Internet Engineering Task Force
Internet Draft
draft-ietf-svrloc-wasrv-01.txt
Nov. 14, 1997
Expires: May 1997

Service Location WG
J.Rosenberg,H.Schulzrinne,B.Suter
Bell Laboratories

Wide Area Network Service Location

STATUS OF THIS MEMO

1 Abstract

We propose extensions to the Service Location Protocol (SLP), which allow for registration and discovery of services scattered across the wide area network. We make use of scalable wide area multicast to allow agents within an administrative domain to learn about services within other domains. We also describe a new agent, the Brokering Agent (BA), which is responsible for providing information about a particular set of services types.

2 Introduction and Motivation

The Service Location Protocol (SLP), recently approved as RFC 2165 [1], with version 2 under development [2], provides an automatic way for clients to discover services within an administrative domain. Clients can select services based on various attributes which describe the service. The basic mechanism involves Service Agents (SA)'s, which represent a service, registering their service attributes with a Directory Agent (DA). User Agents (UA's) query the

DA with a set of desired attributes, and the DA replies with a set of servers meeting the requirements. SLP also specifies operation when there is no DA, by making use of multicast queries.

SLP is designed specifically for use within an administrative domain. Its use of uncontrolled multicast for service queries and DA discovery make it unscalable to wide area usage. However, there are many cases where clients may wish to access services provided outside of their administrative domain. This may be because the client's domain does not provide the service, or because the servers inside the domain do not meet the required criteria. Examples of such services include media servers, telephony gateways, conference bridges, certification authorities, etc.

To facilitate the automatic discovery of such services, explicit protocol support is required. Such a protocol must meet a number of criteria:

1. It must be scalable in terms of network usage; bandwidth consumption in wide area accesses must be limited.

2. It must allow for multi-criteria selection.

3. It must allow a user to locate a service about which it has no apriori hints for its location (such as the domain the service is located in). This is key. If a client wishes to contact a media server, it will not know that the media server is in abc.com or nbc.com; it only cares that the server meets the required service criteria.

4. It must be possible for an administrator to easily control the set of services which can match a client request. This will allow ISP's, for example, to filter services for its users, or prevent its users from accessing certain services.

5. It must allow for universal service. In other words, the protocol should operate so that any host on the Internet can potentially be a client for a service. Of course, local policies can restrict certain clients from accessing a service, but it must be possible to provide universal service.

6. Locating a service should be fast.

7. It should be lightweight.

There are other methods which can be used to locate services in a wide area network. These include the use of SRV records [3] [4], and other approaches which leverage off of DNS, the world wide web, and large

distributed databases, such as X.500 [5] or whois++ [6]. The DNS based approaches generally require the user to either know the domain in which the service is located (which isn't generally known), or require the service to be mapped into some hierarchy based on a single attribute (therefore eliminating multi-criteria selection). The world wide web allows users to search for services. However, the approaches used thus far are not amenable to automation, and require significant human interaction to locate a service. Furthermore, they are generally based on pull technologies, which means that services are often out of date. Hierarchically distributed databases like X.500 have the same problem as DNS - they require services to be mapped into a strict hierarchy, which does not exist for multicriteria selection. Whois++, through centroids, allows for non-hierarchically organized databases, but may require a search through all servers, which is problematic in terms of both search time and server load.

It seems that a sensible solution is to use replicated databases, where each administrative domain keeps a copy of the database of servers providing a particular service, and the attributes which characterize them. Each replicated database can either be flat or hierarchical. However, since it is local to the administrative domain, the load is much smaller than on a single worldwide database, and the time required to search the database is vastly reduced since its components are local. Replication even within an administrative domain can further reduce the loading burden, if needed.

The main difficulty with replicated databases is population of the entries. However, multicast provides a natural framework for populating these databases. It allows servers to register with each database without them needing to know the address of each database (thus meeting the universality requirements and maintaining the simplicity of administration present in SLP), and it is much more efficient than unicasting database entries from the servers to the databases, which causes the traffic volume to grow as the product of the number of databases and servers. Furthermore, techniques like those used in SAP [7] and RTCP [8] [9] [10] can be used to allow for scalability of the multicast advertisements.

