-configured=en-us),

(printer-location=In my home office),

(printer-document-format-supported=application/postscript),

(printer-color-supported=false),

(printer-compression-supported=deflate, gzip)

The example uses the standard syntax for attributes in SLP, only newlines have been added to improve readability.

The definition of a "service:" URL and the allowed attributes for the URL are specified by a *service template*, a formalized description of the URL syntax and the attributes. Service templates are defined in [RFC 2609](https://tools.ietf.org/html/rfc2609).

SLP allows several query types to locate services and obtain information about them:

* It can search for all services with the same service type or abstract service type
* The query can be combined with a query for attributes, using [LDAP](https://en.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol)'s query language.
* Given its URL, the attributes of a service can be requested. In standard SLP the attributes are not returned in the query result and must be fetched separately. The Attribute List Extension ([RFC 3059](https://tools.ietf.org/html/rfc3059)) fixes this problem.
* A list of all service types can be obtained
* A list of all existing scopes can be requested.

Roles

SLP has three different roles for devices. A device can also have two or all three roles at the same time.

* *User Agents* (UA) are devices that search for services
* *Service Agents* (SA) are devices that announce one or more services
* *Directory Agents* (DA) are devices that cache services information. They are used in larger networks to reduce the amount of traffic and allow SLP to scale. The existence of DAs in a network is optional, but if a DA is present, UAs and SAs are required to use it instead of communicating directly.

Today most implementations are [daemons](https://en.wikipedia.org/wiki/Daemon_(computer_software)) that can act both as UA and SA. Usually they can be configured to become a DA as well.

Network protocol

SLP is a packet-oriented protocol. Most packets are transmitted using [UDP](https://en.wikipedia.org/wiki/User_Datagram_Protocol), but [TCP](https://en.wikipedia.org/wiki/Transmission_Control_Protocol) can also be used for the transmission of longer packets. Because of the potential unreliability of UDP, SLP repeats all multicasts several times in increasing intervals until an answer has been received. All devices are required to listen on port 427 for UDP packets, SAs and DAs should also listen for TCP on the same port. [Multicasting](https://en.wikipedia.org/wiki/Multicast) is used extensively by SLP, especially by devices that join a network and need to find other devices.

The operation of SLP differs considerably, depending on whether a Directory Agent (DA) is in the network or not. When a client first joins a network it multicasts a query for DAs on the network. If no DA answers it will assume that it is in a network without DAs. It is also possible to add DAs later, as they multicast a 'heartbeat' packet in a predefined interval that will be received by all other devices. When an SA discovers a DA, it is required to register all services at the DA. When a service disappears the SA should notify the DA and unregister it.

In order to send a query in a network without a DA, the UA sends a multicast UDP packet that contains the query. All SAs that contain matches will send a UDP answer to the UA. If the answer is too large to fit into a single UDP packet, the packet will be marked as "overflown" and the UA is free to send the query directly to the SA using TCP, which can transmit packets of any size.

In order to send a query in a network with a DA, the UA will send the query packet to the DA using either UDP or TCP. As every SA must register all services with the DA, the DA is able to fulfill the request completely and simply sends the result back to the UA.

Security

SLP contains a [public-key cryptography](https://en.wikipedia.org/wiki/Public_key) based security mechanism that allows signing of service announcements. In practice it is rarely used:

* The public keys of every service provider must be installed on every UA. This requirement defeats the original purpose of SLP, being able to locate services without prior configuration.
* Protecting only the services is not enough. Service URLs contain host names or IP addresses, and in a local network it is almost impossible to prevent [IP or DNS spoofing](https://en.wikipedia.org/wiki/Internet_protocol_spoofing). Thus only guaranteeing the authenticity of the URL is not enough if any device can respond to the address.
* As addresses can be spoofed, the authenticity of the device must be proven at a different level anyway, e.g. in the application protocol (e.g. with [SSL](https://en.wikipedia.org/wiki/Secure_Sockets_Layer)) or in the packet layer ([IPsec](https://en.wikipedia.org/wiki/IPsec)). Doing it additionally in SLP does not provide much additional security.

Adoption

* SLP is frequently used for locating printers and supported by printing systems such as [CUPS](https://en.wikipedia.org/wiki/CUPS).
* SLP is often found in LAN-enabled printers, so that they are discoverable out of the box. Some client print drivers can use this for printer discovery.
* [ACN](https://en.wikipedia.org/wiki/Architecture_for_Control_Networks), a protocol being developed for entertainment control, uses SLP to find different devices such as dimmers and intelligent lights.
* The [classic Mac OS](https://en.wikipedia.org/wiki/Classic_Mac_OS), and [Mac OS X](https://en.wikipedia.org/wiki/Mac_OS_X) up to version 10.1, used SLP to locate file shares and other services. However, features introduced with [Mac OS X](https://en.wikipedia.org/wiki/Mac_OS_X) (version 10.2 onwards) use [Zeroconf](https://en.wikipedia.org/wiki/Zeroconf" \o "Zeroconf).
* [Netware Core Protocol](https://en.wikipedia.org/wiki/Netware_Core_Protocol) (NCP) clients in a pure IP environment use SLP to locate [Novell NetWare](https://en.wikipedia.org/wiki/Novell_NetWare) and [Novell Open Enterprise Server](https://en.wikipedia.org/wiki/Novell_Open_Enterprise_Server) (OES) servers.
* [SUSE Linux](https://en.wikipedia.org/wiki/SUSE_Linux) supports SLP for a variety of services since version 9.1.
* [Sun Microsystems](https://en.wikipedia.org/wiki/Sun_Microsystems) supports SLPv1 and SLPv2 including SA, UA and DA functionality.[[1]](https://en.wikipedia.org/wiki/Service_Location_Protocol#cite_note-1)
* The [Distributed Management Task Force](https://en.wikipedia.org/wiki/Distributed_Management_Task_Force) has standardized discovery of [WBEM Services](https://en.wikipedia.org/wiki/Web-Based_Enterprise_Management) via SLP.
* The [Storage Networking Industry Association](https://en.wikipedia.org/wiki/Storage_Networking_Industry_Association) has mandated the use of SLP for services discovery in the [Storage Management Initiative - Specification](https://en.wikipedia.org/wiki/SMI-S).