Service Location Protocol

Erik Guttman

James A. Kempf

Charles E. Perkins

Mark Stein

Pete St. Pierre

Technology Development Group

Sun Microsystems, Inc.

Palo Alto, CA 94303

Introduction to Service Location Protocol (SLP)

SLP allows:

- Clients to find Servers by *type* and desired *attributes*
- Services to advertise themselves
- *Scopes* to organize services, using arbitrary policies
- Hierarchy of services not artificially imposed
- Support for internationalization

Introduction to SLP (cont.)

SLP is:

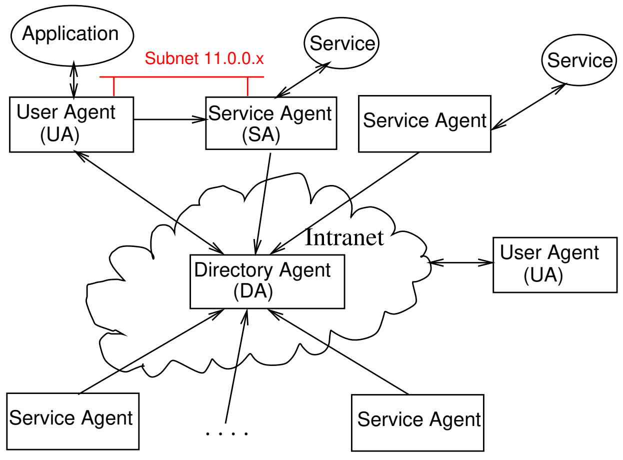
- Low-cost administration and effortless extension to new services
- Decentralized and (after creation) self-administered availability
- Compatible with browser, existing applications, and services (URLs)
- featherweight

Protocol Characteristics

- distributed and self-managing, with numerous service agents
- compatible with other administrative protocols and mobile networking protocols
- string-based (reduced parser complexity)
- easily implemented
- uses existing standards where possible
- expressive query grammar – LDAPv3 query syntax
- *Scalability* a prime motivation, given future explosion of network services and the crowded nature of of enterprise networks

Service Location Discovery Framework

- User Agents (UAs) intercede for applications
- Service Agents (SAs) intercede for services
- UAs and SAs on *nearby* networks can communicate
- Larger deployments use Directory Agents (DAs) (transparent)



URLs (Service Handles)

Standardized way to access a large variety of network resources

General form of URLs:

<scheme>:<scheme-specific-part>

often something like

```
scheme://host:port/opaque
```

Some examples of valid service replies are:

```
nfs://slag.eng.sun.com/src/slp

service:http://www.research.sun.com/

service:lpr://motels.eng.sun.com/MPK15-214
```

Service Requests

- A UA requests a service by *type*, possibly with a *naming_authority*
- from a particular *scope* (or, *default*)
- by *predicate* (a boolean query based on service attributes)

Example:

service type	=	service:printer:lpr
scope list	=	ENGINEERING,QA
predicate	=	(&(location-description=TD Fax/Printer Room)
		(duplex-mode=duplex))

This will be represented in examples as:

<`service-type`[.`na`],`scope`,[`query`]>

Service Request Examples

Some examples of valid Service Requests are:

`<nfs,default,(content=slp-src)>`

`<http.sun,research,homepage>`

`<lpr,local,postscript>`

- In the first example, the attribute *content* has value `slp-src`
- In the second example, the scope is `research`
- In the third example, the scope is the reserved value `local`, and only URLs for printers registered with the keyword `postscript` may be returned

Other SLP Features

- Service Type Request - Browsers can discover all service types
- Dynamic, decentralized registration and deregistration
- Attribute Request - enables (optional!) GUI based profiles and queries
  - Discover all (or subset) of a service's attributes
  - Discover attributes of a service type (to build queries)
- Service attributes can be updated dynamically or deregistered
- NO addresses need to be preconfigured with SLP agents
- GUI tools may provide users with a lot of feedback but are not required. SLP functions give a GUI designer plenty of functionality to support sophisticated interactive discovery.