The Service Location Protocol provides a natural way in which to define this replicated database structure. It already provides the location service within an administrative domain, and the message formats and operations for clients, databases, and servers as defined are largely applicable. SLP also does not require SA's or UA's to have preconfigured information about infrastructure, which makes it more scalable and manageable. Furthermore, having different architectures for discovery of local and remote services is complex and unwieldy. Our extensions do not alter the interface seen by user agents, thus preserving the appearance of a unified mechanism for location of services. It is for this reason

we belive SLP with multicast extensions for wide area service location is a solution to the problem.

This document specifies extensions to SLP which allow for servers scattered across the wide area network to register with DA's in various domains. It makes heavy use of scalable wide area multicast, and develops new entities for improving scalability. It is not meant as a replacement for SLP, but rather as an extension to SLP which co-exists peacefully with existing systems.

3 Overview

3.1 Terms

For clarity, we first define some terms in addition to those specified in [1]:

- Service Location Protocol Domain (SLPD). An SLPD is a collection of UA's, SA's, and DA's under the administration of a single authority. DA's within one SLPD provide service only to those UA's within the SLPD. SA's within one SLPD may register their services using RFC2165 mechanisms only to DA's within their SLPD.

- Advertising Agent (AA). An advertising agent is responsible for advertising information about services within an SLPD. An SA, DA, or BA may act as an AA for some subset of the services it knows about.

- Brokering Agent (BA). A brokering agent collects advertisements learned from AA's in remote SLPD's. A BA is much like a DA, except that it may not collect information about all service types. In a sense, a BA provides brokering services for a specific set of service types that it knows about. Such a device is useful since the scope and range of services on a wide area network can be large. To reduce storage requirements, and allow for optimized processing and software, a BA only worries about one (or several) service types.

3.2 Basic Operation

The multicast extensions allow for both wide area service discovery, and multicast based registrations within an SLPD.

Within a domain, multicast can be used to allow SA's to register themselves with SA's and/or DA's. Multicast within an SLPD further enhances its scalability for very large domains (like aol.com), where there may be a multitude of DA's, each of which shares the load from various SA's.

The operation of the multicast registration within an SLPD is simple. The domain is configured with a single, well-known, administratively scoped multicast group that is used for multicast registrations. Call this the registration group. Note that this group address is different from the ones currently used in SLP to discover services and send out DAAdverts. A multicast capable (that is, version 3) DA listens to this group, and multicast capable SA's send to the group. Optionally, BA's can listen to the group as well. In order to simplify configuration, SA's can readily determine whether there is a multicast capable DA or BA in the SLPD, since these devices will advertise their presence using multicast to the registration group (A version 2 DA will send its DAAdverts to the DA discovery multicast address, which is different). If there are no such devices present, the SA will unicast its registration to the DA. When there are a mix of unicast and multicast capable DA's, the SA will unicast its registrations to those non-multicast capable DA's, and multicast the registration as well. This allows an administrator to gradually upgrade an SLPD from unicast only operation to multicast without complex administration or configuration. The same mechanisms are used for Deregistration.

Some subset of the services available within the SLPD can be advertised onto the wide area network to other SLPD's. This advertisement is accomplished by means of an advertising agent (AA). An SA, DA, or BA within an SLPD can be configured to act as an AA. When the AA is co-resident with the SA, it must be explicitly configured with the set of services known by the SA to advertise. When the AA is coresident with a BA or DA, protected scopes are used to determine which services are to be advertised externally. An administrator assigns an SA to a protected scope if it wishes its service to be advertised externally. An AA is given the private keys for the protected scope, and will advertise any services it learns through protected scopes. This maintains the decentralized model of service configuration, yet allows for administrative controls over which services are advertised.

The AA's then advertise these services onto a wide area multicast group. The AA's must also join the multicast group that they advertise into. This allows them to count the number of other AA's which are advertising, and then scale the rate of their advertisements proportionally in order to control multicast bandwidth usage. Note that the entire network between AA's and BA's need not be multicast. Tunnels can be used to bring multicast to an AA or BA which needs it. This allows an SLPD who has a unicast-only ISP to still use multicast, since the tunnel is unicast as far as the ISP is concerned. As an alternative, the AA could act as an SA, and unicast its service registrations to another AA in a remote SLPD (which it has been authorized to do). That remote AA can then advertise on behalf of the unicast-only AA.

