Internet Engineering Task Force
INTERNET DRAFT

WASRV

Jonathan Rosenberg
Bell Laboratories
Erik Guttman
Sun Microsystems
Ryan Moats
AT&T
Henning Schulzrinne
Columbia U.

draft-rosenberg-wasrv-arch-00.txt
February 15, 1998
Expires: August 15, 1998

WASRV Architectural Principles

STATUS OF THIS MEMO

1 Abstract

This document defines the problem of wide area service location,

describing its key attributes, and giving examples of location problems which do or do not fall under its definition. It also touches on a number of related protocols, and looks at how they fit, or do not fit, the problem of wide area service location.

2 Introduction

There has been recent interest in mechanisms for discovery of services across the global internet. Such interest has manifested itself in numerous papers, standards proposals, and commercial tools. However, what is meant by wide area service location seems to vary in all cases. This document defines the problem of wide area service location explicitly and narrowly. It also looks at how some current protocol architecures may or may not be applied to it.

3 Problem Definition

Wide Area Service Location would allow client software to find services in remote locations in the Internet. Currently, this is very difficult to do. It generally requires the client software to be configured with the network address or domain name of a server. This requires the end-user to have knowledge of the network rather than merely an understanding of what she wishes to achieve.

Wide Area Service Location would enable the discovery of services based on usage criteria as opposed to explicit knowledge of the name or address of the server. The considerations included below express the features which Wide Area Service Location should possess and issues which it must confront.

- Multi-criteria. The client desires a service which can be characterized by a number of attributes. The client should be able to express the desired attributes of the service, and get back a server whose service meets the criteria. There should be no arbitrary restriction on the type, values, or number of attributes which can potentially characterize a service. Some restrictions on typing and some convention with respect to the interpretation of values will exist in any wide area service location protocol.

- Location Independent. The location of the desired server is irrelevent and may not be known a priori. That is, the domain name, administrative domain and network address of the desired service is not generally known in advance and in many cases is not pertinent. In some cases, the location of a server may be important, but no more important than any other attribute of the service.

- Auto Configured. It should be straightfoward to configure new clients and servers to use the Wide Area Service Location Protocol. The new service should automatically be discoverable, requiring little or no setting of addresses or other parameters.

oRapid Availability. When a server is first brought on line, it should become visible and accessible to clients rapidly.

oService Based. The attributes provided by the client provide the attributes of the service, not the content provided by that service. For example, an attribute for a web server might be support of ICP, as this is a characteristic of the web service. Contains web page X is not an attribute of a web server, but rather a description of the content provided by a web server.

oAutomated. The service location process should be automated, and not rely on interactive input from a user to satisfy a reasonable query. However, the protocol should not prevent interactive sessions.

oRapid. The service location process should occur as rapidly as possible. Usually, it is the precursor to further communciations between the client and the server. Service discovery adds to the amount of time required to actually access the service.

oPolicy Support. The agent responsible for satisfying the client's service request should be able to inject its own policy into the process. This policy may disallow various servers from being used by the particular client, for example.

oWorldwide. The location of a matching server can be anywhere, as can be the client. The protocol should be internationalized, supporting queries and attributes in different languages.

oScalable. There can be millions of clients, and thousands of servers for a particular service.

oMildly Dynamic. The attributes which characterize the service provided by some server do not change on small timescales. However, they may change on larger timescales, and this should be handled appropriately.

This definition elimnates quite a number of problems which might otherwise be deemed wide area service location. For example:

oYellow Pages. The yellow pages service allows you to find pizza parlors in San Antonio which deliver, or cleaners in Boise that are open on Saturday. While this resembles the wide area service problem, its focus differs in many respects. The most important distinction is that location (here, geographic) is a key part of the selection process. This allows for the use of a global, distributed database in each geographic locale. YP is therefore focused on locating regional services. Locating wide area

services, on the other hand, is based on multiple attributes, and is global in scope. Furthermore, there is no single attribute on which to hierarchically organize the database.

- White Pages. The white pages service allows you to find some person or service with a specific name, associated with some organization. Like the yellow pages service, it is based on strict hierarchies for the names. It lacks the multicriteria selection required of yellow pages, however. Its main difference from wide area service location is the assumption of a hierarchy of names. Service attributes are much flatter, by comparsion, and change more frequently than names in a WP database.

- Web Pages. The web page location problem is to find a web page (which is on some web server, of course) which contains the word foo. A slightly more advanced version of this would be to find the web page that talks about some subject. This location problem is different from the wide area service location problem in that it is looking for a document, not a service. The documents are usually sorted and indexed based on either content or meta-information. The result is that searching is not automated, but progresses based on interactive input and trial from a human user. Furthermore, web indexing is based on pull of information by webbots. Since webbots have no way of knowing when information on a page has changed, the information returned by a search engine is often stale and out of date.