Service Location Common Message Header

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Version (=2) |    Function   |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|O|Q|U|A|F|R|      reserved     |      Language Tag Length      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Next option, offset      |              XID              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Language Tag (ASCII string) ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Values of the function field (which define the message type):

1	Service Request	6	Attribute Request
2	Service Reply	7	Attribute Reply
3	Service Registration	8	Directory Agent Advertisement
4	Service Deregistration	9	Service Type Request
5	Service Acknowledgement	10	Service Type Reply

Service Request Packet Format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Service Location Header (function =    SrvRqst = 1)    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       length of <PRList>      |  <PRList> String (variable ...)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    length of <service-type>   |   <service-type> (variable ...)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     length of <scope-list>    |<scope-list> String (variable ...)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   length of SrvRqst predicate | SrvRqst predicate (variable ...)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    SrvRqst predicate string, continued (variable length) ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

<`service-type`[.`na`],`scope`,[`query`]>

The <`PRList`> is the *Prev Responders Addr Spec List* for multicast convergence

The *predicate* is either:

- a list of attributes or keywords, OR
- a boolean predicate in prefix notation

Example:

(& (Pages-Per-Minute >= 12)(Unrestricted-Access)(Location = 12th floor))

Service Request for a Directory Agent

<`service:directory-agent,default,`>, which means:

- service scheme = "directory-agent"
- no *query* clause

Could instead have: <`directory-agent.SUN,math-dept,`>

Finding Directory Agents

There are four ways to find Directory Agents:

- Listening for Directory Agent Advertisements
- Multicast/broadcast request for small installations
- Request option 78 from DHCP
- Manual configuration

Note that even the last choice is *MUCH* better than having to configure each service manually time and again.

Option 78 should go through working group Last Call soon.

Service Reply Message Format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Service Location Header (function =   SrvRply = 2)        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Error Code            |         URL Entry Count       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              URL Entry 1 (variable length ...)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              URL Entry 2 (variable length ...)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                              .....
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              URL Entry N (variable length ...)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

A URL entry has the form:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Reserved    |            Lifetime           | # of URL auths|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Length of URL          |   URL (variable length) ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  (if present) URL Authentication Block (variable length) ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Service Registration Message Format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Service Location Header (function =    SrvReg = 3)      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                URL Entry (variable length) ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   length of <service-type>    |<service-type> string (variable)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    length of <scope-list>     | <scope-list> string (variable)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     length of <attr-list>     |<attr-list> string (variable ...)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|# of AttrAuths |  (if present) Attribute Authentication Blocks
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Lifetime of registration is set in the common header.

The *attribute list* is a comma delimited list of

- (`attr` = `value`) , -OR-
- `keyword`

`URL entry` as discussed for the SrvRply

Incremental attribute registration

Service Deregistration

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Service Location Header (function =    SrvDeReg = 5)     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      length of <scope-list>   |<scope-list> string (variable ..)|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Length of URL        |   URL (variable length ...)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| # auth blocks | (if present) Authentication Blocks (variable ...)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      length of <tag-list>     |<tag-list> string (variable ...) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

A <**`tag-list`**> is a list of attribute tags or keywords. Attribute values are not needed in a deregistration.

If keywords and attributes are present in a deregistration, only those are removed from the service registration. Otherwise, the entire service registration is removed.

Service Acknowledge Message Format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Service Location Header (function =    SrvAck = 4)     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Error Code          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

0	Success	1	LANGUAGE_NOT_SUPPORTED
2	PROTOCOL_PARSE_ERROR	3	INVALID_REGISTRATION
4	SCOPE_NOT_SUPPORTED	5	
6	AUTHENTICATION_ABSENT	7	AUTHENTICATION_FAILED
8		9	VER_NOT_SUPPORTED
10	INTERNAL_ERROR	11	DA_BUSY_NOW
12	OPTION_NOT_UNDERSTOOD	13	INVALID_UPDATE

Attribute Request Message Format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Service Location Header (function =   AttrRqst = 6)       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      length of <PRList>       | <PRList> string (variable ...)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Length of URL         |   URL (variable length ...)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    length of <scope-list>     |<scope-list> string (variable ..)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     length of <tag-list>      |<tag-list> string (variable ...)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- Can be used without DAs
- Can request substring matching for attributes

Attribute Reply Message Format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Service Location Header (function =    AttrRply = 7)     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Error Code          |  length of <attr-list> string  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Attribute List (variable length) ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| # of AttrAuths |(if present) Attribute Authentication Blocks  ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- Can be used without DAs
- Should be returned with original upper and lower-case