The BA's within an SLPD join the wide area multicast group corresponding to the service types they wish to broker for. As a result, they will build up a service database of services located in remote SLPD's. An SLPD may have multiple brokers for a particular service type. Furthermore, these brokers may not retain all of the advertisements they receive, dependent on some local policy or policing operations.

BA's also generate service advertisements that they send to the DA's within their SLPD. To do this, the BA acts as an SA. The service type it provides is the abstract type broker, and the concrete type is the one corresponding to the service type they are brokering. The service advertisement contains the service attributes of the brokering services that are provided. The BA, acting as an SA, can register itself with the DA using the unicast mechanisms of RFC2165, or the multicast approaches described here. In this fashion, the DA's within an SLPD know about all of the brokers within the SLPD. The DA's can use this information to decide which BA to contact in order to resolve a query from a UA.

Based on this description, a client can locate a service as follows. Using current SLP methods, the client locates a DA in its domain. It sends out a request for a particular service to the DA. The DA may be able to resolve the query. This may be possible if the required service is within the domain. Otherwise, the DA cannot resolve the service request. However, a DA will always know about the BA's in each SLPD, and know which service types they are providing broker service for. The DA can then contact the BA directly with the query, and then forward the resolution back to the client. In this fashion, client behavior is unchanged between SLPv2 and SLPv3.

Clients may also obtain information about BA's in remote SLPD's through some out of band means, such as an advertisement on a web page, and then contact them directly with queries. This would allow for competitive broker services (for example, a BA with the largest collection of service announcements from a media servers, with the fastest search engines, could offer its services to UA's outside its SLPD, and possibly charge for the brokering services provided).

4 BA URL's and Attributes

Broker agents are both repositories for service information, and service providers themselves. Since they listen to particular multicast groups to build up a service database for particular service types, BA's can be considered as providing broker services for those particular service types.

Consider a BA which collects service advertisements about the service

type remote-fax. This BA can then be considered as providing broker services for remote-fax; the BA can direct users to the right remote-fax based on the attributes of the request. Since this brokering is a service itself, it can be described by a service URL as well. If a broker provides brokering for some service <srvtype>, then the URL for the broker service is [11]:

``service:broker:'' <srvtype> ``://'' <addr-spec>

Where the <addr-spec> is the address of the broker.

Furthermore, like other services, the broker service is characterized by attributes. These attributes are always a superset of the attributes which characterize the service being brokered. When a broker has a particular attribute and value pair which are also a valid attribute value pair for the service, it means that the broker collects service registrations from servers which have that value for the attribute.

For example, consider the following service template for the service type remote-fax:

```
type=remote-fax

version=0.0

lang=en

description=
The remote-fax service allows you to send a fax from a PC to a regular
fax machine.

url-syntax=
  url-path=;

file-format= string M X
tiff
#File formats understood by fax gateway
tiff,gif,postscript,eps

payment-method= string M O
#Payment methods accepted
ecash,visa,mastercard,amex,discover,att-account
```

The concrete service template for the remote-fax broker might then look like:

```
type=service:broker:remote-fax

version=0.0

lang=en

description=
The remote-fax broker service allows you to find remote faxes, which
allow you to send a fax from a PC to a regular fax machine.

url-syntax=
  url-path=;

file-format= string O M
#File formats understood by fax gateway
tiff,gif,postscript,eps

payment-method= string O M
#Payment methods accepted by remote fax
ecash,visa,mastercard,amex,discover,att-account

database-size= integer M
#Size of broker's database - helpful in trying to resolve which broker
to use

broker-fee= string M X
free
#What is the charging model for this broker
free,not-free
```

According to these templates, the remote-fax service type has the attributes file-format and payment-method. The broker:remote-fax service type has the same attributes, plus two more (database-size and broker-fee) which help in choosing the broker.

5 Message Formats

SLPv3 defines no new messages or message syntax beyond what is described in SLPv2 [2].