- User Location. When making an Internet telephone call, it is necessary to find the address where the called user may be found. This address may depend on time of day (I'm at home after 9pm), callee preferences (If Bob calls, forward to my cell phone), caller preferences (I only want him if he's at work), and status (I'm currently on the phone - try my voicemail). This location service is different in that the attributes characterizing the service to be located (here, the person to be called) change on rapid timescales. Furthermore, there is usually a tie to administrative location when finding a user. When I want to call John Doe, I usually know that John Doe works for Lucent, and so he may be found on a machine under their administrative control.

With so many examples of what is not a wide area service location problem, some example of what does quality are useful:

- Gateway Location. I want to make a call from a PC to someone on the PSTN in Zimbabwe. I need a telephony gateway that speaks H.323 and uses the G.729 speech coder. The gateway should accept credit cards for the call.

oMedia Server. I need a media server that speaks RTSP, and has all the latest Warner Brothers films in MPEG2 format.

oBank Server. I wish to complete an electronic transaction on the Internet. I need to contact some server which will accept Visa cards, and then transfer some kind of token representing electronic cash to me, which I can then use to purchase an item at a web site.

oConference Bridge. I need a conference bridge that can handle at least 20 people, and understands H.261.

4 Other Protocols

With the abundance of existing protocols for finding things on the Internet, it is valuable to comment on why these are not appropriate for wide area service location, and why something else is needed.

4.1 Service Location Protocol

SLP [1] defines mechanisms whereby a client can locate a service within an administrative domain. The current architecture is centered around the concept of a directory agent, or DA. This device is a server which collects information about services provided by various service agents (SA's) across the network. The SA's register the services they provide directly with the DA. Clients (also known as user agents, or UA's) wishing to locate a service send a query to the DA, which then searches its database for matches. The DA then returns the list of servers to the client. Clients and SA's must know the IP address of the DA. This can be learned in one of several ways: through static configuration, through DHCP, through multicast advertisements from the DA, or through multicast searches performed by the client or SA.

This protocol does not scale beyond a single administrative domain for several reasons:

1. Registrations of services from SA's to DA's are asynchronous, and the soft-state refresh interval is fixed (the recommended value is around three hours). This means that if thousands of servers attempt to register, the DA may be swamped with bursts of messages. As the number of servers grows, this problem worsens.

2. SA's must know the location of all DA's. In a small administrative domain, this is easy. But in a wide area Internet, this is virtually impossible. Having fixed tables of DA's does not scale well, and is not dynamic enough. Using the multicast

searching technique will cause an overload on the network, since the scope of such searches is neccesarily unbounded. Having DA's advertise their presence works better, but still causes traffic. With fixed soft-state refresh intervals, the bandwidth used grows linearly with the number of DA's present in the network.

3. As the number of servers and services grow, the disk space required for a DA to store information about all of them is excessive. DA's must somehow be able to provide service location for only a subset of services.

4.2 LDAP

The Lightweight Directory Access Protocol [2] was designed as a thin front end for accessing an X.500 database. It's applicability has grown, and it is now a more general mechanism for both client and administrative access to distributed databases, X.500 or otherwise.

Its main limitation for wide area service location is its reliance on indexing of the database on a single key, and distribution of components of the database across the network based on that key. Usually, this key is related to some kind of organizational hierarchy. This makes LDAP databases ideal for things like yellow pages and white pages. However, in wide area service location, there is no single attribute upon which the database can be distributed.

Furthermore, LDAP requires a great deal of regularity in syntax and semantics in order to function properly. All cooperating databases and clients must use exactly the same schema. Furthermore, it is not feasible to store or search for entries for which the schema is not known a priori. This is problematic for wide area service location, where many remote services are involved, and where the client may not know about the schema. By its nature, the directory supports lookup of entities well, but wide area searches by 'type' very poorly. Wide area service location must be more flexible, allowing more looser use of schema. Furthermore, the schema for any service must be extensible in a de-centralized manner.

While a wide area location system based solely on LDAP requires a more robust solution to the problems of server discovery than those given in [3] [4], LDAP could be used by clients to make requests of a wide area service location system.

4.3 DNS

DNS, like LDAP databases, are based on a hierarchical organization. For DNS, this hierarchy is arranged based on domain name. Unlike

LDAP, the queries for the database are based on a single attribute value (the domain name) to be matched. This makes its applicability to wasrv even less than LDAP. DNS is most useful for white pages type of services, which generally rely on a single attribute to match.