Service Type Request

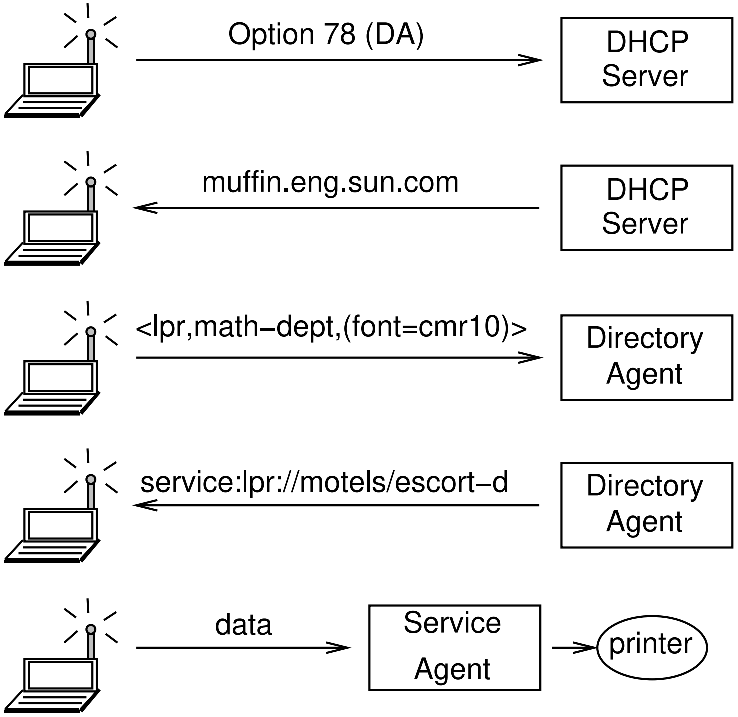
```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Service Location Header (function =    SrvTypeRqst = 9) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       length of <PRList>        | <PRList> string (variable ...)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  length of <naming authority>   |<naming authority> (variable ...)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     length of <scope-list>      |<scope-list> string (variable ..)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- Can be used without DAs
- Only entities loaded with schemes for the indicated Naming Authority may reply
- Likewise for scopes

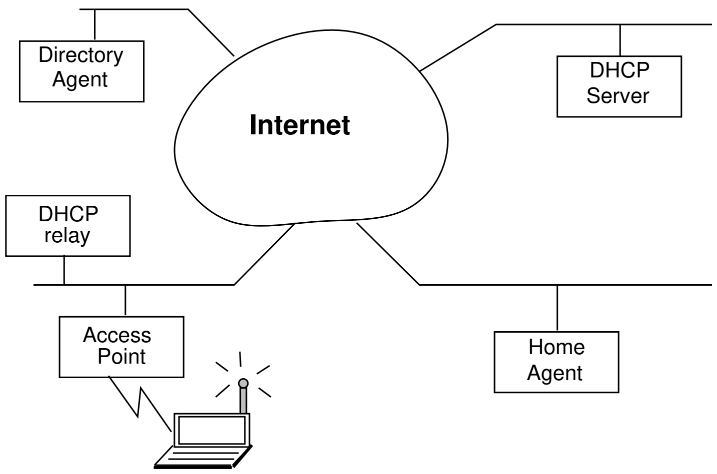
Service Type Reply Message Format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Service Location Header (function =    SrvTypeRply = 10)   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Error Code           |    Number of Service Types    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      length of <SrvType-list> | <SrvType-list> string (variable)
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Discovering a Print Service



- Query *DHCP* for IP address of a Directory Agent
- get back *muffin.eng.sun.com*
- contact *muffin* for URL of printer
- receive *service:lpr://motels/queuename*
- send print jobs to IP address of printer service agent, *motels*



Mobile-IP and Service Location

- Mobile discovers that a cell switch is needed
- Mobile queries DHCP for a *Care-of Address*, and includes the Directory Agent (DA) option 78
- Mobile registers new Care-of Address with home agent
- Mobile queries Directory Agent for local resources

Observations about Mobile Networking and Service Location

- The growth of wireless communications is driving mobile networking
- Mobility is driving the need for automatic resource discovery
- Need to access local resources without preconfiguration
- Mobile users should not have to become system administrators
- While DHCP is useful, it is in no way required for use of SLP either in general or for mobile computers

JNDI Service Provider for SLP

- JNDI service provider for SLP implements the standard JNDI naming interface for SLP
- An SLP-based JNDI driver service allows dynamic registration of drivers for returning service objects
- Drivers are located from the network and linked on demand by the SLP service provider through JNDI
- Design supports legacy services (LPR) as well as newer service designs (IPP)
- Current version supports JNDI 1.1beta, plans are to roll forward to JNDI 1.1 FCS when available.

Scopes

- A *scope* is a set of services
- Services in *protected* scopes provide signed registrations
- Scopes are not hierarchical, and typically not user-selectable
- *default* scope enables smooth transitions, and ability to ignore scopes entirely when not used

Scope Design Points

- Users select services by *attribute*, not by *scope*
- Users can be assigned scopes by the administration in order to conveniently segregate expected access patterns.
- *access definition*, not *access control*
- *local* scope defines a set of services that depends upon the user, and needs interpretation by the DA
- initial deployments will use only *default* scope

Service Provision using Scopes

- User Agents and Service Agents are assigned scopes by administrators
  - Possibly by way of DHCP options 78 or 79
- Policy for set membership is completely determined by administration
- Scoped service agents don't answer service requests from unknown scopes
- Scoped directory agents don't register services from unknown scopes