However, some of the fields have a slightly different semantic. The differences are:

oVersion This protocol document defines version 3 of the Service Location protocol. All multicast registrations from SA's, and advertisements from AA's should have the version number set to 3.

oBitfields. These have the same definition as in SLPv2. For multicast registrations and AA advertisements, the multicast bit is set. The overflow bit may not be sent since all multicast registrations are UDP.

oXID. The XID field is used to distinguish updated registrations from unchanged registrations. A registration message is considered unchanged if the attributes contained within are the same as the ones in a registration which was previously sent. The XID field is set to 0 for the first registration for a particular service URL. If the service attributes for that URL are unchanged the next time the registration is sent, the XID field is not incremented, else it is incremented by 1. This field helps BA's and DA's to decide whether or not to process a message depending on whether they have seen it or not.

6 SA Behavior

Service Agents can operate in one of three modes:

oUnicast registration - registrations are unicast to the DA('s) using only those mechanisms described in SLPv2. This happens when all DA's in the SLPD are v2, or the SA is v2.

oMulticast registration - registrations are multicast to the registration group. This happens when the SA is v3, and ALL DA's are v3.

oHybrid registration - registrations are unicast and multicast. This happens when there is a mix of v3 and v2 DA's, and the SA is v3.

To determine which mode to operate in, the SA listens to the registration group. If, after WAIT_TIMER seconds after joining this group, the SA does not receive any DAAdvert messages or Multicast Service Registrations from a BA which is brokering for the service type provided by the SA, the SA decides to operate in unicast mode.

In unicast registration mode, the SA registers its services using the unicast mechanisms described in SLPv2. The SLPv2 mechanisms will provide the SA with a list of DA's, which we call the unicast DA set, to which it must register. However, the SA continues to listen to the registration group. If, at any time, the SA receives a DAAdvert message or Multicast Service Registration from a relevant broker, it must switch modes

to hybrid.

If, when listening to the multicast address, the SA receives either a DAAdvert or Multicast ServiceReg message from a relevant broker, it switches to hybrid mode. In this mode, the SA begins to multicast its service registrations, based on the rules described in Section 10. The SA also builds up a list of DAAdverts received from the registration group. The set of DA's which are learned through multicast is called the multicast DA set. This set is dynamic. Any DA which has not sent a DAAdvert for more than five times the current transmission interval (this interval is the period between messages from an entity (BA,SA or DA) to the registration group; see section 10) is removed from the set. The SA continues to unicast its registrations to any DA which is in the unicast DA set and not in the multicast DA set. The SA also maintains a partial list of brokers. This list contains those brokers which provide broker service for a service type advertised by the SA. Any broker which has not sent a Multicast ServiceRegistration message for five times the current transmission interval is removed from the list.

If, at any time, the multicast DA set becomes a superset of the unicast set, the SA switches to multicast mode. Furthermore, if the multicast DA set and broker list should become empty, the SA reverts to unicast registration mode.

In multicast registration mode, the SA registers its services using the multicast mechanisms described here. In this mode, all services supported by an SA are registered using multicast. If the multicast DA set should ever cease being a superset of the unicast DA set, the SA reverts to hybrid mode.

These basic rules allow an SLPD to be gradually upgraded from unicast only operation to multicast operation with no configuration. Of course, an administrator can force an SA to always unicast its registrations to some set of DA's if desired, even though the DA's may be multicast capable.

7 DA Behavior

The behavior of a DA is nearly identical to that of SLPv2, with three major differences. The first is that a DA may need to contact a BA to resolve the request. Secondly, a DA must listen to the registration group to collect advertisements, and third, a DA must multicast its DAAdvert messages on the registration group AS WELL as the DA discovery group defined in SLPv2.

7.1 Multicast Listening

Both BA's and SA's may use multicast to register themselves with the

DA. It is therefore neccesary for the DA to listen to the registration group used within the SLPD. By joining this group, the DA will receive multicast service registrations from BA's and SA's.

A DA must keep all service registrations received from brokers. A DA may drop registrations received from SA's, but only if the DA knows of a BA in the SLPD which is providing broker services for that SA. Otherwise, the DA must keep all service registrations received from SA's (unless forbidden by some special administrative policy).