However, unlike LDAP, DNS has some legacy use in service discovery where the domain of the service is known a priori. (see [5]). Therefore, for those systems that use DNS for wide area service location, [6] and [7] provide a discussion on how to accomplish this.

4.4 URN's

Another possibility for a wide area service location is the use/resolution of URNs [8]. URNs can support the multi-criteria, location independence, network level and automation requirements. However, URNs at this time do not provide for auto configuration, instant availability and mildly dynamic requirements needed for a wide area location service. Therefore, while we can consider use/resolution of URNs at the client/server interface, they are not appropriate for the system in its entirety.

4.5 Whois++ or CIP

Whois++ is another type of distributed database [9]. Unlike LDAP, it does not require the use of a strict hierarchy to organize the database. Each database is indexed into a collection of forward information (called a centroid), which is to other databases. Whois++ servers use the information in centroids to aid query routing by returning referrals to clients. The client then uses the referral to determine where next to query for desired information.

This approach has been expanded upon in the Common Indexing Protocol [10], which applies to any type of database, not just whois++. Note that CIP considers only the server to server "back-end" aggregation and distribution of index objects. Client-server protocol exchanges are not considered under the CIP architecture, but are handled with "native" protocols (Whois++, LDAP, etc.)

Since index objects aggregate databases, they may have a place in wide area service discovery. However, scalability issues (such as lengthening referral chains and index object size) make it likely that they would be just one component of an overall architecture.

5 Security Considerations

Wide area service location must be able to provide functions for determining the authenticity of discovered services and their attributes. It is clear by its nature as a wide area discovery

protocol that confidentiality is not a WASRV requirement.

The WASRV framework itself should not be easily subvertable. That is, messages sent between WASRV agents should be able to be authenticated so that attackers cannot easily divert or bring down this wide area service.

The 'autoconfigurability' requirement means that configuration requirements for security should be kept to a minimum even if they cannot be eliminated entirely.

6 Conclusion

We have defined the problem of wide area service location, identifying its key attributes. Examples of service problems which do and do not fit our characterization are described. We then discussed some existing distributed data retrieval protocols which might be used for wide area service location, and shown how they are not appropriate.

7 Full Copyright Statement

8 Bibliography

[1] J. Veizades, E. Guttman, C. Perkins, and S. Kaplan, Service location protocol, Request for Comments (Proposed Standard) 2165, Internet Engineering Task Force, June 1997.

[2] T. Howes, S. Kille, and M. Wahl, Lightweight directory access protocol (v3), Request for Comments (Proposed Standard) 2251, Internet Engineering Task Force, Dec. 1997.

[3] R. Moats, LDAP servers finding other LDAP servers, Internet Draft, Internet Engineering Task Force, Nov. 1997. Work in progress.

[4] R. Moats, LDAP clients finding LDAP servers, Internet Draft, Internet Engineering Task Force, Nov. 1997. Work in progress.

[5] A. Gulbrandsen and P. Vixie, A DNS RR for specifying the location of services (DNS SRV), Request for Comments (Experimental) 2052, Internet Engineering Task Force, Oct. 1996.

[6] M. Hamilton, P. Leach, and R. Moats, Finding stuff (how to discover services), Internet Draft, Internet Engineering Task Force, Oct. 1997. Work in progress.

[7] M. Hamilton and R. Moats, Advertising services (providing information to support service discovery), Internet Draft, Internet Engineering Task Force, Oct. 1997. Work in progress.

[8] K. Sollins, Architectural principles of uniform resource name resolution, Request for Comments (Informational) 2276, Internet Engineering Task Force, Jan. 1998.

[9] C. Weider, J. Fullton, and S. Spero, Architecture of the whois++ index service, Request for Comments (Proposed Standard) 1913, Internet Engineering Task Force, Feb. 1996.

[10] M. Mealling and J. Allen, The architecture of the common indexing protocol (CIP), Internet Draft, Internet Engineering Task Force, Dec. 1997. Work in progress.

9 Authors Addresses

Jonathan Rosenberg
Lucent Technologies, Bell Laboratories
101 Crawfords Corner Rd.
Holmdel, NJ 07733
Rm. 4C-526

email: jdrosen@bell-labs.com

Erik Guttman
Sun Microsystems
Bahnstr. 2
74915 Waibstadt
Germany
email: Erik.Guttman@sun.com

Ryan Moats
AT&T
15621 Drexel Circle
Omaha, NE 68135-2358
email: jayhawk@att.com

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