Service Templates

- Standardized service: URLs to be tabulated by IANA
- Lists all attribute names and enumerations or allowable types for their values
- Service-type name, version number, language, description
- URL syntax
- Grammars (in ABNF) for template fields
- Flags
  - Literal
  - Optional in SrvReg
  - Required in every SrvRqst
  - Multi-valued
- Alternate address families (IPX, Appletalk)

SLP Administrative Tool

- For UAs, configure DA location, attribute selection for desired services, scopes
- For SAs, configure DA location, attribute values for service instance, registration intervals, scopes
- For DAs, configure proxy registrations, advertisement frequency, scopes
- Multicast parameters
- Security associations and key management

Critical: Allow use of Service Templates to simplify user input and perform error checking

LDAP

- Lightweight Directory Access Protocol
  - ⋆ But, ASN.1 is not a lightweight encoding
  - ⋆ SLP is more lightweight – we might say, *Java-weight*
- Derived from X.500 DAP Protocol
- Hierarchical naming scheme
- Centralized administration (one directory per enterprise)
- Pre-defined schemas for directory entries
- Services are statically registered into the directory
- Requires static association between driver and service definition

SLP compared with LDAP

	LDAP	**SLP**
Configuration	Static	Dynamic
Administration	Central	Distributed
Structure	Predefined	Arbitrary
Schemas	Static	Extensible
Language	Single	Internationalized

SLP & LDAP

- SLP can be used to dynamically populate LDAP with service entries
- SLP can be used by LDAP clients to find LDAP services
- SLPv2 queries = LDAP queries, so that LDAP can be SLP back end

SLP Standardization Efforts

- Passed IETF Last Call - May 1997
- Three interoperability workshops held, with a number of participants.
- IBM, Axis, Sun here at March Connectathon (other groups too busy implementing products!)

Differences between SLPv1 and SLPv2

- very much smaller implementations possible
- always UTF8
- no language id
- *Fresh* flag set by SA, not DA
- *MCAST* flag, *Attribute Required* flag
- scopes redefined, *default* scope defined
- DSA (not RSA) is default, multiple authenticators allowed
- better error codes
- <string-list> regularization
- LDAPv3 search filters
- Service-specific multicast attributes gone

IETF Documents

- RFC 2165
- SLP for IPv6
- DHCP options 78, 79
- Interoperation with DNS
- API
- Printer service template
- SLP version 2
- Template recipe format
- Converting LDAP Schemas <==> SLP Templates
- MIB
- *Wired for Management* Systems Management Service Type

Current Status: SLP in the Industry

- Products (beta) shipping from Novell (and Axis and IBM?). Work underway on SLP at Apple, Cisco, Axis, Intel (, HP?).
- Novell is using SLP as underlying basis of new NDS products
- Intel interested in SLPv2 as basis for their "*Wired for Management*" OEM initiative
- Specified as part of MNCRS
- Salutation consortium has adopted SLP for service device discovery

Conclusions

- Service Location Protocol is here now, and can become immediately useful.
- SLP is a natural and now standardized scheme for finding services
- Service Location is one of the next big problem areas for mobile computing
- Using Web technology is appropriate and synergistic for Service Location
- Implementation and deployment can be very simple or much more sophisticated
- For more information, contact cperkins@eng.sun.com,

  or check Service Location Protocol web pages at:

  http://www.srvloc.org

  http://www.ietf.org/html.charters/svrloc-charter.html

This presentation is available at:

http://www.srvloc.org/˜charliep/cthon-slp.ps