This allows a DA to cease storing registrations from SA's as soon as there is a BA which is storing them. Since the DA always knows about BA's, the DA will still be able to satisfy client queries for local services by contacting the BA.

7.2 Contacting BA's

In SLPv2, it is generally assumed that a DA knows about all services (at least within its scope). However, in SLPv3, there are many reasons why a DA will not know about a service:

- The service is local to the SLPD. However, the SLPD is using the multicast mechanisms for registering some of the SA's. This means that there are BA's within the SLPD, and that the DA may therefore not store all service registrations it receives.

- The service is not local to the SLPD, and the BA's are not coresident with the DA. This means that information collected via multicast advertisements from remote SLPD's are not known to the DA.

Even though a DA may not know about a service, it will always know about the BA's within the SLPD which broker for that service type. This is because all brokers must register themselves with all DA's. This registration describes the service types which are being brokered (via the URL), and the characteristics of that brokering service. These attributes describe, among other things, the subset of the services which are brokered (for example, only those SA's providing remote-fax service with FILE-TYPE of TIFF). In that case, the URL path of all service:broker URL's should include the predicate describing the restriction. For example, the aforementioned remote-fax broker would have a service URL:

service:broker:remote-fax://broker3.brokersrus.com/FILE-TYPE=TIFF

When a DA gets a service request which it cannot resolve, since it doesn't know about the service at all, or one which it can resolve, but

for which it has only partial information (since there are brokers in the SLPD for that service), the DA should consult a broker. To determine which brokers to consult, the DA eliminates those brokers which do not broker for the service type and service which was requested. Such an elimination can happen if the service type is not brokered, or because the service type is brokered, but the attributes of the broker indicate that it does not broker for that particular service. The resulting set of brokers can potentially resolve the request. The DA then further filters the list based on any local policy (for example, do not consult brokers which require payment for their services).

The DA then acts as a UA, and sends a ServiceRequest message to each broker. It may send a message to each broker in parallel, or in series. Each BA will answer the request with a ServiceReply, containing a list of URL's for servers which match the query requirements. The DA may then send an AttributeRequest message to each SA to obtain additional attribute information which may be required to resolve the original request. The DA should not cache the resulting URL's and attribute information. The final match is then determined, and a list of URL's is returned to the UA in a ServiceReply message.

Consider the following example. A UA formulates a query as follows:

```
(&(service-type=media-server)
  (movie = eraser)
  (cost<=*))
```

This query is then sent to the DA. The DA doesn't know about the media-server service, but it knows about two BA's in the SLPD providing the media-server-broker service. It then sends the query to each of the two servers. Both respond with one URL matching the query (presumably the cheapest media server each knows about; this should be the same, but may not be since there are no enforced database synchronization rules). However, if the URL's are different, the DA must determine which is actually cheapest. To that end, it sends an AttributeRequest message to each SA containing the URL for the service. This yields a set of attribute value pairs, including cost. The DA then selects the cheapest of the two, and returns the resulting URL to the UA.

In order to improve scalability, the DA must not query the SA's with an AttributeRequest, as above, unless the UA request contains either the <=* or >=* operator on an attribute. If the original request does not contain the MIN or MAX operator, the DA may return the entire list of URL's obtained from the BA's, or it may return some subset as it sees fit.

7.3 Multicasting DAAdverts

As in SLPv2, the DA's will multicast DAAdverts to make themselves known. However, in SLPv3, DA's will also multicast DAAdverts to the registration group. This effectively announces its ability to receive multicast registrations. The rules for how to transmit the DAAdvert into this group are described in section 10.

8 AA Behavior

An Advertising Agent (AA), is responsible for advertising attributes about the services within its SLPD to other SLPD's. An SA, DA or BA may act as an AA for some subset of services which it knows about. When the SA is acting as an AA, it is administratively configured with the set of services to advertise. Other services which the SA administrator would like to have advertised, but not by the SA itself, should be registered to the DA (and possibly BA) with a protected scope. When the BA or DA is acting as an AA, they advertise those services in protected scopes with which they have been provided private keys.

AA's send Multicast Service Registration messages to a wide-area multicast group (called an advertising group). The rules for choosing the address of this group, and for when to send to the group, are described below.

9 BA Behavior

A BA is responsible for collecting multicast advertisements heard about a particular service.

9.1 Receiving Advertisements

A BA is administratively configured to broker for some set of service types, S. To do this, it determines the multicast groups to listen to for each service in S (see the section below on multicast group selection), and then joins them. This will cause the BA to receive advertisements. Since the groups are sometimes determined from hash functions, two services may both share a multicast group. The BA must drop all service registrations received on a multicast group for which it is not brokering.

9.2 Policy

Once the BA has received a registration for a service that it is brokering, it still has the option of dropping this registration. It is within the rights of an administrator to configure a BA to drop advertisements based on any criteria which can be programmed into the

BA. This allows administrators to implement policy, in much the same way BGP policies allow routers to choose which routes to accept. Some examples of policies include:

1. The BA may drop all registrations received from a set of IP addresses.

2. The BA may drop all registrations which have an attribute set to a particular value.

3. The BA may drop all registrations which do not contain authentication blocks for the URL's.

4. The BA may only accept registrations with the attribute PAYMENT-METHOD set to CREDIT-CARD.

5. The BA may only hold the most recently received 100 registrations.

This list is not meant to be at all exhaustive; it is only to illustrate that there is a wide range of possible policies, limited only by the imagination of the administrator.

Note that these policies apply to services learned about from the wide area multicast group. Services learned from the registration group inside an SLPD should not be dropped.

9.3 Policing

An optional mode of operation for a BA is policing. In this mode, the BA will discard advertisements from AA's who are sending messages faster than is allowed by the protocol operation described below. This will hopefully deter AA's from flooding advertisements to wide area group, which is a form of a denial of service attack.

For each distinct AA which sends an advertisement, each BA will maintain a few pieces of information. This information includes the last time an advertisement was received from that AA, and a violation counter, which is initialized at zero. At all points in time, BA's maintain the current minimum transmission interval (described below), which reflects the smallest allowed interval between transmission of a packet from any AA. When a packet is received from an AA, the BA computes the difference between the current time, and the last time an advertisement was received from that AA. If the difference is less than the minimum tranmsission interval, the counter is incremented. In either case, the value for the last time an advertisement was received is updated to the current time.

If the violation counter hits three, the BA should discard any further advertisements from this AA. The BA may eventually reset the counter, and begin accepting advertisements again, after some suitably long interval, at the discretion of the administrator.

10 Sending Multicast Advertisements

Both within an SLPD, and across the wide area network, entitites will periodically transmit multicast packets. This section discusses the rules by which these packets are sent.

10.1 Scheduling Transmission of Advertisements

The transmitting entity is assumed to have some set of packets, S, which it wishes to send to a multicast group. This situation arises when:

- An SA within an SLPD wishes to multicast its service registrtions to the registration group.
- A DA within an SLPD wishes to multicast its DAAdverts to the registration group.
- A BA within an SLPD wishes to multicast its service registrations to the registration group.
- An AA wishes to multicast some of its services to the wide area network.

The rules for transmission in all of these cases are identical:

- An entity wishing to send to some multicast group must be a member of the group. Call the time it first joins the group time t0.
- There must be only one entity sending to a particular multicast group per IP address. This allows the IP address to be used as a unique identifier for that entity. A multi-homed host should choose one interface for sending its messages.
- After joining the group, the entity must continually maintain a list of all of the souce IP addresses seen in packets sent to the group. At any point in time, this list is used to determine the group size estimate, N, which is a count of the number of other entities transmitting to the group. This estimate is most easily obtained by counting the number of IP addresses seen. If storage is limited, the entity may use statistical sampling to reduce the size of the list, and obtain a statistical estimate of the group size estimate.

oA fixed amount of bandwidth, B, is assumed to be available for packet transmissions to the multicast group. By default, B is 8 kbps. In this way, if there are 1000 entities sending to the group, the period between messages from an entity is around 25 minutes on average. For ten-thousand entities it is a little over four hours.

oThe entity selects an element (message) from S for transmission. If this element differs from the information in the element last time it was transmitted, the XID of the message is incremented, and the age of the message is set to zero. If the element is not different, the age of the element is incremented.

oLet lt represent the last time any message from the entity was transmitted. The time of transmission for the next message, tn, is set to:

tn = lt + R(1/2) max(CONFIG_FAST_TIMER * 2^(min(16,age)), N * K / B)

R(1/2) is a random number uniformly distributed between 1/2 and 3/2. K is the size of the message in bits. CONFIG_FAST_TIMER is 1 second. This formula ensures that the total transmission rate of packets to the multicast group never exceeds B, by evenly dividing this rate among all of the N senders in the group. Furthermore, when group sizes are small, the packet rate is limited depending on the age of the message. A new message is transmitted with a small period, and the period increases as the age of the message grows. Eventually, the period increases to about once a day.

oWhen time tn arrives, the entity does not send the packet. It recomputes the above formula, yeilding a new time, tn'. If tn' is less than tn, the packet is transmitted, lt is set to tn, and the entity selects another message to transmit, computes its transmission time tn as above, and the process repeats. If tn' is more than tn, transmission of the message is further delayed until time tn. This algorithm, called reconsideration, helps avoid bursts of packets from entities who join the group initially, and then have a lowball estimate of the group size.

10.2 Timing Out Senders

In addition to maintaining a list of IP addresses of senders to the group, each entity also maintains the last time a packet was received from that sender. Furthermore, each entity maintains a timeout interval, To, which is equal to:

To = 5 * max(CONFIG_FAST_TIMER * 65536, N * K' / B)

Where K' is the MTU for the interface the entity is connected to. Any entity whose last transmission time is earlier than the current time minus To is assumed to no longer be active, and is therefore timed out. Its address is removed from the list of senders, and the group size estimate is decremented by 1.

An entity which sends a Mulicast Deregistration is also removed from the list, and the group size is decremented by 1.

10.3 Minimum Transmission Interval

For the policing operation, each entity may maintain an estimate of the minimum transmission interval, which is:

Tmin = 1/2 max(CONFIG_INTERVAL_13, N * 128 / B)

Where 128 is chosen as the minimum size in bits for an SLP packet.

10.4 Multicast Groups

There are two different cases for choosing the multicast group to send to. In the first case, the entity is an SA, DA, or BA, advertising its availability within in SLPD. In this case, all messages go on a single multicast group, called the registration group. This group must be administratively scoped. It may be a well-known group, whose value is to be determined by IANA. Administrators may also configure the system for usage with any other group, so long as it is administratively scoped.

For AA's, advertisements are sent to wide area multicast groups. Each service is mapped to at least one multicast group. This multicast group is obtained by applying a hash function to the string which describes the service type (remote-fax, for example). The resulting value indicates an index of a multicast group to use. The set of multicast groups at each index are to be determined by IANA.

As an alternative, a single, well known, wide area multicast group can be used to advertise the addresses for particular services. When a new server comes up to offer a service, it first listens to this group. If it does not see any information about its own service, it obtains a multicast address, and then begins to send an advertisement

out indicating that this multicast address is to be used for the particular service. This avoids static allocation of multicast addresses, which results in a waste of address space when there are too few services, and may result in large amounts of hash collisions when there are many services.

Furthermore, AA's may also advertise onto private multicast addresses. This is useful when a set of SLPD's wish to share information about services, but only among themselves.

11 Open Issues

There are many open issues remaining to be resolved. Some of the more important ones include:

1. Since multicast is used for service registrations, UDP must be used. This means that all registrations must fit within the path MTU. However, it is very likely that wide area services (like a media server), may have very large service descriptions. Some kind of mechanism for breaking up these advertisements into multiple packets seems necessary. Use of IP fragmentation does not seem attractive. Rather, the AA should probably break its attributes into non-overlapping sets, and send sets of attributes in each packet. The precise mechanism for doing this remains unsolved.

2. The determination of multicast groups can be done in one of two ways - either using the hash based approach of SLPv2, or using an additional multicast group to advertise the bindings. Which one to use is an open issue. The latter seems more scalable, but is more complex.

3. For certain services, should we allow for multiple multicast groups? The advertisements can be partitioned among the groups based on attributes. This would allow, for example, remote-faxes in Europe to advertise on one address, and remote-faxes in Africa to advertise on another. This allows a BA which isn't interested in all servers for a service to collect only those it wants by joining the appropriate multicast group. This makes the service to address binding problem harder, and further complicates the protocol, but may improve network bandwidth usage.

4. Wide area multicast is not generally available yet. In the interim, application level multicast, like that used in irc, can be used to connect regions of network level multicast. This would help improve the immediate applicability of the protocol, but at the expense of much complication. Bridging

application and network level multicast makes loop detection and duplicate packet generation a big problem. Solving them will largely require a re-invention of existing multicast routing protocols at the application layer. Is it necessary?

5. Obtaining certificates is a big issue for wide area services. Should we require wide area multicast registrations to contain SLP or X.509 certificates? Should they contain a URL which points to the certificate? The latter seems the most sensible solution.

6. Should we allow a DA to send referrals to BA's back to a UA? This is easily accomplished by having the service reply message contain a list of URL's for BA's instead of SA's. However, it requires UA behavior to be modified, whereas the UA behavior is unchanged in these extensions.

7. When a BA boots, it will initially have an empty database. It can build up its database by listening to the group for long enough. However, for a group with many servers, the learning time may be substantial. Should we allow a BA to query another BA via unicast, and download the entire database for initialization? What happens if the policies in the two BA's are different, so that the source rejected entries that the receiver would otherwise keep?

12 Acknowledgements

The authors would like to thank Erik Guttman for his detailed and insightful comments on this work.

13 Bibliography

[1] J. Veizades, E. Guttman, C. Perkins, and S. Kaplan, Service location protocol, Tech. Rep. RFC 2165, Internet Engineering Task Force, June 1997.

[2] E. Guttman, C. Perkins, and J. Veizades, Service location protocol, (internet draft), Internet Engineering Task Force, Oct. 1997. Work in Progress.

[3] A. Gulbrandsen and P. Vixie, A DNS RR for specifying the location of services (DNS SRV), Tech. Rep. RFC 2052, Internet Engineering Task Force, Oct. 1996.

[4] M. Hamilton, P. Leach, and R. Moats, Finding stuff (how to discover services), Internet Draft, Internet Engineering Task Force,

Oct. 1997. Work in progress.

[5] ITU-T, Recommendation X.500: The Directory: Overview of concepts, models, and services , Nov. 1993.

[6] P. Deutsch, R. Schoultz, P. Faltstrom, and C. Weider, Architecture of the WHOIS++ service, Tech. Rep. RFC 1835, Internet Engineering Task Force, Aug. 1995.

[7] M. Handley, SAP: Session announcement protocol, Internet Draft, Internet Engineering Task Force, Nov. 1996. Work in progress.

[8] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, RTP: a transport protocol for real-time applications, Tech. Rep. RFC 1889, Internet Engineering Task Force, Jan. 1996.

[9] J. Rosenberg and H. Schulzrinne, Timer reconsideration for enhanced RTP scalability, Internet Draft, Internet Engineering Task Force, July 1997. Work in progress.

[10] J. Rosenberg and H. Schulzrinne, Timer reconsideration for enhanced rtp scalability, in To appear in Proceedings of IEEE Infocom '98 , 1998.

[11] C. Perkins, E. Guttman, and J. Kempf, Schemes, Internet Draft, Internet Engineering Task Force, Oct. 1997. Work in progress.

14 Authors Addresses

Jonathan Rosenberg
Lucent Technologies, Bell Laboratories
101 Crawfords Corner Rd.
Holmdel, NJ 07733
Rm. 4C-526
email: jdrosen@bell-labs.com

Henning Schulzrinne
Columbia University
M/S 0401
1214 Amsterdam Ave.
New York, NY 10027-7003
email: schulzrinne@cs.columbia.edu

Bernhard Suter
Lucent Technologies, Bell Laboratories
101 Crawfords Corner Rd.

Holmdel, NJ 07733
Rm. 4C-526
email: suter@bell-labs.com